



Universidad Nacional del Sur

TESIS DE DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

Formalización de Agregación de Argumentos:

Semánticas de Aceptabilidad

y Procedimiento de Prueba Dialéctico

Mauro Javier Gómez Lucero

BAHÍA BLANCA — ARGENTINA

2011

Prefacio

Esta Tesis es presentada como parte de los requisitos para optar al grado académico de Doctor en Ciencias de la Computación, de la Universidad Nacional del Sur, y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otras. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el Departamento de Ciencias de la Computación, durante el período comprendido entre el 1 de abril de 2006 y el 15 de febrero de 2011, bajo la dirección del Dr. Carlos I. Chesñevar, Profesor Asociado del Departamento de Ciencias e Ingeniería de la Computación, y el Dr. Guillermo R. Simari, Profesor Titular del Departamento de Ciencias e Ingeniería de la Computación.

Mauro Javier Gómez Lucero

mjg@cs.uns.edu.ar

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

UNIVERSIDAD NACIONAL DEL SUR

Bahía Blanca, 15 de Febrero de 2011.



UNIVERSIDAD NACIONAL DEL SUR
Secretaría General de Posgrado y Educación Continua

La presente tesis ha sido aprobada el .../.../..., mereciendo

la calificación de(.....)

Agradecimientos

En primer lugar, quiero agradecer a mis directores, Carlos Chesñevar y Guillermo Simari, por el conocimiento transmitido durante el desarrollo de mi tesis, pero principalmente, y lo que considero más importante, por brindarme siempre su confianza, su apoyo, y su amistad. Quiero incluir en este agradecimiento a Alejandro García, por haberme motivado y guiado en los inicios de este camino, y haberme transmitido siempre su confianza.

Agradezco a mis compañeros de trabajo y administrativos del Departamento por la buena energía que transmiten, por la solidaridad y predisposición para dar una mano cuando lo que necesité, y por crear un ambiente laboral excelente. En especial, quiero agradecer a mis compañeros de la “salita” de becarios por todos los momentos de diversión compartidos, pero a la vez por prestarme un oído o darme un consejo en las dificultades.

Agradezco a las instituciones que me dieron la posibilidad y el apoyo económico para desarrollar este trabajo: la Universidad Nacional del Sur y el CONICET.

Agradezco especialmente a mi familia por su apoyo y afecto incondicionales, por los valores que me transmitieron con el ejemplo, y por acompañarme plenamente, como siempre, en una etapa más de mi vida.

Agradezco a la familia de Gabi por darme siempre su afecto y por haberme prácticamente adoptado durante todos estos años.

Finalmente quiero agradecer muy especialmente a Gabi por todo lo que significa para mí, por ser siempre un cable a tierra y ayudarme a encontrar y recordar en todo momento lo que realmente importa.

Mauro, mayo de 2011.

Resumen

En los últimos años, la *argumentación* ha evolucionado como un paradigma poderoso para formalizar razonamiento de sentido común, y varios formalismos surgieron basados en la idea de argumentación, encontrando aplicaciones en diversos dominios tales como razonamiento automatizado, toma de decisiones, disputas legales, negociación automatizada, etc. Sin embargo, la mayoría de estos formalismos no permiten modelar la *agregación de argumentos*, identificada inicialmente por Pollock, y cuya importancia ha sido reconocida en incremento en los últimos años.

En esta tesis se define una formalización novedosa de la agregación de argumentos, involucrando una caracterización declarativa de esta noción, y una caracterización operacional asociada abordando su computación. La formalización desarrollada presenta ventajas significativas respecto a los enfoques de agregación existentes, principalmente en relación a las respuestas brindadas, la explicación de respuestas, la evaluación y comparación de agregaciones, y la eficiencia de la computación.

En primer lugar se presentan tres *semánticas de aceptabilidad* para la agregación de argumentos: *a-grounded*, *a-estable* y *a-preferida*, basadas, respectivamente, en las semánticas *grounded*, *estable* y *preferida* propuestas por Dung, y ampliamente reconocidas por la comunidad de argumentación. Adicionalmente se presenta un *procedimiento de prueba dialéctico* para la semántica *a-grounded* siguiendo el enfoque *dialéctico* (*argument games approach*), y basado en el ampliamente reconocido procedimiento dialéctico para la semántica *grounded* propuesto por Dung. Por último se presentan distintas instancias de la formalización propuesta, incorporando el modelamiento de diferentes características, como la regla deductiva *modus tollens*, el tipo de ataque *undercutting*, y la incorporación explícita de *incertidumbre posibilística* a nivel del lenguaje objeto, entre otras.

Abstract

In the last years, *argumentation* has evolved as a powerful paradigm to formalize commonsense reasoning, and several argumentation based formalisms emerged, finding application in different domains such as automated reasoning, decision making, legal dispute, automated negotiation, etc. However, most of this formalisms do not model the notion of *argument accrual*, initially identified by Pollock, and which has been recently gaining increasing importance.

In this thesis a novel formalization of argument accrual is defined, involving a declarative characterization of this notion, and an associated operational characterization addressing computation. The proposed formalization has significant advantages over existing argument accrual approaches, mainly with respect to the answers obtained, answer explanation, accrual evaluation and comparison, and efficiency of computation.

Firstly, three semantics of *argument accrual* are presented: *a-grounded*, *a-stable* and *a-preferred*, respectively based on the *grounded*, *stable* and *preferred* argumentation semantics proposed by Dung, widely recognized by the argumentation community. Additionally, a *dialectical proof procedure* for a-grounded semantics is presented, following the *dialectical* approach (also known as *argument games approach*), and based on the widely recognized dialectical proof procedure proposed by Dung for grounded semantics. Finally some instances of the proposed formalization are presented, incorporating different features like the *modus ponens* inference rule, the *undercutting* attack, and the explicit treatment of *possibilistic uncertainty* at object-language level, among others.

Índice general

1. Introducción	1
1.1. Ejemplo motivador de agregación	2
1.2. Un marco conceptual para sistemas argumentativos	5
1.3. Contribuciones de esta Tesis	11
1.4. Publicaciones surgidas del desarrollo de esta Tesis	13
1.5. Organización de la Tesis	14
2. Programación en Lógica Rebatible: Argumentos, Conflicto y Derrota	17
2.1. Lenguaje de Representación de Conocimiento y Argumentos	17
2.2. Conflictos y Derrotas entre Argumentos	21
3. Semánticas de Aceptabilidad para Argumentos	25
3.1. Asignaciones de Status	26
3.2. Semánticas grounded, preferida y estable	32
3.2.1. Análisis comparativo y propiedades de las semánticas	37
3.2.2. Equivalencia con el enfoque basado en extensiones	38
3.2.3. Semántica grounded caracterizada mediante la noción de Punto Fijo	41
3.3. Procedimiento de Prueba Dialéctico para la Semántica Grounded	42
3.3.1. Refinamiento considerando subargumentos	47
3.4. Conclusiones	49

4. Agregación de argumentos: Estructuras agregadas, Conflicto y Derrota	52
4.1. Estructuras Agregadas	52
4.2. Conflicto y Derrota entre Estructuras Agregadas	59
4.2.1. Ataque Parcial	59
4.2.2. Evaluación y Derrota Parcial	60
4.2.3. Ataque Combinado y Derrota Combinada	62
4.3. Conclusiones	66
5. Semánticas de Aceptabilidad para la Agregación de argumentos	68
5.1. Asignación de Status para Estructuras Agregadas	69
5.1.1. Asignación de status única	74
5.1.2. Situaciones de Bloqueo y múltiples asignaciones de status	77
5.1.3. Ciclos de longitud impar	80
5.1.4. Orden parcial de asignaciones de status por grado de compromiso	81
5.2. Semántica A-Grounded	83
5.3. Semántica A-Estable	85
5.4. Semántica A-Preferida	89
5.5. Semántica A-Grounded caracterizada mediante la noción de Punto Fijo	90
5.6. Propiedades de Consistencia	95
5.7. Modelando el ejemplo del alquiler	97
5.8. Conclusiones	103
6. Procedimiento de Prueba Dialéctico para la Semántica A-Grounded	104
6.1. Líneas dialécticas agregadas	104
6.2. Árboles dialécticos agregados	108
6.3. Vínculo entre la semántica y el procedimiento de prueba	112
6.4. Conclusiones	115

7. Generalización del Lenguaje y algunas instancias relevantes	117
7.1. Framework Argumentativo con Agregación	117
7.2. Modelando undercutting	126
7.3. Refinamiento para evitar la agregación de razones menos específicas	129
7.4. Incorporación de manejo explícito de incertidumbre posibilística	133
7.4.1. Argumentación con incertidumbre posibilística: P-DeLP	133
7.4.2. Agregación con incertidumbre posibilística	135
7.5. Conclusiones	140
8. Trabajo Relacionado	142
8.1. Enfoque de Prakken	142
8.1.1. Principios de la Agregación	142
8.1.2. Formalización de Prakken	145
8.2. Enfoque de Verheij (CumulA)	148
8.3. Análisis de los principios de Prakken para nuestro enfoque	152
8.4. Características salientes del enfoque propuesto: discusión	155
8.4.1. Aceptabilidad: comparación de respuestas	155
8.4.2. Evaluación de agregaciones: determinando la relación de derrota	156
8.4.3. Complejidad del análisis argumentativo	157
8.4.4. Explicación de respuestas	159
8.4.5. Computación	159
9. Conclusiones	162
A. Pruebas de Lemas y Teoremas	166

Capítulo 1

Introducción

La *argumentación* es un mecanismo que los seres humanos generalmente empleamos para debatir acerca de alguna cuestión, ya sea con otros seres humanos o internamente con nosotros mismos. En un sentido general, es un proceso de razonamiento en el cual se consideran *argumentos* que sustentan distintas conclusiones. Un argumento para una conclusión puede ser cuestionado o *atacado* por otro argumento, ya sea porque este segundo soporta una conclusión contraria a la del primero o porque lo contradice en algún otro punto (de ahí el término *Argumentación Rebatible*). De esta forma, durante el proceso de argumentación una conclusión originalmente justificada por un argumento puede dejar de estarlo al considerar nuevos argumentos. El propósito final de la argumentación es determinar las conclusiones *justificadas*.

La Inteligencia Artificial (IA) se ha enfrentado desde hace tiempo con el desafío de modelar formalmente y automatizar el *razonamiento de sentido común*, llevado a cabo cotidianamente por los seres humanos, y que casi siempre ocurre a la luz de información incompleta y potencialmente inconsistente [CML00]. Dentro de la IA, varios formalismos surgieron para abordar este desafío. Muchos de ellos se basan en la idea de argumentación, modelando diversos aspectos del proceso argumentativo previamente descrito. Actualmente, formalismos basados en argumentación han sido aplicados con éxito a diferentes problemas de inteligencia artificial, tales como negociación, toma de decisiones, razonamiento legal, sistemas de recomendación y conciliación de ontologías, entre otros.

La noción de *agregación de argumentos*, inicialmente identificada por Pollock (1991), ha comenzado a cobrar cada vez más importancia en los últimos años. Esta noción se basa en la idea intuitiva de que varias razones (argumentos) soportando una misma conclusión

suelen proveer, en conjunto, un soporte más fuerte para la conclusión que cada una de las razones individuales por separado. A pesar de ser una característica natural de la argumentación, y que afecta, en general, el resultado final de un proceso argumentativo (como se verá en la siguiente sección), la mayoría de los formalismos de argumentación existentes no permiten modelar naturalmente la agregación.

Actualmente existen en la literatura principalmente dos formalizaciones de argumentación contemplando agregación (aparte de nuestro enfoque, presentado parcialmente en [LCS09b, LCS09a]), concretamente el sistema CumulA de Verheij [Ver96, Ver95] y la formalización de Prakken [Pra05]. Modelar agregación de argumentos plantea varios desafíos, y en este sentido Prakken ha identificado en [Pra05] tres principios deseables que “todo tratamiento formal de agregación debería satisfacer”. Finalmente Prakken ha estudiado en [Pra05] la importancia de la agregación para modelar razonamiento evidencial y en [BCP06] se aplica agregación de argumentos para el razonamiento acerca de acciones (*practical reasoning*).

En esta tesis se presenta una formalización novedosa de la agregación de argumentos, involucrando una caracterización declarativa de esta noción (semánticas de aceptabilidad para agregación), y una caracterización operacional asociada abordando su computación (procedimiento de prueba dialéctico). Además de satisfacer los principios de agregación propuestos por Prakken, la formalización desarrollada presenta ventajas significativas respecto a los enfoques de agregación existentes, principalmente en relación a las respuestas brindadas, la explicación de respuestas, la evaluación y comparación de agregaciones, la complejidad del análisis argumentativo, y la eficiencia de la computación.

En la sección siguiente se presenta un ejemplo ilustrando la noción de agregación, y la importancia de modelar esta característica al formalizar la argumentación.

1.1. Ejemplo motivador de agregación

Alicia está buscando un departamento para alquilar, y frente a uno de los departamentos candidatos analiza distintos argumentos en favor y en contra de alquilarlo:

- (A) el departamento se encuentra en una muy buena ubicación (de acuerdo a sus intereses), y por lo tanto debería alquilarlo.

(B_1) el departamento es algo pequeño, luego no alquilarlo.

(B_2) el departamento manifiesta problemas de humedad, luego no alquilarlo.

(B_3) los inquilinos del edificio son en su mayoría estudiantes, luego es probable que sean habituales los disturbios, por lo tanto no alquilarlo.

Supongamos que Alicia considera que la ubicación es más importante que cada uno de los otros aspectos por separado, y por lo tanto, que el argumento A para alquilar es más fuerte que cada uno de los argumentos individuales B_1 - B_3 para no alquilar. Sin embargo, Alicia considera que los tres argumentos B_1 - B_3 , en conjunto, tienen más fuerza que A (ver Fig. 1.1¹).

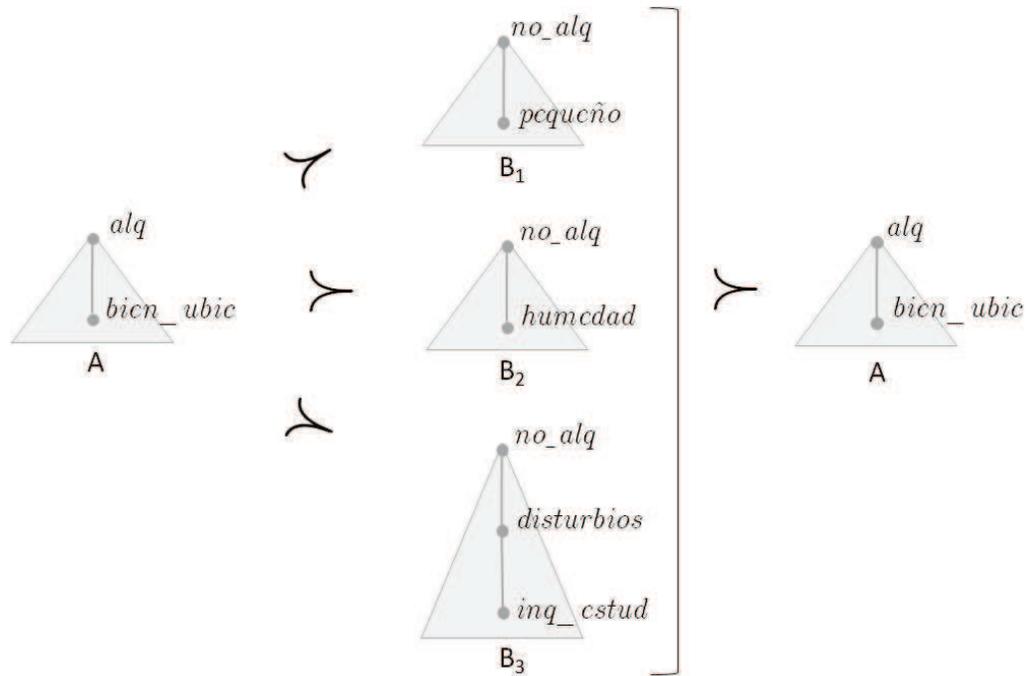


Figura 1.1: Preferencia entre argumentos según el criterio de Alicia (notado con el símbolo ' \succ ', denotando mayor).

La situación planteada no es modelada naturalmente por los formalismos de argumentación tradicionales, que consideran únicamente argumentos individuales, y donde los argumentos en conflicto se comparan de a pares. Para capturar correctamente esta

¹Como es usual en la literatura, los argumentos se representan gráficamente como triángulos, con la conclusión del argumento en la punta, y mostrando en su interior el encadenamiento de razones (inferencia) que sustenta la conclusión

situación es necesario modelar agregación, particularmente que los argumentos B_1 - B_3 soportando ‘no alquilar’ agregan sus fuerzas, logrando superar (o derrotar) al argumento A para ‘alquilar’.

Consideremos ahora una extensión del ejemplo con el propósito de ilustrar otro caso donde se presenta naturalmente la agregación. Supongamos que Alicia decide indagar un poco más acerca de los supuestos disturbios en el edificio, consultando a algunos vecinos. A partir de dicha interacción con los vecinos surgen nuevos argumentos:

- (D_1) Juan, un vecino del edificio, dice que no son habituales los disturbios, luego seguramente no sean habituales los disturbios.
- (D_2) Pablo, un vecino del edificio, dice que no son habituales los disturbios, luego seguramente no sean habituales los disturbios.

Adicionalmente, distinguiremos el subargumento C de B_3 soportando la existencia de disturbios en el edificio:

- (C) los inquilinos del edificio son en su mayoría estudiantes, luego es probable que sean habituales los disturbios (ver Fig. 1.2)

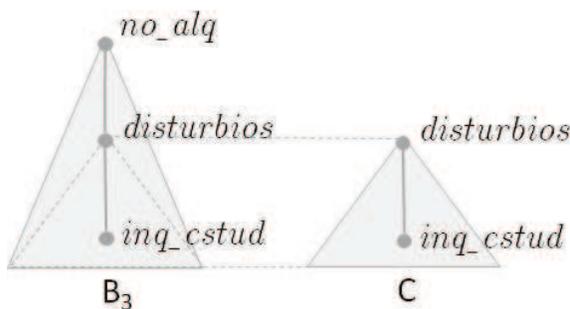


Figura 1.2: Subargumento C de B_3 .

Alicia considera que el testimonio de un vecino (desestimando los disturbios) no es suficiente para derrotar a C , ya que podría justo tratarse de una persona que presta poca atención a los ruidos, o ser de sueño pesado. Sin embargo, si dos vecinos dicen lo mismo, existe una razón más fuerte para creer en lo que dicen. De esta forma Alicia considera que D_1 y D_2 , en conjunto, prevalecen sobre C (ver Fig. 1.3).

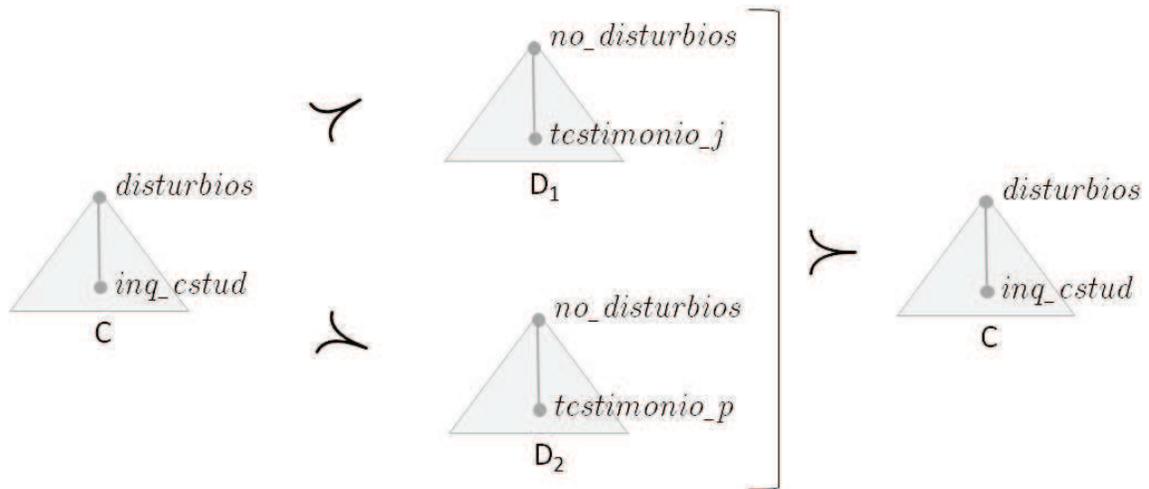


Figura 1.3: Preferencia entre argumentos según el criterio de Alicia.

Analicemos ahora la situación completa, representada gráficamente en la Fig. 1.4, a modo de percibir alguna de las dificultades que se presentan al intentar modelar la noción de agregación. Al agregar D_1 y D_2 el argumento C resultará derrotado. Dado que el argumento B_3 involucra a C como subargumento, B_3 también resulta derrotado al considerar la agregación de D_1 y D_2 , y por lo tanto su fuerza ya no debería agregarse con la de B_1 y B_2 , en contra de A . Este comportamiento es identificado por Prakken en [Pra05], mediante la postulación del siguiente principio general de la agregación: “argumentos individuales que resultan derrotados (consecuencia de la derrota de un subargumento) no deben agregarse”. Finalmente, solo B_1 y B_2 deberían agregarse, y asumiendo que Alicia considera que la ubicación es más importante que las deficiencias representadas por B_1 y B_2 en conjunto, entonces A logrará derrotar a la agregación de B_1 y B_2 , y la decisión de alquilar terminará prevaleciendo (ver Fig. 1.4, parte inferior).

Modelar agregación de argumentos plantea otras dificultades y desafíos, que iremos identificando y abordando al ir introduciendo la formalización.

1.2. Un marco conceptual para sistemas argumentativos

Prakken ha identificado un marco conceptual dentro del cual pueden caracterizarse la mayoría de los sistemas de argumentación existentes ([PV02]). De acuerdo a este marco

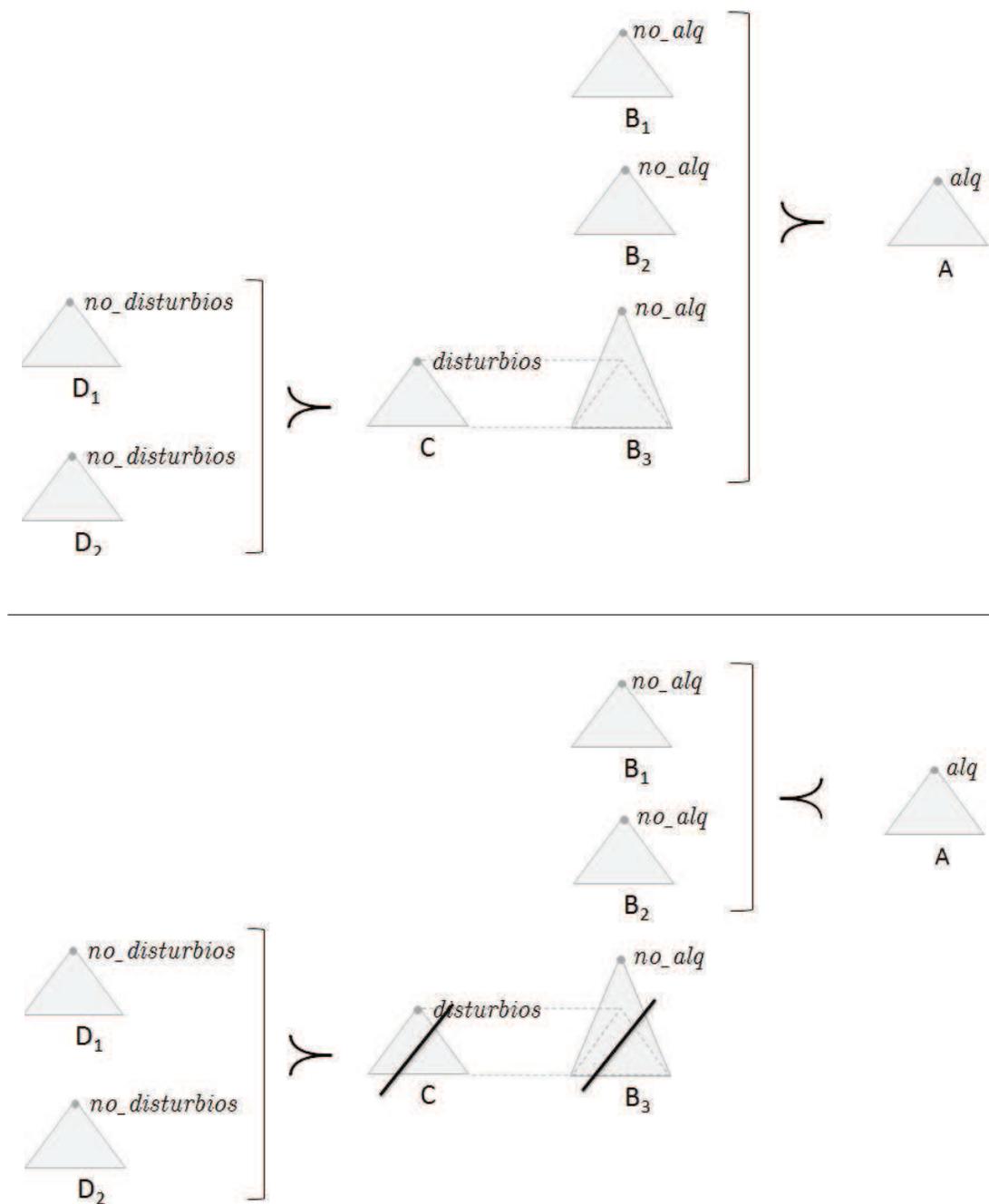


Figura 1.4: Argumentos individuales derrotados (por la derrota de un subargumento) no deben agregarse.

conceptual, todo sistema de argumentación rebatible presenta los siguientes cinco elementos (en algunos casos implícitamente):

E1. un *lenguaje lógico* subyacente;

- E2.** una definición de *argumento*;
- E3.** una definición de *conflicto* entre argumentos;
- E4.** una definición de *derrota* entre argumentos;
- E5.** una definición de *aceptabilidad* de argumentos.

A continuación se analiza en mayor detalle cada uno de los elementos del marco conceptual, describiendo, para cada elemento, distintas alternativas adoptadas por diferentes formalismos argumentativos existentes.

E1. Lenguaje lógico subyacente

Los sistemas argumentativos cuentan con un lenguaje lógico subyacente, que constituye el medio para expresar (formalmente) la información acerca del dominio en que se basará la argumentación. Asociado a este lenguaje lógico se define una noción de consecuencia lógica, pilar para la definición de la noción de argumento (siguiente elemento). Esta noción de consecuencia lógica es monótona ([Dav89]), es decir, nuevas premisas no invalidan consecuencias previas, sino que darán lugar a nuevos argumentos que estarán en conflicto con los argumentos asociados a dichas consecuencias previas (tercer elemento).

Algunos sistemas argumentativos adoptan una lógica particular, mientras que otros sistemas dejan la lógica subyacente parcialmente o completamente sin especificar. Estos sistemas pueden ser instanciados con diferentes lógicas alternativas, y por lo tanto son considerados más bien *frameworks* antes que sistemas.

E2. Definición de argumento

La noción de argumento corresponde a una prueba en la lógica subyacente, empleando la noción de consecuencia lógica descrita previamente. Respecto a la representación formal adoptada para los argumentos, se destacan principalmente tres alternativas en la literatura: como un árbol de prueba con base en las premisas [LS89, Vre92], como una secuencia de prueba (o derivación) [PS97a], o como un par *premisas-conclusión*, dejando implícito que existe una prueba para la conclusión a partir de las premisas en la lógica subyacente [SL92, GS04, Dun93a].

Como se mencionó anteriormente, algunos formalismos argumentativos especifican parcialmente, o directamente no especifican, la lógica subyacente. El sistema de Dung [Dun95] es el ejemplo más extremo de esta característica, donde la lógica subyacente, y

consecuentemente la estructura interna de los argumentos, se encuentran sin especificar. Dung trata la noción de argumento como primitiva y asume que los conflictos entre argumentos están preestablecidos de antemano (es decir, no se derivan de la estructura de los argumentos), lo que le permite concentrarse en el quinto elemento: la interacción entre argumentos y el status final de los mismos.

E3. Conflicto entre argumentos

Intuitivamente, la argumentación presupone desacuerdo en algún sentido. Esto se captura por la noción de conflicto entre argumentos, también llamada *contraargumentación* o *ataque*.

En la literatura existen tres tipos de conflicto. El primer tipo de conflicto, conocido como *rebutting attack* (o ataque por refutación), se da cuando dos argumentos poseen conclusiones contradictorias². Retomando el ejemplo del alquiler de un departamento (Sec. 1.1), este tipo de ataque ocurre entre el argumento A para alquilar y el argumento B_1 para no alquilar. Claramente, el *rebutting attack* es simétrico.

Un segundo tipo de conflicto, conocido como *assumption attack* (o ataque a una suposición), se da cuando un argumento dado involucra una suposición de “no existencia de prueba” (es decir, sustenta la conclusión bajo la suposición de que una cierta fórmula no sea probable, empleando algún tipo de negación default provista por el lenguaje lógico subyacente), y otro argumento prueba la conclusión que fue asumida como “no probable” por el primero ([Dun93b, DKT06]). Este tipo de ataque es asimétrico.

Finalmente, un tercer tipo de conflicto, conocido como *undercutting attack* (e identificado inicialmente por Pollock [Pol70]), se da cuando un argumento cuestiona una regla de inferencia de otro argumento (por supuesto, esta regla debe ser no deductiva o rebatible). A modo de ejemplo, consideremos un argumento que concluye que un cierto objeto es rojo dado que a la vista parece rojo. Si además se sabe que el objeto está siendo iluminado por una luz roja, podría construirse otro argumento que basado en esta información, concluya que el hecho de que el objeto se vea rojo no constituye una razón para creer que es rojo. Nótese que este último argumento no contradice la conclusión del primero, sino que cuestiona la regla de inferencia que conecta premisa y conclusión. Este tipo de ataque es asimétrico.

²En el resto de la tesis se emplearán siempre los términos en inglés para los tres tipos de ataque, por no existir una traducción al castellano ampliamente aceptada.

Tanto el *rebutting* como el *undercutting* tienen una versión directa e indirecta. En la versión directa del *rebutting* el ataque se dirige contra la conclusión final, mientras que en la indirecta se dirige contra una sub-conclusión o conclusión intermedia. Nótese que, estrictamente, el *rebutting* indirecto no es simétrico, ya que el argumento que ataca no es atacado recíprocamente por la víctima, sino por uno de sus subargumentos. En la versión directa del *undercutting* el ataque se dirige contra el paso o regla de inferencia final del argumento, mientras que en el indirecto se dirige contra un paso intermedio.

E4. Derrota entre argumentos

La noción de conflicto no conlleva ninguna forma de evaluación. El cuarto elemento consiste entonces en la evaluación comparativa de pares de argumentos en conflicto para determinar si un ataque dado tiene éxito o no. Esta noción de “ataque exitoso” se formaliza mediante una relación binaria entre argumentos, comúnmente denominada *derrota*, y definida como “ataca y no es más débil”, o en una versión más fuerte como “ataca y es más fuerte” (también denominada derrota estricta).

Distintos sistemas de argumentación adoptan diferentes criterios de evaluación de argumentos. Uno de los criterios más populares en Inteligencia Artificial es el criterio de especificidad, que prefiere argumentos basados en información más específica. Sin embargo, algunos investigadores, por ejemplo, Vreeswijk [Vre91], Pollock [Pol95] y Prakken&Sartor [PS96], consideran que no constituye un principio general del razonamiento de sentido común, sino simplemente un criterio más que podría o no adoptarse. Otros investigadores sostienen que no existen principios generales, independientes del dominio, o que son demasiado débiles (conllevando indecisión en la mayoría de los casos), y que la información acerca del dominio constituye la herramienta más importante para decidir entre argumentos en conflicto [Kon88, Vre91]. Por esta razón varios sistemas de argumentación se encuentran parametrizados respecto al criterio de comparación, que se espera sea provisto por el usuario en relación al dominio de aplicación. Finalmente, otros investigadores (Prakken&Sartor [PS99]) incluso sostienen que los criterios de evaluación son parte de la teoría del dominio, y por lo tanto sujetos a discusión, por lo que que los sistemas argumentativos deberían permitir la construcción de argumentos acerca del criterio.

E5. Aceptabilidad de argumentos

El propósito final de un sistema de argumentación es determinar qué argumentos son aceptables. La aceptabilidad de un argumento depende de la interacción (a través de la relación de derrota) entre todos los argumentos: por ejemplo, un argumento A puede

ser derrotado por un argumento B , pero a su vez B podría resultar derrotado por un argumento C . Esta situación se conoce como *reinstatement*, donde el argumento C reinstaura (la aceptabilidad de) A .

La noción de aceptabilidad varía en general de un formalismo a otro, aunque existen ciertas intuiciones básicas comunes, tales como que un argumento que no es derrotado por ningún otro argumento es aceptable, y la noción de *reinstatement* identificada previamente. Algunos sistemas distinguen dos tipos de argumentos no aceptados: argumentos decididamente no aceptados, y argumentos tales que existe indecisión acerca de aceptarlos o no (esta indecisión es generalmente consecuencia de argumentos en conflicto considerados igualmente preferidos o incomparables de acuerdo al criterio de evaluación – elemento E4.) Diferentes sistemas de argumentación utilizan diferente terminología para hacer referencia a los distintos tipos de argumentos. En esta tesis se empleará la terminología argumento *garantizado* (para referirnos a un argumento definitivamente aceptado), argumento *discutible* (para referirnos a un argumento tal que existe indecisión acerca de su status) y argumento *rechazado* (para referirnos a un argumento decididamente no aceptado).

La aceptabilidad de argumentos puede definirse en forma declarativa o procedural. De acuerdo a la forma declarativa simplemente se declaran o especifican condiciones necesarias y suficientes que debe verificar el conjunto de argumentos considerados aceptados, sin definir un procedimiento para verificar si un argumento dado pertenece o no al conjunto. En la literatura suele emplearse el término *semántica de aceptabilidad* para hacer referencia a una definición declarativa de aceptabilidad. Por otro lado, la forma procedural consiste en brindar un procedimiento para determinar si un argumento se encuentra o no aceptado. En la literatura suele emplearse el término *procedimiento o teoría de prueba* para hacer referencia a una definición operacional de aceptabilidad.

Dung se concentró en [Dun95] en la definición y el análisis de diferentes semánticas de aceptabilidad. Para ello presentó la noción de *framework de argumentación abstracto*, abstrayéndose por completo de la estructura interna de los argumentos (y del lenguaje lógico subyacente), del mecanismo de definición de conflictos y de la evaluación de los argumentos. Concretamente, Dung asume la noción de argumento como primitiva, y que la relación de derrota entre argumentos viene dada, concentrándose así en el quinto elemento: la definición de aceptabilidad. Las semánticas propuestas por Dung en [Dun95] son ampliamente aceptadas por la comunidad de argumentación, y muchos de los sis-

temas argumentativos actuales se definen instanciando el framework abstracto de Dung con definiciones particulares de lenguaje lógico, argumento, conflicto y derrota, beneficiándose así automáticamente de las semánticas de aceptabilidad definidas para dicho framework abstracto.

Tres semánticas fueron introducidas originalmente por Dung, denominadas *grounded* (grounded semantics), *estable* (stable semantics) y *preferida* (preferred semantics). Estas tres semánticas serán presentadas y analizadas en detalle en el Cap. 3, y son centrales al enfoque de agregación propuesto en esta tesis. Posteriormente varios autores introdujeron otras semánticas, generalmente con el propósito de reparar alguna limitación o mejorar algún comportamiento no deseado de las tradicionales. Entre ellas se destacan la semi-estable [Cam06b], la ideal [DMT06], CF2 [BGG05] y la prudente [CMDM05].

Hasta ahora hemos hablado de la forma declarativa de definir aceptabilidad. Respecto a la forma procedural, la mayoría de las definiciones procedurales de aceptabilidad existentes se basan en el llamado enfoque dialéctico, o de juegos argumentales [GS04, CDM03, DMT07, DBC03, JV99, Vre93a, VP00]. La naturaleza dialéctica de la argumentación sugiere naturalmente la formulación de un procedimiento de prueba como una simulación de una discusión (o juego argumental) entre dos personas o jugadores ficticios, que suelen llamarse proponente y oponente, donde el proponente comienza presentando el argumento a ser testado, y luego oponente y proponente van atacando sucesivamente cada argumento propuesto por el jugador contrario. Finalmente, se determinará que el argumento inicial está aceptado cuando, luego de una discusión exhaustiva, el proponente resulte vencedor.

Varias de las semánticas mencionadas previamente para el framework abstracto de Dung cuentan con procedimientos de prueba dialécticos asociados. Particularmente, procedimientos dialécticos para las semánticas *grounded*, *estable*, *preferida* e *ideal* fueron presentados en [MC09, TDH09]. En la Sec. 3.3 se presenta un procedimiento dialéctico para la semántica *grounded* similar al de [MC09].

1.3. Contribuciones de esta Tesis

Se presenta una formalización novedosa de la agregación de argumentos, definiendo tres semánticas de aceptabilidad para la agregación y un procedimiento de prueba asociado a una de ellas. Se destacan ventajas significativas de nuestro enfoque respecto a las

principales formalizaciones de agregación existentes (concretamente, las propuestas por Prakken [Pra05] y Verheij [Ver95, Ver96]).

En primer lugar, se introduce la noción de *estructura agregada* o *a-estructura*, denotando la agregación de diferentes argumentos soportando la misma conclusión. Con el propósito de simplificar la presentación del enfoque, en una primera instancia se adoptó el lenguaje de representación y la noción de argumento provistos por un formalismo de argumentación particular: la *Programación en Lógica Rebatible* (DeLP). Sin embargo, posteriormente se muestra que la formalización propuesta funciona sin modificaciones para un lenguaje más general. La noción de *narrowing*, inicialmente propuesta por [Ver95, Ver96], es adaptada para a-estructuras, donde una a-estructura dada es un *narrowing* de otra si representa un subconjunto de las razones presentes en la primera. Se introducen además las nociones de *ataque parcial* y *derrota parcial*, como versiones para agregación de las nociones tradicionales de ataque y derrota, donde el adjetivo ‘parcial’ se debe a que es posible que afecten solo a una parte (*narrowing*) de la a-estructura víctima. Finalmente se introducen las nociones de *ataque combinado* y *derrota combinada*, capturando que el ataque de varias a-estructuras, consideradas conjuntamente, pueden provocar un ‘daño mayor’ sobre una a-estructura víctima, que si fueran consideradas individualmente.

Se presentan tres semánticas de aceptabilidad para estructuras agregadas: *a-grounded*, *a-estable* y *a-preferida*. Dichas semánticas para agregación se basan, respectivamente, en las semánticas de argumentación *grounded*, *estable* y *preferida* propuestas por Dung [Dun95], y ampliamente reconocidas por la comunidad de argumentación. La caracterización de estas semánticas para agregación contempla las interacciones (conflictos) entre a-estructuras y la noción de derrota combinada, sancionando para cada a-estructura tres *narrowings* complementarios representando su “status de aceptabilidad”: *narrowing garantizado* (denotando las razones definitivamente aceptadas), *narrowing rechazado* (denotando las razones definitivamente no aceptadas) y *narrowing discutible* (representando las razones que presentan indecisión). La distinción de estos tres *narrowings* significa una gran ventaja a la hora de interpretar las respuestas brindadas por el sistema. Finalmente se enuncian y demuestran propiedades deseables de consistencia satisfechas por la caracterización propuesta.

Adicionalmente se presenta un procedimiento de prueba dialéctico para la semántica *a-grounded* siguiendo el enfoque dialéctico (argument games approach – [CDM03, DMT07, DBC03, JV99, Vre93a, VP00]), donde la aceptabilidad de un argumento se determina me-

dante la simulación de una discusión entre dos personas o jugadores ficticios. Una ventaja de este enfoque es que la propia representación formal del análisis dialéctico realizado constituye en sí misma una explicación intuitiva del resultado obtenido. Particularmente, el procedimiento desarrollado se basa en el ampliamente reconocido procedimiento dialéctico para la semántica grounded ([TDH09]).

Como se señaló anteriormente, con el propósito de simplificar la presentación del enfoque de agregación, se adoptó en una primera instancia el lenguaje de representación y la noción de argumento de DeLP. De esta forma, por último se presenta la noción de *Framework Argumentativo con Agregación*, definiendo un lenguaje de representación y noción de argumento más generales, en base a los cuales se aplica directamente toda la formalización de agregación presentada en el contexto de DeLP. Asimismo se definen distintos sistemas como instancias del framework, incorporando el modelamiento de diferentes características, como la regla deductiva *modus tollens*, el tipo de ataque *undercutting*, la discriminación de argumentos redundantes en una agregación y la incorporación explícita de *incertidumbre posibilística* a nivel del lenguaje objeto.

Finalmente se reseñan los principales formalismos de agregación existentes, y se destacan ventajas significativas de nuestro enfoque en relación a las respuestas brindadas, la explicación de respuestas, la evaluación y comparación de agregaciones (para lo cual nuestra formalización permite el uso explícito de incertidumbre posibilística), la complejidad del análisis argumentativo, y la eficiencia de la computación.

1.4. Publicaciones surgidas del desarrollo de esta Tesis

Gran parte de los resultados obtenidos durante el desarrollo de esta tesis han sido publicados en diferentes workshops y conferencias internacionales. Las publicaciones son listadas a continuación:

1. “Formalizing accrual in defeasible logic programming” (Mauro J. Gómez Lucero, Carlos I. Chesñevar, and Guillermo R. Simari). Proceedings of the 12th International Workshop on Nonmonotonic Reasoning (NMR 2008), pp. 122-130.

2. “On the accrual of arguments in defeasible logic programming” (Mauro J. Gómez Lucero, Carlos I. Chesñevar, and Guillermo R. Simari). Proceedings of Twenty-first International Joint Conference on Artificial Intelligence (IJCAI 2009), pp. 804-809. Cabe destacar que IJCAI es considerada la principal conferencia internacional en Inteligencia Artificial, con un índice de aceptación de un 25 %.
3. “Modelling argument accrual in possibilistic defeasible logic programming” (Mauro J. Gómez Lucero, Carlos I. Chesñevar, and Guillermo R. Simari). Proceedings of 10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2009), pp. 131-143.

En 1 y 2 se introdujeron las nociones presentadas en el Cap. 4 de esta tesis y una versión preliminar del procedimiento dialéctico para la semántica a-grounded presentado en el Cap. 6. En 3 se introdujo una formalización de agregación incorporando el manejo explícito de *incertidumbre possibilística*, en la cual se basa el enfoque presentado en el Cap. 7 de esta tesis. Finalmente, se encuentra en evaluación el artículo

“Modelling Argument Accrual with Possibilistic Uncertainty in a Logic Programming Setting” (Mauro J. Gómez Lucero, Carlos I. Chesñevar, and Guillermo R. Simari),

enviado a la revista International Journal of Approximate Reasoning (Elsevier), donde, en base a la formalización de agregación presentada en 3 (contemplando incertidumbre possibilística), se presenta el procedimiento de prueba dialéctico para la semántica a-grounded del Cap. 6, y que resulta de refinar aquel propuesto preliminarmente en 1 y 2.

1.5. Organización de la Tesis

Previamente se introdujo la noción de agregación de argumentos, ilustrando la importancia de contemplar esta característica al modelar formalmente la argumentación. Adicionalmente, se presentó un marco conceptual propuesto por Prakken [PV02] para la formalización de argumentación, identificando cinco elementos o componentes generales (E1-E5) presentes en toda formalización de argumentación existente. La organización de

los conceptos presentados en esta tesis se basa fuertemente en este modelo conceptual. A continuación se describe brevemente el contenido de los restantes capítulos:

Capítulo 2.

Se presentan los elementos E1-E4 del formalismo de argumentación DeLP (Programación en Lógica Rebatible – Defeasible Logic Programming, [GS04]), que son empleados como base para la formalización de agregación desarrollada.

Capítulo 3.

Se presentan y analizan comparativamente las tres semánticas de aceptabilidad propuestas originalmente por Dung [Dun95] (E5 – enfoque declarativo), ampliamente adoptadas por la comunidad de argumentación: semánticas grounded, estable y preferida. Se presenta además un procedimiento de prueba dialéctico para la semántica grounded (E5 – enfoque procedural). Nuevas versiones que contemplan agregación para estas tres semánticas y el procedimiento de prueba dialéctico serán introducidas en el Cap. 5.

Capítulo 4.

En este capítulo se definen los elementos E1-E4 del marco conceptual para nuestro sistema con agregación. Adoptando como lenguaje lógico subyacente el de DeLP (E1), se introduce la noción de estructura agregada (E2), denotando la agregación de diferentes argumentos DeLP que sustentan la misma conclusión. Se introducen las nociones de ataque parcial (E2) y derrota parcial (E3), como versiones para agregación de las nociones de ataque y derrota en DeLP. Finalmente se introducen las nociones de ataque combinado y derrota combinada, que pueden pensarse como un primer paso hacia la caracterización del análisis de aceptabilidad (E5).

Capítulo 5.

Se presentan las semánticas a-grounded, a-estable y a-preferida para la agregación (E5 – enfoque declarativo), basadas en las correspondientes semánticas de aceptabilidad propuestas por Dung, analizándolas comparativamente a la luz de diferentes situaciones. Se enuncian y demuestran propiedades deseables de consistencia satisfechas por la caracterización propuesta.

Capítulo 6.

En este capítulo se presenta un procedimiento de prueba dialéctico para la semántica a-grounded (E5 – enfoque procedural), basado en el procedimiento dialéctico para la semántica grounded presentado en el Cap. 3.

Capítulo 7.

Con el propósito de simplificar la presentación del enfoque de agregación, se adoptó en una primera instancia (Caps. 4, 5 y 6) el lenguaje de representación y la noción de argumento de DeLP. En este capítulo se presenta la noción de *Framework Argumentativo con Agregación*, definiendo un lenguaje de representación y noción de argumento más generales, en base a los cuales se aplica directamente toda la formalización de agregación presentada. Asimismo se definen distintos sistemas como instancias del framework, incorporando el modelamiento de diferentes características, como la regla deductiva modus tollens, el tipo de ataque undercutting, la discriminación de argumentos redundantes en una agregación y la incorporación explícita de incertidumbre posibilística a nivel del lenguaje objeto.

Capítulo 8.

Se describen los dos principales formalismos de agregación existentes, concretamente [Pra05, Ver95], y se destacan ventajas significativas de nuestro enfoque en relación a estos últimos.

Capítulo 9.

Se detallan los principales resultados y conclusiones obtenidas.

Apéndice A.

Se presentan pruebas de proposiciones y teoremas enunciados a lo largo de la tesis.

Capítulo 2

Programación en Lógica Rebatible: Argumentos, Conflicto y Derrota

La *Programación en Lógica Rebatible* (DeLP– [GS04]) es un formalismo de representación de conocimiento y razonamiento que combina resultados de la *Programación en Lógica* y la *Argumentación Rebatible*. y que ha sido aplicado exitosamente en diferentes dominios concretos (por ej. [CMS06]). El formalismo DeLP respeta el modelo conceptual presentado en la Sec. 1.2 para sistemas de argumentación, adoptando como lenguaje lógico subyacente (elemento E1) una extensión de la Programación en Lógica para incluir negación fuerte y representar información rebatible (además de estricta).

En este capítulo se presenta el lenguaje lógico, y las nociones de argumento, conflicto y derrota (primeros cuatro elementos del modelo conceptual) de DeLP, a ser empleados como base para la formalización de agregación desarrollada.

2.1. Lenguaje de Representación de Conocimiento y Argumentos

A continuación se introduce el lenguaje de DeLP, junto a una noción de consecuencia lógica rebatible asociada (elemento E1 del esquema conceptual de argumentación).

Definición 1 (Lenguaje de DeLP) *En el lenguaje de DeLP, un literal ‘q’ es un átomo fijo ‘a’ o un átomo fijo negado ‘ $\sim a$ ’, donde ‘ \sim ’ representa la negación fuerte. El lenguaje de DeLP se define en términos de tres tipos de cláusulas:*

- hechos, que son literales,
- reglas estrictas, de la forma $q_0 \leftarrow q_1, \dots, q_k$, y
- reglas rebatibles, de la forma $q_0 \prec q_1, \dots, q_k$

donde q_0, q_1, \dots, q_k , con $k > 0$, son literales.

Pragmáticamente, los hechos y reglas estrictas se emplean para representar conocimiento seguro, libre de excepciones. Por ejemplo, para expresar conocimiento como “los perros son mamíferos” se empleará la siguiente regla estricta:

$$\text{mamífero} \leftarrow \text{perro}$$

Por otra lado, las reglas rebatibles se emplearán para representar información tentativa o débil, susceptible de ser cuestionada. Por ejemplo, para expresar que “las aves usualmente vuelan” se empleará la siguiente regla rebatible:

$$\text{vuela} \prec \text{ave}$$

Definición 2 (Programa DeLP) *Un programa DeLP es un conjunto finito de hechos, reglas estrictas y reglas rebatibles. En un programa \mathcal{P} dado distinguiremos el conjunto Π de hechos y reglas estrictas, y el conjunto Δ de reglas rebatibles. Cuando sea conveniente se denotará a \mathcal{P} como (Π, Δ) .*

Aunque el lenguaje de DeLP solo admite literales fijos en las reglas, en algunos ejemplos se utilizarán variables solamente como una forma de denotar esquemas de reglas (al igual que se hace en [GS04]).

Ejemplo 1 *A continuación se presenta un programa \mathcal{P} que modela el ejemplo del alquiler presentado en el Cap. 1, con algunos agregados para ilustrar el uso de reglas estrictas. Concretamente, se sabe que el departamento tiene una cochera debajo del edificio (cochera_subsuelo), y que si la cochera se encuentra debajo del edificio entonces con certeza es cubierta (cochera_cub \leftarrow cochera_subsuelo). Además, el hecho de contar con una cochera cubierta constituye una razón para alquilar el departamento (alq \prec cochera_cub)*

$alq \prec b_ubic$	b_ubic
$alq \prec cochera_cub$	
$\sim alq \prec pequeño$	$pequeño$
$\sim alq \prec humedad$	$humedad$
$\sim alq \prec disturb$	inq_estud
$disturb \prec inq_estud$	$dice_no_dist(j)$
$\sim disturb \prec dice_no_dist(V)$	$dice_no_dist(p)$
$cochera_cub \prec cochera_subsuelo$	$cochera_subsuelo$

Como noción de consecuencia lógica se emplea la noción de *derivación rebatible*, una especie de *derivación SLD* (*Programación en Lógica*) extendida apropiadamente para manejar literales negados de la forma $\sim q$ como si fuesen nuevos predicados no_q . La derivación se dice rebatible dado que aunque un literal L pueda ser derivado, puede existir en el programa información que contradiga a L , y luego L no será necesariamente aceptado como una creencia válida del programa.

Definición 3 (Derivación rebatible) Sea \mathcal{P} un programa y q un literal fijo. Una derivación rebatible de q a partir de \mathcal{P} , denotado $\mathcal{P} \mid \sim q$, consiste de una secuencia finita $q_1, \dots, q_n = q$ de literales tal que para cada i , $1 \leq i \leq n$, vale que:

1. q_i es un hecho en \mathcal{P} , o
2. existe una regla R_i en \mathcal{P} (estricta o rebatible) con cabeza q_i y cuerpo b_1, \dots, b_k y cada literal en el cuerpo de la regla es un elemento q_j de la secuencia apareciendo antes que q_i ($j < i$).

En adelante, dado un literal q , usaremos \bar{q} para denotar el complemento de q con respecto a ‘ \sim ’, i.e., \bar{q} denota “ $\sim a$ ” si $q \equiv a$, y “ a ” si $q \equiv \sim a$. Además, diremos que un conjunto dado de cláusulas DeLP es contradictorio si y solo si existen derivaciones rebatibles para un par de literales complementarios a partir de dicho conjunto. Finalmente, para todo programa $\mathcal{P} = (\Pi, \Delta)$ se asume que el conjunto Π (representando información estricta) es no contradictorio.

A continuación se presenta la noción de argumento de DeLP (elemento E2 del esquema conceptual de argumentación).

Definición 4 (Argumento) Sea $\mathcal{P} = (\Pi, \Delta)$ un programa DeLP. Diremos que $\langle \mathcal{A}, h \rangle$ es un argumento para un literal h a partir de \mathcal{P} , si \mathcal{A} es un conjunto de reglas rebatibles de Δ , tal que:

1. $\Pi \cup \mathcal{A} \mid \sim h$,
2. $\Pi \cup \mathcal{A}$ es no contradictorio, y
3. \mathcal{A} es minimal: no existe $\mathcal{A}' \subset \mathcal{A}$ tal que \mathcal{A}' satisface las condiciones (1) y (2).

En resumen, un argumento para una conclusión (literal) h es un par $\langle \mathcal{A}, h \rangle$ donde \mathcal{A} es un conjunto minimal y no contradictorio de reglas rebatibles que permiten derivar rebatiblemente h . Nótese que las reglas estrictas no forman parte de un argumento.

Ejemplo 2 A continuación se listan algunos de los argumentos que es posible obtener a partir de \mathcal{P} . Además, dichos argumentos se encuentran representados gráficamente en la Fig. 2.1.

$$\begin{aligned} \langle \mathcal{A}_1, alq \rangle &= \langle \{(alq \prec b_ubic)\}, alq \rangle \\ \langle \mathcal{A}_2, alq \rangle &= \langle \{(alq \prec cochera_cub)\}, alq \rangle \\ \langle \mathcal{A}_3, \sim alq \rangle &= \langle \{(\sim alq \prec disturb), (disturb \prec inq_estud)\}, \sim alq \rangle \\ \langle \mathcal{A}_4, \sim disturb \rangle &= \langle \{(\sim disturb \prec dice_no_dist(j))\}, \sim disturb \rangle \end{aligned}$$

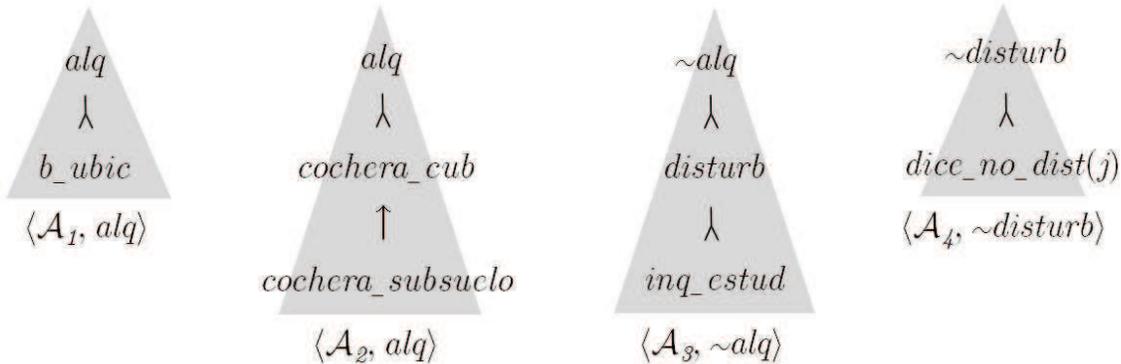


Figura 2.1: Argumentos

Aclaración: Aunque el soporte \mathcal{A} de un argumento $\langle \mathcal{A}, h \rangle$ está formado únicamente por reglas rebatibles, en la representación gráfica de $\langle \mathcal{A}, h \rangle$ también mostraremos las reglas estrictas involucradas en la derivación subyacente de h a partir de \mathcal{A} .

Definición 5 (Subargumento) *Un argumento $\langle \mathcal{B}, q \rangle$ es un subargumento de un argumento $\langle \mathcal{A}, h \rangle$ si $\mathcal{B} \subseteq \mathcal{A}$.*

Ejemplo 3 *La Fig. 2.2 muestra un subargumento $\langle \mathcal{S}_1, disturb \rangle$ del argumento $\langle \mathcal{A}_3, \sim alq \rangle$.*

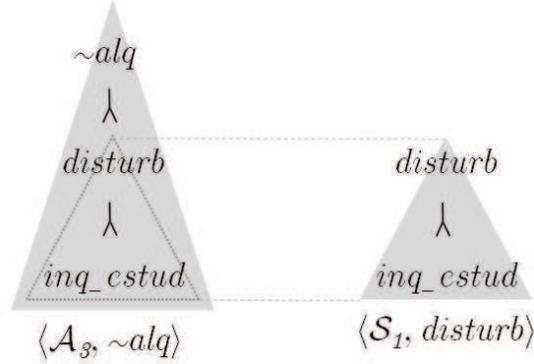


Figura 2.2: Subargumento

2.2. Conflictos y Derrotas entre Argumentos

Definición 6 (Ataque) *Sea \mathcal{P} un programa DeLP, y sean $\langle \mathcal{A}, h \rangle$ y $\langle \mathcal{B}, k \rangle$ argumentos con respecto a \mathcal{P} . Diremos que $\langle \mathcal{B}, k \rangle$ ataca a $\langle \mathcal{A}, h \rangle$ (en el literal \bar{k}) sssi existe un subargumento (llamado subargumento de desacuerdo) $\langle \mathcal{S}, \bar{k} \rangle$ de $\langle \mathcal{A}, h \rangle$.*

Aclaración: La definición de ataque presentada es una simplificación de la introducida originalmente en [GS04], donde el conflicto entre argumentos se define en términos de la noción de *literales en desacuerdo*, en lugar considerar simplemente literales complementarios. Dos literales k y k_1 están en desacuerdo sssi el conjunto $\Pi \cup \{k, k_1\}$ es contradictorio. Luego, la noción de desacuerdo detecta más conflictos que la de literales complementarios, ya que dos literales complementarios k y \bar{k} están trivialmente en desacuerdo, mientras que dos literales a y b podrían estar en desacuerdo, a pesar de no ser complementarios (por ejemplo, si $\Pi = \{(h \leftarrow a), (\sim h \leftarrow b)\}$ o si $\Pi = \{(h \leftarrow a, b), \sim h\}$). A pesar de perder, en una primera instancia, la detección de ciertos conflictos originados a partir de la información estricta, en el Cap. 7 se presenta una generalización del lenguaje de representación que permite, entre otras cosas, modelar la regla de inferencia deductiva *modus tollens*. Empleando esta regla de inferencia, todos los conflictos implicados por la información estricta son detectados.

Intuitivamente, un argumento ataca a otro si la conclusión de este primero contradice la conclusión final del segundo, o alguna de sus conclusiones intermedias. La Fig. 2.3 muestra un esquema ilustrando la definición de ataque presentada. Nótese que la relación de ataque se representa gráficamente mediante una flecha punteada.

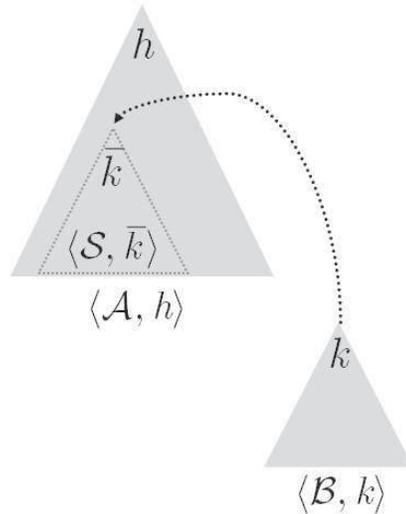


Figura 2.3: Noción de ataque

Ejemplo 4 El argumento $\langle \mathcal{A}_4, \sim disturb \rangle$ ataca al argumento $\langle \mathcal{A}_3, \sim alq \rangle$ en la conclusión intermedia $disturb$, donde el subargumento de desacuerdo asociado es $\langle \mathcal{S}_1, disturb \rangle$ (Fig. 2.4a). El argumento $\langle \mathcal{A}_1, alq \rangle$ ataca al argumento $\langle \mathcal{A}_3, \sim alq \rangle$ en su conclusión final, donde el argumento de desacuerdo es el propio argumento $\langle \mathcal{A}_3, \sim alq \rangle$ (Fig. 2.4b).

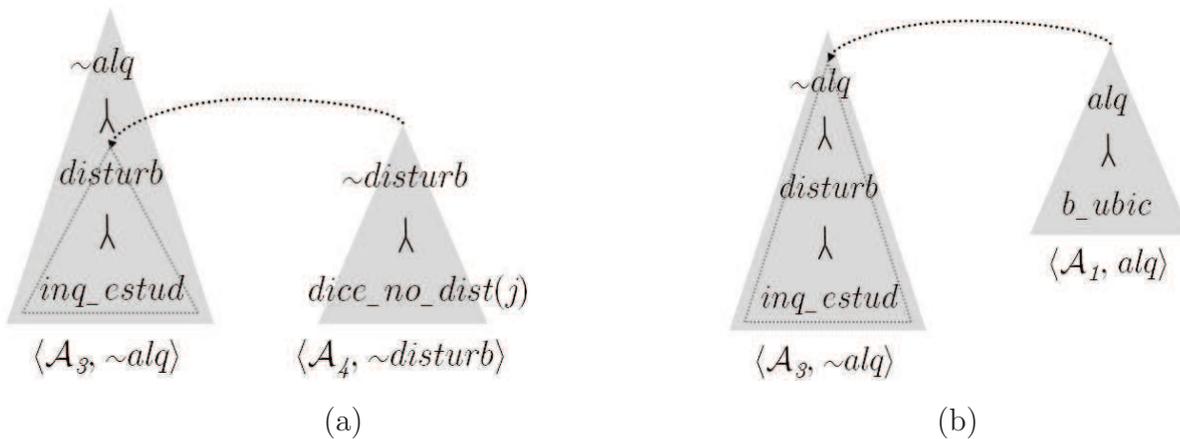


Figura 2.4: Ejemplos de ataque

Para decidir si un ataque realmente tiene éxito y constituye una derrota se necesita de un criterio que establezca la fuerza relativa de los argumentos en conflicto. Como se discutió en la Sec. 1.2, en general, tal criterio debe definirse de acuerdo al dominio de aplicación concreto. En adelante nos abstraeremos del criterio de comparación, asumiendo la existencia de una relación de preferencia binaria ‘ \succsim ’ entre argumentos, significando ‘tanto a más preferido que’ (preferencia débil). A partir de ‘ \succsim ’ puede derivarse una versión estricta ‘ \succ ’ (significando ‘estrictamente más preferido que’), donde $\langle \mathcal{A}_1, h_1 \rangle \succ \langle \mathcal{A}_2, h_2 \rangle$, cuando $\langle \mathcal{A}_1, h_1 \rangle \succsim \langle \mathcal{A}_2, h_2 \rangle$, y $\langle \mathcal{A}_2, h_2 \rangle \not\prec \langle \mathcal{A}_1, h_1 \rangle$. Asimismo diremos informalmente que dos argumentos $\langle \mathcal{A}_1, h_1 \rangle$ y $\langle \mathcal{A}_2, h_2 \rangle$ son igualmente preferidos de acuerdo a ‘ \succsim ’ cuando $\langle \mathcal{A}_1, h_1 \rangle \succsim \langle \mathcal{A}_2, h_2 \rangle$ y $\langle \mathcal{A}_2, h_2 \rangle \succsim \langle \mathcal{A}_1, h_1 \rangle$, y que son incomparables cuando $\langle \mathcal{A}_1, h_1 \rangle \not\prec \langle \mathcal{A}_2, h_2 \rangle$ y $\langle \mathcal{A}_2, h_2 \rangle \not\prec \langle \mathcal{A}_1, h_1 \rangle$.

Definición 7 (Derrota) Sea \mathcal{P} un programa DeLP, y sean $\langle \mathcal{A}, h \rangle$ y $\langle \mathcal{B}, k \rangle$ argumentos con respecto a \mathcal{P} . Diremos que $\langle \mathcal{B}, k \rangle$ derrota a $\langle \mathcal{A}, h \rangle$ (o de manera equivalente que $\langle \mathcal{B}, k \rangle$ es un derrotador para $\langle \mathcal{A}, h \rangle$) si 1) $\langle \mathcal{B}, k \rangle$ ataca $\langle \mathcal{A}, h \rangle$, donde $\langle \mathcal{S}, \bar{k} \rangle$ es el subargumento de desacuerdo asociado, y 2) no es el caso que $\langle \mathcal{S}, \bar{k} \rangle \succ \langle \mathcal{B}, k \rangle$.

La Fig. 2.3 muestra un esquema ilustrando la definición de derrota presentada, donde la relación de derrota se representa gráficamente mediante una flecha continua. La comparación para determinar si el ataque constituye o no una derrota involucra al argumento que ataca y al subargumento del atacado soportando la conclusión complementaria (subargumento de desacuerdo)¹. Luego, el ataque constituye una derrota si no es el caso que el subargumento de desacuerdo $\langle \mathcal{S}, \bar{k} \rangle$ es preferido ante $\langle \mathcal{B}, k \rangle$, ya que si así fuera $\langle \mathcal{S}, \bar{k} \rangle$ resistiría el ataque. En otras palabras, $\langle \mathcal{B}, k \rangle$ derrotará a $\langle \mathcal{A}, h \rangle$ siempre y cuando $\langle \mathcal{B}, k \rangle$ sea tanto o más preferido, o incomparable, a $\langle \mathcal{S}, \bar{k} \rangle$ respecto a la relación de preferencia ‘ \succsim ’.

Ejemplo 5 La Fig. 2.6a muestra un ataque que no constituye una derrota, dado que de acuerdo a ‘ \succsim ’ (reflejando para este ejemplo el criterio de Alicia) el subargumento de desacuerdo es preferido ante el argumento que ataca. La Fig. 2.6b muestra una derrota, dado que de acuerdo a ‘ \succsim ’ el argumento que ataca es preferido ante el subargumento de desacuerdo (en este caso, el propio argumento atacado).

¹No obstante debe quedar claro que todo el argumento atacado se encuentra afectado por el conflicto (no solo el subargumento de desacuerdo), ya que una derivación no debería ser admitida si uno de sus pasos es rebatido.

Capítulo 3

Semánticas de Aceptabilidad para Argumentos

En este capítulo se presentan y analizan comparativamente las tres semánticas de aceptabilidad propuestas originalmente por Dung [Dun95] (elemento E5 del modelo conceptual – enfoque declarativo), ampliamente adoptadas por la comunidad de argumentación: semánticas *grounded*, *estable* y *preferida*.

En la literatura pueden identificarse dos estilos o enfoques comúnmente empleados para definir semánticas de argumentación. Por un lado el enfoque basado en *extensiones* (*extension-based approach*), adoptado originalmente por Dung en [Dun95] para definir las semánticas *grounded*, *estable* y *preferida*, y que consiste en caracterizar uno o varios conjuntos de argumentos, denominados extensiones, que intuitivamente se consideran colectivamente aceptados de acuerdo a la semántica. Por otro lado se encuentra el enfoque basado en *asignaciones de status* (*status assignment approach* – también conocido en la literatura como *labelling-based approach*), propuesto originalmente por Pollock [Pol95], y empleado más recientemente en [MC09, Cam06a, Cam07, Ver07, Vre06]. De acuerdo a este enfoque, la definición de una semántica consiste en caracterizar una o varias asignaciones de status para los argumentos, donde una asignación de status asocia a cada argumento una de varias posibles etiquetas (o status), típicamente *In* (denotando que el argumento se encuentra aceptado de acuerdo a la asignación), *Out* (denotando que el argumento se encuentra decididamente no aceptado de acuerdo a la asignación), y en ocasiones se incluye una etiqueta *Undecided* (denotando indecisión). Como se analiza en [BG09], el enfoque basado en asignaciones de status es tanto o más expresivo que el basado en extensiones.

De hecho, en [Cam06a] se presentan definiciones basadas en el enfoque de asignaciones de status para las semánticas *grounded*, *estable* y *preferida* propuestas por Dung.

La caracterización de las semánticas *grounded*, *estable* y *preferida* presentada en este capítulo se basa en el enfoque de asignaciones de status, y es similar a la propuesta en [Cam06a]. Como sustento a la caracterización presentada se establece y demuestra la equivalencia con el enfoque basado en extensiones.

Adicionalmente se describe una caracterización de la semántica *grounded* basada en la noción de punto fijo, introducida por Dung en [Dun95], y que sugiere un mecanismo simple para computar el conjunto de argumentos aceptados de acuerdo a dicha semántica.

Finalmente se presenta un procedimiento de prueba para la semántica *grounded* (elemento E5 – enfoque procedural) basado en el enfoque dialéctico, o de juegos argumentales (*argument games approach*), donde la aceptabilidad de un argumento se determina mediante la simulación de una discusión entre dos personas o jugadores ficticios.

3.1. Asignaciones de Status

En primer lugar se introduce el framework de argumentación abstracto de Dung [Dun95], en base al cual se definirán las asignaciones de status y las semánticas de aceptabilidad.

Definition 1 (Framework de Argumentación [Dun95]) *Un framework de argumentación (argumentation framework, AF) es un par $(Args, def)$, donde $Args$ es un conjunto de argumentos y $def \subseteq Args \times Args$ es una relación binaria representando una relación de derrota (defeat) entre argumentos.*

Si $A_1 def A_2$, entonces se dirá que A_1 es un *derrotador* (*defeater*) para A_2 .

Un AF denota un grafo dirigido cuyos nodos son los argumentos de $Args$ y los arcos conectando nodos reflejan la relación de derrota ‘ def ’ entre argumentos. En la literatura suele hacerse referencia a este grafo como *grafo de derrotas* asociado al AF . En adelante hablaremos de grafo de derrota o de AF intercambiabilmente.

Definición 8 (Asignación de Status para un AF) *Sea $\Gamma = (Args, def)$ un AF. Una asignación de status \mathbb{S} para Γ es una asignación de un status *IN* o *OUT* (pero no ambos) a algunos (posiblemente a todos) los argumentos de $Args$, verificando que:*

- $A \in \text{Args}$ se encuentra *IN* sssi para todo $B \in \text{Args}$ tal que B def A , B está *OUT*.
- $A \in \text{Args}$ se encuentra *OUT* sssi existe $B \in \text{Args}$ tal que B def A y B está *IN*.

Se empleará $\text{IN}_{\mathbb{S}}$ [$\text{OUT}_{\mathbb{S}}$] para denotar el conjunto de todos los argumentos de *Args* asignados *IN* [*OUT*] de acuerdo a \mathbb{S} .

Una asignación de status dada \mathbb{S} podría dejar argumentos sin asignar, reflejando una postura de indecisión acerca de considerarlos *IN* o *OUT*. Diremos que dichos argumentos sin asignar se encuentran *undecided* de acuerdo a \mathbb{S} ¹.

Aclaración sobre notación gráfica. Los argumentos en el grafo de derrotas de acuerdo a su status asociado. Concretamente, los argumentos *IN* se pintarán de blanco, los *OUT* de negro (gris oscuro), y los *undecided* a rayas blancas y negras. Cuando se desee mostrar solo un grafo de derrotas, sin especificar una asignación de status, los argumentos se pintarán de gris claro.

Considere el grafo de derrotas (*AF*) de la Fig. 3.1a, donde un argumento A es derrotado por un argumento B , que no tiene derrotadores. Claramente existe una única asignación de status posible: el argumento B es asignado *IN* por tener todos sus derrotadores *OUT* (en particular, no tiene derrotadores), y A es asignado *OUT* por tener un derrotador *IN* (B). El grafo de la Fig. 3.1b extiende la situación anterior incorporando un argumento C como derrotador de B . Esta situación se conoce como *reinstatement*, donde el argumento C “reinstaura” A mediante la derrota de su (único) derrotador, B . El argumento C es asignado *IN* por tener todos sus derrotadores *OUT* (en particular, no tiene derrotadores), B es asignado *OUT* por tener un derrotador *IN* (C), y A es asignado *IN* por tener todos sus derrotadores (B) *OUT*.

Consideremos ahora que A tiene otro derrotador, D , que a su vez no tiene derrotadores (Fig. 3.1c). En este caso A será asignado *OUT* dado que tiene un derrotador que se encuentra *IN*: D . Esta situación sugiere una versión más general de la noción de reinstauración: para reinstaurar un argumento dado deben derrotarse *todos* sus derrotadores.

¹Pollock ([Pol95]) utiliza el término *asignación de status parcial* para hacer referencia a este tipo de asignaciones de status, donde ciertos argumentos pueden quedar sin asignar, reflejando indecisión. En el enfoque de [MC09], en cambio, se requiere que todos los argumentos sean asignados, pero se incluye explícitamente una etiqueta adicional ‘UNDEC’ denotando indecisión. Ambas alternativas resultan equivalentes.

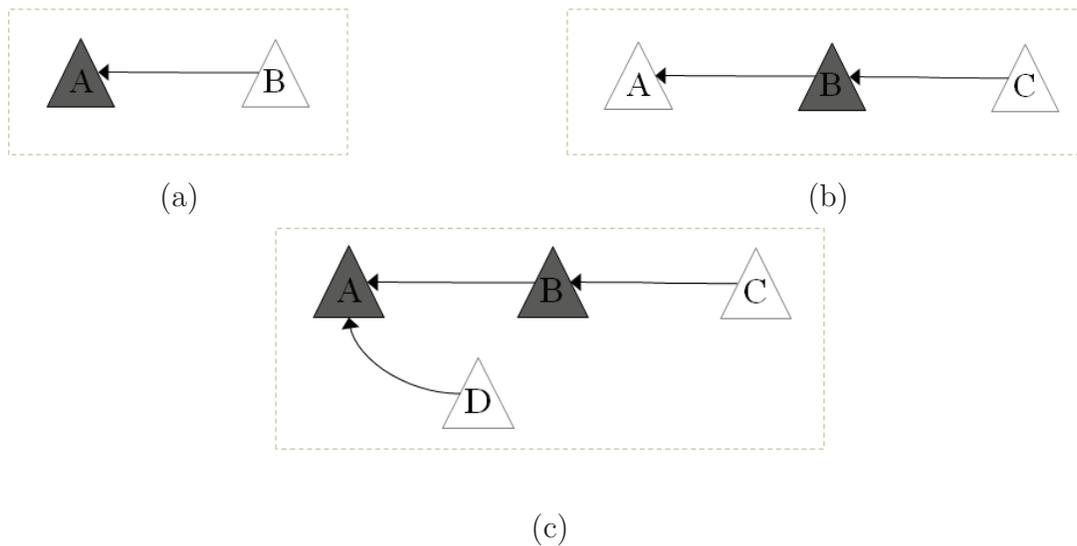


Figura 3.1: Ejemplos de asignaciones de status

Por ejemplo, incorporando un derrotador E para D al grafo de la Fig. 3.1c conseguiremos reinstaurar A , y en ese caso se dice que A es reinstaurado por el conjunto de argumentos $\{C, E\}$.

La Fig. 3.2 muestra una situación comúnmente conocida como *bloqueo*, donde dos argumentos (de igual preferencia o incomparables) se derrotan mutuamente. Como se ilustra en la figura, una situación de bloqueo induce tres asignaciones de status posibles, una de ellas (\mathbb{S}_1) considerando a ambos argumentos como *undecided* (asignación “menos comprometida”), y las otras dos (\mathbb{S}_2 y \mathbb{S}_3) considerando a uno de los argumentos como IN a expensas del otro, considerado OUT (estas asignaciones son “más comprometidas”, representando formas arbitrarias y alternativas de resolver el bloqueo). A modo de verificar que \mathbb{S}_1 satisface las condiciones de la Def. 8, analicemos el status de A (el caso de B es análogo). Por un lado, A no está IN dado que no es el caso que todos sus derrotadores estén OUT (particularmente B no posee status). Por otro lado, A no está OUT dado que carece de un derrotador IN. Respecto a las asignaciones \mathbb{S}_2 y \mathbb{S}_3 , el argumento asignado IN en cada caso tiene efectivamente todos sus derrotadores OUT, y el asignado OUT tiene un derrotador IN, cumpliendo así con la Def. 8.

Consideremos ahora una situación algo más compleja involucrando un bloqueo. La Fig. 3.3 muestra un grafo formado por cuatro argumentos, A, B, C y D , donde B defiende a D de la derrota de C , pero a la vez B se encuentra en bloqueo con A . Las asignaciones de status asociadas son $\mathbb{S}_1, \mathbb{S}_2$ y \mathbb{S}_3 , ilustradas en la figura, donde los argumentos A y

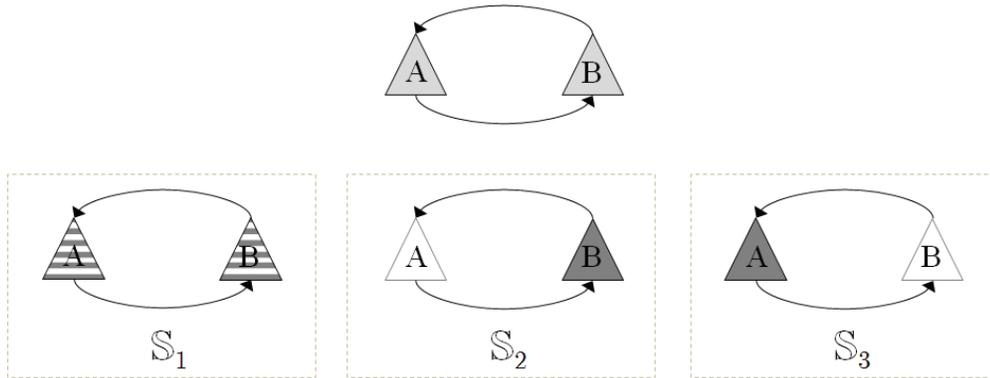


Figura 3.2: Situación de bloqueo y múltiples asignaciones de status

B involucrados en el bloqueo se asignan de acuerdo a las tres posibilidades ilustradas en la Fig. 3.2, determinando en cada caso la asignación para C y D . Nótese que D es reinstaurado en la asignación que favorece a su defensor B sobre A (S_3).

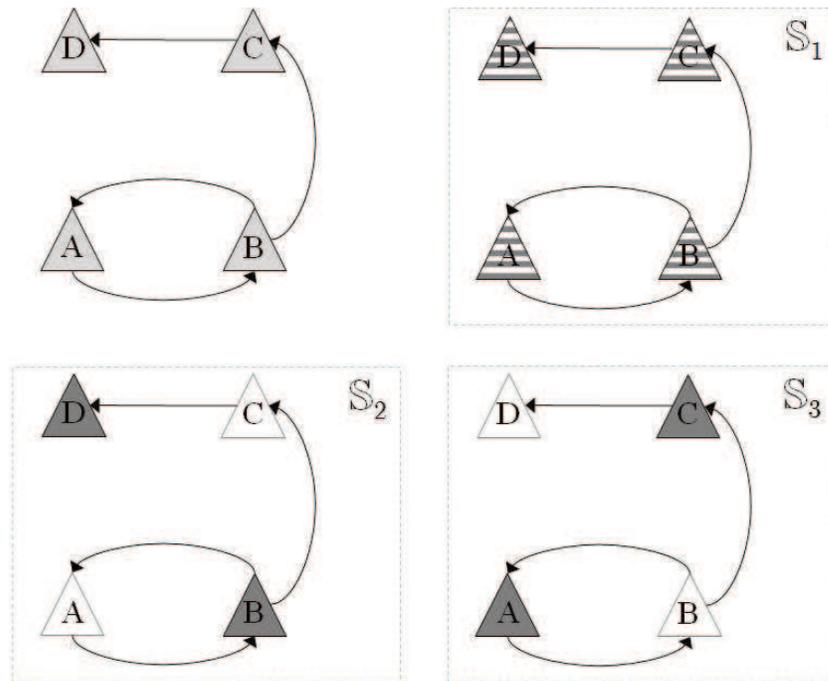


Figura 3.3: Situación involucrando un bloqueo y múltiples asignaciones de status

El efecto de una situación de bloqueo, analizado previamente, se generaliza a cualquier ciclo de longitud par. La Fig. 3.4 muestra las tres asignaciones de status posibles para un ciclo de longitud 4. Nótese que un ciclo par puede verse como definiendo dos bandos (conjuntos) de argumentos (en este caso $\{A, C\}$, y $\{B, D\}$), donde los argumentos de cada

bando se defienden entre ellos de derrotas de argumentos del bando contrario. Las tres posibilidades de asignación surgen entonces de favorecer alternativamente a uno de los dos grupos (\mathbb{S}_2 y \mathbb{S}_3), o considerar todos los argumentos *undecided* (\mathbb{S}_1).

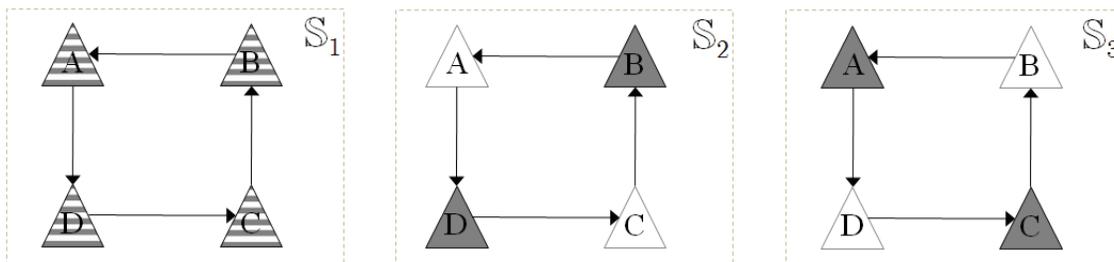


Figura 3.4: Ciclo de longitud par y múltiples asignaciones de status

Analicemos ahora los ciclos de longitud impar. La Fig. 3.5b muestra un grafo que involucra tres argumentos, A , B y C , donde A derrota B , B derrota C y C derrota A (ciclo de longitud 3). Puede verificarse que la única posibilidad de asignación para este grafo, de acuerdo a la Def. 8, consiste en considerar a todos los argumentos como *undecided* (Fig. 3.5b). Veamos que no es posible asignar IN o OUT a A . Por un lado, si consideramos que A se encuentra IN, dado que A derrota a C , C debe estar OUT. Luego, por ser C el único derrotador de B , B estará IN. Dado que B está IN y derrota a A , A estará OUT, arribando a una contradicción. Por otro lado, si consideramos que A se encuentra OUT, por ser el único derrotador de C , C debe estar IN. Luego, dado que C derrota a B , B estará OUT. Por ser C el único derrotador de A , A deberá estar IN, arribando nuevamente a una contradicción. Finalmente, la única posibilidad es considerar a A como *undecided*, y consecuentemente también a B y a C .

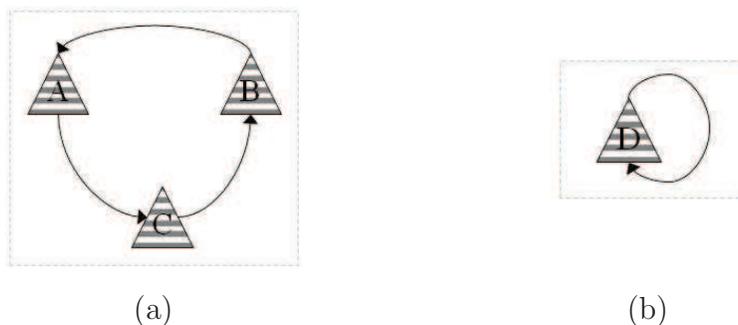


Figura 3.5: Ciclos de longitud impar

Un caso particular de ciclo de longitud impar es el argumento que se derrota a sí mismo (o self-defeating). La Fig. 3.5 ilustra esta situación, junto a la única asignación de status,

considerando el argumento como *undecided*.

Finalmente cabe aclarar que un ciclo en un grafo (ya sea par o impar) no siempre implica indecisión o asignaciones alternativas. La indecisión inherente a un ciclo se verá anulada cuando uno de sus integrantes sufra una derrota de un argumento externo (no involucrado en el ciclo) que se encuentra no derrotado. Las Figs. 3.6a y 3.6b ilustran esta situación para un ciclo par y uno impar, respectivamente, mostrando la única asignación de status posible en cada caso.

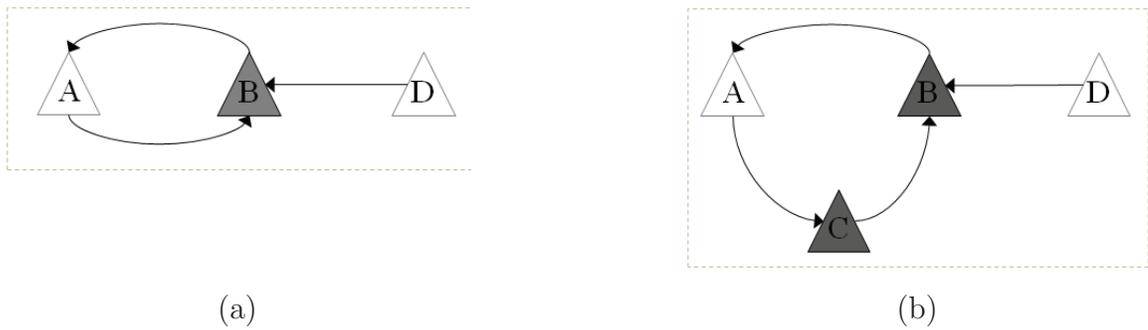


Figura 3.6: Ciclos sin indecisión

Cuando se analizó la situación de bloqueo de la Fig. 3.2, dando lugar a múltiples asignaciones de status alternativas, se habló informalmente de asignación de status “menos comprometida” al referirnos a aquella que considera ambos argumentos como *undecided*, y de asignaciones de status “más comprometidas” al referirnos a aquellas que resuelven el bloqueo arbitrariamente, en una u otra dirección. A continuación se introduce una relación de orden parcial ‘ \prec ’ entre asignaciones de status formalizando esta noción de grado de compromiso relativo entre asignaciones de status.

Definición 9 Sean \mathbb{S}_1 y \mathbb{S}_2 dos asignaciones de status para un AF dado Γ . Diremos que \mathbb{S}_1 es tanto o menos comprometida que \mathbb{S}_2 (o que \mathbb{S}_2 es tanto o más comprometida que \mathbb{S}_1), notado $\mathbb{S}_1 \preceq \mathbb{S}_2$, si $IN_{\mathbb{S}_1} \subseteq IN_{\mathbb{S}_2}$ y $OUT_{\mathbb{S}_1} \subseteq OUT_{\mathbb{S}_2}$. Además notaremos $\mathbb{S}_1 \prec \mathbb{S}_2$ (estrictamente menos comprometido) cuando $\mathbb{S}_1 \preceq \mathbb{S}_2$ pero $\mathbb{S}_2 \not\preceq \mathbb{S}_1$.

Intuitivamente, una asignación \mathbb{S}_2 es más comprometida que otra \mathbb{S}_1 si puede obtenerse a partir de esta última asignando status IN o OUT a argumentos *undecided*. En otras palabras, \mathbb{S}_2 adopta una determinada postura de asignación (arbitraria) frente a conflictos entre argumentos (ciclos pares) considerados *undecided* de acuerdo a \mathbb{S}_1 .

Ejemplo 6 *Considere las asignaciones de status \mathbb{S}_1 , \mathbb{S}_2 y \mathbb{S}_3 de la Fig. 3.3. Entonces vale que $\mathbb{S}_1 \prec \mathbb{S}_2$ y $\mathbb{S}_1 \prec \mathbb{S}_3$. Además vale que $\mathbb{S}_2 \not\prec \mathbb{S}_3$ y $\mathbb{S}_3 \not\prec \mathbb{S}_2$.*

Proposición 1 *Sean \mathbb{S}_1 y \mathbb{S}_2 dos asignaciones de status arbitrarias para un AF dado Γ . Si $\mathbb{S}_1 \preceq \mathbb{S}_2$ y $\mathbb{S}_2 \preceq \mathbb{S}_1$, entonces $\mathbb{S}_1 = \mathbb{S}_2$.*

3.2. Semánticas grounded, preferida y estable

Como se mencionó en la Sec. 1.2, una semántica de aceptabilidad establece cuando considerar a un argumento como aceptado o *garantizado*. En esta sección se presenta una caracterización en términos de asignaciones de status, de las semánticas *grounded*, *estable* y *preferida*, originalmente introducidas por Dung en [Dun95],

Definición 10 (Asignación de status grounded) *Sea \mathbb{S} una asignación de status para un AF dado Γ . Diremos que \mathbb{S} es una asignación de status grounded si $\mathbb{S} \preceq \mathbb{S}'$ para toda otra asignación de status \mathbb{S}' para Γ (es decir, \mathbb{S} es la menor asignación de status con respecto a \preceq .)*

Observación: Se ha probado en [Dun95] que para cualquier AF existe siempre exactamente una extensión grounded, y por lo tanto exactamente una asignación de status grounded asociada (en la Sec. 3.2.2 se prueba la equivalencia entre el enfoque basado en extensiones y el nuestro, basado en asignaciones de status). Nótese además que la unicidad de asignación de status grounded se sigue trivialmente a partir de la definición y de la proposición 1.

Intuitivamente, la asignación de status grounded es la menos comprometida de todas las asignaciones de status, es decir, aquella que realiza la asignación de status a argumentos mínima necesaria para respetar la relación de derrota (*def*), evitando así adoptar postura alguna frente a conflictos de bloqueo (o ciclos pares en general). A modo de ejemplo, los tres grafos de la Fig. 3.1 tienen, cada uno, una única asignación de status, y por lo tanto, grounded. Lo mismo sucede con los grafos de la Fig. 3.5 (ciclos impares), y los de la Fig. 3.6 (ciclos anulados). Para los grafos de las Figs. 3.2, 3.3 y 3.4, todos con múltiples asignaciones de status asociadas, la asignación de status grounded en cada caso es aquella denotada como \mathbb{S}_1 .

A continuación se define cuándo un argumento es considerado aceptado o *garantizado* de acuerdo a la semántica grounded. Asimismo, se definen dos categorías de argumentos no garantizados: los *rechazados*, que representan argumentos decididamente no aceptados, y los *discutibles*, manifestando indecisión.

Definición 11 (Semántica Grounded) Sea $\Gamma = (Args, def)$ un AF. Diremos que, de acuerdo a la semántica grounded, un argumento $A \in Args$ es un argumento

- *garantizado sssi es asignado IN por la asignación de status grounded para Γ .*
- *rechazado sssi es asignado OUT por la asignación de status grounded para Γ .*
- *discutible sssi no se encuentra ni garantizado ni rechazado.*

A continuación se presenta la semántica *estable*. A diferencia de la semántica grounded, que se define en base a la asignación de status con menor grado de compromiso, la semántica estable considera las distintas alternativas de asignación “totalmente” comprometidas (*i.e.*, sin argumentos *undecided*), y que surgen de asumir la resolución de cada situación de bloqueo en una u otra dirección, estableciendo a un argumento como garantizado cuando se encuentra IN en todas las posibles asignaciones.

Definición 12 (Asignación de status estable) Sea \mathbb{S} una asignación de status para un AF dado Γ . Diremos que \mathbb{S} es una asignación de status estable si $IN_{\mathbb{S}} \cup OUT_{\mathbb{S}} = Args$ (*i.e.*, \mathbb{S} es “totalmente comprometida”).

Definición 13 (Semántica Estable) Sea $\Gamma = (Args, def)$ un AF. Diremos que, de acuerdo a la semántica estable, un argumento $A \in Args$ es un argumento

- *garantizado sssi es asignado IN por todas la asignaciones de status estables para Γ ,*
- *rechazado sssi es asignado OUT por todas las asignaciones de status estables para Γ , y*
- *discutible sssi no se encuentra ni garantizado ni rechazado,*

Los tres grafos de la Fig. 3.1 tienen, cada uno, una única asignación de status, que es a la vez grounded y estable. Obviamente ambas semánticas coinciden para estos grafos, definiendo los mismos conjuntos de argumentos garantizados y rechazados. El grafo de la Fig. 3.2 tiene dos asignaciones de status estables, S_2 y S_3 . Dado que tanto A como B son asignados IN en una de las las asignaciones y OUT en la otra, ambos son considerados discutibles por la semántica estable, coincidiendo nuevamente con la grounded. Algo similar ocurre con el grafo de la Fig. 3.3, donde todos los argumentos son considerados discutibles.

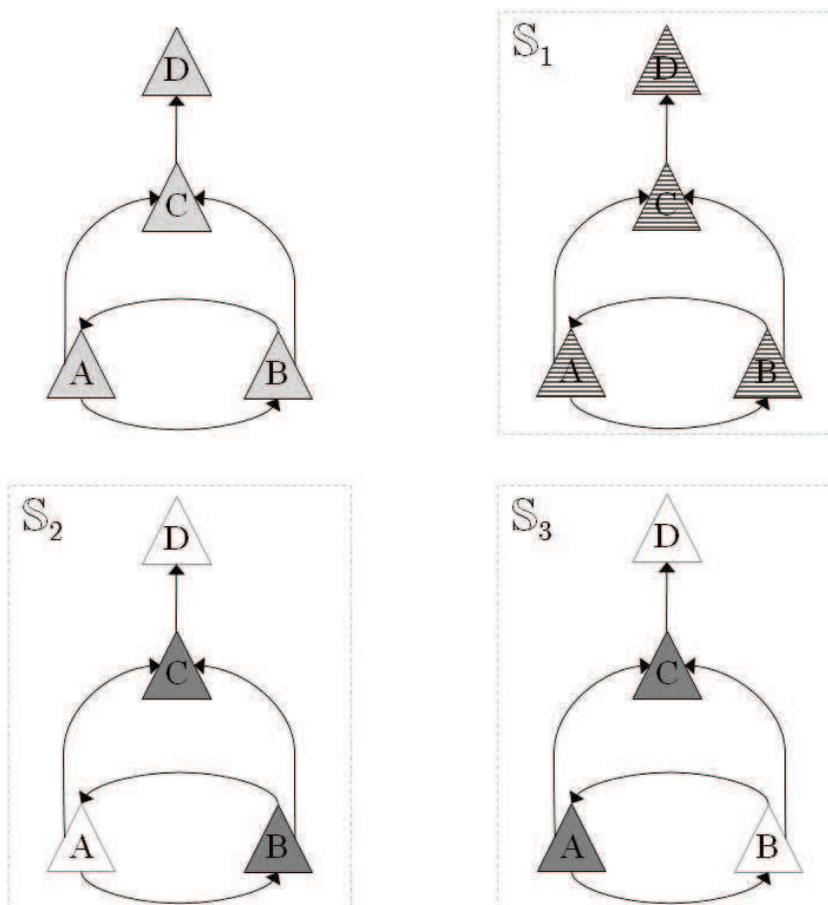


Figura 3.7: Argumentos flotantes

Finalmente el grafo de la Fig. 3.7 ilustra una situación identificada en la literatura como *argumentos flotantes*, donde las semánticas grounded y estable difieren. De acuerdo a la semántica grounded (determinada por la asignación grounded S_1), todos los argumentos son discutibles. La indecisión general en S_1 es consecuencia de la situación de bloqueo entre A y B , que afecta directa o indirectamente a todos los argumentos. Por otro lado, existen

dos asignaciones estables, \mathbb{S}_2 y \mathbb{S}_3 , resultado de resolver el bloqueo alternativamente en favor de A o B . Aunque \mathbb{S}_2 y \mathbb{S}_3 difieren en el status asignado a A y B , coinciden en asignar OUT a C e IN a D , quedando C rechazado y D garantizado de acuerdo a la semántica estable.

La principal limitación de la semántica estable tiene que ver con los ciclos impares: dado que generalmente un ciclo impar solo admite que sus miembros sean considerados *undecided*, un grafo conteniendo un ciclo impar podría no tener asignación de status estable, provocando que en estos casos la semántica no provea respuestas (o provea respuestas sin sentido, de acuerdo a como se enunció la Def. 8). Esta falencia resulta más evidente en casos como el de la Fig. 3.8, donde un ciclo impar ocasiona la inexistencia de asignaciones estables (y la consecuente ausencia de respuesta), aún cuando existen argumentos que el ciclo no afecta.

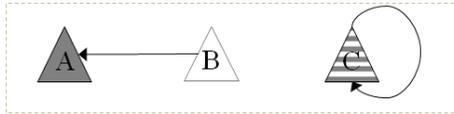


Figura 3.8: Única asignación de status, no estable.

Además de la ausencia de respuestas, en ocasiones la semántica estable puede brindar respuestas no “seguras” (*i.e.*, que reflejan una tendencia o preferencia arbitraria) como resultado de la interacción de ciclos impares con ciclos pares. La Fig. 3.9 muestra las tres asignaciones de status asociadas a un grafo donde un argumento auto-derrotado C es derrotado por un argumento A que a su vez se encuentra involucrado en un bloqueo con otro argumento B . Es claro que A y B deberían considerarse discutibles dado que se derrotan mutuamente y no son afectados por ningún otro argumento del grafo (de hecho así ocurre de acuerdo a la semántica grounded). Sin embargo, por ser \mathbb{S}_2 la única asignación de status estable, de acuerdo a esta semántica A se considera garantizado y B rechazado.

A continuación se presenta la semántica *preferida*, que soluciona las limitaciones de la estable relajando el requerimiento de considerar asignaciones de status *totalmente* comprometidas a considerar asignaciones de status *maximalmente* comprometidas, permitiéndole de esta forma abordar situaciones con argumentos imposibles de asignar (ciclos impares).

Definición 14 (Asignación de status preferida) Sea \mathbb{S} una asignación de status para un AF dado Γ . Diremos que \mathbb{S} es una asignación de status preferida si \mathbb{S} es maximal con

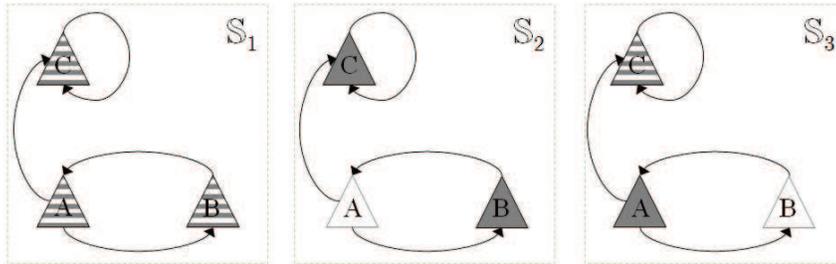


Figura 3.9: Única asignación estable: \mathbb{S}_2 .

respecto a \preceq (o maximalmente comprometida), *i.e.*, si no existe asignación de status \mathbb{S}' tal que $\mathbb{S}' \succ \mathbb{S}$.

Definición 15 (Semántica Preferida) Sea $\Gamma = (\text{Args}, \text{def})$ un AF. Diremos que, de acuerdo a la semántica preferida, un argumento $A \in \text{Args}$ es un argumento

- garantizado *ssi* es asignado *IN* por todas las asignaciones de status preferidas para Γ .
- rechazado *ssi* es asignado *OUT* por todas las asignaciones de status preferidas para Γ .
- discutible *ssi* no se encuentra ni garantizado ni rechazado.

La asignación de status mostrada en la Fig. 3.8 (única para el grafo, y no estable) es preferida, dado que no existe otra asignación posible más comprometida (*i.e.*, es maximalmente comprometida). Luego, de acuerdo a la semántica preferida el argumento A se considera rechazado, B garantizado y C discutible.

Considérese el caso de la Fig. 3.9. Aunque solo la asignación \mathbb{S}_2 es estable, tanto \mathbb{S}_2 como \mathbb{S}_3 son preferidas, evitando así que prevalezca una de las posibilidades de resolución del bloqueo y se brinden respuestas “no seguras”: los tres argumentos se consideran discutible de acuerdo a la semántica preferida.

Finalmente, para la situación de argumentos flotantes (Fig. 3.7) las semánticas preferida y estable coinciden, difiriendo de la grounded.

3.2.1. Análisis comparativo y propiedades de las semánticas

A continuación se listan algunas propiedades referidas a las semánticas de aceptabilidad introducidas en la sección anterior, enunciadas y demostradas originalmente por Dung en [Dun95]². La presentación de estas propiedades es a modo de ganar una mayor intuición.

- Todo AF posee al menos una asignación de status.
- Todo AF posee asignación de status grounded (semántica universalmente definida)
- Todo AF posee al menos una asignación de status preferida (semántica universalmente definida)
- Existen AF s para los que no existe asignación de status estable (semántica no universalmente definida)
- La asignación de status grounded es siempre menor (con respecto a ‘ \preceq ’) que todas las asignaciones de status preferidas y estables (por definición, la asignación de status grounded es la menor de todas las asignaciones de status).
- Toda asignación de status estable es preferida, pero no vice versa. Es claro que una asignación de status ‘totalmente comprometida’ (*i.e.*, que asigna IN o OUT a todos los argumentos) es en particular ‘maximalmente comprometida’. La recíproca no vale, consecuencia de los ciclos impares, imposibles de asignar (como ilustra el caso de la Fig. 3.8).
- Todo AF *well-founded* (*i.e.*, tal que no hay caminos infinitos ni ciclos en el grafo de derrotas asociado) posee exactamente una asignación de status, que es grounded, preferida y estable. Nótese que los ciclos de longitud par y los caminos infinitos son los causantes de la multiplicidad de asignaciones de status, y que los ciclos impares son los responsables de la inexistencia de asignación de status estable.

²Aunque la caracterización de las semánticas grounded, estable y preferida presentada por Dung en [Dun95] se basa en la noción de extensión, existe una correspondencia directa con el enfoque basado en asignaciones de status (como se muestra en la sección 3.2.2), y por lo tanto las propiedades enunciadas originalmente en [Dun95] se trasladan directamente.

- Todo argumento garantizado de acuerdo a la semántica *grounded* está garantizado de acuerdo a la preferida. La recíproca no vale, como ilustra el caso de la Fig. 3.7 (argumentos flotantes).

A pesar de que tanto la semántica *grounded* como la preferida definen la noción de argumento discutible, la preferida resulta más conveniente para identificar conjuntos compatibles de argumentos, es decir, que pueden asignarse **IN** simultáneamente. Mientras que de acuerdo a la semántica *grounded* un argumento es considerado discutible en un sentido individual, la semántica preferida considera a un argumento como discutible por existir una asignación de status (maximalmente comprometida) que lo considera **IN** junto a otros argumentos. Esto resulta de especial importancia si se tiene en cuenta que aunque dos argumentos discutibles no se relacionen directamente mediante la relación de derrota, podrían de todas formas ser incompatibles en el sentido de que no exista posibilidad de considerarlos ambos **IN**. El caso de la Fig. 3.3 ilustra esta situación: todos los argumentos son considerados discutibles de acuerdo a ambas semánticas; sin embargo, al considerar las asignaciones de status preferidas se evidencia una incompatibilidad entre los argumentos *A* y *D*, a pesar de no estar directamente relacionados mediante la noción de derrota. Esta incompatibilidad es difícil de vislumbrar a partir de la única asignación de status *grounded*.

3.2.2. Equivalencia con el enfoque basado en extensiones

Las semánticas *grounded*, *estable* y *preferida* fueron introducidas por primera vez por Dung en [Dun95]. A diferencia de la caracterización presentada en la sección anterior, que se basa en la noción de asignación de status, las definiciones presentadas por Dung se basan en la noción de extensión (extension-based approach). En un sentido general, una extensión *E* para un *AF* dado (*Args*, *def*) es un subconjunto de argumentos de *Args* ($E \subseteq \text{Args}$) considerados “colectivamente aceptables” teniendo en cuenta la relación de derrota *def*. De esta forma, de acuerdo al enfoque basado en extensiones (y en contraste con el adoptado en la sección anterior) una semántica se define en términos de un conjunto de extensiones (en lugar de asignaciones de status) satisfaciendo ciertos requerimientos. El propósito de esta sección es hacer explícita la equivalencia entre las dos versiones.

En primer lugar se presentará la noción de *extensión completa*, que subyace a todas las semánticas propuestas por Dung, y que se basa en las nociones de libertad (o ausencia) de conflicto y defensa presentadas a continuación. Sea $S \subseteq Args$, diremos que:

- S es *libre de conflicto* (conflict-free) si no existen $A, B \in S$ tal que $A \text{ def } B$.
- S *defiende* a un argumento $A \in Args$ si todos los derrotadores de A son derrotados por argumentos de S . Formalmente: $\forall B \in Args$ tq. $B \text{ def } A$ vale que $\exists C \in S$ tq. $C \text{ def } B$.

Definición 16 *Un conjunto $E \subseteq Args$ constituye una extensión completa si es libre de conflicto y vale que $A \in E$ sssi A es defendido por E .*

Informalmente, una extensión completa es un conjunto de argumentos libres de conflictos, donde cada argumento del conjunto es “defendido” de sus derrotadores por argumentos del mismo conjunto (conjunto auto-defendido – $A \in E$ solo si A es defendido por E), y tal que incluye a todo argumento defendido por argumentos del conjunto ($A \in E$ si A es defendido por E). En base a la noción de extensión completa se definen las extensiones grounded, estables y preferidas.

Definición 17 *Un conjunto E constituye una extensión*

- *grounded*, si es la menor (con respecto a \subseteq) de las extensiones completas.
- *preferida*, si es maximal (con respecto a \subseteq) tal que es completa.
- *estable*, si es completa y derrota a todos los argumentos de $Args$ que no pertenecen a E ($A \notin E$ entonces $\exists B \in E$ tq. $B \text{ def } A$).

La siguiente proposición establece la equivalencia entre las nociones de *asignación de status* y *extensión completa*, pilares para las definiciones de las semánticas grounded, estable y preferida de acuerdo a los enfoques basado en etiquetas y en extensiones, respectivamente.

Proposición 2 *Sea $\Gamma = (Args, \text{def})$ un AF, y sea $E \subseteq Args$. E es una extensión completa sssi existe una asignación de status \mathbb{S} tal que $E = IN_{\mathbb{S}}$.*

Prueba.

(\leftarrow) Sea \mathbb{S} una asignación de status. Verifiquemos que $\text{IN}_{\mathbb{S}}$ es una extensión completa, es decir, que 1) $\text{IN}_{\mathbb{S}}$ es libre de conflicto y 2) $A \in \text{IN}_{\mathbb{S}}$ sssi A es defendido por $\text{IN}_{\mathbb{S}}$. 1) $\text{IN}_{\mathbb{S}}$ es libre de conflicto, dado que si $A \in \text{IN}_{\mathbb{S}}$ entonces para todo B tal que $B \text{ def } A$ vale que $B \in \text{OUT}_{\mathbb{S}}$ (primer item, Def. 8). 2) $A \in \text{IN}_{\mathbb{S}}$ sssi para todo B tq. $B \text{ def } A$ vale que $B \in \text{OUT}_{\mathbb{S}}$ (primer item Def. 8), y $B \in \text{OUT}_{\mathbb{S}}$ sssi existe un $C \in \text{IN}_{\mathbb{S}}$ tq. $C \text{ def } B$ (segundo item de la Def. 8).

(\rightarrow) Sea E una extensión completa. Considérese el conjunto $E' \subseteq \text{Args}$ de todos los argumentos derrotados por argumentos en E . Probaremos que la asignación de IN exactamente a los argumentos de E y OUT exactamente a los de E' constituye efectivamente una asignación de status de acuerdo a la Def. 8. Dado que E es libre de conflicto, $E \cap E' = \emptyset$, lo que indica que no estaremos asignando IN y OUT a un mismo argumento. Verifiquemos ahora que la asignación considerada satisface las condiciones de los dos items de la Def. 8:

- (item 1) A se encuentra IN sssi $A \in E$ sssi A es defendido por E , sssi para todo B tq. $B \text{ def } A$ vale que existe $C \in E$ tal que $C \text{ def } B$ sssi para todo B tq. $B \text{ def } A$, $B \in E'$ sssi para todo B tq. $B \text{ def } A$, B se encuentra OUT .
- (item 2) A se encuentra OUT sssi $A \in E'$ sssi existe $B \in E$ tq. $B \text{ def } A$ sssi existe B tal que B está IN y $B \text{ def } A$.

Establecida la correspondencia entre las nociones de asignación de status y extensión completa, resulta directo mostrar la equivalencia entre los distintos tipos de asignaciones de status (grounded, estable y preferida) y los correspondientes tipos de extensiones.

Proposición 3 *Sea $\Gamma = (\text{Args}, \text{def})$ un AF , y sea $E \subseteq \text{Args}$. E es una extensión grounded [estable/preferida] sssi existe una asignación de status grounded [estable/preferida] \mathbb{S} tal que $E = \text{IN}_{\mathbb{S}}$.*

Prueba. Puede verse fácilmente que, en el marco de la equivalencia entre asignaciones de status y extensiones completas, la relación ' \preceq ' (tanto o menos comprometida que) entre asignaciones de status se corresponde a la relación ' \subseteq ' (inclusión) entre extensiones. Formalmente, $\mathbb{S}_1 \preceq \mathbb{S}_2$ sssi $\text{IN}_{\mathbb{S}_1} \subseteq \text{IN}_{\mathbb{S}_2}$. Luego, E es la menor (con respecto a \subseteq) de las extensiones completas sssi existe asignación de status \mathbb{S} tal que $E = \text{IN}_{\mathbb{S}}$ y \mathbb{S} es la

menor (con respecto a ‘ \preceq ’) de las asignaciones de status (equivalencia para la semántica grounded). Además, E es completa maximal (con respecto a \subseteq) sssi existe asignación de status \mathbb{S} tal que $E = \text{IN}_{\mathbb{S}}$ y \mathbb{S} es maximal (con respecto a ‘ \preceq ’) (equivalencia para la semántica preferida). Finalmente, E es completa tal que si $A \notin E$ entonces existe $B \in E$ tal que $B \text{ def } A$ (extensión estable) sssi existe asignación de status \mathbb{S} tal que $E = \text{IN}_{\mathbb{S}}$ donde si $A \notin \text{IN}_{\mathbb{S}}$ entonces existe $B \in \text{IN}_{\mathbb{S}}$ tal que $B \text{ def } A$ sssi existe asignación de status \mathbb{S} tal que $E = \text{IN}_{\mathbb{S}}$ donde si $A \notin \text{IN}_{\mathbb{S}}$ entonces $B \in \text{OUT}_{\mathbb{S}}$ (equivalencia para la semántica estable).

3.2.3. Caracterización de la Semántica Grounded basada en la noción de Punto Fijo

La definición de semántica grounded propuesta originalmente por Dung en [Dun95] se basa en la noción de punto fijo para una función denominada función característica, definida a continuación:

Definición 18 Sea $\Gamma = (\text{Args}, \text{def})$ un AF. La función característica de Γ , denotada F_{Γ} , se define como sigue:

$$F_{\Gamma}(S) : 2^{\text{Args}} \rightarrow 2^{\text{Args}},$$

$$F_{\Gamma}(S) =_{\text{def}} \{A \in \text{Args} \mid A \text{ es defendido por } S\}.$$

Observación: Dado que siempre nos referimos a un AF arbitrario pero fijo Γ , a menudo usaremos F en lugar de F_{Γ} , a modo de abreviación.

Definición 19 (extensión grounded, [Dun95]) La extensión grounded de un AF Γ es el menor punto fijo (con respecto a \subseteq) de su función característica F_{Γ} .

La idea intuitiva que subyace a esta definición es la siguiente: la extensión grounded (única) puede construirse incrementalmente comenzando con el conjunto de aquellos argumentos que no tienen derrotadores ($F(\emptyset)$), agregando luego aquellos argumentos defendidos por los que no tienen derrotadores ($F^2(\emptyset)$), y así siguiendo. Este proceso termina cuando $F^i(\emptyset) = F^{i+1}(\emptyset)$, es decir, se alcanza un punto fijo.

Dung demostró que si la cantidad de argumentos de AF es finita, entonces $F^i(\emptyset) = F^{i+1}(\emptyset)$ para algún i , y en ese caso $F^i(\emptyset)$ coincide con el menor punto fijo de F (extensión grounded). La siguiente proposición, enunciada en [Dun95], establece como obtener el menor punto fijo (extensión grounded) para el caso más general de AF finitarios, *i.e.*, donde cada argumento tiene una cantidad finita de derrotadores (aunque la cantidad de argumentos puede ser infinita).

Proposición 4 *Sea Γ un AF finitario. Entonces la extensión grounded E asociada a Γ coincide con $\bigcup_{i=0}^{\infty} F_{\Gamma}^i(\emptyset)$.*

Dung establece la equivalencia entre las dos definiciones de extensión grounded presentadas (Defs. 17 y 19), probando que un conjunto de argumentos E libre de conflicto constituye una extensión completa sssi E es un punto fijo de F ($E = F(E)$). Nótese que de acuerdo a la definición de la función característica, $F(S)$ es el conjunto de argumentos defendidos por S , y por lo tanto $E \subseteq F(E)$ equivale a decir que todos los elementos de E son defendidos por E (auto-defensa), y $E \supseteq F(E)$, a decir que E contiene a todos los elementos que defiende.

3.3. Procedimiento de Prueba Dialéctico para la Semántica Grounded

En esta sección se describe un procedimiento de prueba para la semántica grounded basado en el enfoque dialéctico, o de juegos argumentales (argument games approach). El enfoque dialéctico es ampliamente adoptado en la literatura en la caracterización de procedimientos de prueba para la argumentación, destacándose los trabajos de [CDM03, DMT07, DBC03, JV99, Vre93a, VP00]. En general, un procedimiento de prueba dialéctico consiste en la elaboración (o simulación) de una discusión entre dos personas o jugadores ficticios, llamados PRO (por proponente) y OPP (por oponente, o contrincante), donde PRO comienza presentando el argumento A a ser testeado, y luego tanto OPP como PRO van presentando sucesivamente argumentos en “respuesta” (es decir, atacando) a argumentos presentados previamente por el contrincante. Finalmente, se determinará que el argumento inicial A está aceptado cuando PRO resulte vencedor en una discusión exhaustiva acerca

de A (*i.e.*, una discusión donde todos los argumentos defendiendo y atacando directa e indirectamente a A fueron contemplados).

Existen procedimientos de prueba dialécticos para diferentes semánticas de Dung [MC09, TDH09], y todos ellos resultan de imponer distintas restricciones (dependiendo de la semántica) sobre las movidas que PRO y OPP pueden realizar al entablar la discusión. Aquí nos concentraremos en la semántica grounded. A continuación se introducen una serie de definiciones con el propósito de formalizar esta noción de discusión, y de la evaluación de una discusión para determinar su resultado, en el contexto de la semántica grounded.

En primer lugar se presenta la noción formal de línea dialéctica. Una línea dialéctica representa una “línea de discusión” entre PRO y OPP acerca de un argumento dado A . Concretamente, es una secuencia de argumentos (comenzando con A) donde cada uno derrota al previo en la línea, y tal que cada argumento que aparece en una posición impar (entre ellos A) se considera como presentado por PRO (y por lo tanto, en favor de A), y cada argumento en una posición par se considera presentado por OPP (y por lo tanto, en contra de A).

Definición 20 (Línea Dialéctica) *Sea $(Args, def)$ un AF y $A \in Args$. Una línea dialéctica acerca de A (o simplemente línea dialéctica) es una secuencia finita de argumentos $[A_1, A_2, \dots, A_n]$, con $A_1 = A$, tal que:*

(d1) A_i def A_{i-1} , $1 < i \leq n$,

(d2) si A_i y A_j son argumentos PRO, $i \neq j$, entonces $A_i \neq A_j$,

donde A_i se dice que es un argumento PRO de la línea si i es impar, o un argumento OPP de la línea si i es par.

La condición d1 establece que cada argumento en la línea (excepto el primero) derrota al previo. La condición d2 evita la repetición de argumentos PRO en la línea, y constituye una restricción específica para capturar la semántica grounded, como se explica a continuación.

Intuitivamente, una línea dialéctica acerca de un argumento dado A terminada con un argumento PRO representa una “línea de discusión” ganada por PRO, y por lo tanto

defendiendo la garantía de A , mientras que una terminada con un argumento OPP representa una línea de discusión ganada por OPP, y por lo tanto, que no logra defender A . Sin embargo, un ciclo en el grafo de argumentos podría conducir a una línea de discusión infinita, y por lo tanto denotando indecisión acerca de su resultado (defendiendo o no defendiendo el argumento inicial). Como vimos en la Sec. 3.2, la semántica grounded adopta la postura de “menor compromiso” frente a ciclos en el grafo, considerando a los argumentos afectados por un ciclo como no garantizados. Teniendo esto en cuenta, la condición d2 puede considerarse como imponiendo esta postura de “no compromiso” frente a ciclos en el grafo de derrotas: evitando la repetición de argumentos PRO y a la vez permitiendo la repetición de argumentos OPP, una línea de discusión infinita consecuencia de un ciclo dará lugar a una línea dialéctica finita terminada con un argumento OPP, y por lo tanto, que no logra defender (la garantía de) el argumento inicial.

Dado que queremos modelar una discusión exhaustiva acerca de A , nos interesa capturar líneas de discusión exhaustivas, es decir, donde el último argumento de la línea, propuesto por uno de los jugadores, no puede ser respondido por su contrincante.

Definición 21 (Línea Dialéctica Exhaustiva) *Sea $\Lambda = [A_1, A_2, \dots, A_n]$ una línea dialéctica. Diremos que Λ es exhaustiva si no existe argumento B tal que $[A_1, A_2, \dots, A_n, B]$ es una línea dialéctica.*

El conjunto de todas las líneas dialécticas exhaustivas acerca de un argumento dado A modela una discusión exhaustiva acerca de A , y puede ser representada como un árbol, donde cada línea corresponde a una rama en el árbol.

Definición 22 (Árbol Dialéctico) *Sea $A \in \text{Args}$. Un árbol dialéctico acerca de A , denotado T_A , se define como sigue:*

1. *Los nodos se etiquetan con argumentos de Args .*
2. *$\Lambda = [A_1, A_2, \dots, A_n]$ es la secuencia de etiquetas de una rama del árbol (camino desde la raíz hasta una hoja) sssi Λ es una línea dialéctica exhaustiva acerca de A .*
3. *No existen nodos hermanos (hijos del mismo nodo) M y M' en el árbol etiquetados con el mismo argumento.*

El propósito de la condición 3 es factorizar tanto como se pueda en el árbol los prefijos comunes de las líneas dialécticas acerca de A .

Finalmente, la evaluación de una discusión se formaliza mediante la noción de árbol dialéctico evaluado, presentada a continuación, donde los nodos de un árbol dialéctico se van marcando como ‘D’ (defeated) o ‘U’ (undefeated), comenzando por las hojas (trivialmente no derrotadas) y propagando el marcado hasta el argumento inicial, determinando así el resultado final de la discusión.

Definición 23 (Árbol Dialéctico Evaluado) *Sea T_A una árbol dialéctico. El correspondiente árbol dialéctico evaluado, denotado T_A^* , se obtiene marcando cada nodo N en T_A como ‘U’ (undefeated) o ‘D’ (defeated) de acuerdo a las siguientes reglas:*

1. *si N es una hoja, entonces N se marca como ‘U’.*
2. *si todos los hijos M_i de N están marcados como ‘D’, entonces N se marca como ‘U’.*
3. *si existe un hijo M_i de N marcado como ‘U’, entonces N se marca como ‘D’.*

La siguiente proposición establece formalmente la conexión entre el procedimiento de prueba dialéctico presentado y la noción de argumento garantizado de acuerdo a la semántica grounded.

Proposición 5 *Sea $\Gamma = (Args, def)$ un AF, y sea $A \in Args$. A está garantizado de acuerdo a la semántica grounded sssi A está marcado ‘U’ en T_A^* .*

Prueba. (\leftarrow) Sea $GE \subseteq Args$ el conjunto de todos los argumentos garantizados de acuerdo a la semántica grounded (o extensión grounded). La estrategia consiste en probar que cada argumento PRO X marcado ‘U’ en T_A^* (en particular la raíz A) pertenece a GE haciendo inducción sobre la altura n del subárbol de T_A^* con raíz X .

caso base ($n = 0$): X es hoja en T_A^* . Dado que no existen restricciones para OPP en una línea dialéctica, puede asegurarse que X no tiene derrotadores en $Args$ (de lo contrario aparecerían como hijos OPP de X), y por lo tanto $X \in GE$.

caso inductivo ($n > 0$): dado que X está marcado con U, para cada hijo Y de X (marcado con D) existe un hijo Z de Y marcado con U. Dado que el subárbol con raíz Z tiene

altura $n' < n$, por hipótesis inductiva, $Z \in GE$. Luego, GE defiende a X de todos sus derrotadores, y por completitud de GE , vale que $X \in GE$.

(\rightarrow) Se plantean dos casos:

caso 1) A no tiene derrotadores, y entonces trivialmente A estará marcado con U en T_A^* .

caso 2) (A tiene derrotadores). Para este caso se apelará a la propiedad de admisibilidad, satisfecha por GE : para cada $Y \in Args$ tal que $Y \text{ def } X$, existe un $Z \in GE$ tal que $Z \text{ def } Y$ (intuitivamente, GE defiende a sus elementos de derrotas externas). Esto implica que para cada hijo OPP Y de A en T_A^* existirá un hijo PRO Z de Y , tal que $Z \in GE$. Luego si probáramos que Z está U en T_A^* , de acuerdo a las reglas de marcado, A se marcará con U . Pero el razonamiento para Z es análogo al razonamiento para A (casos 1 y 2). Dado que T_A^* tiene altura finita (por la condición d2 de la Def. 20), eventualmente se llegará a las hojas (contempladas por caso 1), para las cuales se puede asegurar trivialmente que están marcadas con U , lo que confirma el marcado U de los nodos PRO considerados en el análisis, en particular A \square

La siguiente proposición establece formalmente la estrategia para determinar si un argumento dado se encuentra rechazado de acuerdo a la semántica grounded.

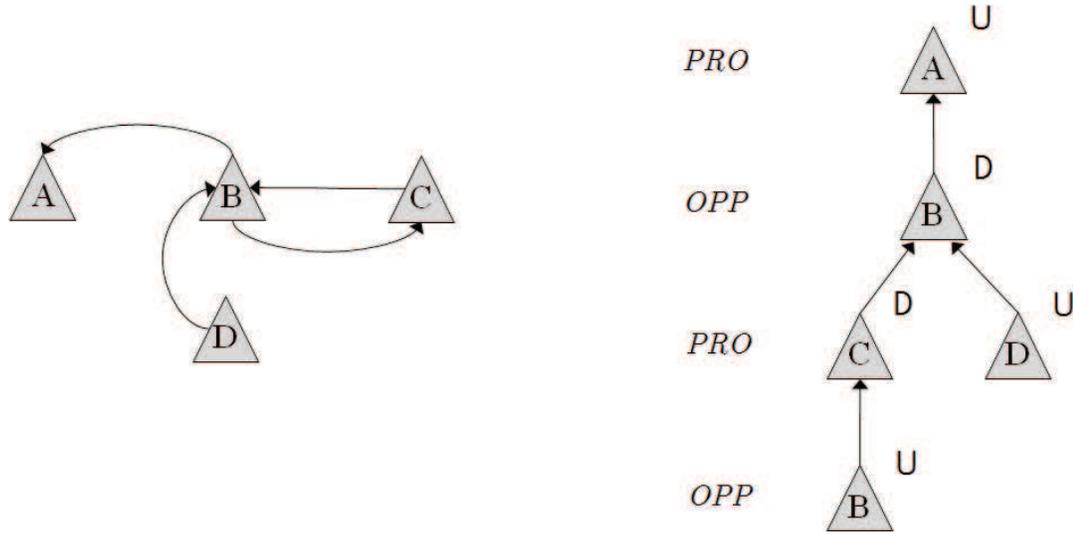
Proposición 6 *Sea $\Gamma = (Args, \text{def})$ un AF, y sea $A \in Args$. A está rechazado de acuerdo a la semántica grounded sssi existe $B \in Args$ tal que $B \text{ def } A$ y B está marcado ‘U’ en T_B^* ³.*

Finalmente para determinar si un argumento es discutible basta con verificar que no sea garantizado ni rechazado.

La Fig. 3.10 muestra un grafo de derrotas involucrando un argumento A , junto al árbol dialéctico evaluado para A , de acuerdo al cual A se encuentra marcado con U , y por lo tanto garantizado de acuerdo a la semántica grounded. Nótese que el árbol efectivamente representa todas las líneas dialécticas exhaustivas acerca de A . Aunque a partir de la línea dialéctica $[A, B, C, B]$ (denotando un ciclo) PRO no logra defender a A de la derrota de B , finalmente consigue hacerlo al introducir a D como derrotador para B (línea $[A, B, D]$).

Considérese ahora el grafo que resulta de eliminar el argumento D del grafo anterior (Fig. 3.11a). En este caso, el árbol dialéctico para A (Fig. 3.11b) está formado por una única línea dialéctica exhaustiva, donde A queda marcado como D , indicando que

³La demostración es trivial a partir de la definición de asignación de status y la proposición 5.


 Figura 3.10: Árbol Dialéctico Evaluado: argumento A garantizado

no está garantizado. Finalmente, dado que el único derrotador de A (argumento B) no está garantizado (árbol dialéctico evaluado de la Fig. 3.11c) entonces A tampoco se encuentra rechazado, y por lo tanto es discutible.

3.3.1. Refinamiento del procedimiento dialéctico considerando subargumentos

El procedimiento dialéctico presentado previamente se define para el framework de argumentación abstracto AF propuesto por Dung. En esta sección, el AF se instancia con el lenguaje y las nociones de argumento, ataque y derrota de DeLP (Cap. 2), y se presenta un refinamiento del procedimiento dialéctico resultante para tomar ventaja de la noción de subargumento de desacuerdo asociado a una derrota. Esta versión refinada será usada como base para definir el procedimiento de prueba dialéctico para la semántica a-grounded, contemplando agregación (Cap. 6).

Definición 24 (Línea Dialéctica, refinada) Sea \mathcal{P} un programa DeLP. Sea $\Gamma_{\mathcal{P}} = (Args, def)$, donde $Args$ es el conjunto de todos los argumentos DeLP construibles a partir de \mathcal{P} , y para cualquier par de argumentos $\langle \mathcal{A}_1, h_1 \rangle, \langle \mathcal{A}_2, h_2 \rangle \in Args$, $\langle \mathcal{A}_1, h_1 \rangle def \langle \mathcal{A}_2, h_2 \rangle$ sssi $\langle \mathcal{A}_1, h_1 \rangle$ derrota $\langle \mathcal{A}_2, h_2 \rangle$ (Def. 7). Sea $\langle \mathcal{A}, h \rangle \in Args$. Una línea dialéctica acerca de $\langle \mathcal{A}, h \rangle$ (o simplemente línea dialéctica) es una secuencia finita de argumentos $[\langle \mathcal{A}_1, h_1 \rangle, \langle \mathcal{A}_2, h_2 \rangle, \dots, \langle \mathcal{A}_n, h_n \rangle]$, con $\langle \mathcal{A}_1, h_1 \rangle = \langle \mathcal{A}, h \rangle$, tal que:

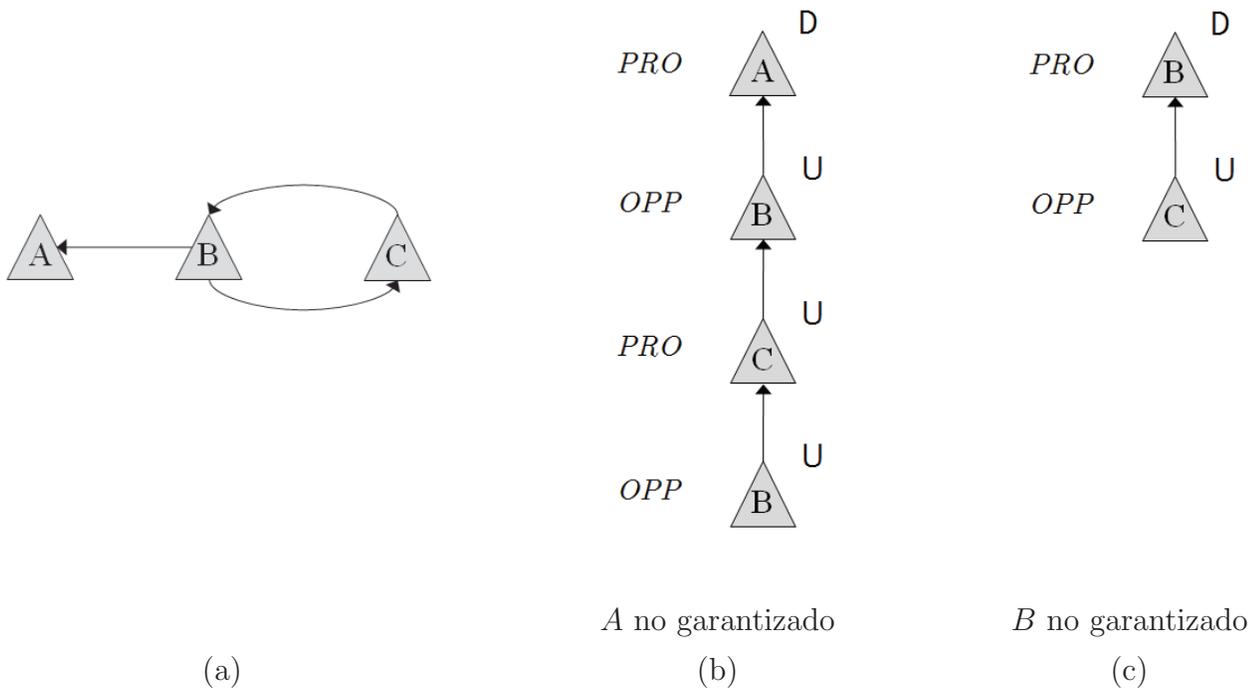


Figura 3.11: *A* discutible de acuerdo a la semántica grounded

(d1') $\langle \mathcal{A}_i, h_i \rangle$ def $\langle \mathcal{A}_{i-1}, h_{i-1} \rangle$, $1 < i \leq n$,

(d2') si $\langle \mathcal{A}_i, h_i \rangle$ es PRO y Σ es el conjunto de todos los subargumentos de desacuerdo asociados con los ataques contra argumentos PRO apareciendo antes de $\langle \mathcal{A}_i, h_i \rangle$ en la línea, entonces ningún argumento en Σ es un subargumento de $\langle \mathcal{A}_i, h_i \rangle$,

donde $\langle \mathcal{A}_i, h_i \rangle$ se dice que es un argumento PRO de la línea si i es impar, o un argumento OPP de la línea si i es par.

La condición d1' es simplemente una instancia de la condición d1 en la Def. 20. La condición d2' evita la ocurrencia de un argumento PRO $\langle \mathcal{A}_i, h_i \rangle$ si este involucra el subargumento de desacuerdo de un argumento PRO $\langle \mathcal{A}_j, h_j \rangle$ apareciendo previamente en la línea ($j < i$). La Fig. 3.12 muestra un esquema ilustrando el efecto de la condición d2'. Nótese que si tal argumento $\langle \mathcal{A}_i, h_i \rangle$ existe, luego el derrotador $\langle \mathcal{A}_{j+1}, h_{j+1} \rangle$ (argumento OPP) de $\langle \mathcal{A}_j, h_j \rangle$ será también un derrotador de $\langle \mathcal{A}_i, h_i \rangle$, implicando la existencia de un ciclo en el grafo de derrotas. Al evitar $\langle \mathcal{A}_i, h_i \rangle$ se favorece a OPP frente al ciclo, logrando así imponer la postura de menor compromiso correspondiente a la semántica grounded sin la necesidad de introducir $\langle \mathcal{A}_i, h_i \rangle$, ni reintroducir $\langle \mathcal{A}_{j+1}, h_{j+1} \rangle$. Puede verse además que d2' evita la repetición de argumentos PRO al igual que d2, considerando simplemente $\langle \mathcal{A}_j, h_j \rangle$ como la ocurrencia previa de $\langle \mathcal{A}_i, h_i \rangle$ en la línea.

3.4. Conclusiones

En este capítulo se presentaron y analizaron comparativamente las tres semánticas de aceptabilidad propuestas por Dung en [Dun95] (grounded, estable y preferida), ampliamente adoptadas por las formalizaciones de argumentación existentes para definir la aceptabilidad de argumentos (quinto elemento del modelo conceptual de argumentación). A diferencia de la caracterización propuesta originalmente por Dung, basada en la noción de extensión, la presentada aquí se basa en el enfoque de asignaciones de status, adoptado por Pollock [Pol95] y [MC09], que resulta más conveniente para la posterior presentación de las semánticas para agregación. La equivalencia entre las dos caracterizaciones fue establecida y demostrada.

Adicionalmente se describió una caracterización de la semántica grounded basada en la noción de punto fijo, introducida por Dung en [Dun95], y que sugiere un mecanismo simple para computar el conjunto de argumentos aceptados de acuerdo a dicha semántica.

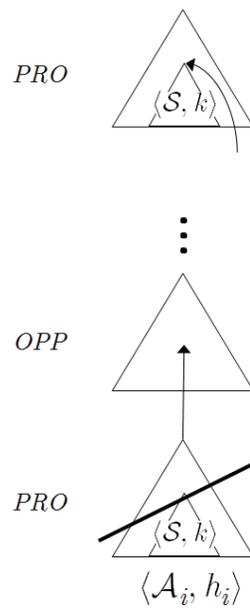


Figura 3.12: Esquema ilustrando efecto de la condición $d2'$

Finalmente se presentó un procedimiento de prueba para la semántica grounded basado en el enfoque dialéctico, o de juegos argumentales (argument games approach), el cual posteriormente se refinó para tomar ventaja de la noción de subargumento de desacuerdo asociado a una derrota.

En capítulos siguientes se presentarán versiones contemplando agregación de las distintas caracterizaciones presentadas en este capítulo.

Capítulo 4

Agregación de Argumentos: Estructuras Agregadas, Conflicto y Derrota

En este capítulo se definen los primeros cuatro elementos del marco conceptual para argumentación propuesto por Prakken, correspondientes a nuestra formalización de agregación. Concretamente, adoptando como lenguaje lógico subyacente el de DeLP (E1), se introduce la noción de estructura agregada (E2), denotando la agregación de diferentes argumentos DeLP soportando la misma conclusión. Se introducen además las nociones de ataque parcial (E2) y derrota parcial (E3), como versiones para agregación de las nociones de ataque y derrota en DeLP. Finalmente se definen las nociones de ataque combinado y derrota combinada, que pueden pensarse como un primer paso hacia la caracterización del análisis de aceptabilidad (E5). La caracterización presentada en este capítulo se encuentra publicada en [LCS09b].

4.1. Estructuras Agregadas

A continuación se introduce la noción de *estructura agregada* con el propósito de modelar la agregación de diferentes argumentos para la misma conclusión.

Definition 2 (Estructura Agregada) Sea \mathcal{P} un programa DeLP, y sea Ω un conjunto de argumentos en \mathcal{P} que sustentan la misma conclusión h , i.e., $\Omega = \{\langle \mathcal{A}_1, h \rangle, \dots, \langle \mathcal{A}_n, h \rangle\}$.

La estructura agregada para h (o simplemente a -estructura) obtenida a partir del conjunto Ω (denotada $\text{Accrual}(\Omega)$) es un par $[\Phi, h]$, donde $\Phi = \mathcal{A}_1 \cup \dots \cup \mathcal{A}_n$. Cuando $\Omega = \emptyset$ se obtiene la estructura agregada especial $[\emptyset, \epsilon]$, representando la agregación del conjunto vacío de argumentos.

Ejemplo 7 Considere un programa DeLP \mathcal{P} donde:

$$\mathcal{P} = \left\{ \begin{array}{cccc} a \prec b, c & b \prec d & f \prec b & c \\ a \prec g & b \prec e & d & h \\ a \prec f & g \prec h & e & \end{array} \right\}$$

Sean $\langle \mathcal{A}_1, a \rangle = \langle \{(a \prec b, c), (b \prec d)\}, a \rangle$,

$\langle \mathcal{A}_2, a \rangle = \langle \{(a \prec b, c), (b \prec e)\}, a \rangle$,

$\langle \mathcal{A}_3, a \rangle = \langle \{(a \prec g)\}, a \rangle$ y

$\langle \mathcal{A}_4, a \rangle = \langle \{(a \prec f), (f \prec b), (b \prec e)\}, a \rangle$ argumentos en \mathcal{P} .

Entonces $\text{Accrual}(\{\langle \mathcal{A}_1, a \rangle, \langle \mathcal{A}_3, a \rangle\}) = [\Phi_1, a]$ donde

$\Phi_1 = \{(a \prec b, c), (b \prec d), (a \prec g)\}$ (Fig. 4.1),

$\text{Accrual}(\{\langle \mathcal{A}_1, a \rangle, \langle \mathcal{A}_4, a \rangle\}) = [\Phi_2, a]$ donde

$\Phi_2 = \{(a \prec b, c), (b \prec d), (a \prec f), (f \prec b), (b \prec e)\}$ (Fig. 4.2),

$\text{Accrual}(\{\langle \mathcal{A}_1, a \rangle, \langle \mathcal{A}_2, a \rangle\}) = [\Phi_3, a]$ donde

$\Phi_3 = \{(a \prec b, c), (b \prec d), (b \prec e)\}$ (Fig. 4.3),

$\text{Accrual}(\{\langle \mathcal{A}_2, a \rangle\}) = [\Phi_4, a]$ donde $\Phi_4 = \mathcal{A}_2$,

$\text{Accrual}(\{\langle \mathcal{A}_1, a \rangle, \langle \mathcal{A}_2, a \rangle, \langle \mathcal{A}_3, a \rangle\}) = [\Phi_5, a]$ donde

$\Phi_5 = \{(a \prec b, c), (b \prec d), (b \prec e), (a \prec g)\}$ (Fig. 4.4).

Una a -estructura para una conclusión h puede verse como una clase especial de argumento que involucra diferentes encadenamientos de razones que proveen soporte para h . Por ejemplo, la a -estructura $[\Phi_1, a]$ (Fig. 4.1) involucra dos encadenamientos de razones alternativos soportando a , cada uno proveniente de uno de los argumentos acumulados: por un lado, a porque b y c , y a su vez b porque d , y por otro lado, a porque g y g porque h .

El caso de $[\Phi_2, a]$ en el Ej. 7 (Fig. 4.2) ilustra un aspecto importante de nuestra noción formal de agregación, y de la forma en que interpretaremos las a -estructuras.

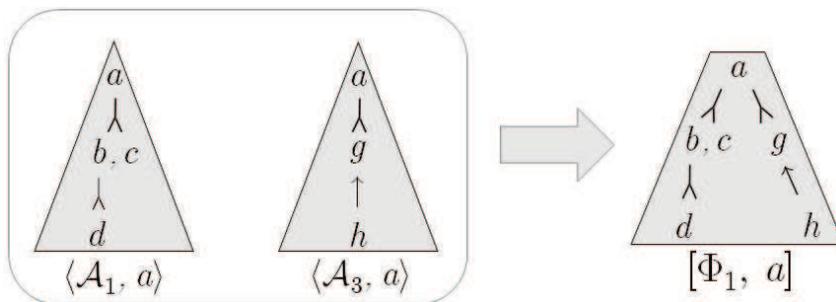


Figura 4.1: Estructura Agregada $[\Phi_1, a]$

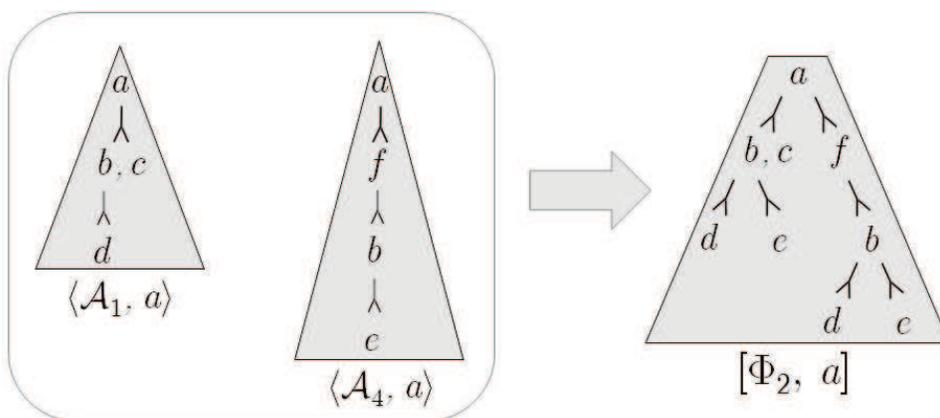


Figura 4.2: Estructura Agregada $[\Phi_2, a]$

Dado que el soporte de una a-estructura resulta simplemente de unir los soportes de los argumentos agregados, si dos argumentos para la misma conclusión (en este caso, $\langle \mathcal{A}_1, a \rangle$ y $\langle \mathcal{A}_2, a \rangle$) comparten alguna conclusión intermedia (b), pero la soportan de diferentes maneras, entonces al agregarlos (obteniendo $[\Phi_2, a]$) se considera que las razones para la conclusión intermedia también se acumulan. En otras palabras, la agregación se produce tanto a nivel de conclusión final, como a nivel de conclusión intermedia.

Adicionalmente, el caso de $[\Phi_2, a]$ evidencia una particularidad de la noción de agregación presentada. A pesar de que los argumentos agregados representan, cada uno, un encadenamiento de razones soportando a , la a-estructura $[\Phi_2, a]$ resultante representa cuatro encadenamientos de razones distintos para a (dos de los cuales no están presentes explícitamente en los argumentos individuales agregados).

Finalmente, el caso de $[\Phi_3, a]$ en el Ej. 7 (Fig. 4.3) ilustra una situación similar a la previa, donde los argumentos involucrados comparten no solo la conclusión intermedia b , sino que además comparten su parte superior (más precisamente la regla $a \prec b, c$). En

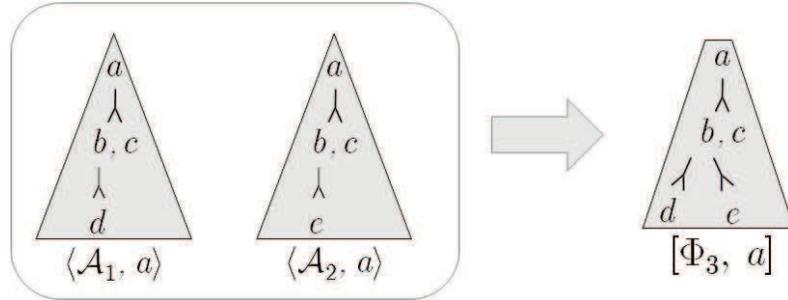


Figura 4.3: Estructura Agregada $[\Phi_3, a]$

este caso la agregación se produce solo a nivel de la conclusión intermedia b , aunque por supuesto, esto fortalece indirectamente el soporte para la conclusión final a .

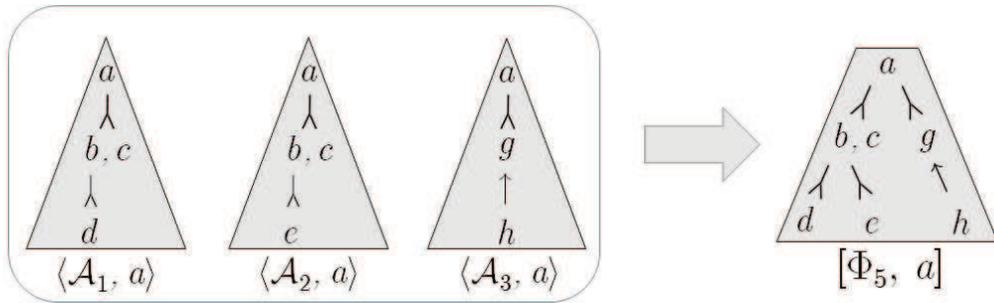


Figura 4.4: Estructura Agregada $[\Phi_5, a]$

Una pregunta importante que surge naturalmente al considerar la forma en que definimos la agregación es qué sucede si agregamos dos argumentos cuyos soportes están en conflicto (por ejemplo, involucran conclusiones intermedias contradictorias.) Retomaremos esta cuestión más adelante.

Definición 25 Sea $[\Phi, h]$ una a -estructura. Entonces el conjunto de argumentos en $[\Phi, h]$, denotado $Args([\Phi, h])$, es el conjunto de todos los argumentos $\langle \mathcal{A}_i, h \rangle$ tal que $\mathcal{A}_i \subseteq \Phi$. Nótese que $Args([\emptyset, \epsilon]) = \emptyset$.

Ejemplo 8 Considere los argumentos y a -estructuras presentados en el Ej. 7. Entonces $Args([\Phi_1, a]) = \{\langle \mathcal{A}_1, a \rangle, \langle \mathcal{A}_3, a \rangle\}$ y $Args([\Phi_2, a]) = \{\langle \mathcal{A}_1, a \rangle, \langle \mathcal{A}_2, a \rangle, \langle \mathcal{A}_4, a \rangle, \langle \{(a \prec f), (f \prec b), (b \prec d)\}, a \rangle\}$.

A pesar de que $Accrual(\cdot)$ y $Args(\cdot)$ no son operaciones recíprocas (como ilustra el caso de $[\Phi_2, a]$ en los Ejs. 7 y 8), podemos asegurar que los argumentos agregados siempre

estarán entre los argumentos en la a-estructura resultante. Además podemos asegurar que al agregar los argumentos en una a-estructura dada $[\Psi, k]$ ($Args([\Psi, k])$) siempre se obtiene $[\Psi, k]$ como resultado.

Propiedad 1 *Sea Ω un conjunto de argumentos para una conclusión dada h . Entonces $Args(Accrual(\Omega)) \supseteq \Omega$. Además, para cada a-estructura $[\Psi, k]$ vale que $Accrual(Args([\Psi, k])) = [\Psi, k]$.*

Prueba. La demostración de $Args(Accrual(\Omega)) \supseteq \Omega$ resulta trivial considerando las definiciones de $Accrual(\cdot)$ y $Args(\cdot)$: $Accrual(\Omega)$ denota la a-estructura que resulta de unir los conjuntos soporte de los argumentos en Ω , y $Args(Accrual(\Omega))$ es el conjunto de argumentos cuyo soporte está incluido en dicha unión.

Para demostrar que $Accrual(Args([\Psi, k])) = [\Psi, k]$ probaremos por doble inclusión que $\bigcup_{\{\mathcal{A}_i \mid \langle \mathcal{A}_i, k \rangle \in Args([\Psi, k])\}} = \Psi$. ‘ \subseteq ’: trivial, ya que, por definición de $Args(\cdot)$, para todo $\langle \mathcal{A}_i, k \rangle \in Args([\Psi, k])$ vale que $\mathcal{A}_i \subseteq \Psi$.

‘ \supseteq ’: por definición de a-estructura podemos asegurar que existe un conjunto de argumentos Ω tal que $Accrual(\Omega) = [\Psi, k]$. Aplicando $Args(\cdot)$ a ambos miembros, y apelando a la afirmación demostrada anteriormente, tenemos que $Args(Accrual(\Omega)) = Args([\Psi, k]) \supseteq \Omega$. Luego vale que $\bigcup_{\{\mathcal{A}_i \mid \langle \mathcal{A}_i, k \rangle \in Args([\Psi, k])\}} \supseteq \bigcup_{\{\mathcal{A}_i \mid \langle \mathcal{A}_i, k \rangle \in \Omega\}}$, donde $\bigcup_{\{\mathcal{A}_i \mid \langle \mathcal{A}_i, k \rangle \in \Omega\}} = \Psi$ (por $Accrual(\Omega) = [\Psi, k]$) \square

Entre todas las a-estructuras respecto a un programa DeLP \mathcal{P} , distinguiremos a-estructuras maximales, que son aquellas a-estructuras (no vacías) obtenidas como resultado de acumular todos los argumentos que soportan una conclusión dada h .

Definición 26 (a-estructura maximal) *Sea \mathcal{P} un programa DeLP y sea Ω_h el conjunto de todos los argumentos en \mathcal{P} que soportan una conclusión dada h , $\Omega_h \neq \emptyset$. Diremos que la a-estructura $[\Phi, h] = Accrual(\Omega_h)$ es una a-estructura maximal.*

A partir de la definición de a-estructura maximal resulta evidente que, dado un literal h , existe a lo sumo una a-estructura maximal que soporta h .

Ejemplo 9 *Considere el programa DeLP \mathcal{P} y las a-estructuras del Ej. 7. Entonces $[\Phi_6, a] = Accrual(\{\langle \mathcal{A}_1, a \rangle, \langle \mathcal{A}_2, a \rangle, \langle \mathcal{A}_3, a \rangle, \langle \mathcal{A}_4, a \rangle, \langle \{(a \prec f), (f \prec b), (b \prec d)\}, a \rangle\})$ es una a-estructura maximal en \mathcal{P} , mientras que $[\Phi_1, a]$ y $[\Phi_2, a]$ no son a-estructuras maximales.*

A continuación se introduce la noción de *narrowing* de una a-estructura, que es análoga a la noción de narrowing en [Ver96], y que constituye un elemento central en nuestra formalización de agregación ¹. Intuitivamente, un narrowing de una a-estructura $[\Phi, h]$ es una a-estructura $[\Theta, h]$ representando un subconjunto de $Args([\Phi, h])$.

Definición 27 (Narrowing de una a-estructura) Sean $[\Phi, h]$ y $[\Theta, h]$ dos a-estructuras. Diremos que $[\Theta, h]$ es un narrowing de $[\Phi, h]$, denotado $[\Theta, h] \sqsubseteq [\Phi, h]$, sssi $Args([\Theta, h]) \subseteq Args([\Phi, h])$.

Ejemplo 10 Considere las a-estructuras del Ej. 7. Entonces $[\Phi_1, a]$, $[\Phi_3, a]$, y la propia a-estructura $[\Phi_5, a]$, son narrowings de $[\Phi_5, a]$.

A continuación se introduce la noción de sub-estructura agregada, que es análoga a la noción de subargumento pero para a-estructuras. Intuitivamente, una sub-estructura agregada de una a-estructura $[\Phi, h]$ es una a-estructura soportando una conclusión intermedia k de $[\Phi, h]$ y representando el conjunto de todas las razones soportando k en $[\Phi, h]$.

Definición 28 (a-subestructura) Sean $[\Phi, h]$ y $[\Theta, k]$ dos a-estructuras. Diremos que $[\Theta, k]$ es una sub-estructura agregada (o simplemente a-subestructura) de $[\Phi, h]$ sssi $[\Theta, k]$ es maximal verificando que $\Theta \subseteq \Phi$, es decir, para toda otra a-estructura $[\Theta', k]$ vale que $\Theta' \subset \Theta$.

Ejemplo 11 Considere la a-estructura $[\Phi_3, a]$ del Ej. 7. La a-estructura $\{(b \prec d), (b \prec e)\}, b]$ y la propia $[\Phi_3, a]$ son a-subestructuras de $[\Phi_3, a]$, mientras que $\{(b \prec d)\}, b]$ no es a-subestructura de $[\Phi_3, a]$.

Finalmente se introducen operadores de unión (\sqcup), intersección (\sqcap) y diferencia (\setminus) entre a-estructuras soportando una misma conclusión.

Definición 29 (operadores \sqcup , \sqcap y \setminus para a-estructuras) Sean $[\Phi_1, h]$ y $[\Phi_2, h]$ dos a-estructuras (soportando la misma conclusión h). Entonces:

¹Se decidió emplear el término original “narrowing”, en inglés, dado que no se encontró una traducción al castellano que resulte adecuada.

- $[\Phi_1, h] \sqcup [\Phi_2, h] =_{def} Accrual(Args([\Phi_1, h]) \cup Args([\Phi_2, h]))$
- $[\Phi_1, h] \sqcap [\Phi_2, h] =_{def} Accrual(Args([\Phi_1, h]) \cap Args([\Phi_2, h]))$
- $[\Phi_1, h] \setminus [\Phi_2, h] =_{def} Accrual(Args([\Phi_1, h]) \setminus Args([\Phi_2, h]))$

Ejemplo 12 Considere las a -estructuras del Ej. 7. Entonces $[\Phi_1, a] \sqcup [\Phi_4, a] = [\Phi_5, a]$ (Fig. 4.5a), $[\Phi_1, a] \sqcap [\Phi_3, a] = [\Phi_7, a]$ (Fig. 4.5b), $[\Phi_1, a] \sqcap [\Phi_4, a] = [\emptyset, \epsilon]$, $[\Phi_5, a] \sqcap [\Phi_3, a] = [\Phi_3, a]$, y $[\Phi_5, a] \setminus [\Phi_4, a] = [\Phi_1, a]$ (Fig. 4.5c).

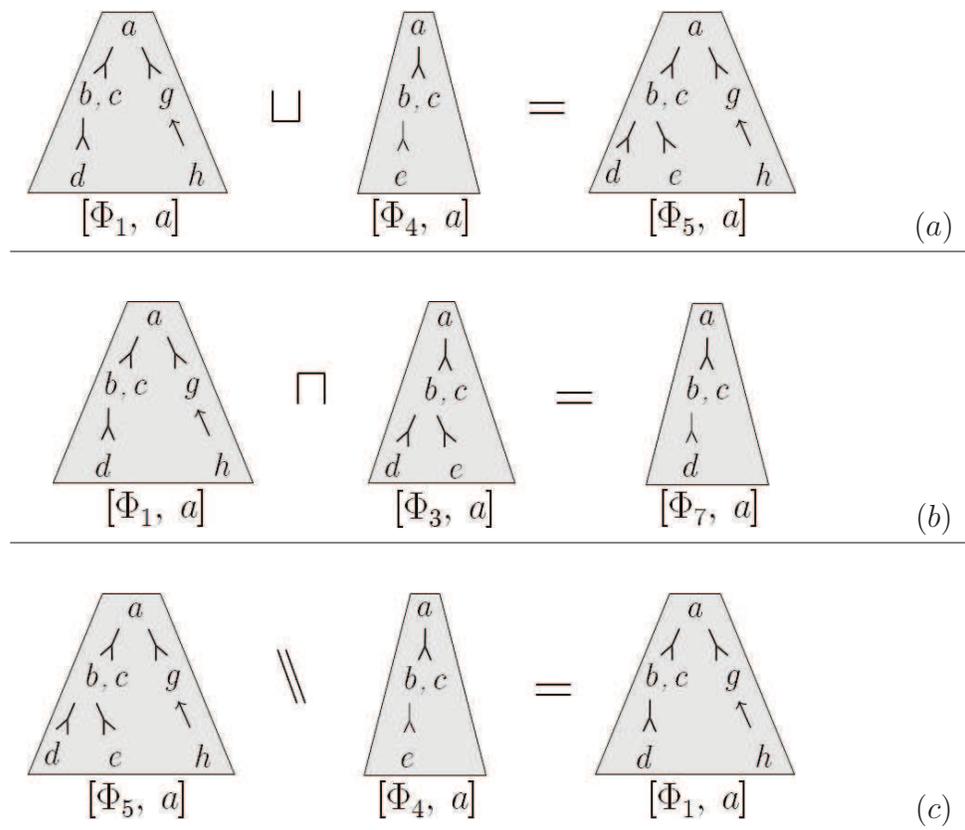


Figura 4.5: Unión, intersección y diferencia para a -estructuras

4.2. Conflicto y Derrota entre Estructuras Agregadas

4.2.1. Ataque Parcial

A continuación se formaliza la noción de ataque entre a-estructuras, que difiere de la noción de ataque para argumentos en varios aspectos. Primero, una a-estructura $[\Phi, h]$ representa (posiblemente) varias cadenas de razonamiento (argumentos) soportando la conclusión h . Además, algunas conclusiones intermedias de $[\Phi, h]$ podrían ser compartidas por algunos, aunque no necesariamente todos los argumentos en $[\Phi, h]$. Por lo tanto, dadas dos a-estructuras $[\Phi, h]$ y $[\Psi, k]$, si $[\Phi, h]$ involucra una conclusión intermedia \bar{k} , y en ese caso $[\Psi, k]$ contradice a $[\Phi, h]$ en ese punto, solo aquellos argumentos en $Args([\Phi, h])$ involucrando \bar{k} se verán afectados por el conflicto.

A continuación se define la noción de *ataque parcial*, donde la a-estructura que ataca generalmente afecta solo un narrowing de la a-estructura atacada (aquel conteniendo exactamente los argumentos en la a-estructura atacada afectados por el conflicto), y haremos referencia a dicho narrowing como el *narrowing atacado*.

Definición 30 (Ataque Parcial y Narrowing Atacado) Sean $[\Phi, h]$ y $[\Psi, k]$ dos a-estructuras. Diremos que $[\Psi, k]$ ataca parcialmente a $[\Phi, h]$ (en el literal \bar{k}), sssi existe una a-subestructura $[\Theta, \bar{k}]$ de $[\Phi, h]$. La a-subestructura $[\Theta, \bar{k}]$ se denominará a-subestructura de desacuerdo. Diremos además que $[\Lambda, h] \sqsubseteq [\Phi, h]$ es el narrowing atacado de $[\Phi, h]$ asociado con el ataque sssi $[\Lambda, h] = Accrual(\{\langle \mathcal{A}, h \rangle \in Args([\Phi, h]) \mid \text{existe un subargumento } \langle \mathcal{S}, \bar{k} \rangle \text{ de } \langle \mathcal{A}, h \rangle\})$.

La Fig. 4.6 muestra un esquema ilustrando la definición de ataque parcial presentada. Como muestra la figura, la relación de ataque parcial será denotada gráficamente con una flecha punteada.

Ejemplo 13 Considere un programa DeLP \mathcal{P} donde:

$$\mathcal{P} = \left\{ \begin{array}{cccc} x \prec z & y \prec u & \sim x \prec q & u \\ x \prec y & \sim z \prec w & \sim y \prec p & w \\ z \prec t & \sim z \prec s & t & p \\ z \prec v & s \prec p & v & q \end{array} \right\}$$

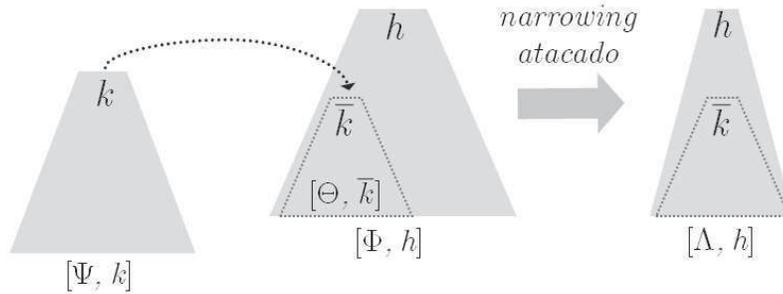


Figura 4.6: Noción de ataque parcial

Considere las *a*-estructuras $[\Phi, x]$ y $[\Psi_1, \sim z]$ en la Fig. 4.7. La *a*-estructura $[\Psi_1, \sim z]$ ataca parcialmente a $[\Phi, x]$ con *a*-subestructura de desacuerdo $[\Theta, z] = [\{(z \prec t), (z \prec v)\}, z]$. El *narrowing atacado* de $[\Phi, x]$ es $[\{(x \prec z), (z \prec t), (z \prec v)\}, x]$.

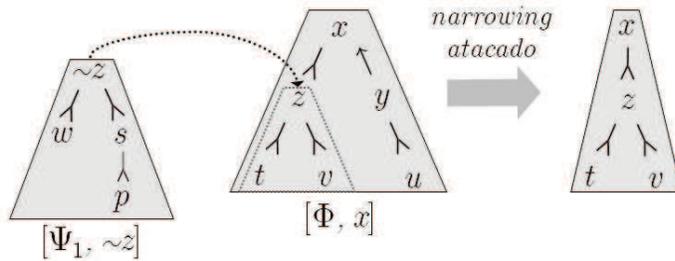


Figura 4.7: Ataque Parcial

4.2.2. Evaluación y Derrota Parcial

Para decidir si un ataque parcial dado realmente tiene éxito y constituye una derrota necesitamos un criterio para determinar la fuerza relativa de aquellas *a*-estructuras en conflicto. En general, tal criterio de comparación debe definirse de acuerdo al dominio de aplicación. En adelante, nos abstraeremos del criterio asumiendo la existencia de una relación de preferencia binaria ‘ \gg ’ entre *a*-estructuras.

Definición 31 (relación de preferencia ‘ \gg ’) Una relación de preferencia binaria \gg , que significa ‘tanto o más preferida que’, es un preorden sobre *a*-estructuras que verifica la siguiente condición:

$$\text{si } [\Phi_1, h] \sqsubseteq [\Phi_2, h] \text{ entonces } [\Phi_2, h] \gg [\Phi_1, h].$$

Informalmente, la condición impuesta sobre \succcurlyeq establece que si una a-estructura dada $[\Phi_2, h]$ soporta h con las mismas razones o más que otra a-estructura $[\Phi_1, h]$ ($[\Phi_1, h] \sqsubseteq [\Phi_2, h]$) entonces $[\Phi_2, h]$ es al menos tan fuerte como $[\Phi_1, h]$ de acuerdo a \succcurlyeq . En otras palabras, incorporar (agregar) nuevas razones nunca debilita.

A partir de ‘ \succcurlyeq ’ podemos derivar otras dos relaciones importantes:

- relación de preferencia estricta ‘ \gg ’, definida como $[\Phi, h] \gg [\Psi, k]$ sssi $[\Phi, h] \succcurlyeq [\Psi, k]$ pero no vale $[\Psi, k] \succcurlyeq [\Phi, h]$.
- relación de igualdad de preferencia ‘ \approx ’, definida como $[\Phi, h] \approx [\Psi, k]$ sssi $[\Phi, h] \succcurlyeq [\Psi, k]$ y $[\Psi, k] \succcurlyeq [\Phi, h]$.

Definición 32 (Derrotador Parcial) Sean $[\Phi, h]$ y $[\Psi, k]$ dos a-estructuras. Diremos que $[\Psi, k]$ es un derrotador parcial de $[\Phi, h]$ (o de forma equivalente que $[\Psi, k]$ es un ataque exitoso sobre $[\Phi, h]$) sssi 1) $[\Psi, k]$ ataca parcialmente a $[\Phi, h]$, donde $[\Theta, \bar{k}]$ es la a-subestructura de desacuerdo asociada, y 2) no es el caso que $[\Theta, \bar{k}] \gg [\Psi, k]$.

La Fig. 4.8 muestra un esquema ilustrando la definición derrota parcial presentada. Como muestra la figura, la relación de derrota parcial será denotada gráficamente con una flecha continua.

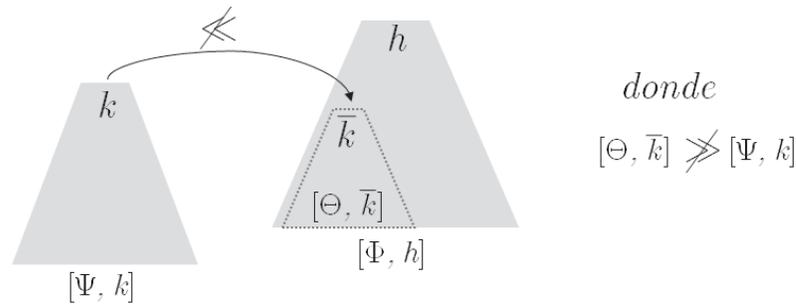


Figura 4.8: Noción de derrota parcial

Ejemplo 14 Considere el ataque parcial de $[\Psi_1, \sim z]$ contra $[\Phi, x]$ con a-subestructura de desacuerdo $[\Theta, z]$ en el Ej. 13 (Fig. 4.7), y asumamos que $[\Psi_1, \sim z] \gg [\Theta, z]$. Entonces el ataque tiene éxito, constituyendo una derrota.

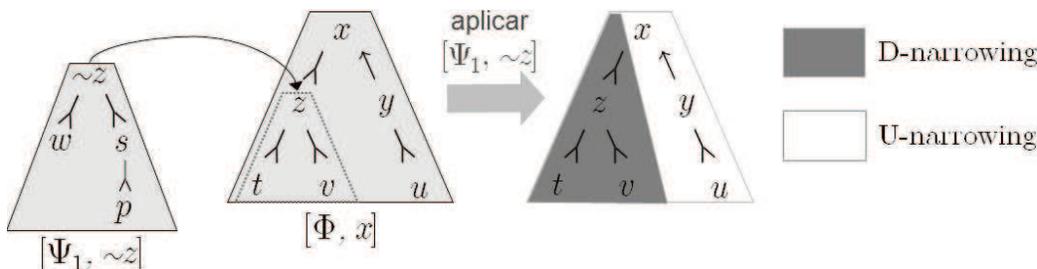


Figura 4.9: narrowings derrotado y no derrotado

Dada una relación de ataque, identificaremos dos narrowings complementarios asociados con la a-estructura atacada: el narrowing que resulta derrotado consecuencia del ataque, y el narrowings que resulta no derrotado.

Definition 3 (U/D-narrowings) Sean $[\Phi, h]$ y $[\Psi, k]$ dos a-estructuras tal que $[\Psi, k]$ ataca parcialmente a $[\Phi, h]$. Sea $[\Lambda, h]$ el narrowing atacado de $[\Phi, h]$. El D-narrowing (defeated narrowing) de $[\Phi, h]$ asociado con el ataque, denotado $DNwg([\Phi, h], [\Psi, k])$, se define por casos como sigue: 1) $DNwg([\Phi, h], [\Psi, k]) =_{def} [\Lambda, h]$, si $[\Psi, k]$ es un derrotador parcial de $[\Phi, h]$, o 2) $DNwg([\Phi, h], [\Psi, k]) =_{def} [\emptyset, \epsilon]$, en caso contrario. El U-narrowing (undefeated narrowing) de $[\Phi, h]$ asociado con el ataque, denotado $UNwg([\Phi, h], [\Psi, k])$, es la a-estructura $[\Phi, h] \setminus DNwg([\Phi, h], [\Psi, k])$.

Ejemplo 15 La Fig. 4.9 ilustra un ataque exitoso de $[\Psi_1, \sim z]$ contra $[\Phi, x]$, así como el U-narrowing y el D-narrowing de $[\Phi, x]$ asociados con el ataque. La Fig. 4.10 ilustra el ataque de $[\Psi_2, \sim x]$ contra $[\Phi, x]$, con la propia a-estructura $[\Phi, x]$ como a-subestructura de desacuerdo, y donde $[\Phi, x] \gg [\Psi_2, \sim x]$. En este caso el ataque no tiene éxito, y por lo tanto $[\emptyset, \epsilon]$ es el D-narrowing y $[\Phi, x]$ es el U-narrowing.

4.2.3. Ataque Combinado y Derrota Combinada

Hasta ahora solo se han considerado ataques *individuales*. Cuando un ataque individual tiene éxito, un narrowing no vacío de la a-estructura atacada resulta derrotado. Sin embargo, dos o más a-estructuras podrían atacar simultáneamente a otra, posiblemente afectando diferentes narrowings de la a-estructura *víctima*, y causando así la derrota de un narrowing mayor (comparado con los D-narrowings asociados a los ataques individuales). La Fig. 4.11 ilustra un ataque *combinado* de las a-estructuras $[\Psi_1, \sim z]$ y $[\Psi_3, \sim y]$ contra

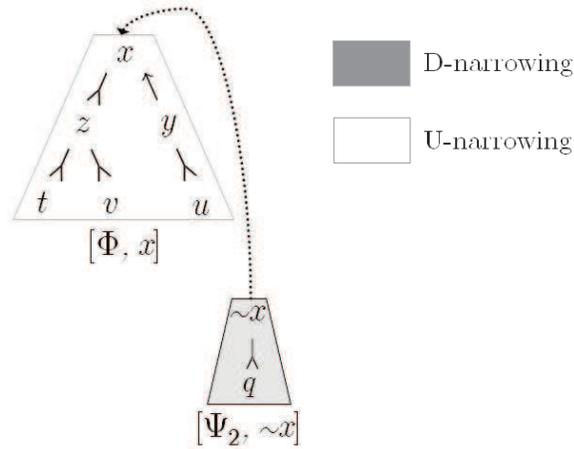


Figura 4.10: narrowings derrotado y no derrotado

$[\Phi, x]$. A pesar de que cada una de las a-estructuras que ataca solo derrota un narrowing propio de $[\Phi, x]$, la a-estructura $[\Phi, x]$ completa resulta derrotada luego de aplicar *ambos* ataques.

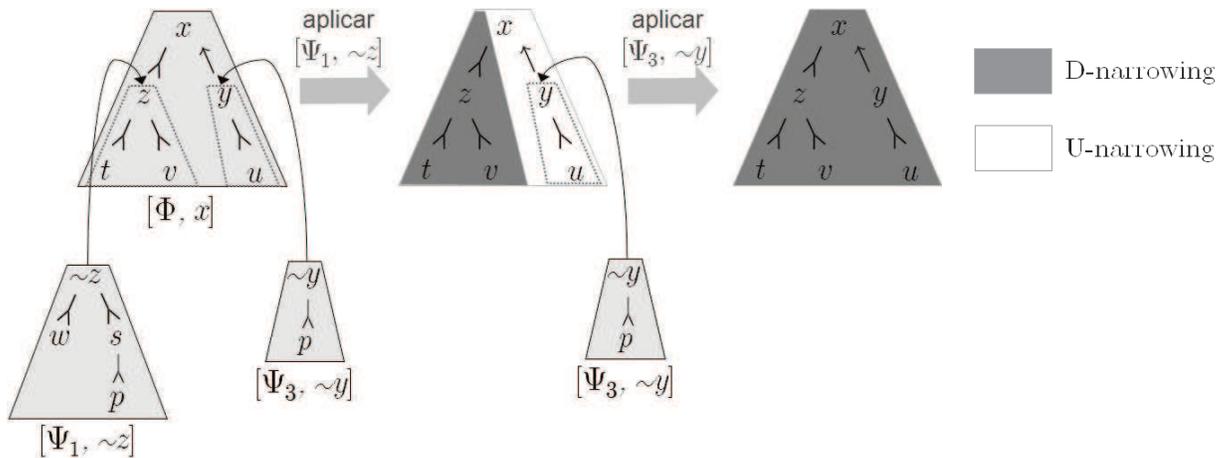


Figura 4.11: Derrota Combinada

Considérese ahora el ataque combinado contra $[\Phi, x]$ mostrado en la Fig. 4.12. Una de las a-estructuras que atacan ($[\Psi_1, \sim z]$) derrota por sí sola un narrowing de $[\Phi, x]$, mientras que la otra ($[\Psi_2, \sim x]$) solo ataca $[\Phi, x]$. Supongamos que, a pesar de que $[\Phi, x]$ es más fuerte que $[\Psi_2, \sim x]$ de acuerdo al criterio de preferencia ' \gg ', $[\Psi_2, \sim x]$ es más fuerte que $[\Phi', x] = [\{y \prec y\}, x]$, narrowing propio de $[\Phi, x]$. Luego, como se muestra en la Fig. 4.12, cuando las a-estructuras $[\Psi_1, \sim z]$ and $[\Psi_2, \sim x]$ combinan sus ataques, ocasionan la derrota de la a-estructura $[\Phi, x]$ completa. La razón es que el ataque exitoso de $[\Psi_1, \sim z]$ debilita la a-estructura víctima, permitiendo que el ataque de $[\Psi_2, \sim x]$ tenga éxito. La

Fig. 4.13 ilustra un ataque combinado de $[\Psi_2, \sim x]$ y $[\Psi_3, \sim y]$ contra $[\Phi, x]$. En este caso, un narrowing no vacío de la a-estructura atacada permanece no derrotado.

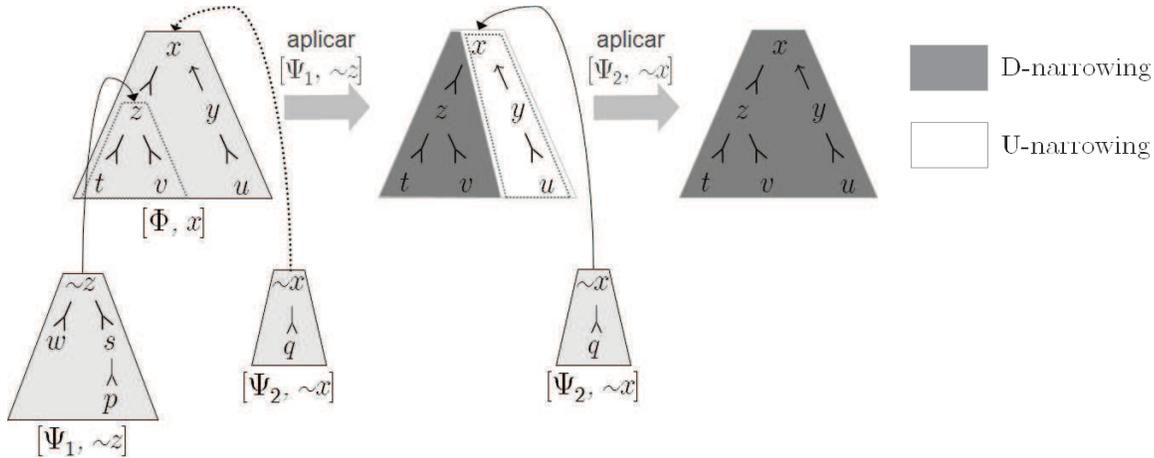


Figura 4.12: Derrota Combinada

Las Figs. 4.11, 4.12 y 4.13 sugieren el siguiente pseudo-algoritmo para computar el U-narrowing asociado con el ataque combinado de un conjunto Σ de a-estructuras contra una a-estructura $[\Phi, h]$. La variable UN (por U-narrowing) almacena en todo momento el narrowing de $[\Phi, h]$ que aún permanece no derrotado.

UN \Leftarrow $[\Phi, h]$

Repetir mientras haya derrotadores de UN en Σ :

1. Seleccionar un derrotador $[\Psi_i, k_i] \in \Sigma$ de UN
2. UN \Leftarrow UNwg($[\Phi, h], [\Psi_i, k_i]$)

Retornar(UN)

De acuerdo a este procedimiento, derrotadores de $[\Phi, h]$ son aplicados en secuencia (en algún orden), y cada derrotador causa que un narrowing de la a-estructura víctima resulte derrotado. En otras palabras, la a-estructura $[\Phi, h]$ es *secuencialmente degradada* a través de la aplicación de derrotadores. La siguiente definición provee una caracterización formal (declarativa) del pseudo-algoritmo presentado.

Definición 33 (Degradación Secuencial) Sea $[\Phi, h]$ una a-estructura y sea Σ un conjunto de a-estructuras atacando $[\Phi, h]$. Una degradación secuencial de $[\Phi, h]$, asociada con el ataque combinado de las a-estructuras en Σ , consiste de una secuencia finita Υ de narrowings de $[\Phi, h]$:

$$[\Phi_1, h], [\Phi_2, h], \dots, [\Phi_{m+1}, h]$$

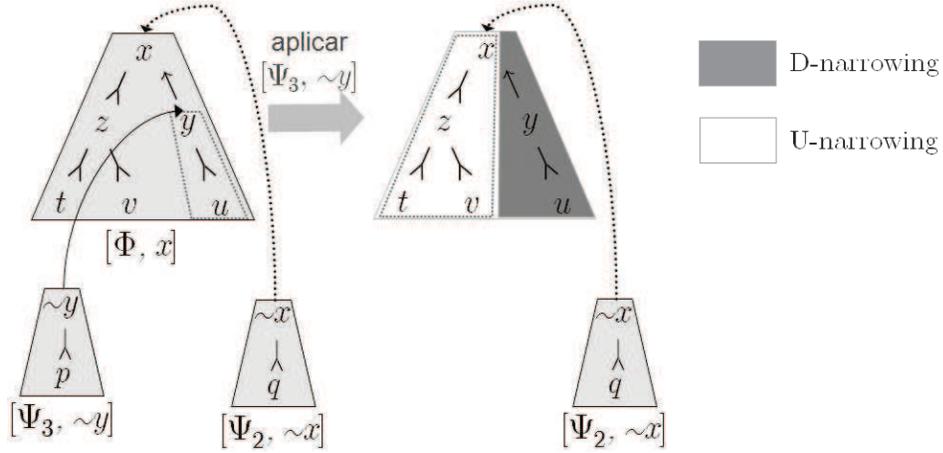


Figura 4.13: Derrota Combinada

provisto de que exista una secuencia finita de a -estructuras de Σ :

$$[\Psi_1, k_1], [\Psi_2, k_2], \dots, [\Psi_m, k_m]$$

donde $[\Phi_1, h] = [\Phi, h]$, para cada i , $1 \leq i \leq m$, $[\Psi_i, k_i]$ es un derrotador parcial de $[\Phi_i, h]$ con U -narrowing asociado $[\Phi_{i+1}, h]$, y $[\Phi_{m+1}, h]$ no tiene derrotadores en Σ .

Dado un ataque combinado contra una a -estructura $[\Phi, h]$, podrían existir varios posibles órdenes de aplicación de derrotadores, y por lo tanto, más de una degradación secuencial asociada con el ataque combinado. Considérese el ataque combinado de la Fig. 4.11. Es posible aplicar los derrotadores en el otro orden (primero $[\Psi_3, \sim y]$ y luego $[\Psi_1, \sim z]$), obteniendo así una degradación secuencial diferente. Afortunadamente, puede mostrarse que todas las degradaciones secuenciales asociadas con un ataque combinado dado convergen en la misma a -estructura.

Teorema 1 (Convergencia) Sea $[\Phi, h]$ una a -estructura y sea Σ un conjunto de a -estructuras que atacan $[\Phi, h]$. Sean $\Upsilon = [\Phi_1, h], \dots, [\Phi_m, h]$ y $\Upsilon' = [\Phi'_1, h], \dots, [\Phi'_m, h]$ dos degradaciones secuenciales de $[\Phi, h]$ asociadas con el ataque combinado de las a -estructuras en Σ . Entonces $[\Phi_m, h] = [\Phi'_m, h]$ ².

Finalmente, el U -narrowing de una a -estructura $[\Phi, h]$ respecto a un ataque combinado se definirá como el último elemento de una degradación secuencial de $[\Phi, h]$ asociada con dicho ataque combinado.

²Las pruebas de los teoremas y lemas se incluyen en el apéndice A.

Definición 34 (Narrowings asociados con un Ataque Combinado) Sea $[\Phi, h]$ una a-estructura y sea Σ un conjunto de a-estructuras atacando $[\Phi, h]$. Sea $[\Phi_1, h], [\Phi_2, h], \dots, [\Phi_{m+1}, h]$ una degradación secuencial de $[\Phi, h]$ asociada con el ataque combinado de las a-estructuras en Σ . Entonces $[\Phi_{m+1}, h]$ es el U-narrowing de $[\Phi, h]$ (notado $UNwg([\Phi, h], \Sigma)$) asociado con el ataque combinado, y $[\Phi, h] \setminus\setminus UNwg([\Phi, h], \Sigma)$ es su D-narrowing (notado $DNwg([\Phi, h], \Sigma)$).

Ejemplo 16 Considere el ataque combinado de $[\Psi_1, \sim z]$ y $[\Psi_2, \sim x]$ contra $[\Phi_3, x]$ (Fig. 4.12). El U-narrowing de $[\Phi, x]$ asociado con el ataque es $[\emptyset, \epsilon]$, es decir, la a-estructura $[\Phi, x]$ completa resulta derrotada. Por otro lado, cuando $[\Psi_2, \sim x]$ y $[\Psi_3, \sim y]$ atacan $[\Phi, x]$ (Fig. 4.13), el U-narrowing asociado es $[\{(x \prec z), (z \prec y), (z \prec v)\}, x]$

4.3. Conclusiones

En este capítulo se definieron los elementos E1-E4 del marco conceptual para argumentación propuesto por Prakken, correspondientes a nuestra formalización de agregación. Adoptando como lenguaje lógico subyacente el de DeLP (E1), se introdujo la noción de *estructura agregada* o *a-estructura* (E2), denotando la agregación de diferentes argumentos DeLP soportando la misma conclusión. Adicionalmente se adoptó la noción de *narrowing*, inicialmente propuesta por [Ver95, Ver96], asociada a una a-estructura dada, consistiendo básicamente de una a-estructura representando un subconjunto de las razones presentes en la primera. Con el propósito de simplificar la formalización de nociones subsiguientes (del presente capítulo y de los Caps. 5 y 6), se presentaron operaciones de *unión*, *intersección* y *diferencia* entre a-estructuras soportando la misma conclusión.

Se introdujeron las nociones de *ataque parcial* (E3) y *derrota parcial* (E4), como versiones para agregación de las nociones de ataque y derrota en DeLP. El adjetivo ‘parcial’ se debe a que, a diferencia de las nociones tradicionales de ataque y derrota, es posible que afecten solo a una parte (narrowing) de la a-estructura víctima. Finalmente se introdujeron las nociones de *ataque combinado* y *derrota combinada*, que constituyen un primer paso hacia la caracterización del análisis de aceptabilidad (E5). Estas nociones capturan que el ataque de varias a-estructuras (soportando diferentes conclusiones), al ser consideradas conjuntamente, pueden provocar un ‘daño mayor’ sobre una a-estructura víctima que si fueran consideradas individualmente. El efecto de un ataque combinado contra una

a-estructura (narrowing de la a-estructura atacada que resulta derrotado y narrowing que permanece no derrotado) se define en términos de la noción de *degradación secuencial*, que sugiere directamente un procedimiento para computarlo.

Capítulo 5

Semánticas de Aceptabilidad para la Agregación de argumentos

En el Cap. 4 se presentaron las *estructuras agregadas* (o a-estructuras) y se caracterizaron los conflictos y derrotas entre ellas (elementos E2-E4 del esquema conceptual de argumentación presentado en la introducción). En este capítulo se presentan las semánticas de aceptabilidad *a-grounded*, *a-estable* y *a-preferida* para la agregación (elemento E5), que pueden verse como extensiones naturales para contemplar agregación de las correspondientes semánticas de aceptabilidad propuestas por Dung en [Dun95] y ampliamente aceptadas por la comunidad de argumentación (presentadas y analizadas en detalle en el Cap. 3). Se propone además una caracterización alternativa para la semántica a-grounded basada en la noción de punto fijo, inspirada en la caracterización original de la semántica *grounded* desarrollada por Dung ([Dun95]), y que sugiere un procedimiento efectivo para computar la semántica a-grounded. Luego se enuncian y demuestran propiedades deseables de consistencia satisfechas por las semánticas propuestas, y finalmente se ilustra la formalización desarrollada aplicándola al ejemplo del alquiler presentado en la introducción.

5.1. Asignación de Status para Estructuras Agregadas

En esta sección se introducirá la noción de *asignación de status para a-estructuras*, que tiene una fuerte correspondencia con la noción de asignación de status para argumentos individuales introducida en la sección 3.1. Una asignación de status para argumentos individuales asocia un status IN o OUT a algunos (posiblemente a todos) los argumentos, donde el status de un argumento dado A depende de los asociados a sus derrotadores. Aquellos argumentos sin status asociado son considerados *undecided*. Análogamente, una asignación de status para a-estructuras (respecto a un programa DeLP dado \mathcal{P}) asocia a cada a-estructura maximal $[\Phi, h]$ (de \mathcal{P}), dos narrowings disjuntos de $[\Phi, h]$, no necesariamente complementarios, denominados *narrowing IN* y *narrowing OUT*, y éstos dependen de los narrowings IN y OUT asociados a las a-estructuras que atacan $[\Phi, h]$. Dado que los narrowings IN y OUT pueden no ser complementarios, distinguiremos un tercer narrowing de $[\Phi, h]$, al que haremos referencia como *narrowing undecided*, conformado por aquellos argumentos de $[\Phi, h]$ que no pertenecen a los narrowings IN ni OUT.

Antes de definir la noción de asignación de status, se definirá una versión preliminar, más relajada, que asocia narrowings IN y OUT a cada a-estructura, ignorando los ataques entre ellas.

Definición 35 (Asignación de Status Arbitraria) *Sea \mathcal{P} un programa DeLP. Una asignación de status arbitraria \mathbb{S} para \mathcal{P} es una asignación, para cada a-estructura maximal $[\Phi, h]$ de \mathcal{P} , de dos narrowings de $[\Phi, h]$, $IN_{\mathbb{S}}([\Phi, h])$ (narrowing IN de $[\Phi, h]$ de acuerdo a \mathbb{S}) y $OUT_{\mathbb{S}}([\Phi, h])$ (narrowing OUT de $[\Phi, h]$ de acuerdo a \mathbb{S}) tal que $IN_{\mathbb{S}}([\Phi, h]) \sqcap OUT_{\mathbb{S}}([\Phi, h]) = [\emptyset, \epsilon]$. Distinguimos una asignación de status arbitraria especial \circlearrowleft (“vacía”), donde $IN_{\circlearrowleft}([\Phi, h]) = [\emptyset, \epsilon]$ y $OUT_{\circlearrowleft}([\Phi, h]) = [\emptyset, \epsilon]$, para toda a-estructura maximal $[\Phi, h]$.*

Aclaraciones sobre notación:

- A modo de simplificación, usaremos $\neg OUT_{\mathbb{S}}([\Phi, h])$ para hacer referencia a la a-estructura $[\Phi, h] \setminus OUT_{\mathbb{S}}([\Phi, h])$, representando el narrowing de $[\Phi, h]$ que no se encuentra OUT.

- En ocasiones, cuando estemos analizando una asignación de status \mathbb{S} en particular, y no exista ambigüedad al respecto, emplearemos simplemente $IN([\Phi, h])$ y $OUT([\Phi, h])$, en lugar de $IN_{\mathbb{S}}([\Phi, h])$ y $OUT_{\mathbb{S}}([\Phi, h])$, respectivamente.

Definición 36 (Asignación de Status) Sea \mathcal{P} un programa DeLP y sea \mathbb{S} una asignación de status arbitraria para \mathcal{P} . Diremos que \mathbb{S} es una asignación de status propiamente dicha (o simplemente asignación de status), si para cada a-estructura maximal $[\Phi, h]$ de \mathcal{P} vale que:

- $IN_{\mathbb{S}}([\Phi, h]) = UNwg([\Phi, h], \Sigma)$, donde $\Sigma = \{-OUT_{\mathbb{S}}([\Psi, k]) \mid [\Psi, k] \text{ es a-estructura maximal que ataca parcialmente } [\Phi, h], \text{ y } \neg OUT_{\mathbb{S}}([\Psi, k]) \neq [\emptyset, \epsilon]\}$.
- $OUT_{\mathbb{S}}([\Phi, h]) = DNwg([\Phi, h], \Sigma')$, donde $\Sigma' = \{IN_{\mathbb{S}}([\Psi, k]) \mid [\Psi, k] \text{ es a-estructura maximal que ataca parcialmente } [\Phi, h], \text{ e } IN_{\mathbb{S}}([\Psi, k]) \neq [\emptyset, \epsilon]\}$.

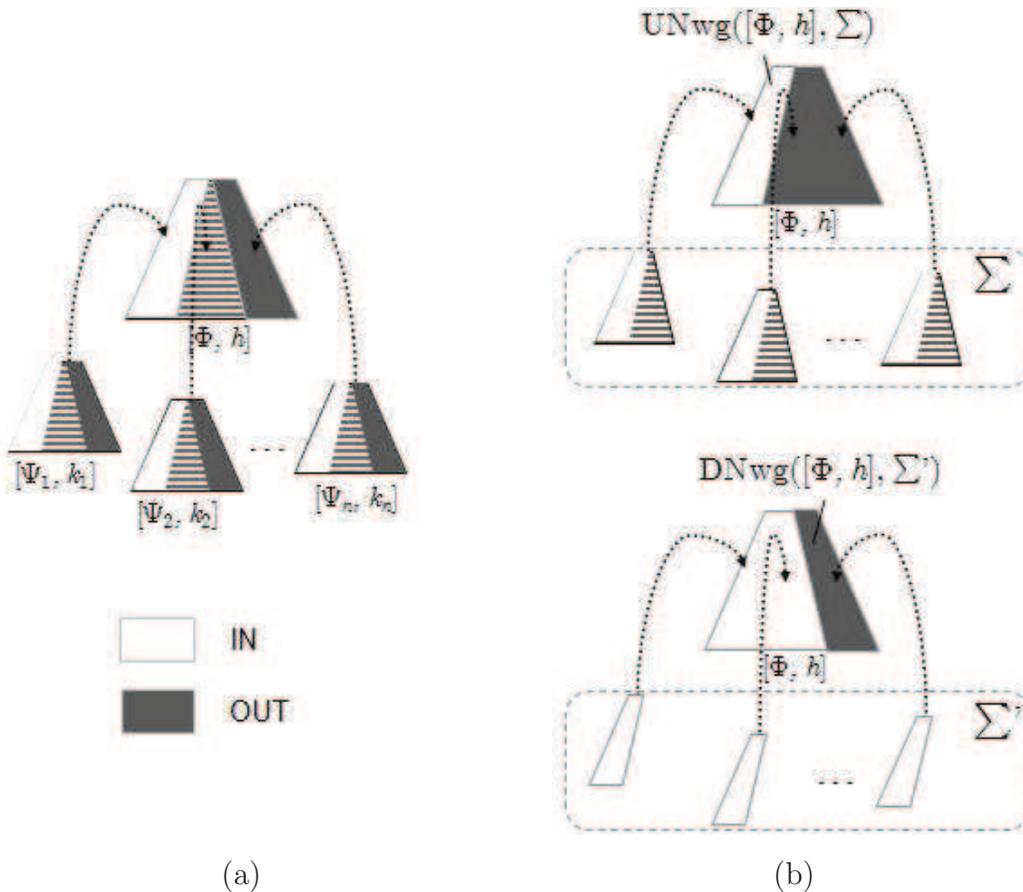


Figura 5.1: Asignación de Status: narrowings IN y OUT de una a-estructura $[\Phi, h]$

Intuitivamente, una asignación de status asocia narrowings IN y OUT a cada a-estructura maximal $[\Phi, h]$, respetando las interacciones (ataques y derrotas combinadas) entre a-estructuras: los narrowings IN y OUT de una a-estructura $[\Phi, h]$ se definen en términos de los narrowings IN y OUT de las a-estructuras que atacan combinadamente $[\Phi, h]$, empleando la noción de degradación secuencial. Para entender como una asignación de status asocia narrowings IN y OUT en base a la relación de ataque, consideremos una a-estructura $[\Phi, h]$ y las a-estructuras $[\Psi_1, k_1], \dots, [\Psi_m, k_m]$ que atacan $[\Phi, h]$. Como se muestra en el esquema de la Fig. 5.1, los narrowings IN y OUT de $[\Phi, h]$ se definen como resultado de considerar dos ataques combinados contra $[\Phi, h]$, cuya diferencia radica en la consideración o no de los narrowings *undecided* de los $[\Psi_i, k_i]$ en el ataque: para el cálculo del IN se considera el ataque de los narrowings \neg OUT (que incluyen los narrowings *undecided*) de los $[\Psi_i, k_i]$ (Σ) y para el OUT se considera el ataque de los narrowings IN de los $[\Psi_i, k_i]$ (Σ'). Un narrowing *undecided* denota un conjunto de razones o argumentos individuales que la asignación de status no establece si considerarlos IN o OUT. De esta forma, el ataque combinado del conjunto Σ , involucrando los narrowings \neg OUT, constituye el ataque combinado más “fuerte” que es posible concebir frente a la indecisión en los $[\Psi_i, k_i]$, y puede pensarse como el resultado de asumir que todo argumento *undecided* (en los $[\Psi_i, k_i]$) en realidad está IN. Luego resulta natural adoptar como narrowing IN de $[\Phi, h]$ a aquel que queda sin derrotar resultado de este ataque más fuerte ($UNwg([\Phi, h], \Sigma)$), dado que también quedará sin derrotar en cualquier otro ataque posible (más débil) que se pueda concebir dado los *undecided*. Análogamente, el ataque combinado del conjunto Σ' , que considera para cada $[\Psi_i, k_i]$ solo el narrowing IN, constituye el ataque combinado más “débil” que es posible concebir frente a la indecisión en los $[\Psi_i, k_i]$, y puede pensarse como el resultado de asumir que todo argumento *undecided* en realidad está OUT. Por lo tanto, resulta natural adoptar como narrowing OUT de $[\Phi, h]$ a aquel que queda derrotado resultado de este ataque más débil ($DNwg([\Phi, h], \Sigma')$), dado que también quedará derrotado (más precisamente, estará incluido en el narrowing derrotado) en cualquier otro ataque posible (más fuerte).

Otra forma de entender la asignación de status asociada a una a-estructura $[\Phi, h]$ a partir de los dos posibles ataques combinados extremos es considerando los argumentos individuales que conforman cada narrowing de $[\Phi, h]$. Los argumentos que conforman el narrowing IN de $[\Phi, h]$ son exactamente los que integran los U-narrowings de $[\Phi, h]$ en ambos ataques combinados extremos. Análogamente, los argumentos que conforman el narrowing OUT de $[\Phi, h]$ son exactamente los que integran los D-narrowings de $[\Phi, h]$

en ambos ataques combinados extremos. Finalmente, los argumentos que conforman el narrowing *undecided* son exactamente aquellos que integran el U-narrowing de acuerdo a uno de los ataques (el más débil), y el D-narrowing de acuerdo al otro (el más fuerte).

Cuando, de acuerdo a una asignación de status dada, no hay indecisión entre las a-estructuras que atacan, tampoco habrá indecisión en la a-estructura atacada. A continuación, esta propiedad se establece formalmente.

Proposición 7 *Sea \mathbb{S} una asignación de status para un programa dado \mathcal{P} . Sea $[\Phi, h]$ una a-estructura maximal de \mathcal{P} tal que para toda a-estructura $[\Psi_i, k_i]$ que ataca parcialmente a $[\Phi, h]$ vale que $IN_{\mathbb{S}}([\Psi_i, k_i]) \sqcup OUT_{\mathbb{S}}([\Psi_i, k_i]) = [\Psi_i, k_i]$. Entonces vale que $IN_{\mathbb{S}}([\Phi, h]) \sqcup OUT_{\mathbb{S}}([\Phi, h]) = [\Phi, h]$.*

Prueba. Al ser $IN_{\mathbb{S}}([\Psi_i, k_i])$ y $OUT_{\mathbb{S}}([\Psi_i, k_i])$ complementarios para todo $[\Psi_i, k_i]$, entonces $\Sigma = \Sigma'$ en la Def. 36 (dado que en ese caso $IN_{\mathbb{S}}([\Psi_i, k_i]) = [\Psi_i, k_i] \setminus OUT_{\mathbb{S}}([\Psi_i, k_i]) =_{def} \neg OUT_{\mathbb{S}}([\Psi_i, k_i])$). Por lo tanto $IN_{\mathbb{S}}([\Phi, h]) \sqcup OUT_{\mathbb{S}}([\Phi, h]) = UNwg([\Phi, h], \Sigma) \sqcup DNwg([\Phi, h], \Sigma) = [\Phi, h]$.

La Fig. 5.2 ilustra la definición de los narrowings IN y OUT de una a-estructura dada $[\Phi, h]$ (de acuerdo a la Def. 36) para el caso particular en el cual las a-estructuras que atacan $[\Phi, h]$ no presentan indecisión.

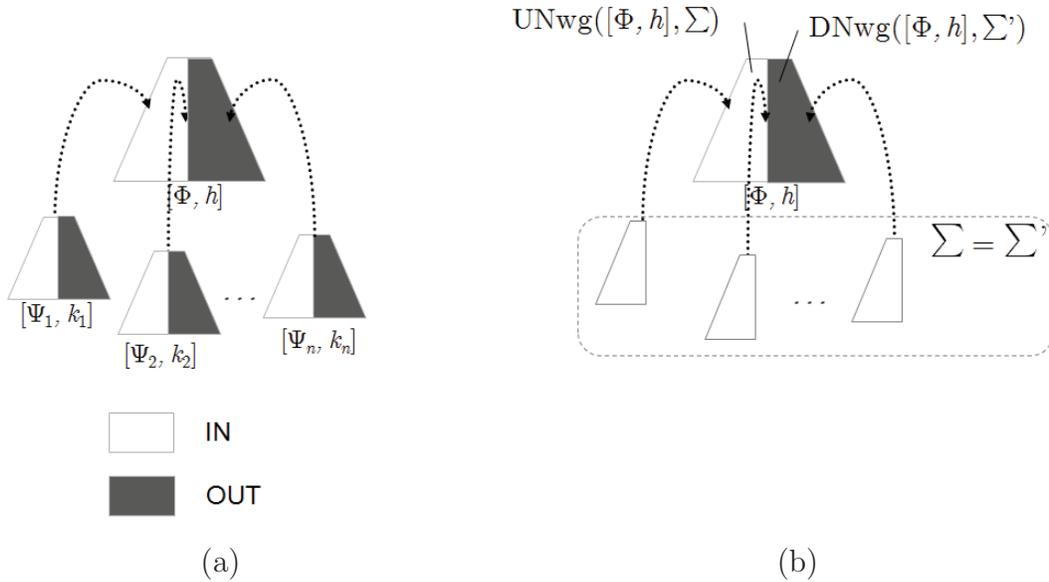


Figura 5.2: Caso particular: ausencia de indecisión en a-estructuras que atacan

Aclaraciones:

- Cuando el narrowing **IN** de una a-estructura dada $[\Phi, h]$ sea la propia a-estructura $[\Phi, h]$ diremos informalmente que $[\Phi, h]$ se encuentra *completamente IN*. De forma análoga usaremos la terminología informal *completamente OUT* y *completamente undecided*.
- En adelante hablaremos de grafo de ataques asociado a un programa \mathcal{P} para referirnos al grafo cuyos nodos representan todas las a-estructuras maximales de \mathcal{P} , y donde los arcos reflejan la relación de ataque parcial entre a-estructuras.
- Por practicidad, para los ejemplos presentaremos directamente grafos de ataques (de forma gráfica) en lugar de especificar programas, y hablaremos de “asignación de status para un grafo” para referirnos a una asignación de status para el programa denotado por el grafo.
- Finalmente, en las representaciones gráficas de los grafos de ataques se omitirá en ocasiones la representación de algunas a-estructuras maximales cuando éstas no participan de ataques (*i.e.*, cuando no atacan ni sean atacadas). Estas a-estructuras siempre estarán completamente **IN** cualquiera sea la asignación de status, y nunca afectarán la asignación de las demás a-estructuras. A modo de ejemplo, considere el grafo de ataques de la Fig. 5.3a (asociado a un programa dado \mathcal{P}). La a-estructura $[\Phi, a]$ es maximal, y por lo tanto la a-subestructura $[\{o \prec p\}, o]$ de $[\Phi, a]$ también es una a-estructura maximal. Sin embargo, dado que $[\{o \prec p\}, o]$ no participa en ningún ataque, se decidió omitirla.

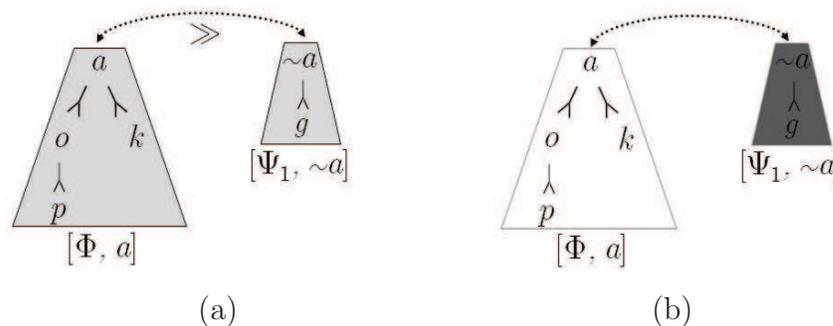


Figura 5.3: Asignación de Status para ataque ‘rebut’

5.1.1. Asignación de status única

Considere el grafo de ataques de la Fig. 5.3a (respecto a un programa dado \mathcal{P}), conformado por dos a-estructuras, $[\Phi, a]$ y $[\Psi_1, \sim a]$, soportando conclusiones contrarias, y por lo tanto que se atacan mutuamente (rebut), donde $[\Phi, a]$ es ‘más fuerte’ que $[\Psi_1, \sim a]$ de acuerdo a ‘ \gg ’. La Fig. 5.3b muestra la única asignación de status posible para dicho grafo, donde la a-estructura más fuerte ($[\Phi, a]$) está completamente **IN** y la más débil ($[\Psi_1, \sim a]$) está completamente **OUT**.

Verifiquemos primero que constituye efectivamente una asignación de status. Por un lado, $[\Phi, a]$ está completamente **IN** dado que la única a-estructura que la ataca, $[\Psi_1, \sim a]$, está completamente **OUT**. Formalmente, $\text{IN}([\Phi, a]) =_{def} \text{UNwg}([\Phi, a], \emptyset) = [\Phi, a]$. Por otro lado, $[\Psi_1, \sim a]$ está completamente **OUT** dado que es atacada por $[\Phi, a]$, cuyo narrowing **IN** (la propia a-estructura $[\Phi, a]$) es más fuerte que $[\Psi_1, \sim a]$ de acuerdo a ‘ \gg ’. Formalmente, $\text{OUT}([\Psi_1, \sim a]) =_{def} \text{DNwg}([\Psi_1, \sim a], \{\text{IN}([\Phi, a])\}) = \text{DNwg}([\Psi_1, \sim a], \{[\Phi, a]\}) = [\Psi_1, \sim a]$, considerando que $[\Phi, a] \gg [\Psi_1, \sim a]$. Además, puede comprobarse que es la única asignación de status posible. Nótese que cualquiera fueran los narrowings **IN** y **OUT** asignados a la a-estructura $[\Psi_1, \sim a]$ (la más débil), nunca lograría evitar que la a-estructura $[\Phi, a]$ (más fuerte) quede completamente **IN**. Formalmente, cualquiera sea $\neg\text{OUT}([\Psi_1, \sim a])$, vale que $\neg\text{OUT}([\Psi_1, \sim a]) \ll [\Psi_1, \sim a] \ll [\Phi, a]$, y por lo tanto $\text{IN}([\Phi, a]) =_{def} \text{UNwg}([\Phi, a], \{\neg\text{OUT}([\Psi_1, \sim a])\}) = [\Phi, a]$. Finalmente, si $[\Phi, a]$ está completamente **IN**, y dado que $[\Phi, a] \gg [\Psi_1, \sim a]$, necesariamente $[\Psi_1, \sim a]$ debe estar completamente **OUT**.

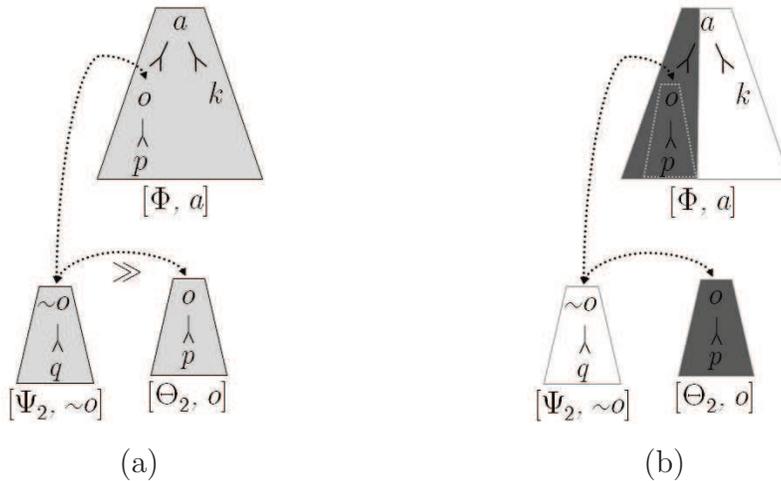


Figura 5.4: Asignación de Status

Considere el grafo de ataques de la Fig. 5.4a (respecto a un programa dado \mathcal{P}), donde la a-estructura $[\Psi_2, \sim o]$ ataca parcialmente a la a-estructura $[\Phi, a]$ en la conclusión intermedia ‘ o ’ (rebut de una a-subestructura). Dado que la a-subestructura de desacuerdo asociada con este ataque es en sí misma una a-estructura, también aparece como un nodo en el grafo ($[\Theta_2, o]$), entablando una relación de ataque mutuo con $[\Psi_2, \sim o]$. La Fig. 5.4b muestra la única asignación de status posible para dicho grafo, de acuerdo a la cual la a-estructura $[\Psi_2, \sim o]$ (completamente IN) prevalece sobre $[\Theta_2, o]$ (completamente OUT), causando la derrota del narrowing de $[\Phi, a]$ involucrando a $[\Theta_2, o]$.

Dado que $[\Psi_2, \sim o]$ y $[\Theta_2, o]$ se atacan mutuamente (en rebut), donde $[\Psi_2, \sim o]$ es preferida sobre $[\Theta_2, o]$, y ninguna otra a-estructura las afecta en forma directa ni indirecta, análogamente a lo explicado para el grafo de la Fig. 5.3, la única posibilidad es considerar a $[\Psi_2, \sim o]$ completamente IN y a $[\Theta_2, o]$ completamente OUT. Luego, los narrowings IN y OUT de $[\Phi, a]$ quedan determinados (en base a la Def. 36) por los narrowings asociados a la única a-estructura que la ataca: $[\Psi_2, \sim o]$. Informalmente, dado que el narrowing IN de $[\Psi_2, \sim o]$ (la propia $[\Psi_2, \sim o]$) es preferido a la a-subestructura de desacuerdo soportando o , el narrowing de $[\Phi, a]$ involucrando o queda OUT ($DNwg([\Phi, a], \{[\Psi_2, \sim o]\})$), mientras que el narrowing complementario queda IN ($UNwg([\Phi, a], \{[\Psi_2, \sim o]\})$).

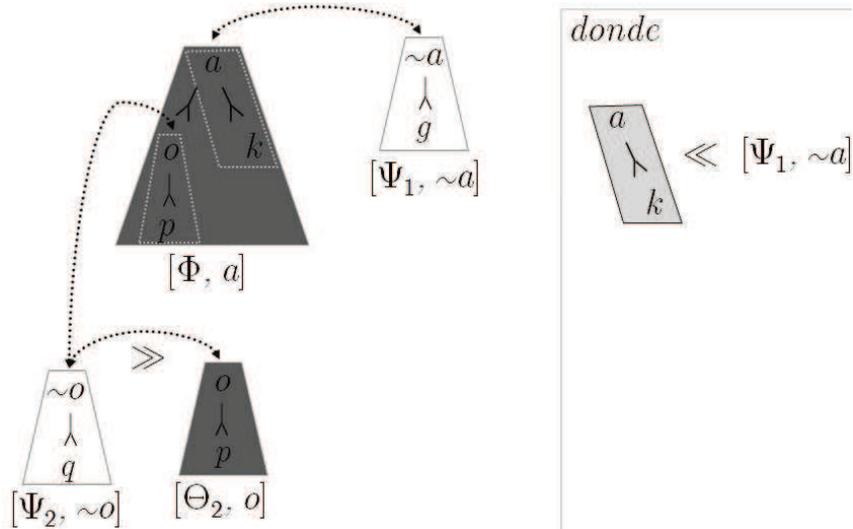


Figura 5.5: Asignación de Status

El grafo de ataques de la Fig. 5.5 resulta de combinar el grafo de la Fig. 5.3, donde la a-estructura $[\Phi, a]$ (completamente IN) prevalece sobre la a-estructura $[\Psi_1, \sim a]$ (completamente OUT), con el grafo de la Fig. 5.4, donde la a-estructura $[\Psi_2, \sim o]$ derrota parcialmente a $[\Phi, a]$. Adicionalmente, se muestra la única asignación de status posible

para el nuevo grafo, reflejando que las a-estructuras $[\Psi_2, \sim o]$ y $[\Psi_1, \sim a]$ logran, combinadamente, derrotar a $[\Phi, a]$ en su totalidad, dejándola completamente **OUT**. Nótese que de acuerdo al criterio de preferencia ‘ \gg ’ adoptado, la a-estructura $[\Psi_1, \sim a]$ es preferida ante el narrowing $\{\{a \prec k\}, a\}$ de $[\Phi, a]$ (especificado sobre el lado derecho de la Fig. 5.5).

A continuación verificaremos que la asignación cumple efectivamente con la Def. 36. Las a-estructuras $[\Psi_2, \sim d]$ y $[\Theta_2, d]$ se atacan mutuamente (en rebut), donde $[\Psi_2, \sim d]$ es preferida sobre $[\Theta_2, d]$, y ninguna otra a-estructura las afecta ni directa ni indirectamente, por lo tanto la única posibilidad es considerarlas completamente **IN** y completamente **OUT**, respectivamente, tal como se analizó en el ejemplo de la Fig. 5.3. La a-estructura $[\Phi, a]$ se encuentra completamente **OUT**, dado que queda completamente derrotada resultado del ataque combinado de $\text{IN}([\Psi_2, \sim o])(= [\Psi_2, \sim o])$ e $\text{IN}([\Psi_1, \sim a])(= [\Psi_1, \sim a])$. Por otro lado, la a-estructura $[\Psi_1, \sim a]$ se encuentra completamente **IN**, dado que todos sus atacantes ($[\Phi, a]$) se encuentran completamente **OUT** (formalmente, $\text{IN}([\Psi_1, \sim a]) =_{def} \text{UNwg}([\Psi_1, \sim a], \emptyset)$). Verifiquemos ahora que no existe otra asignación de status posible para el grafo. A partir del razonamiento efectuado en el ejemplo de la Fig. 5.3, resulta claro que no existe otra posibilidad de asignar narrowings **IN** y **OUT** a las a-estructuras $[\Psi_2, \sim o]$ y $[\Theta_2, o]$. Dado que $[\Psi_2, \sim o]$ está completamente **IN** y derrota parcialmente a $[\Phi, a]$, puede asegurarse que al menos el narrowing de $[\Phi, a]$ derrotado por $[\Psi_2, \sim o]$ estará **OUT**. Además, dado que de acuerdo a ‘ \gg ’ el narrowing restante de $[\Phi, a]$ ($\{\{a \prec k\}, a\}$) no constituye un derrotador para $[\Psi_1, \sim a]$, esta última a-estructura deberá asignarse completamente **IN**. Finalmente, $[\Phi, a]$ queda completamente derrotado por el ataque combinado de $\text{IN}([\Psi_2, \sim o])(= [\Psi_2, \sim o])$ e $\text{IN}([\Psi_1, \sim a])(= [\Psi_1, \sim a])$.



Figura 5.6: Reinstatement de argumentos

El ejemplo anterior ilustra un comportamiento similar al de la *reinstauración* (*reinstatement*) para argumentos individuales (Fig. 5.6), donde la a-estructura $[\Psi_1, \sim a]$, que resulta en principio derrotada por $[\Phi, a]$ (Fig. 5.3), es defendida o reinstaurada por $[\Psi_2, \sim o]$,

que derrota parcialmente y en consecuencia debilita a la a-estructura $[\Phi, a]$. Sin embargo, el reinstatement de estructuras agregadas es algo más complejo que para argumentos individuales, y esto tiene que ver con la noción de derrota combinada. Como muestra la Fig. 5.7, defender a una a-estructura dada $[\Phi, h]$ de una derrota por parte de una única a-estructura $[\Psi, k]$ podría requerir del ataque combinado de un conjunto de a-estructuras $[\Psi_1, k_1], \dots, [\Psi_n, k_n]$.

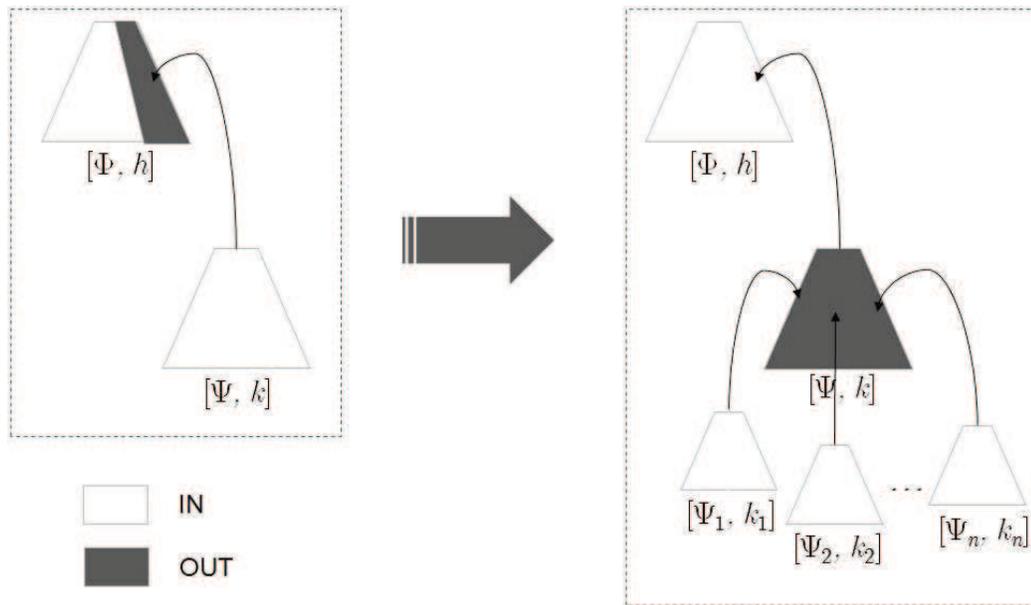


Figura 5.7: Reinstatement de a-estructuras

5.1.2. Situaciones de Bloqueo y múltiples asignaciones de status

Hasta ahora solo se han considerado grafos de ataques para los cuales solo existe una posible asignación de status, y que no presenta indecisión (narrowings *undecided*).

Consideremos ahora dos a-estructuras que soportan conclusiones complementarias, $[\Theta_3, c]$ y $[\Psi_3, \sim c]$, y por lo tanto que se atacan mutuamente (rebut), tal que son igualmente preferidas (o incomparables) de acuerdo a ‘ \gg ’. Análogo a lo que ocurre con las asignaciones de status para argumentos individuales (Sec. 3.1), esta situación dará lugar a tres asignaciones de status posibles, tal como se muestra en la Fig. 5.8: una de ellas (\mathbb{S}_1) considerando a ambas a-estructuras como *undecided* (asignación “menos comprometida”), y las otras dos (\mathbb{S}_2 y \mathbb{S}_3) considerando a una de las a-estructuras como completamente IN a expensas de la otra, considerada completamente OUT (estas asignaciones son “más

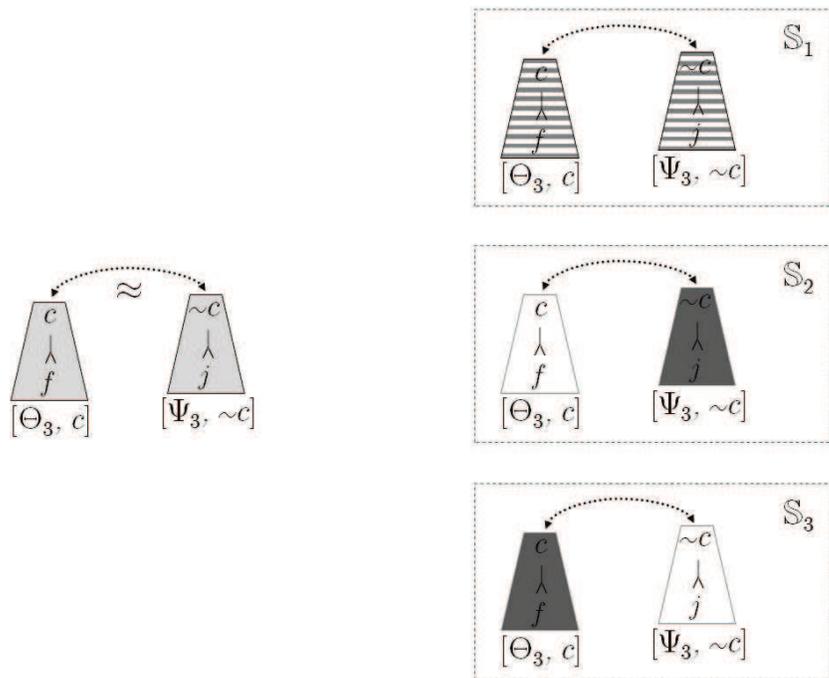


Figura 5.8: Asignación de Status

comprometidas”, representando formas arbitrarias y alternativas de resolver el bloqueo). Verifiquemos que \mathbb{S}_1 efectivamente respeta la Def. 36. Consideremos los narrowings IN y OUT de $[\Theta_3, c]$ (el razonamiento para los narrowings de $[\Psi_3, \sim c]$ es análogo). Por un lado, el narrowing IN de $[\Theta_3, c]$ ($[\emptyset, \epsilon]$) es en efecto el narrowing que queda *sin derrotar* resultado del ataque del \neg OUT de $[\Psi_3, \sim c]$ ($[\Psi_3, \sim c]$). Nótese que, de acuerdo a la Def. 32 de derrota parcial, ante igual preferencia, hay derrota. Por otro lado, el narrowing OUT de $[\Theta_3, c]$ ($[\emptyset, \epsilon]$) es en efecto el narrowing que queda *derrotado* resultado del ataque del IN de $[\Psi_3, \sim c]$ ($[\emptyset, \epsilon]$). Para las asignaciones \mathbb{S}_2 y \mathbb{S}_3 el razonamiento es análogo al del ejemplo de la Fig. 5.3, donde la a-estructura que está completamente OUT es, en efecto, derrotada por una a-estructura (de igual preferencia) que está completamente IN, y recíprocamente la a-estructura que está completamente IN tiene todos sus derrotadores completamente OUT.

El ejemplo anterior muestra el comportamiento de un bloqueo aislado. A continuación se analizará una situación de bloqueo en interacción con otras a-estructuras.

La Fig. 5.9 muestra un grafo de ataques junto con las tres asignaciones de status posibles (\mathbb{S}_1 , \mathbb{S}_2 y \mathbb{S}_3), donde, al igual que en el grafo de la Fig. 5.8, la multiplicidad de asignaciones es consecuencia de una situación de bloqueo. Las a-estructuras $[\Theta_3, c]$ y

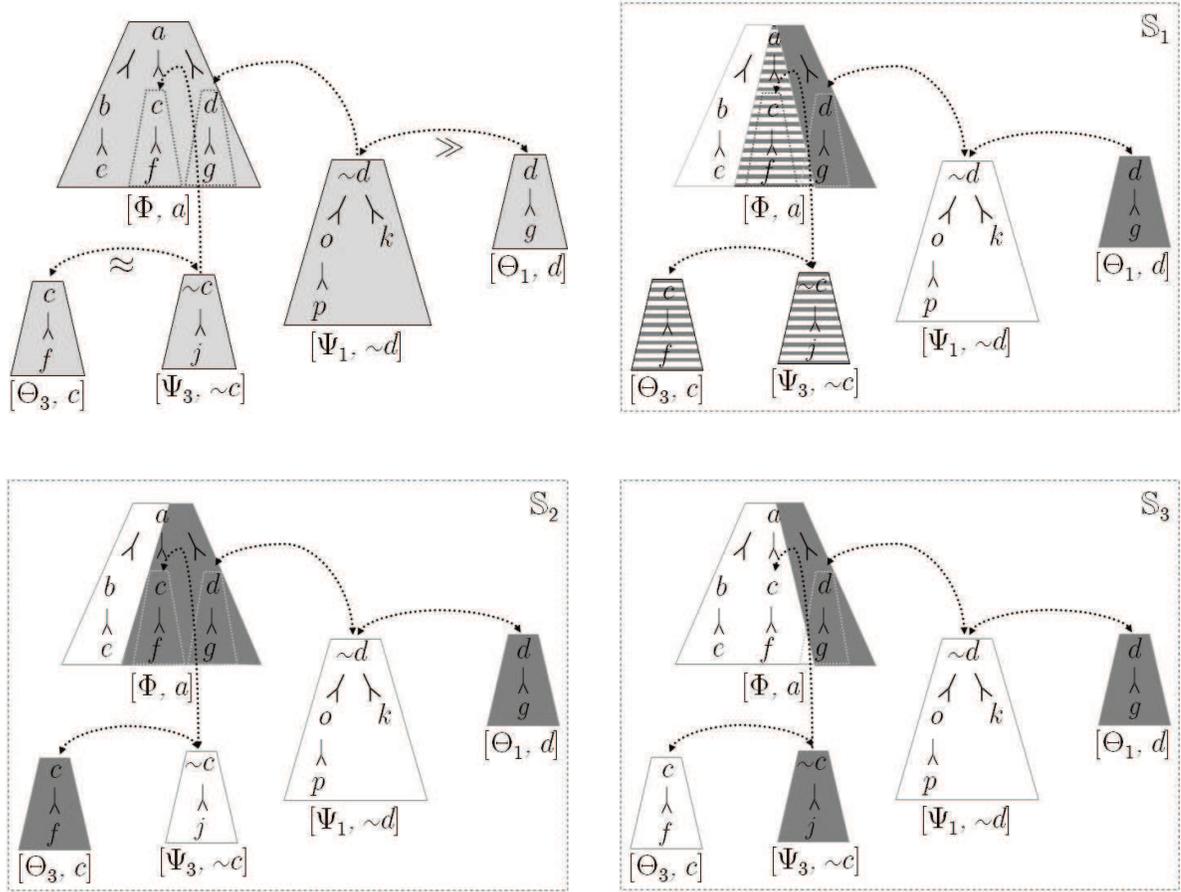


Figura 5.9: Asignación de Status

$[\Psi_3, \sim c]$ se atacan mutuamente, son igualmente preferidas, y ninguna otra a-estructura las afecta (bloqueo), por lo tanto se asignan de acuerdo a las tres posibilidades ilustradas en la Fig. 5.8, dando lugar a las tres asignaciones de status. La a-estructura $[\Phi, a]$ tiene como a-subestructura a una de las a-estructuras involucradas en el bloqueo ($[\Theta_3, c]$), causando que las tres asignaciones difieran también en el status asignado a $[\Phi, a]$. Concretamente, el narrowing $[\Phi', a] = \{(a \prec c), (c \prec f)\}, a$ de $[\Phi, a]$ (involucrando a $[\Theta_3, c]$) forma parte del narrowing IN en S_3 , del OUT en S_2 , y de ninguno de los dos (*undecided*) en S_1 , guardando correspondencia con lo asignado a $[\Theta_3, c]$. Considerando la definición de asignación de status, resulta evidente cómo el status asignado a $[\Psi_3, \sim c]$ en las asignaciones S_2 y S_3 determina el status del narrowing $[\Phi', a]$ (afectado por $[\Psi_3, \sim c]$) en cada caso. Respecto a la asignación S_1 , el narrowing $[\Phi', a]$ no puede asignarse como IN dado que es derrotado por el narrowing \sim OUT de $[\Psi_3, \sim c]$ (la propia $[\Psi_3, \sim c]$), y tampoco puede asignarse como OUT porque no es derrotado por el narrowing IN de $[\Psi_3, \sim c]$ ($[\emptyset, \epsilon]$).

Finalmente, las situaciones de bloqueo no siempre se dan de manera tan evidente, sino que pueden surgir al considerar otras a-estructuras, como ilustra el caso de la Fig. 5.10. La derrota de $[\Psi_2, \sim o]$ contra $[\Phi, a]$ hace emerger el bloqueo entre el narrowing no derrotado de $[\Phi, a]$ y $[\Psi_1, \sim a]$.

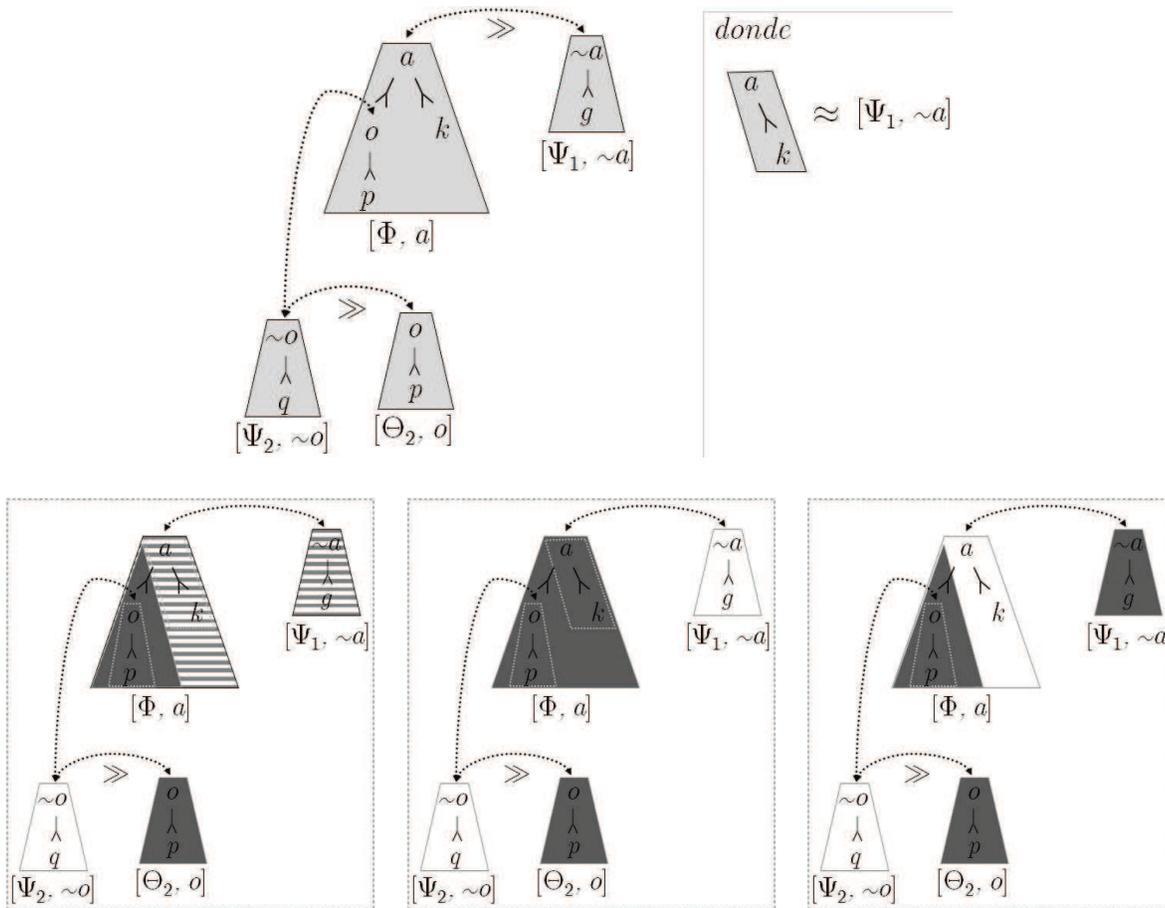


Figura 5.10: Bloqueo implícito

5.1.3. Ciclos de longitud impar

En esta sección analizaremos cómo se comporta la noción de asignación de status frente a ciclos de longitud impar. La Fig. 5.11 muestra un grafo donde tres a-estructuras, $[\Psi_1, \sim a]$, $[\Psi_2, \sim b]$ y $[\Psi_3, \sim c]$, se atacan conformando un ciclo (de longitud impar). Nótese que las a-subestructuras de desacuerdo asociadas a cada uno de los tres ataques del ciclo aparecen también como nodos en el grafo. Además, de acuerdo al criterio de preferencia '»' los tres ataques del ciclo constituyen derrotas. Análogo a lo que ocurría para argu-

mentos individuales, la única asignación de status posible para este grafo es aquella que considera a todas las a-estructuras completamente *undecided* (Fig. 5.11). Si por ejemplo consideráramos completamente IN a $[\Psi_1, \sim a]$, dado que $[\Psi_1, \sim a]$ derrota a $[\Psi_2, \sim b]$, esta última quedará completamente OUT. Luego, por ser $[\Psi_2, \sim b]$ el único derrotador de $[\Psi_3, \sim c]$, esta última quedará completamente IN. Dado que $[\Psi_3, \sim c]$ está completamente IN y derrota a $[\Psi_1, \sim a]$, esta última estará OUT, arribando a una contradicción. Por otro lado, si consideráramos completamente OUT a $[\Psi_1, \sim a]$, mediante un razonamiento análogo al anterior arribaríamos a una contradicción. En conclusión, la única posibilidad es considerar a $[\Psi_1, \sim a]$ como *undecided*, y consecuentemente también a $[\Psi_2, \sim b]$ y $[\Psi_3, \sim c]$.

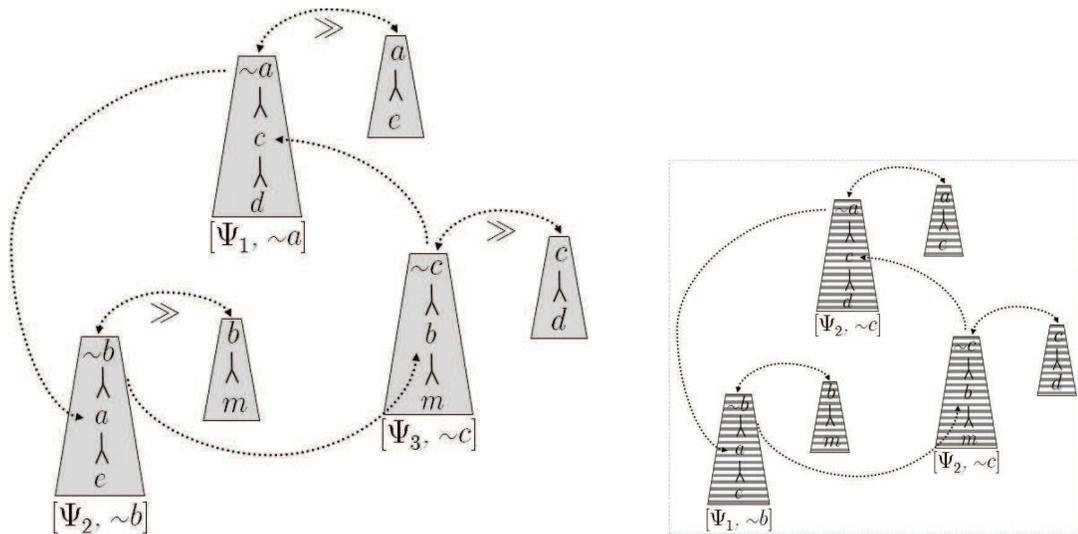


Figura 5.11: Ciclo impar y única asignación de status

Por supuesto, un ciclo impar podría no implicar indecisión. Considérese que se incorpora una nueva razón (argumento) para $\sim a$ al grafo de la Fig. 5.11, tal que por sí misma supera a la única razón para a ($a \prec c$). Como muestra la Fig. 5.12, el ciclo de longitud impar resultante ya no da lugar a indecisión.

5.1.4. Orden parcial de asignaciones de status por grado de compromiso

A continuación se define una relación de orden parcial ' \preceq ' entre asignaciones de status reflejando el grado de compromiso relativo entre pares de asignaciones. Esta relación es

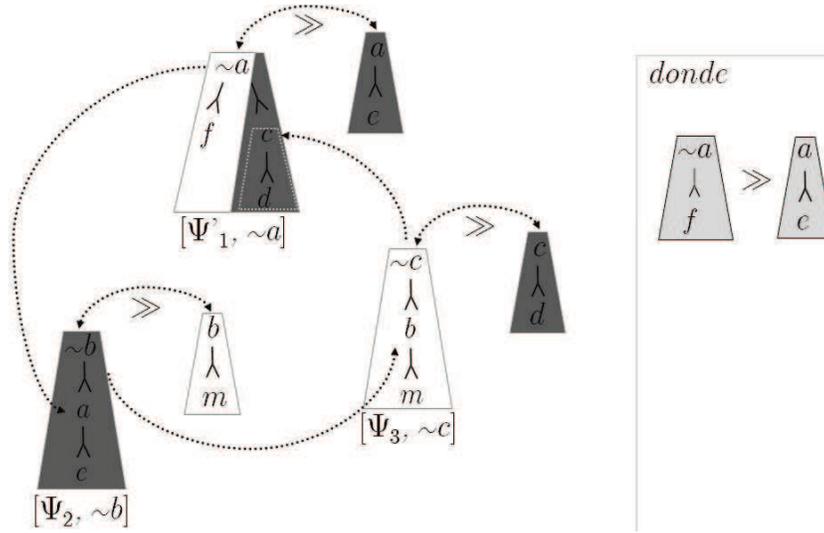


Figura 5.12: Ciclo impar sin indecisión

análoga a la definida en la Sec. 3.1 para asignaciones de status para argumentos, y se empleará para definir las distintas semánticas de agregación.

Definición 37 Sean \mathbb{S}_1 y \mathbb{S}_2 dos asignaciones de status arbitrarias para un programa dado \mathcal{P} . Diremos que \mathbb{S}_1 es tanto o menos comprometida que \mathbb{S}_2 (o que \mathbb{S}_2 es tanto o más comprometida que \mathbb{S}_1), notado $\mathbb{S}_1 \preceq \mathbb{S}_2$, si para toda a-estructura maximal $[\Phi, h]$ de \mathcal{P} vale que $IN_{\mathbb{S}_1}([\Phi, h]) \subseteq IN_{\mathbb{S}_2}([\Phi, h])$ y $OUT_{\mathbb{S}_1}([\Phi, h]) \subseteq OUT_{\mathbb{S}_2}([\Phi, h])$.

Intuitivamente, una asignación \mathbb{S}_2 es más comprometida que otra \mathbb{S}_1 si puede obtenerse a partir de esta última “ampliando” narrowings IN y OUT asignados a a-estructuras. En otras palabras, razones individuales consideradas *undecided* de acuerdo a \mathbb{S}_1 (i.e., formando parte de narrowings *undecided*), son consideradas IN o OUT de acuerdo a \mathbb{S}_2 (i.e., forman parte de narrowings IN o OUT).

Ejemplo 17 Considere las asignaciones de status \mathbb{S}_1 , \mathbb{S}_2 y \mathbb{S}_3 de la Fig. 5.9. Entonces vale que $\mathbb{S}_1 \prec \mathbb{S}_2$ y $\mathbb{S}_1 \prec \mathbb{S}_3$. Además vale que $\mathbb{S}_2 \not\prec \mathbb{S}_3$ y $\mathbb{S}_3 \not\prec \mathbb{S}_2$.

Proposición 8 (orden parcial) La relación ‘ \preceq ’ es un orden parcial. Es decir, sean \mathbb{S}_1 , \mathbb{S}_2 y \mathbb{S}_3 asignaciones de status arbitrarias para un programa dado \mathcal{P} , entonces vale que:

- $\mathbb{S}_1 \preceq \mathbb{S}_1$ (reflexividad),

- si $\mathbb{S}_1 \preceq \mathbb{S}_2$ y $\mathbb{S}_2 \preceq \mathbb{S}_1$, entonces $\mathbb{S}_1 = \mathbb{S}_2$ (antisimetría), y
- si $\mathbb{S}_1 \preceq \mathbb{S}_2$ y $\mathbb{S}_2 \preceq \mathbb{S}_3$, entonces $\mathbb{S}_1 \preceq \mathbb{S}_3$ (transitividad).

La demostración de esta proposición es trivial a partir de la definición de ‘ \preceq ’.

5.2. Semántica A-Grounded

Definición 38 (Asignación de Status A-Grounded) Sea \mathbb{S} una asignación de status para un programa dado \mathcal{P} . Diremos que \mathbb{S} es una asignación de status a-grounded si $\mathbb{S} \preceq \mathbb{S}'$ para toda otra asignación de status \mathbb{S}' para \mathcal{P} (es decir, \mathbb{S} es la menos comprometida de todas las asignaciones de status).

Observación: Similar a lo establecido para la semántica grounded (para argumentos individuales), puede asegurarse que para cualquier grafo de ataques (o programa \mathcal{P}) existe siempre exactamente una asignación de status a-grounded. En particular, la unicidad de asignación de status a-grounded se sigue trivialmente a partir de la definición y de la antisimetría de ‘ \preceq ’ (proposición 8), y para grafos con cantidad finita de a-estructuras la existencia de asignación de status a-grounded se establece y demuestra en la Sec. 5.5 (proposición 13).

Intuitivamente, la asignación de status a-grounded es la menos comprometida de todas las asignaciones de status, es decir, aquella que presenta el compromiso mínimo e indispensable para respetar las relaciones de ataque y derrota entre a-estructuras.

Particularmente, si un grafo dado posee una única asignación de status, entonces dicha asignación de status será a-grounded. A modo de ejemplo, los grafos de las Figs. 5.3, 5.4 y 5.5 tienen, cada uno, una única asignación de status, y por lo tanto, a-grounded. Consideremos el caso de un grafo con múltiples asignaciones de status consecuencia de una situación de bloqueo (ataque mutuo entre dos a-estructuras de igual preferencia). La asignación a-grounded es aquella que adopta la postura menos comprometida, considerando a ambas a-estructuras involucradas en el bloqueo como completamente *undecided*. La asignación de status \mathbb{S}_1 de la Fig. 5.8 es a-grounded, considerando a ambas a-estructuras involucradas en el bloqueo como *undecided*. Nótese que $\mathbb{S}_1 \preceq \mathbb{S}_i$, $i \in \{1, 2, 3\}$. Análogamente, la asignación de status \mathbb{S}_1 de la Fig. 5.9 es a-grounded, de acuerdo a la cual las

a-estructuras $[\Theta_3, c]$ y $[\Psi_3, \sim c]$, en situación de bloqueo, se consideran completamente *undecided*, determinando así que un narrowing no vacío de la a-estructura $[\Phi, a]$ también quede *undecided*.

Las semánticas de aceptabilidad para argumentos individuales (presentadas en la Sec. 3.2) clasifican los argumentos de acuerdo a tres categorías disjuntas: argumentos garantizados (o aceptados), discutibles (denotando indecisión), y rechazados (o decididamente no aceptados). Asimismo una semántica de aceptabilidad para agregación establecerá, para cada a-estructura maximal $[\Phi, h]$, tres narrowings disjuntos de $[\Phi, h]$, denominados narrowings garantizado, discutible y rechazado. A continuación se define la semántica a-grounded, constituyendo la versión para agregación de la semántica grounded:

Definición 39 (Semántica A-Grounded) *Sea \mathcal{P} un programa DeLP y sea $[\Phi, h]$ una a-estructura maximal de \mathcal{P} . Sea \mathbb{S} la asignación de status a-grounded asociada a \mathcal{P} . Diremos que, de acuerdo a la semántica a-grounded,*

- $IN_{\mathbb{S}}([\Phi, h])$ es el narrowing garantizado de $[\Phi, h]$,
- $OUT_{\mathbb{S}}([\Phi, h])$ es el narrowing rechazado de $[\Phi, h]$,
- $[\Phi, h] \setminus\setminus (IN_{\mathbb{S}}([\Phi, h]) \sqcup OUT_{\mathbb{S}}([\Phi, h]))$ es el narrowing discutible de $[\Phi, h]$.

En adelante, cuando el narrowing garantizado de una a-estructura dada $[\Phi, h]$ sea la propia a-estructura $[\Phi, h]$ diremos informalmente que $[\Phi, h]$ se encuentra *completamente garantizada*. De forma análoga usaremos la terminología informal *completamente rechazada* y *completamente discutible*.

Considere el grafo de la Fig. 5.3. De acuerdo a la semántica a-grounded, la a-estructura $[\Phi, a]$ se encuentra completamente garantizada y $[\Psi_1, \sim a]$ completamente rechazada. Para el grafo de la Fig. 5.4, $[\{a \prec k\}, a]$ es el narrowing garantizado de $[\Phi, a]$, $[\{(a \prec o), (o \prec p)\}, a]$ es el narrowing rechazado de $[\Phi, a]$ y $[\emptyset, \epsilon]$ es el narrowing discutible de $[\Phi, a]$, de acuerdo a la semántica a-grounded. Para el grafo de la Fig. 5.8 (situación de bloqueo), tanto $[\Theta_3, c]$ como $[\Psi_3, \sim c]$ son considerados completamente discutibles de acuerdo a la semántica a-grounded. Finalmente, para el grafo de la Fig. 5.9, con asignación de status a-grounded \mathbb{S}_1 , los narrowings garantizado, discutible y rechazado asociados a $[\Phi, a]$ son $[\{(a \prec b), (b \prec c)\}, a]$, $[\{(a \prec c), (c \prec f)\}, a]$ y $[\{(a \prec d), (d \prec g)\}, a]$, respectivamente.

5.3. Semántica A-Estable

A continuación se presenta la semántica *a-estable*. A diferencia de la semántica *a-grounded*, que se define en base a la asignación de status con menor grado de compromiso, la semántica *a-estable* considera las distintas alternativas de asignación “totalmente” comprometidas (*i.e.*, sin narrowings *undecided*), y que surgen de asumir la resolución de cada situación de bloqueo en una u otra dirección. Luego, la semántica *a-estable* establece como narrowing garantizado para una *a-estructura* maximal $[\Phi, h]$ a la intersección de los narrowings IN de $[\Phi, h]$ en todas las posibles asignaciones totalmente comprometidas.

Definición 40 (Asignación de Status A-Estable) *Sea \mathbb{S} una asignación de status para un programa dado \mathcal{P} . Diremos que \mathbb{S} es una asignación de status *a-estable* si $IN_{\mathbb{S}}([\Phi, h]) \sqcup OUT_{\mathbb{S}}([\Phi, h]) = [\Phi, h]$ para toda *a-estructura* maximal $[\Phi, h]$ de \mathcal{P} (es decir, \mathbb{S} es totalmente comprometida).*

Definición 41 (Semántica A-Estable) *Sea \mathcal{P} un programa DeLP y sea $[\Phi, h]$ una *a-estructura* maximal de \mathcal{P} . Sea Ω el conjunto de todas las asignaciones de status *a-estables* asociadas a \mathcal{P} . Diremos que, de acuerdo a la semántica *a-estable*,*

- $[\Phi', h] = \bigcap_{\mathbb{S} \in \Omega} IN_{\mathbb{S}}([\Phi, h])$ es el narrowing garantizado de $[\Phi, h]$,
- $[\Phi'', h] = \bigcap_{\mathbb{S} \in \Omega} OUT_{\mathbb{S}}([\Phi, h])$ es el narrowing rechazado de $[\Phi, h]$,
- $[\Phi, h] \setminus\setminus ([\Phi', h] \sqcup [\Phi'', h])$ es el narrowing discutible de $[\Phi, h]$.

El grafo de la Fig. 5.5 tiene una única asignación de status, que es a la vez *a-grounded* y *a-estable*. Obviamente ambas semánticas coinciden para este grafo, definiendo los mismos narrowings garantizados, rechazados y discutibles para todas las *a-estructuras* maximales. El grafo de la Fig. 5.8 tiene dos asignaciones de status *a-estables*, \mathbb{S}_2 y \mathbb{S}_3 . Dado que tanto $[\Theta_3, c]$ como $[\Psi_3, \sim c]$ son asignados completamente IN en una de las asignaciones y completamente OUT en la otra, ambos son considerados completamente discutibles por la semántica *a-estable*. Analicemos formalmente el caso de $[\Theta_3, c]$ (el análisis para $[\Psi_3, \sim c]$ es análogo.) El narrowing garantizado de $[\Theta_3, c]$ es $IN_{\mathbb{S}_2}([\Theta_3, c]) \cap IN_{\mathbb{S}_3}([\Theta_3, c]) = [\emptyset, \epsilon]$ y el narrowing rechazado de $[\Theta_3, c]$ es $OUT_{\mathbb{S}_2}([\Theta_3, c]) \cap OUT_{\mathbb{S}_3}([\Theta_3, c]) = [\emptyset, \epsilon]$, y por lo tanto el narrowing discutible de $[\Theta_3, c]$ es $[\Theta_3, c] \setminus\setminus ([\emptyset, \epsilon] \sqcup [\emptyset, \epsilon]) = [\Theta_3, c]$. Nuevamente las semánticas *a-grounded* y *a-estable* coinciden para este grafo.

Consideremos el grafo de la Fig. 5.9, con dos asignaciones de status estables asociadas: \mathbb{S}_2 y \mathbb{S}_3 . El narrowing garantizado de $[\Phi, a]$ es $\text{IN}_{\mathbb{S}_2}([\Phi, a]) \sqcap \text{IN}_{\mathbb{S}_3}([\Phi, a]) = [\{(a \prec b), (b \prec c)\}, a]$, el narrowing rechazado de $[\Phi, a]$ es $\text{OUT}_{\mathbb{S}_2}([\Phi, a]) \sqcap \text{OUT}_{\mathbb{S}_3}([\Phi, a]) = [\{(a \prec d), (d \prec g)\}, a]$ y el narrowing discutible de $[\Phi, a]$ es $[\Phi, a] \setminus\setminus ([\{(a \prec b), (b \prec c)\}, a] \sqcup [\{(a \prec d), (d \prec g)\}, a]) = [\{(a \prec c), (c \prec f)\}, a]$. Una vez más las semánticas a-grounded y a-estable coinciden para este grafo.

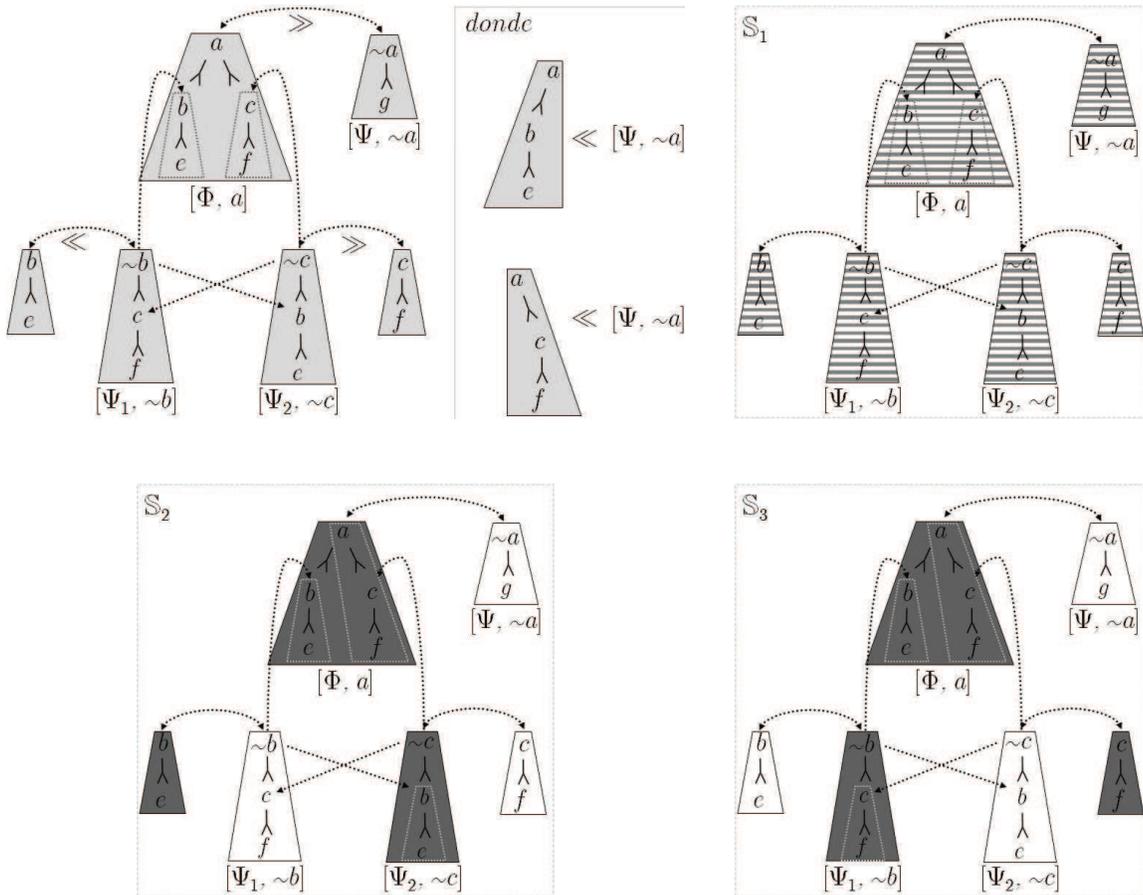


Figura 5.13: Asignación de Status

La Fig. 5.13 ilustra una situación donde las semánticas a-grounded y a-estable difieren. Esta situación puede verse como una versión para agregación de la situación conocida como *argumentos flotantes* presentada en la Sec. 3.2, para la cual las semánticas *grounded* y *estable* difieren. De acuerdo a la semántica a-grounded (determinada por la asignación \mathbb{S}_1), todos los argumentos son discutibles. Nótese que $[\Psi_1, \sim b]$ y $[\Psi_2, \sim c]$ se encuentran en situación de bloqueo (por derrota cruzada) y por lo tanto la asignación de status menos comprometida (\mathbb{S}_1) los considerará a ambos completamente *undecided*. Luego, la

indecisión en $[\Psi_1, \sim b]$ y $[\Psi_2, \sim c]$ determina que tanto $[\Phi, a]$ como $[\Psi, \sim a]$ sean consideradas completamente *undecided*, y la razón se explica informalmente a continuación. De acuerdo a la definición de asignación de status, el status de una a-estructura dada se determina en base a dos ataques, que surgen de asumir posiciones extremas respecto a la indecisión en los atacantes: considerar los narrowings *undecided* de los atacantes como OUT (ataque más débil), o como IN (ataque más fuerte). Dado que las a-estructuras $[\Psi_1, \sim b]$ y $[\Psi_2, \sim c]$ atacando a $[\Phi, a]$ se encuentran completamente *undecided*, en uno de los extremos se considerarán ambas completamente OUT (ataque más débil), y en este caso $[\Phi, a]$ quedará completamente no derrotada, independientemente de lo asignado a $[\Psi, \sim a]$. En el otro extremo, tanto $[\Psi_1, \sim b]$ como $[\Psi_2, \sim c]$ se considerarán completamente IN (ataque más fuerte), provocando que $[\Phi, a]$ quede completamente derrotada. Dado que $[\Phi, a]$ queda completamente no derrotada resultado del ataque más débil, y completamente derrotada de acuerdo al ataque más fuerte, entonces se encuentra completamente *undecided*, determinando la misma asignación para $[\Psi, \sim a]$

Consideremos ahora la semántica a-estable. El grafo presenta dos asignaciones de status a-estables, S_1 y S_2 , resultado de resolver el bloqueo alternativamente en favor de $[\Psi_1, \sim b]$ o $[\Psi_2, \sim c]$. En ambas asignaciones, una de la a-estructuras del ataque cruzado (aquella considerada IN) logra debilitar a $[\Phi, a]$, que termina siendo derrotada por $[\Psi, \sim a]$. De esta forma, de acuerdo a la semántica a-estable $[\Phi, a]$ se encuentra completamente rechazado y $[\Psi, \sim a]$ completamente garantizado, a pesar de que $[\Psi_1, \sim b]$ y $[\Psi_2, \sim c]$ se consideran completamente discutibles.

La razón de esta diferencia entre las dos semánticas se explica a continuación. La semántica a-grounded se basa en la asignación de status menos comprometida, y por lo tanto, que maximiza los narrowings *undecided*. Además, cuando existe indecisión entre los atacantes de una a-estructura dada, el status de esta última se determina considerando dos ataques extremos respecto a la indecisión en los atacantes. Esta estrategia supone independencia entre los narrowings *undecided* de los atacantes, concretamente, supone que es posible concebir una situación donde todos los *undecided* se encuentren IN o todos se encuentren OUT. Sin embargo, cuando existe dependencia entre los narrowings *undecided*, estas situaciones extremas podrían no ser viables en realidad. Más precisamente, la suposición de independencia frente a los *undecided* en un ataque lleva a considerar ataques posiblemente más extremos que los reales, lo que conduce a una mayor indecisión (de la necesaria) en la a-estructura atacada. La semántica a-estable soluciona esta situación

considerando todas las posibilidades de asignación totalmente comprometidas para los atacantes, en lugar de asumir dos posturas extremas respecto a los *undecided*.

La semántica a-estable hereda de la semántica estable la limitación para manejar ciclos impares: dado que generalmente un ciclo impar solo admite que sus miembros sean considerados *undecided*, un grafo conteniendo un ciclo impar podría no tener asignación de status a-estable, causando la ausencia de respuestas, aún cuando existen a-estructuras que el ciclo no afecta. El grafo de la Fig. 5.14 ilustra esta situación, donde la presencia de un ciclo impar causa la inexistencia de asignación de status a-estable, y la consecuente ausencia de respuesta, incluso acerca de las a-estructuras $[\Phi, a]$ y $[\Psi_1, \sim a]$, no afectadas por el ciclo impar.

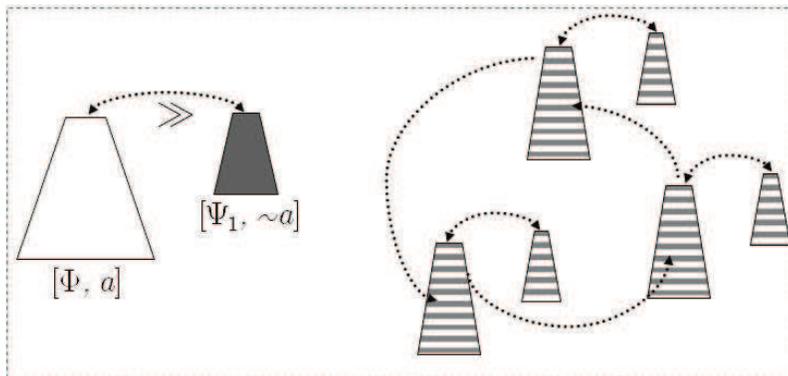


Figura 5.14: Única asignación de status, no a-estable

Además de la ausencia de respuestas, y análogamente a lo explicado en la Sec. 3.2 para la semántica estable, la interacción de ciclos impares con ciclos pares puede provocar que la semántica a-estable brinde respuestas no seguras (es decir, que reflejan una tendencia o preferencia arbitraria frente a situaciones de bloqueo). La Fig. 5.15 muestra las tres asignaciones de status asociadas a un grafo involucrando un ciclo impar, donde uno de las a-estructuras del ciclo es derrotada por una a-estructura $[\Theta_1, a]$, que a su vez se encuentra en situación de bloqueo con otra a-estructura $[\Theta_2, \sim a]$. Es claro que $[\Theta_1, a]$ y $[\Theta_2, \sim a]$ deberían considerarse completamente discutibles, dado que se derrotan mutuamente y no son afectados por ningún otro argumento del grafo (de hecho así ocurre de acuerdo a la semántica grounded). Sin embargo, por ser \mathbb{S}_2 la única asignación de status a-estable, de acuerdo a esta semántica $[\Theta_1, a]$ se considera completamente garantizada y $[\Theta_2, \sim a]$ completamente rechazada.

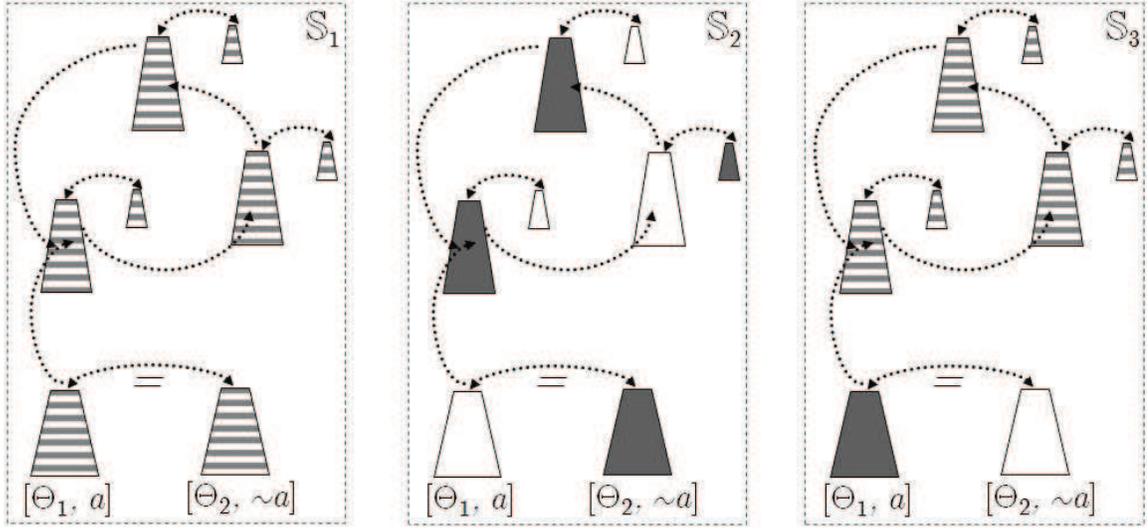


Figura 5.15: Única asignación a-estable: S_2 .

5.4. Semántica A-Preferida

A continuación se presenta la semántica *a-preferida*, que soluciona las limitaciones de la a-estable relajando el requerimiento de considerar asignaciones de status *totalmente* comprometidas a considerar asignaciones de status *maximalmente* comprometidas, permitiéndole de esta forma abordar situaciones con a-estructuras imposibles de asignar (ciclos impares).

Definición 42 (Asignación de Status A-Preferida) Sea S una asignación de status para un programa dado \mathcal{P} . Diremos que S es una asignación de status *a-preferida* si S es maximal con respecto a ' \preceq ' (o maximalmente comprometida), es decir, si no existe asignación de status S' tal que $S' \succ S$.

Definición 43 (Semántica A-Preferida) Sea \mathcal{P} un programa DeLP y sea $[\Phi, h]$ una a-estructura maximal de \mathcal{P} . Sea Ω el conjunto de todas las asignaciones de status a-preferidas asociadas a \mathcal{P} . Diremos que, de acuerdo a la semántica a-preferida,

- $[\Phi', h] = \bigcap_{S \in \Omega} IN_S([\Phi, h])$ es el narrowing garantizado de $[\Phi, h]$,
- $[\Phi'', h] = \bigcap_{S \in \Omega} OUT_S([\Phi, h])$ es el narrowing rechazado de $[\Phi, h]$,
- $[\Phi, h] \parallel ([\Phi', h] \sqcup [\Phi'', h])$ es el narrowing discutible de $[\Phi, h]$.

La asignación de status mostrada en la Fig. 5.14 (única para el grafo, y no a-estable) es a-preferida, dado que no existe otra asignación posible más comprometida (*i.e.*, es maximalmente comprometida). Luego, de acuerdo a la semántica a-preferida la a-estructura $[\Phi, a]$ se considera completamente garantizada, $[\Psi_1, \sim a]$ completamente rechazada y las a-estructuras involucradas en el ciclo impar, completamente discutibles.

Considérese el caso de la Fig. 5.15. Aunque solo la asignación \mathbb{S}_2 es a-estable, tanto \mathbb{S}_2 como \mathbb{S}_3 son a-preferidas, evitando así que prevalezca una de las posibilidades de resolución del bloqueo y se brinden respuestas “no seguras”: todas las a-estructuras del grafo se consideran completamente discutibles de acuerdo a la semántica a-preferida.

Finalmente, para la situación de la Fig. 5.13 las semánticas a-preferida y a-estable coinciden, difiriendo de la a-grounded.

Proposición 9 *Sea \mathcal{P} un programa DeLP. Toda asignación de status a-estable \mathbb{S} para \mathcal{P} es una asignación de status a-preferida¹.*

5.5. Caracterización de la semántica A-Grounded basada en la noción de punto Fijo

En esta sección se propone una caracterización alternativa para la semántica a-grounded basada en la noción de punto fijo, inspirada en la caracterización original de la semántica *grounded* desarrollada por Dung ([Dun95]) y presentada en la Sec. 3.2.3. En primer lugar se presenta un operador sobre asignaciones de status arbitrarias (Def. 35) denominado operador característico, similar a la función característica presentada en [Dun95].

Definición 44 (Operador $F_{\mathcal{P}}$) *Sea \mathcal{P} un programa DeLP y sea \mathbb{S} una asignación de status arbitraria para \mathcal{P} . El operador $F_{\mathcal{P}}(\cdot)$ se define como sigue: $F_{\mathcal{P}}(\mathbb{S})$ es una asignación de status arbitraria tal que, para toda a-estructura maximal $[\Phi, h]$ de \mathcal{P} ,*

- $IN_{F_{\mathcal{P}}(\mathbb{S})}([\Phi, h]) = UNwg([\Phi, h], \Sigma)$, donde $\Sigma = \{\neg OUT_{\mathbb{S}}([\Psi, k]) \mid [\Psi, k] \text{ es a-estructura maximal que ataca parcialmente } [\Phi, h], \text{ y } \neg OUT_{\mathbb{S}}([\Psi, k]) \neq [\emptyset, \epsilon]\}$, y

¹La demostración es directa, ya que una asignación de status ‘totalmente comprometida’ es en particular ‘maximalmente comprometida’.

- $OUT_{F_{\mathcal{P}}(\mathbb{S})}([\Phi, h]) = DNwg([\Phi, h], \Sigma)$, donde $\Sigma = \{IN_{\mathbb{S}}([\Psi, k]) \mid [\Psi, k] \text{ es a-estructura maximal que ataca parcialmente } [\Phi, h], e IN_{\mathbb{S}}([\Psi, k]) \neq [\emptyset, \epsilon]\}$.

Observación: Dado que siempre nos referimos a un programa \mathcal{P} arbitrario pero fijo, a menudo usaremos F en lugar de $F_{\mathcal{P}}$, a modo de abreviación.

El operador $F(\cdot)$ se define para asignaciones de status arbitrarias, es decir, asignaciones que no necesariamente respetan las relaciones de ataque del grafo (salvo que se trate en particular de una asignación de status propiamente dicha.) Informalmente, como resultado de aplicar $F(\cdot)$ a una asignación de status arbitraria \mathbb{S} se obtiene otra asignación de status arbitraria $F(\mathbb{S})$, donde $F(\mathbb{S})$ asigna narrowings IN y OUT a cada a-estructura de acuerdo a la asignación (*i.e.*, a los narrowings IN y OUT) de sus atacantes en \mathbb{S} , y siguiendo la misma regla de la definición de asignación de status (es decir, considerando los dos ataques extremos dada la indecisión en los atacantes.) La Fig. 5.16 ilustra como se asigna una a-estructura dada $[\Phi, h]$ en $F(\mathbb{S})$, a partir de las asignación de las a-estructuras que atacan $[\Phi, h]$ en \mathbb{S} .

La siguiente proposición establece formalmente la relación entre la noción de asignación de status y el operador $F(\cdot)$ sobre asignaciones de status arbitrarias.

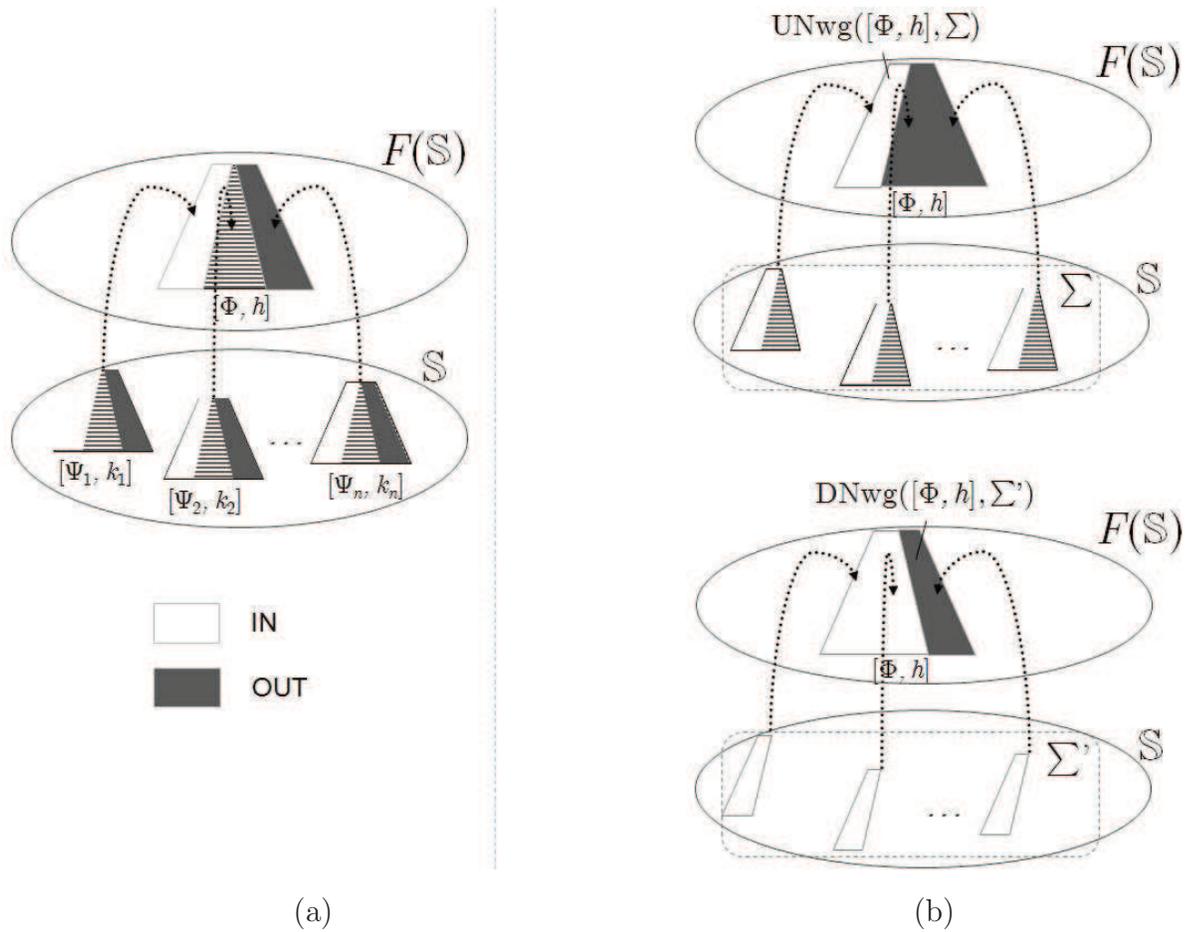
Proposición 10 *Sea \mathcal{P} un programa y sea \mathbb{S} una asignación de status arbitraria para \mathcal{P} . Entonces \mathbb{S} es una asignación de status para \mathcal{P} sssi $F(\mathbb{S}) = \mathbb{S}$, es decir, \mathbb{S} es un punto fijo del operador $F(\cdot)$.*

La demostración de la proposición anterior es directa, ya que cambiando $F(\mathbb{S})$ por \mathbb{S} en la definición del operador (asumiendo $F(\mathbb{S}) = \mathbb{S}$ como hipótesis) se obtienen exactamente las condiciones caracterizando los narrowings IN y OUT en la Def. 36 de asignación de status. El mismo ejercicio puede hacerse sobre el esquema de la Fig. 5.16, obteniendo exactamente el de la Fig. 5.1, esquematizando la definición de asignación de status.

De la proposición anterior se sigue trivialmente que el menor (con respecto a ‘ \preceq ’) punto fijo de $F_{\mathcal{P}}(\cdot)$ es una asignación de status a-grounded.

Proposición 11 *Sea \mathcal{P} un programa y sea \mathbb{S} una asignación de status arbitraria para \mathcal{P} . Entonces \mathbb{S} es una asignación de status a-grounded sssi \mathbb{S} es el menor punto fijo (con respecto a ‘ \preceq ’) del operador $F(\cdot)$ ².*

²La demostración es directa a partir de la proposición 10 y la definición de asignación de status.

Figura 5.16: Operador característico F

A continuación se enuncian dos proposiciones caracterizando formalmente un procedimiento para obtener el menor punto fijo del operador $F(\cdot)$, análogo al propuesto por Dung para argumentos individuales. Concretamente, el menor punto fijo del operador $F(\cdot)$ puede ser obtenido mediante la aplicación iterada de $F(\cdot)$ a partir de la asignación de status arbitraria \emptyset (o “vacía”).

Proposición 12 Sea \mathcal{P} un programa DeLP. Sea \emptyset una asignación de status arbitraria vacía para \mathcal{P} . Considere la siguiente secuencia de asignaciones de status arbitrarias para \mathcal{P} :

$$F^0(\emptyset), F^1(\emptyset), F^2(\emptyset), \dots$$

donde

- $F^0(\emptyset) =_{def} \emptyset$, y

- $F^{i+1}(\circlearrowleft) =_{def} F(F^i(\circlearrowleft))$.

Entonces vale que, para todo $i \in \mathbb{N} \cup \{0\}$:

- $F^i(\circlearrowleft) \preceq F^{i+1}(\circlearrowleft)$
- $F^i(\circlearrowleft) \preceq \mathbb{S}_f$, para cualquier punto fijo \mathbb{S}_f de $F(\cdot)$ ³.

Proposición 13 Sea \mathcal{P} un programa, y sea $F^0(\circlearrowleft), F^1(\circlearrowleft), F^2(\circlearrowleft), \dots$ una secuencia de asignaciones de status arbitrarias para \mathcal{P} tal como fue definida en la proposición 12. Si existe solo un número finito de a-estructuras construibles a partir de \mathcal{P} , entonces existe n tal que $F^n(\circlearrowleft) = F^{n+1}(\circlearrowleft)$, y $F^n(\circlearrowleft)$ coincide con el menor punto fijo del operador $F(\cdot)$.

Finalmente, a continuación se presenta un ejemplo que ilustra el procedimiento para calcular la semántica a-grounded sugerido por la proposición 13.

Ejemplo 18 La Fig. 5.18 muestra la secuencia de $F^i(\circlearrowleft)$ asociada al grafo de ataques de la Fig. 5.17.

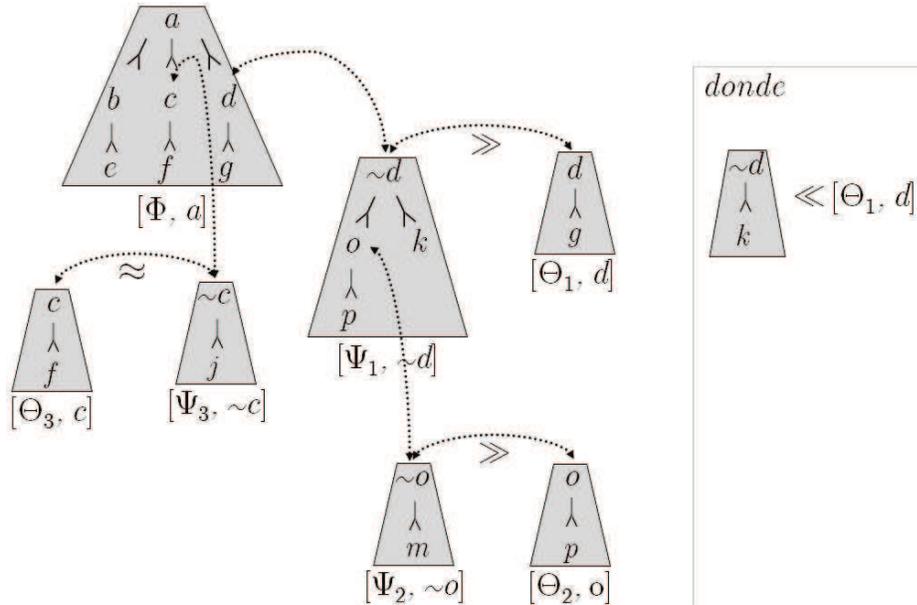


Figura 5.17: Grafo de ataques

³Las demostraciones de esta proposición y de la siguiente se encuentran en el apéndice A.

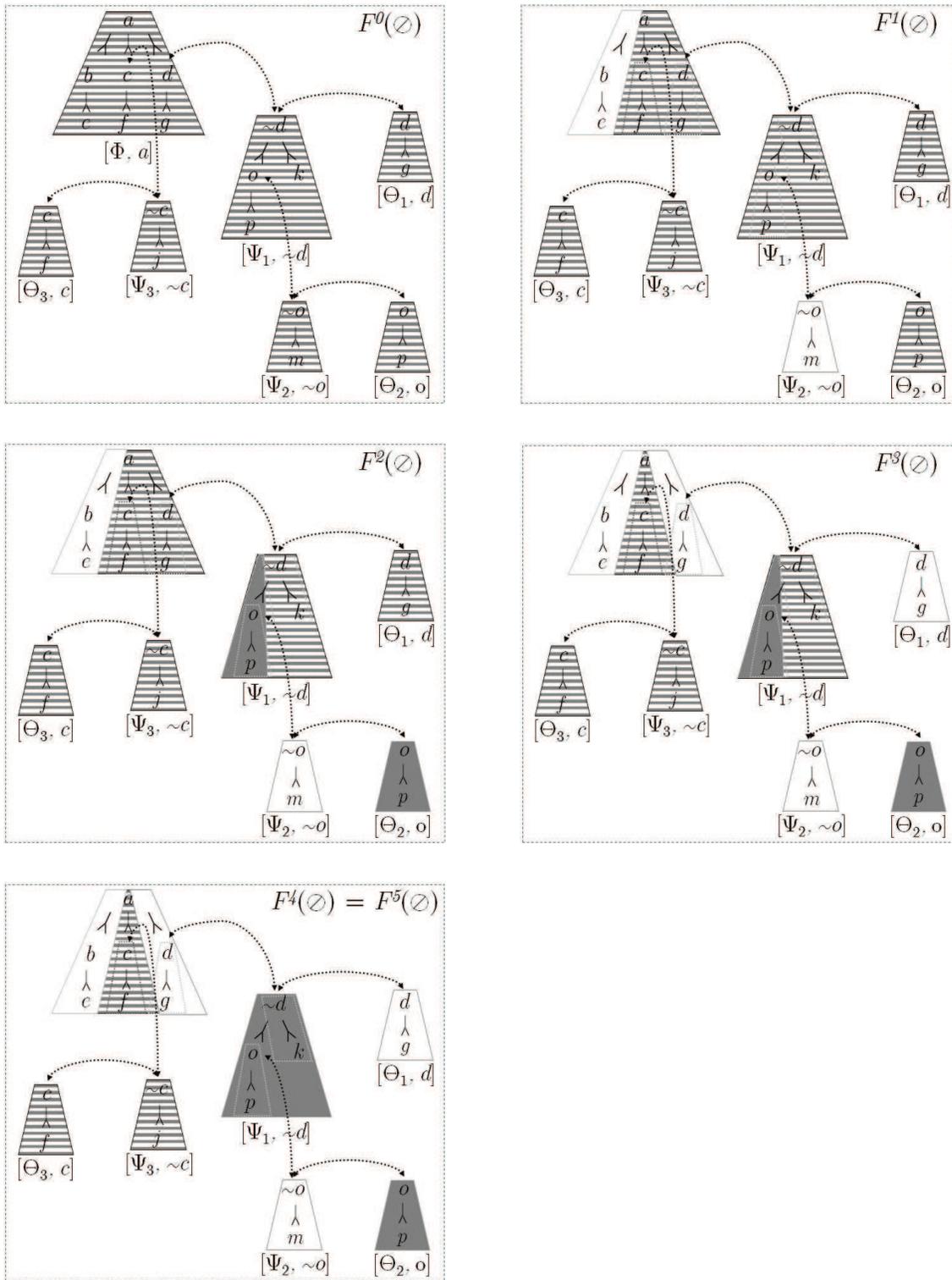


Figura 5.18: Punto fijo del operador característico F : asignación de status a-grounded

La a -estructura $[\Psi_2, \sim o]$ se asigna completamente **IN** en $F^1(\circlearrowleft)$ dado que resiste el ataque del narrowing \neg **OUT** de $[\Theta_2, o]$ en $F^0(\circlearrowleft)$ (ataque más fuerte posible). De manera similar, el narrowing $[\{(a \prec b), (b \prec c)\}, a]$ de $[\Phi, a]$ resiste el ataque de los narrowings \neg **OUT** (ataque más fuerte) asociados a $[\Psi_3, \sim c]$ y $[\Psi_1, \sim d]$ en $F^0(\circlearrowleft)$, dado que no es afectado por dicho ataque. Las demás a -estructuras, al igual que el narrowing restante de $[\Phi, a]$, quedan undecided, dado que resultan derrotadas por los ataques más fuertes, y no derrotados por los más débiles, respecto a la indecisión de los atacantes en $F^0(\circlearrowleft)$. En particular, puede verificarse que $[\Theta_3, c]$ y $[\Psi_3, \sim c]$ permanecerán undecided en los $F^i(\circlearrowleft)$ subsiguientes, dado que se atacan mutuamente, son igualmente preferidas y no reciben ataques de otras a -estructuras (situación de bloqueo).

En $F^2(\circlearrowleft)$, $[\Theta_2, o]$ se asigna completamente **OUT**, dado que resulta derrotada por el narrowing **IN** de $[\Psi_2, \sim o]$ en $F^1(\circlearrowleft)$ (la propia $[\Psi_2, \sim o]$). Por la misma razón se asigna **OUT** al narrowing de $[\Psi_1, \sim d]$ involucrando a 'o'. Luego, la a -estructura $[\Theta_1, d]$ se asigna completamente **IN** en $F^3(\circlearrowleft)$, dado que debido a la derrota de un narrowing de $[\Psi_1, \sim d]$ en $F^2(\circlearrowleft)$, ahora logra resistir el ataque de este último (concretamente, del narrowing \neg **OUT** de $[\Psi_1, \sim d]$ en $F^2(\circlearrowleft)$). Por la misma razón, el narrowing de $[\Phi, a]$ involucrando a $[\Theta_1, d]$ se asigna **IN**. Nótese que el narrowing $[\{(\sim d \prec k)\}, a]$ de $[\Psi_1, \sim d]$ sigue undecided en $F^3(\circlearrowleft)$, dado que la a -estructura que lo ataca ($[\Theta_1, d]$) aún se encuentra completamente undecided en $F^2(\circlearrowleft)$.

Finalmente, dado que $[\Theta_1, d]$ se asigna completamente **IN** en $F^3(\circlearrowleft)$, $[\Psi_1, \sim d]$ resulta completamente **OUT** en $F^4(\circlearrowleft)$. La asignación $F^4(\circlearrowleft)$ constituye un punto fijo de $F(\cdot)$, y por lo tanto la asignación de status a -grounded asociada al grafo.

5.6. Propiedades de Consistencia

En esta sección se enuncian propiedades deseables de consistencia satisfechas por la caracterización propuesta. Las demostraciones (incluyendo lemas) se encuentran en el apéndice. El siguiente teorema establece que los narrowings **IN** asociados por una asignación de status dada a las diferentes a -estructuras maximales, son consistentes dos a dos.

Teorema 2 (consistencia dos a dos de **IN-narrowings)** Sea \mathbb{S} una asignación de status para un programa dado \mathcal{P} . Sean $[\Phi_1, h_1]$ y $[\Phi_2, h_2]$ a -estructuras maximales de \mathcal{P} .

Entonces podemos asegurar que $IN_{\mathbb{S}}([\Phi_1, h_1])$ e $IN_{\mathbb{S}}([\Phi_2, h_2])$ son consistentes, es decir, no es el caso que existan a-subestructuras $[\Theta_1, q] \neq [\emptyset, \epsilon]$ y $[\Theta_2, \bar{q}] \neq [\emptyset, \epsilon]$ de $IN_{\mathbb{S}}([\Phi_1, h_1])$ e $IN_{\mathbb{S}}([\Phi_2, h_2])$, respectivamente.

A pesar de que los argumentos DeLP son consistentes internamente por definición, dos argumentos distintos para una misma conclusión podrían tener soportes en conflicto, es decir, podrían involucrar conclusiones intermedias contradictorias. Luego, como se planteó en la Sec. 4.1, la a-estructura que resulta de agregarlos estará soportando la conclusión mediante razones individuales en conflicto. El siguiente corolario del teorema 2 asegura que razones en conflicto en una misma a-estructura nunca serán asignadas simultáneamente IN.

Corolario 1 (consistencia interna de IN-narrowings) *Sea \mathbb{S} una asignación de status para un programa dado \mathcal{P} . Sea $[\Phi, h]$ una a-estructura maximal de \mathcal{P} . Entonces podemos asegurar que $IN_{\mathbb{S}}([\Phi, h])$ es consistente internamente, es decir, no es el caso que existan a-subestructuras $[\Theta_1, q] \neq [\emptyset, \epsilon]$ y $[\Theta_2, \bar{q}] \neq [\emptyset, \epsilon]$ de $IN_{\mathbb{S}}([\Phi, h])$ ⁴.*

La definición de a-estructura adoptada en la presente formalización, junto al resultado anterior, reflejan la siguiente estrategia para tratar con argumentos inconsistentes soportando la misma conclusión: se permite en principio que conformen la misma a-estructura, pero luego competirán entre sí en el análisis de aceptabilidad, de manera que a lo sumo uno de los dos terminará integrando finalmente el soporte IN de la conclusión.

Corolario 2 (IN-narrowings libres de conflictos) *Sea \mathbb{S} una asignación de status para un programa dado \mathcal{P} . Entonces podemos asegurar que no existen a-estructuras maximales $[\Phi_1, h_1]$ y $[\Phi_2, h_2]$ tal que $IN_{\mathbb{S}}([\Phi_1, h_1])$ ataca a $IN_{\mathbb{S}}([\Phi_2, h_2])$.*

El corolario anterior surge trivialmente de considerar la definición de ataque parcial, que implica la existencia de una a-subestructura (de desacuerdo) soportando la conclusión contraria de la a-estructura que ataca, lo cual, de acuerdo al teorema 2, es imposible para narrowings IN.

El siguiente teorema establece, para las tres semánticas de agregación desarrolladas, la consistencia dos a dos de narrowings garantizados.

⁴El corolario anterior surge de considerar $[\Phi_1, h_1] = [\Phi_2, h_2]$ en el teorema 2.

Teorema 3 (consistencia dos a dos de narrowings garantizados) Sean $[\Phi_1, h_1]$ y $[\Phi_2, h_2]$ a -estructuras maximales de \mathcal{P} , y sean $[\Phi'_1, h_1]$ y $[\Phi'_2, h_2]$ los narrowings garantizados de $[\Phi_1, h_1]$ y $[\Phi_2, h_2]$, respectivamente, de acuerdo a la semántica a -grounded [/a-estable / a -preferida]. Entonces podemos asegurar que $[\Phi'_1, h_1]$ y $[\Phi'_2, h_2]$ son consistentes, es decir, no es el caso que existan a -subestructuras $[\Theta_1, q] \neq [\emptyset, \epsilon]$ y $[\Theta_2, \bar{q}] \neq [\emptyset, \epsilon]$ de $[\Phi'_1, h_1]$ y $[\Phi'_2, h_2]$, respectivamente.

El siguiente corolario establece la consistencia interna de narrowings garantizado para las tres semánticas, y surge de considerar $[\Phi_1, h_1] = [\Phi_2, h_2]$ en el teorema 3.

Corolario 3 (consistencia interna de narrowings garantizados) Sea $[\Phi, h]$ una a -estructura maximal de \mathcal{P} , y sea $[\Phi', h]$ el narrowing garantizado de $[\Phi, h]$ de acuerdo a la semántica a -grounded [/a-estable / a -preferida]. Entonces podemos asegurar que $[\Phi', h]$ es consistente internamente, es decir, no es el caso que existan a -subestructuras $[\Theta_1, q] \neq [\emptyset, \epsilon]$ y $[\Theta_2, \bar{q}] \neq [\emptyset, \epsilon]$ de $[\Phi', h]$.

Teorema 4 Sean $[\Phi_1, h]$ y $[\Phi_2, \bar{h}]$ dos a -estructuras maximales. De acuerdo a la semántica a -grounded [/a-estable / a -preferida], si el narrowing garantizado de $[\Phi_1, h]$ es distinto de $[\emptyset, \epsilon]$, entonces el narrowing rechazado de $[\Phi_2, \bar{h}]$ es $[\Phi_2, \bar{h}]$.

El teorema anterior establece, para las tres semánticas de agregación, que si una a -estructura soportando h tiene un narrowing garantizado no vacío, la a -estructura para la conclusión contraria \bar{h} (en caso de existir) se encontrará completamente rechazada, es decir, no solo su narrowing garantizado será $[\emptyset, \epsilon]$ (implicado por el teorema 3), sino también su narrowing discutible.

5.7. Modelando el ejemplo del alquiler

En esta sección se modela el ejemplo motivador presentado en la introducción, empleando la formalización de agregación desarrollada. Sea un \mathcal{P} un programa DeLP que representa la información que se tiene acerca del problema de alquilar un departamento dado presentado en el Cap. 1. Concretamente, \mathcal{P} contendrá los siguientes elementos:

$alq \prec b_ubic$	b_ubic
$\sim alq \prec pequeño$	$pequeño$
$\sim alq \prec humedad$	$humedad$
$\sim alq \prec disturb$	inq_estud
$disturb \prec inq_estud$	$dice_no_dist(j)$
$\sim disturb \prec dice_no_dist(V)$	$dice_no_dist(p)$

La Fig. 5.19 muestra el grafo de ataques asociado a \mathcal{P} junto a la única asignación de status asociada, de acuerdo a la cual la a-estructura soportando alq se encuentra completamente IN. La relación de preferencia \succcurlyeq refleja el criterio de Alicia, de acuerdo al cual la a-estructura soportando $\sim disturb$ tiene más fuerza que aquella para $disturb$, causando así la derrota de esta última, y del narrowing de la a-estructura para $\sim alq$ que la involucra. Luego, a pesar de que la a-estructura soportando $\sim alq$ es preferida ante aquella soportando alq (de acuerdo a \succcurlyeq), al sufrir la derrota del narrowing involucrando $disturb$, se debilita, resultando completamente derrotada, y prevaleciendo así la a-estructura para alq . Esta asignación de status única es a-grounded, a-preferida y a-estable, y por lo tanto las tres semánticas coinciden, considerando la a-estructura para alq como garantizada.

Supongamos ahora que Alicia consigue otro testimonio de un habitante del edificio, concretamente de Flavia, que afirma percibir disturbios regularmente. Para tener en cuenta la nueva información considérese el programa \mathcal{P}_1 , que resulta de incorporar a \mathcal{P} los siguientes elementos:

$$\begin{aligned} & disturb \prec dice_dist(V) \\ & dice_dist(f) \end{aligned}$$

El testimonio de Flavia da lugar a un nuevo argumento para $disturb$ que se agrega junto con aquel involucrando $disturb \prec inq_estud$. Supongamos además que Alicia considera que las a-estructuras para $disturb$ y $\sim disturb$ son ahora igual de fuertes. La Fig. 5.20 muestra el grafo de ataques asociado a \mathcal{P} junto a la asignación de status a-grounded asociada, estableciendo que todas las a-estructuras son discutibles de acuerdo a esta semántica. Intuitivamente, la indecisión acerca de $disturb$ (situación de bloqueo entre las a-estructuras que soportan $disturb$ y $\sim disturb$) provoca indecisión en el narrowing de la a-estructura para $\sim alq$ involucrando a $disturb$. Dado que de acuerdo al criterio de Alicia este narrowing resulta decisivo en la disputa entre alq y $\sim alq$ ($[\Theta, alq] \preceq [\Phi_1, \sim alq]$), sin

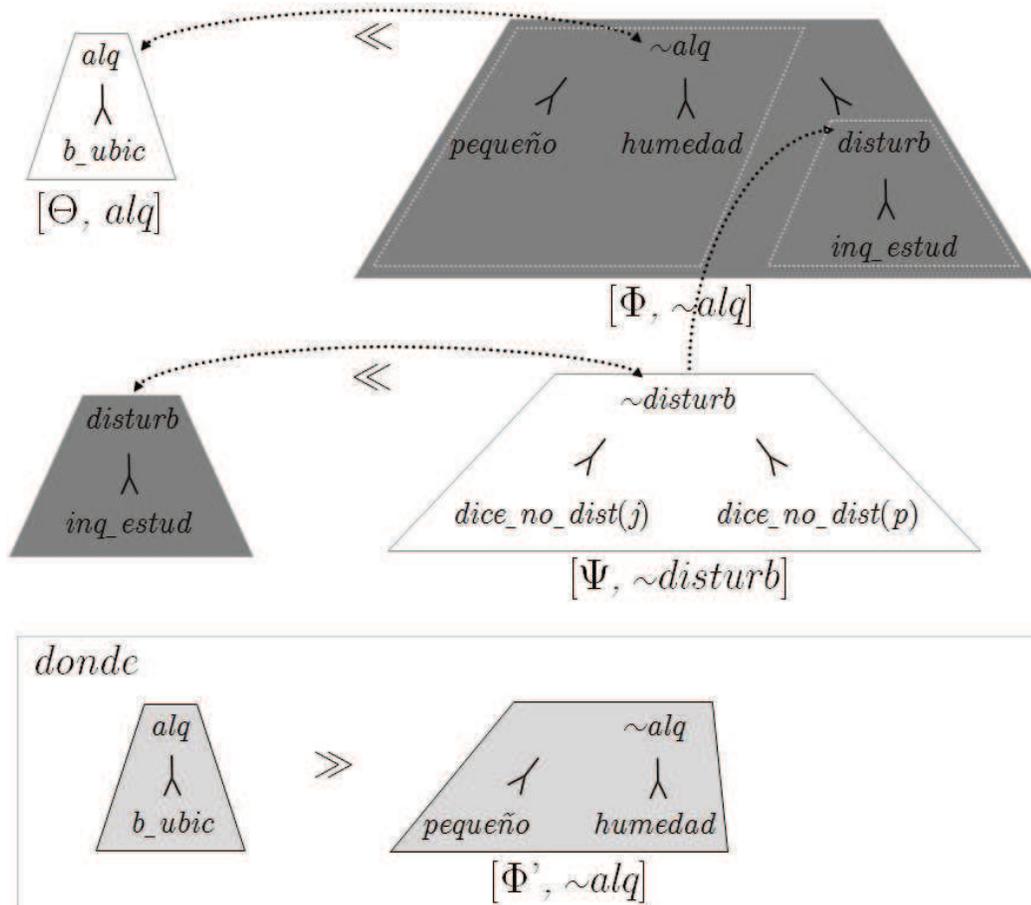


Figura 5.19: Asignación de Status para \mathcal{P}

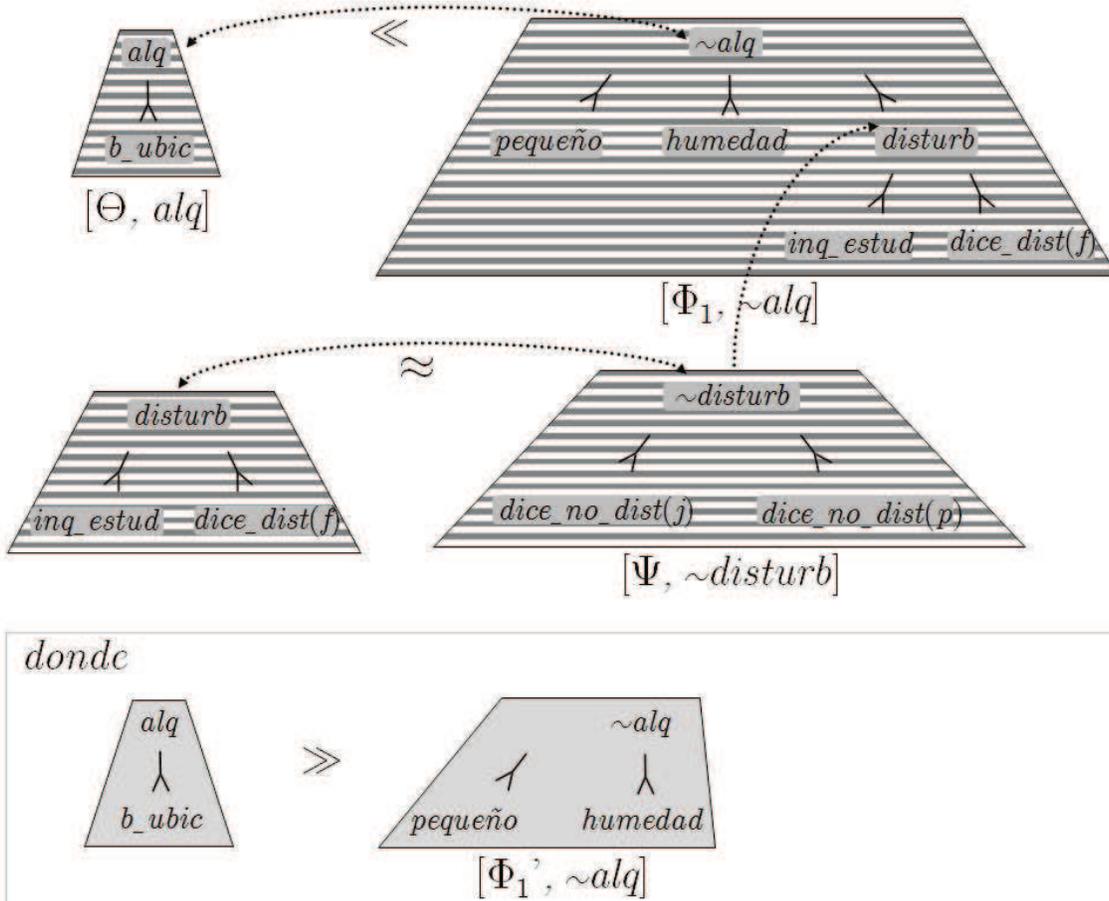


Figura 5.20: Asignación de status a-grounded para \mathcal{P}_1

embargo $[\Theta, alq] \ggg [\Phi_1', \sim alq]$ ambas a-estructuras resultan completamente discutibles. Verifiquemos ahora que se trata de una asignación de status válida. Las a-estructuras soportando *disturb* y $\sim disturb$ se atacan mutuamente, son igualmente preferidas y ninguna otra a-estructura las afecta (bloqueo), y por lo tanto se consideran completamente discutibles. La asignación de $[\Phi_1, \sim alq]$ resulta de considerar las dos posibilidades extremas respecto a la indecisión en sus atacantes. Por un lado, de acuerdo al ataque combinado más fuerte posible, considerando a sus dos atacantes completamente **IN**, la a-estructura $[\Phi_1, \sim alq]$ resulta completamente derrotada. Por otro lado, de acuerdo al ataque combinado más débil posible, considerando a sus dos atacantes completamente **OUT**, $[\Phi_1, \sim alq]$ resulta completamente no derrotada. Mediante un análisis similar, la indecisión en $[\Theta, alq]$ se explica a partir de la indecisión en su único atacante $[\Phi_1, \sim alq]$.

Las semánticas a-estable y a-preferida (idénticas para este grafo) consideran las dos alternativas de asignación (totalmente comprometidas) que surgen de resolver el bloqueo

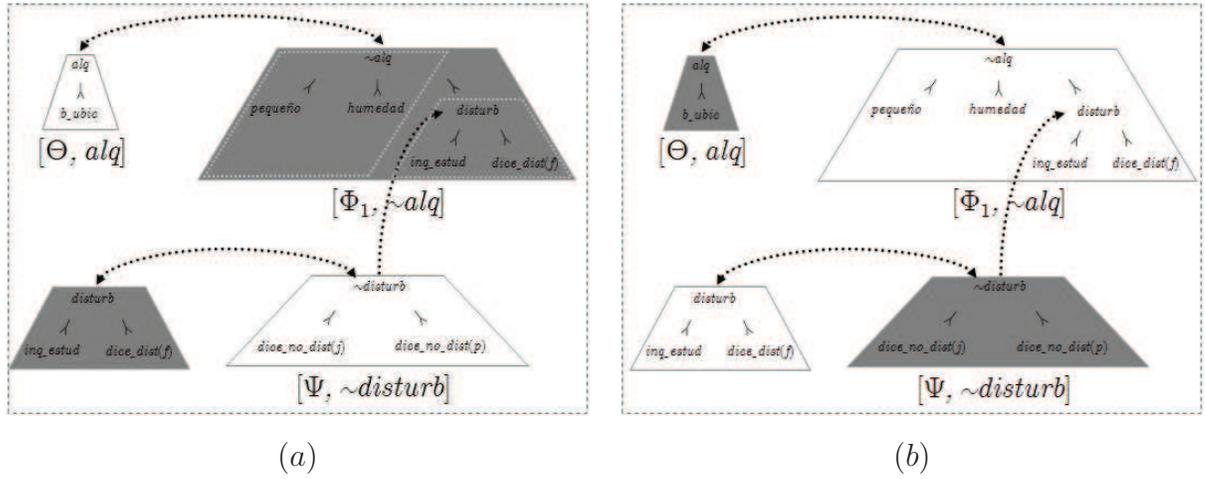


Figura 5.21: Asignaciones de status a-preferidas (/a-estables) para \mathcal{P}_1

en una y otra dirección (Fig. 5.21). Al igual que en la a-grounded, todas las a-estructuras son consideradas discutibles. Sin embargo, las múltiples asignaciones alternativas propuestas por las semánticas a-estable y a-preferida revelan cómo influye exactamente la situación de bloqueo en el status para alq .

Finalmente supongamos que Alicia considera que el alquiler del departamento es un tanto caro, y que esto significa una razón para no alquilarlo. Consideremos un programa \mathcal{P}_2 que tiene en cuenta la nueva información incorporando a \mathcal{P}_1 los siguientes elementos:

$$\begin{aligned} \sim alq &\prec caro \\ caro & \end{aligned}$$

El programa \mathcal{P}_2 da lugar a un nuevo argumento para $\sim alq$ que se agrega junto a las otras tres razones para la misma conclusión. La Fig. 5.22 muestra el grafo de ataques correspondiente a \mathcal{P}_2 junto a la asignación de status a-grounded asociada, bajo la suposición de que, de acuerdo al criterio de Alicia, las razones *pequeño*, *humedad* y *caro* para no alquilar son preferidas a la única razón para alquilar. Nótese que $\sim alq$ prevalece sobre alq , a pesar de que la a-estructura $[\Phi_2, \sim alq]$ involucra una razón (narrowing) discutible. Este ejemplo ilustra la utilidad de definir las respuestas de las semánticas de agregación en términos de tres tipos de narrowings. En este caso la semántica nos está diciendo que las razones para $\sim alq$ involucrando *pequeño*, *humedad* y *caro* se consideran *definitivamente aceptadas* (narrowing garantizado), mientras que existe *indecisión* acerca de la razón para $\sim alq$ involucrando *disturb* (narrowing discutible).

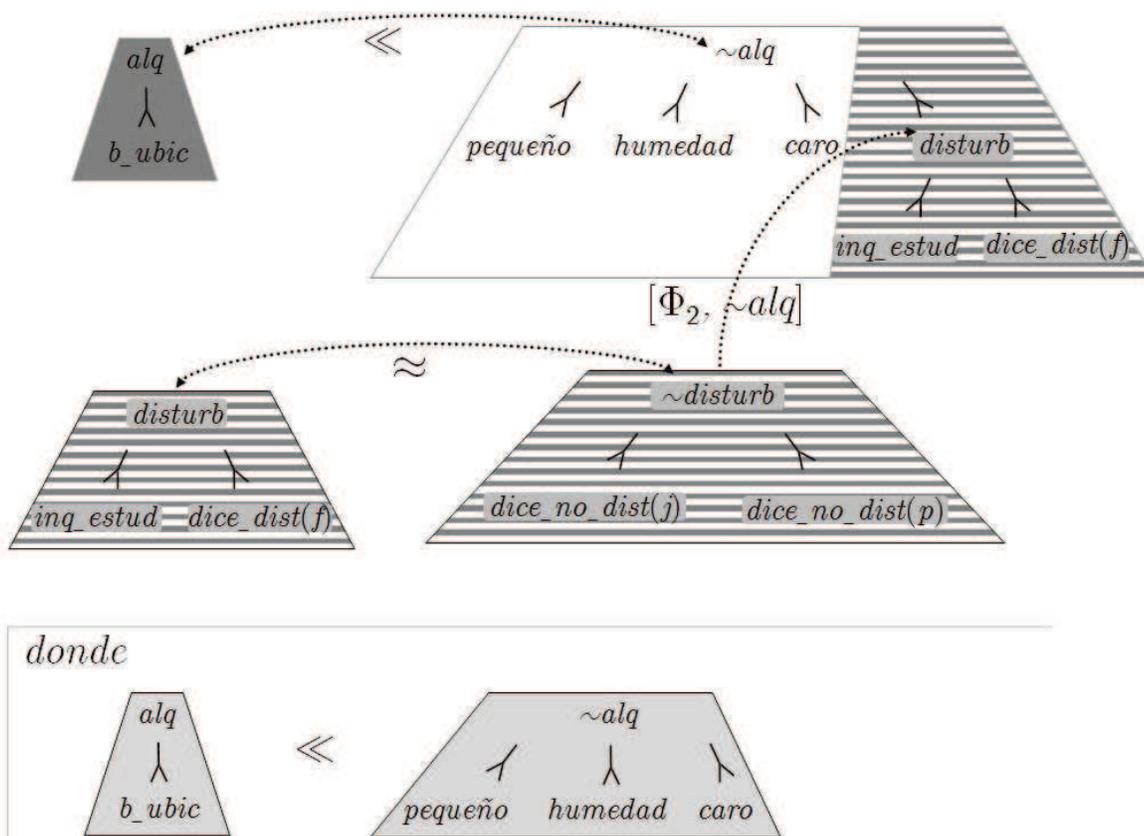


Figura 5.22: Asignación de status a-grounded para \mathcal{P}_2

Nuevamente las semánticas a-estable y a-preferida coinciden con la a-grounded para este ejemplo.

5.8. Conclusiones

En este capítulo se presentaron las semánticas a-grounded, a-estable y a-preferida para la agregación (E5 – enfoque declarativo), basadas en las correspondientes semánticas de aceptabilidad propuestas por Dung, y se analizaron comparativamente a la luz de diferentes situaciones.

La caracterización de estas semánticas contempla las interacciones (conflictos) entre a-estructuras y la noción de derrota combinada, sancionando para cada a-estructura tres narrowings complementarios representando su “status de aceptabilidad”: narrowing garantizado (denotando las razones definitivamente aceptadas), narrowing rechazado (denotando las razones definitivamente no aceptadas) y narrowing discutible (representando las razones que presentan indecisión). La distinción de esto tres narrowings significa una gran ventaja a la hora de interpretar las respuestas brindadas por el sistema, como se ilustró en la última sección a través del ejemplo del alquiler presentado en la introducción.

Se propuso además una caracterización alternativa para la semántica a-grounded basada en la noción de punto fijo, inspirada en la caracterización original de la semántica *grounded* desarrollada por Dung ([Dun95]). Esta caracterización resulta de particular interés dado que sugiere un procedimiento efectivo para computar la semántica a-grounded.

Finalmente se enunciaron y demostraron propiedades deseables de consistencia satisfechas por la caracterización propuesta.

Capítulo 6

Procedimiento de Prueba Dialéctico para la Semántica A-Grounded

En la Sección 3.3 se presentó un procedimiento de prueba dialéctico para la semántica grounded (argumentos individuales), que fue a su vez refinado en la Sec. 3.3.1 para tomar ventaja de la noción de subargumento de desacuerdo asociado a la definición de derrota. A continuación se presenta un procedimiento de prueba dialéctico para la semántica a-grounded, que se basa fuertemente en aquel presentado en la Sec. 3.3.1 para la semántica grounded. Concretamente, este procedimiento de prueba se basa en la construcción de un árbol dialéctico, pero donde los nodos representan a-estructuras en lugar de argumentos individuales. En primer lugar se presenta la noción de *línea dialéctica agregada*, que resulta de adaptar la noción de línea dialéctica para contemplar a-estructuras.

6.1. Líneas dialécticas agregadas

Definición 45 (Línea Dialéctica Agregada) Sea \mathcal{P} un programa DeLP y sea $[\Phi, h]$ una a-estructura maximal en \mathcal{P} . Una línea dialéctica agregada acerca de $[\Phi, h]$ (o simplemente ADL — accrued dialectical line) es una secuencia finita no vacía de a-estructuras $[[\Phi_1, h_1], [\Phi_2, h_2], \dots, [\Phi_n, h_n]]$, con $[\Phi_1, h_1] = [\Phi, h]$, tal que:

(a1) $[\Phi_i, h_i]$ ataca parcialmente a $[\Phi_{i-1}, h_{i-1}]$, $1 < i \leq n$,

- (a2) si $[\Phi_i, h_i]$ es PRO y Σ es el conjunto de todas las a -subestructuras de desacuerdo asociadas a ataques contra a -estructuras PRO apareciendo antes de $[\Phi_i, h_i]$ en la línea, entonces $[\Phi_i, h_i]$ es maximal (w.r.t. \sqsubseteq) verificando que ningún narrowing de una a -estructura en Σ es a -subestructura de $[\Phi_i, h_i]$,
- (a3) si $[\Phi_i, h_i]$ es OPP, entonces es una a -estructura maximal,

donde $[\Phi_i, h_i]$ se dice que es una a -estructura PRO de la línea si i es impar, o una a -estructura OPP de la línea si i es par.

Observación: Dado que cada a -estructura $[\Phi_i, h_i]$ en la línea o bien ataca o es atacada (o ambas cosas) por otra a -estructura en la línea (condición a1), entonces $[\Phi_i, h_i] \neq [\emptyset, \epsilon]$.

A continuación analizaremos cada condición a1-a3 en la Def. 45, comparándolas con las condiciones d1' y d2' de la Def. 24. Mientras que la condición d1' requiere que cada argumento en la línea derrote al argumento inmediato anterior, la condición a1 requiere que cada a -estructura en la línea *ataque parcialmente* a la a -estructura previa. La razón para considerar ataques parciales, en lugar de derrotadores parciales, es que, como se mostró en la sección 4.2.3, una a -estructura que ataca parcialmente a una a -estructura víctima dada $[\Phi, h]$, pero no la derrota, puede pasar a ser un derrotador al considerar otras a -estructuras que atacan $[\Phi, h]$ (derrota combinada).

La condición a2 constituye la adaptación natural para a -estructuras de la condición d2'. Recordemos que d2' evita la inclusión de una argumento PRO $\langle \mathcal{A}, h \rangle$ cuando éste contiene al subargumento de desacuerdo $\langle \mathcal{S}, k \rangle$ de un argumento PRO previo (ver esquema de la Fig. 6.1). Como se explicó en la Sec. 3.3.1, el propósito de d2' es favorecer a OPP (postura de “menor compromiso”) frente a ciclos en el grafo de derrotas. Considerando la metáfora de una discusión entre PRO y OPP, d2' evita que PRO se defienda de un ataque previo de OPP, apelando a la propia a -subestructura (de desacuerdo) cuestionada por OPP ($\langle \mathcal{S}, k \rangle$), en cuyo caso PRO estaría incurriendo en una falacia. De manera análoga, y considerando que una a -estructura representa generalmente varios argumentos para la misma conclusión, cada uno involucrando diferentes subargumentos, la condición a2 exige que las a -estructuras introducidas por PRO sean maximales (con respecto a ' \sqsubseteq ') tal que no involucren a -subestructuras de desacuerdo previamente cuestionadas por OPP.

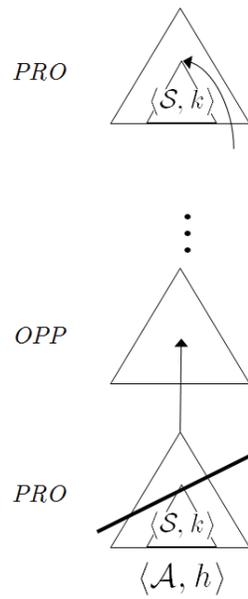


Figura 6.1: Esquema ilustrando efecto de la condición d2'

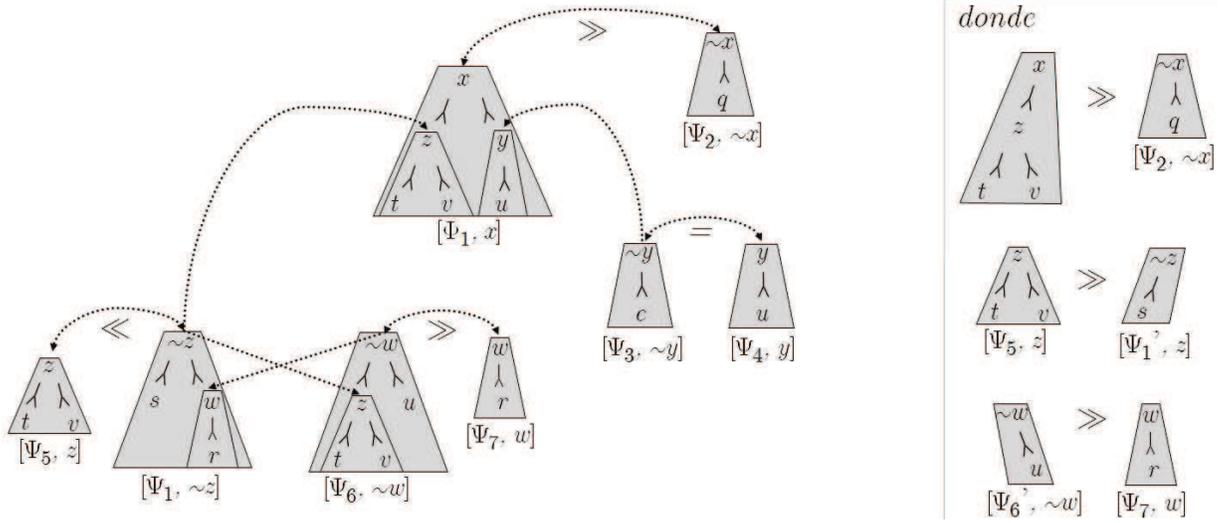


Figura 6.2: Grafo de ataques

Finalmente, la condición a3, que no tiene correspondiente en la Def. 24, establece que las a-estructuras OPP son maximales, reflejando que OPP no posee restricciones en relación a otras a-estructuras OPP apareciendo previamente en la línea.

Ejemplo 19 La Fig. 6.2 muestra un grafo de ataques respecto a un programa dado \mathcal{P} . La Fig. 6.3a muestra una ADL acerca de $[\Phi_1, x]$ respecto al grafo de la Fig. 6.2, distinguiendo a-estructuras PRO y OPP. Resulta evidente que las condiciones a1 y a3 son satisfechas. Consideremos la condición a2. La primera a-estructura (PRO) en la secuencia, $[\Phi_1, x]$, es una a-estructura maximal (nótese que aparece en el grafo de ataques, involucrando exclusivamente a-estructuras maximales). Como regla general, de acuerdo a la condición a2 la primera a-estructura en una ADL dada es siempre una a-estructura maximal, dado que $\Sigma = \emptyset$ en ese caso. Veamos que $[\Psi'_6, \sim w]$ (la otra a-estructura PRO en la línea) satisface a2. Considere la a-estructura $[\Psi_5, z]$, única a-subestructura de desacuerdo de una a-estructura PRO apareciendo antes que $[\Psi'_6, \sim w]$ (i.e., $\Sigma = \{[\Psi_5, z]\}$). En primer lugar, $[\Psi'_6, \sim w]$ no tiene como a-subestructura a $[\Psi_5, z]$, ni a ningún narrowing de $[\Psi_5, z]$. En segundo lugar, $[\Psi'_6, \sim w]$ es maximal verificando tal restricción, dado que todas las a-estructuras “mayores” que $[\Psi'_6, \sim w]$ de acuerdo a \sqsubseteq' ($[\Psi_6, \sim w]$, $[\Psi''_6, \sim w]$ y $[\Psi'''_6, \sim w]$ en Fig. 6.3b) tienen un narrowing de $[\Psi_5, z]$ como a-subestructura.

La Fig. 6.4 muestra una secuencia de a-estructuras que no constituye una ADL, e ilustra que, como caso particular de la condición a2, PRO no puede introducir una a-subestructura de desacuerdo ($[\Psi_5, z]$) de una a-estructura PRO previa ($[\Phi_1, x]$). Nótese

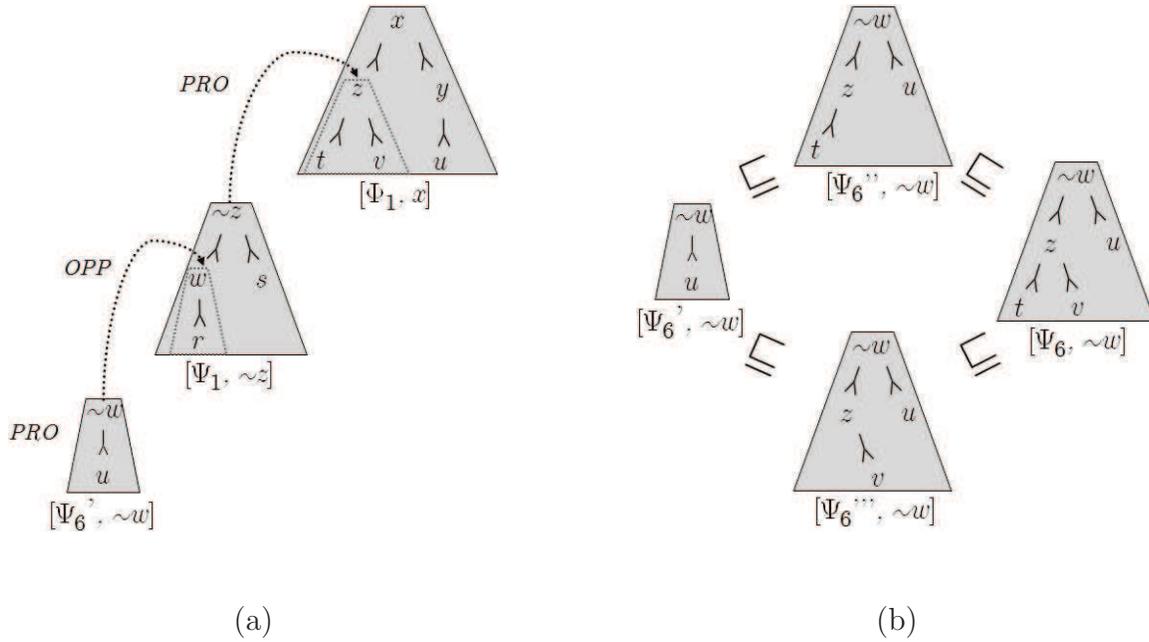


Figura 6.3: Línea Dialéctica Agregada

que la última a-estructura en la secuencia, $[\Psi_5, z]$, viola la condición a2, dado que tiene como a-subestructura a un narrowing de una a-subestructura de desacuerdo PRO previa (la propia $[\Psi_5, z]$).

6.2. Árboles dialécticos agregados

A continuación se introducen las nociones de *ADL exhaustiva* y *árbol dialéctico agregado*, que se obtienen directamente de las correspondientes nociones de línea dialéctica exhaustiva y árbol dialéctico, cambiando argumentos por a-estructuras.

Definición 46 (ADL Exhaustiva) Sea $\Lambda = [[\Phi_1, h_1, \alpha_1], [\Phi_2, h_2, \alpha_2], \dots, [\Phi_n, h_n, \alpha_n]]$ una ADL. Diremos que Λ es exhaustiva cuando no exista a-estructura $[\Theta, q, \delta]$ tal que $[[\Phi_1, h_1, \alpha_1], [\Phi_2, h_2, \alpha_2], \dots, [\Phi_n, h_n, \alpha_n], [\Theta, q, \delta]]$ es una ADL.

Definición 47 (Árbol Dialéctico Agregado) Sea $[\Phi, h]$ una a-estructura. Un árbol dialéctico agregado (o simplemente ADT — accrued dialectical tree) para $[\Phi, h]$, denotado $T_{[\Phi, h]}$, se define como sigue:

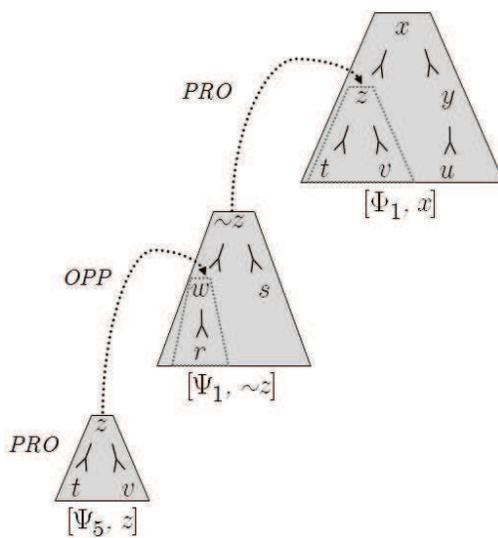


Figura 6.4: Secuencia que no constituye una Línea Dialéctica Agregada

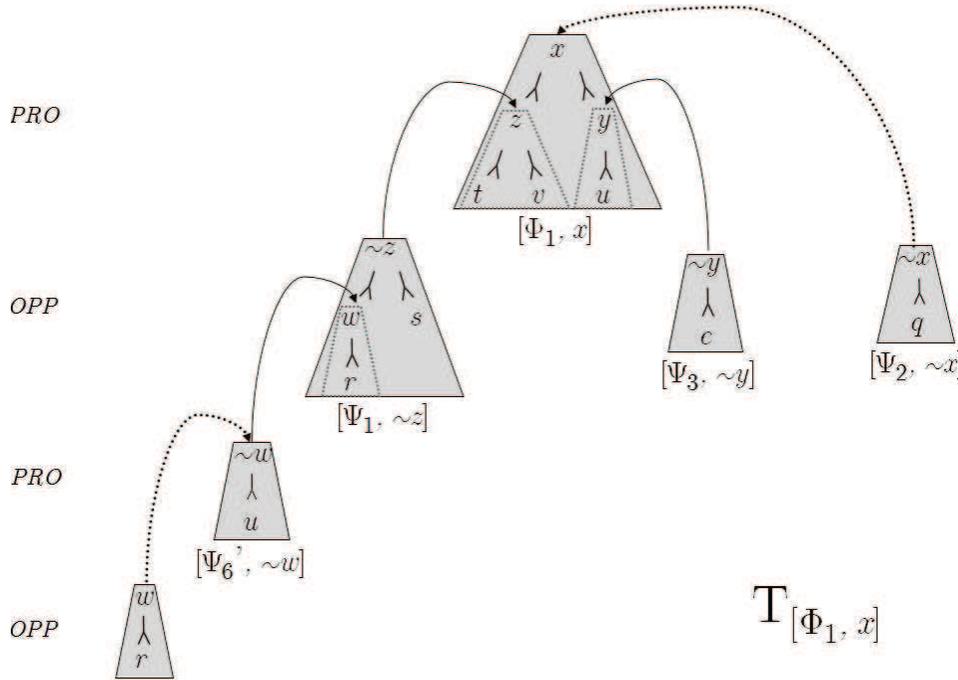


Figura 6.5: Árbol Dialéctico Agregado

1. Los nodos se etiquetan con a-estructuras.
2. $\Lambda = [[\Phi_1, h_1], [\Phi_2, h_2], \dots, [\Phi_n, h_n]]$ es la secuencia de etiquetas de una rama del árbol (camino desde la raíz a una hoja) sssi Λ es una ADL exhaustiva acerca de $[\Phi, h]$.
3. No existen nodos hermanos (hijos del mismo nodo) M y M' en el árbol etiquetados con la misma a-estructura.

Una vez que el ADT fue construido, este debe ser evaluado para determinar el resultado final de la discusión entre PRO y OPP. En el caso del análisis dialéctico para argumentos, cada nodo del árbol es marcado con ‘D’ (defeated) o ‘U’ (undefeated), comenzando por las hojas (trivialmente no derrotadas), y propagando el marcado hacia arriba. Para el caso de las a-estructuras, un marcado binario no es apropiado, dado que los ataques y derrotas entre a-estructuras son parciales, afectando porciones (narrowings) de las a-estructuras atacadas. De esta forma, la evaluación de un ADT consistirá en asociar a cada a-estructura $[\Phi_i, h_i]$ (nodo) del árbol, un *narrowing no derrotado* (o U-narrowing) de $[\Phi_i, h_i]$, comenzando por las hojas (completamente no derrotadas), y analizando cada ataque combinado, desde los más profundos hasta aquel involucrando a la raíz, para determinar los U-narrowings de los restantes nodos del árbol.

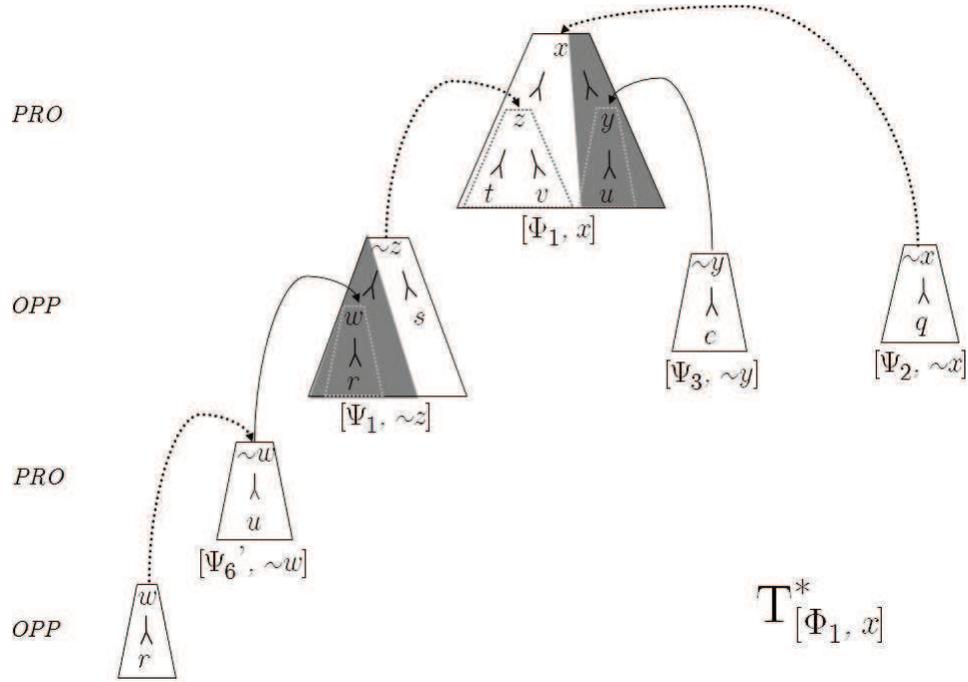


Figura 6.6: ADT evaluado

Definition 4 (ADT Evaluado) Sea $T_{[\Phi, h]}$ un ADT. El correspondiente ADT evaluado, denotado $T_{[\Phi, h]}^*$, se obtiene asociando a cada nodo N en $T_{[\Phi, h]}$, etiquetado con una a-estructura $[\Theta, k]$, un narrowing (posiblemente vacío) de $[\Theta, k]$ que se denominará U -narrowing de N (notado $UNwg(N)$), y definido como sigue:

1. Si N es una hoja, entonces $UNwg(N) =_{def} [\Theta, k]$.
2. En caso contrario (i.e., N es un nodo interno), sean M_1, \dots, M_n los hijos de N , y sea $\Sigma = \{UNwg(M_i) \mid UNwg(M_i) \neq [\emptyset, \epsilon], 1 \leq i \leq n\}$. Entonces $UNwg(N) =_{def} UNwg([\Theta, k], \Sigma)$.

Ejemplo 20 La Fig. 6.5 muestra el ADT para $[\Phi_1, x]$ ($T_{[\Phi_1, x]}$). La Fig. 6.6 muestra el ADT evaluado para $[\Phi_1, x]$ ($T_{[\Phi_1, x]}^*$), donde los U -narrowings de cada nodo se encuentran resaltados en blanco.

6.3. Vínculo entre la semántica y el procedimiento de prueba

La siguiente proposición establece formalmente el vínculo entre la semántica a-grounded y el procedimiento de prueba dialéctico presentado.

Proposición 14 Sea \mathcal{P} un programa y sea $[\Phi, h]$ una a-estructura maximal en \mathcal{P} . Sea \mathbb{S} una asignación de status grounded para \mathcal{P} . Entonces $IN_{\mathbb{S}}([\Phi, h])$ coincide con el U-narrowing de la raíz de $T_{[\Phi, h]}^*$ (Fig. 6.7a).

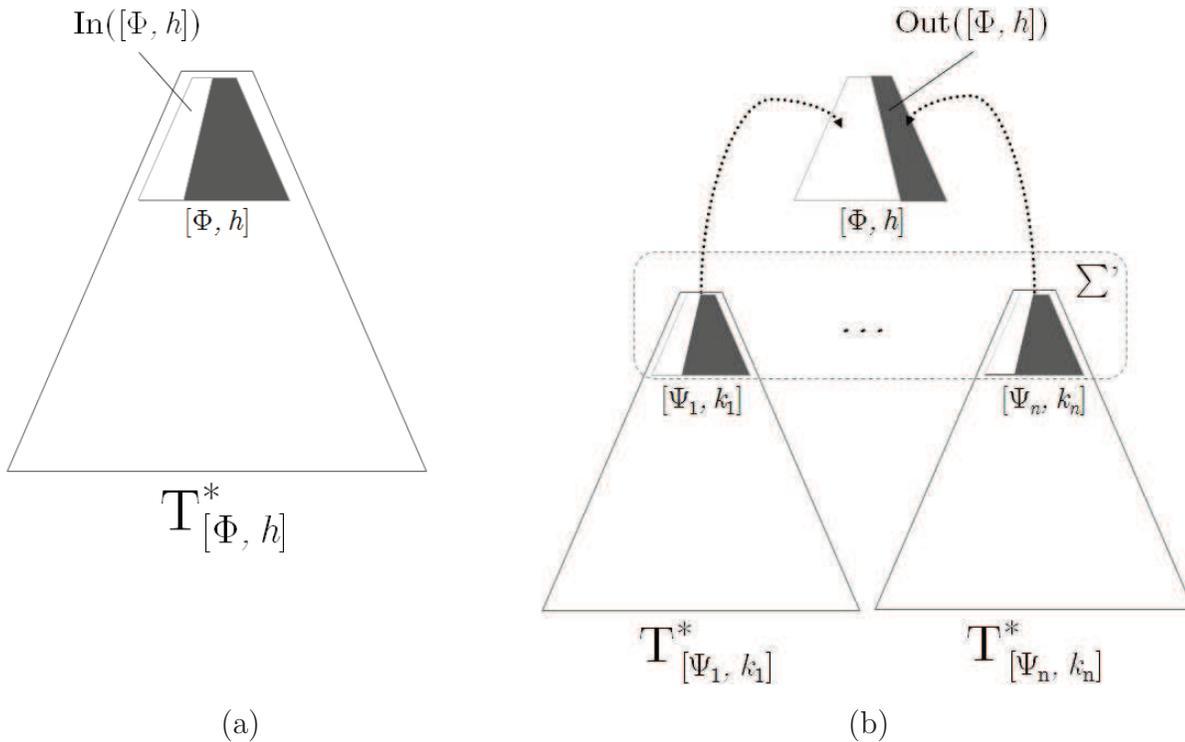


Figura 6.7: Calculando narrowings IN y OUT mediante el procedimiento dialéctico

La proposición anterior no fue demostrada, sin embargo su validez puede entenderse informalmente considerando el siguiente razonamiento ¹. La asignación de status a-grounded toma la postura de menor compromiso frente a bloqueos, *i.e.*, considera a las a-estructuras involucradas en un bloqueo como completamente *undecided*. Además, de acuerdo a la definición de asignación de status, el narrowing IN de una a-estructura maximal dada

¹La demostración formal de la proposición 14 se abordará como parte del trabajo futuro.

$[\Phi, h]$ se calcula considerando los narrowings \rightarrow OUT de las a-estructuras que la atacan, es decir, considerando el ataque “más fuerte” que es posible concebir dada la indecisión en los atacantes. Resumiendo las dos afirmaciones anteriores, de acuerdo a la asignación de status a-grounded, el narrowing IN de una a-estructura dada $[\Phi, h]$ se calcula inclinando la indecisión (consecuencia de bloqueos) en favor de los atacantes de $[\Phi, h]$. Esta política de inclinar la indecisión en favor de los atacantes se refleja en el procedimiento dialéctico propuesto a través de la condición a2 de la noción de línea dialéctica agregada, cuyo efecto es el de resolver las situaciones de bloqueo en favor de OPP, particularmente en favor de los hijos (atacantes) de $[\Phi, h]$ en el árbol.

La Fig. 6.8 muestra un grafo de ataques junto a la asignación de status a-grounded asociada. De acuerdo a esta asignación de status, $[\{(a \prec b), (b \prec e)\}, a]$ constituye el narrowing IN de $[\Phi, a]$, dado que no recibe ataques, $[\{(a \prec c), (c \prec f)\}, a]$ el narrowing *undecided*, dado que involucra a la a-subestructura $[\Theta_3, c]$, en situación de bloqueo con $[\Psi_3, \sim c]$, y $[\{(a \prec d), (d \prec g)\}, a]$ el narrowing OUT, dado que involucra a la a-subestructura $[\Theta_1, d]$, derrotada estrictamente por $[\Psi_1, \sim d]$.

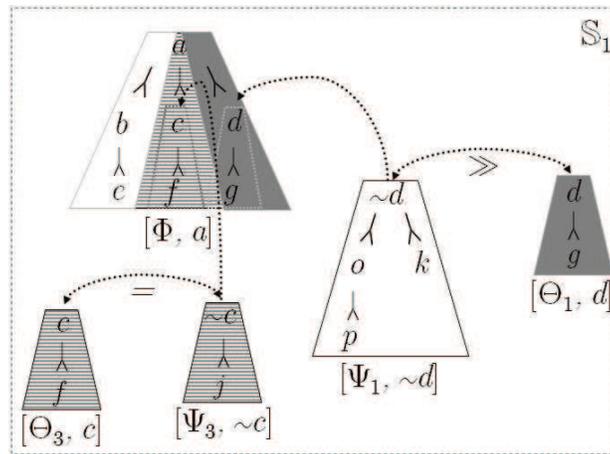


Figura 6.8: Asignación de status a-grounded

La Fig. 6.9 muestra el *ADT* evaluado $T_{[\Phi, a]}^*$ para la a-estructura $[\Phi, a]$ de acuerdo al grafo anterior. Nótese que la condición a2 evita la inclusión de $[\Theta_3, c]$ (a-subestructura de desacuerdo PRO) como hijo de $[\Psi_3, \sim c]$ en el árbol, inclinando el bloqueo entre $[\Theta_3, c]$ y $[\Psi_3, \sim c]$, y la consiguiente indecisión, en favor de OPP, es decir, en favor del ataque hacia $[\Phi, a]$. Finalmente esto causa que el narrowing de $[\Phi, a]$ afectado por el bloqueo quede fuera del narrowing IN.

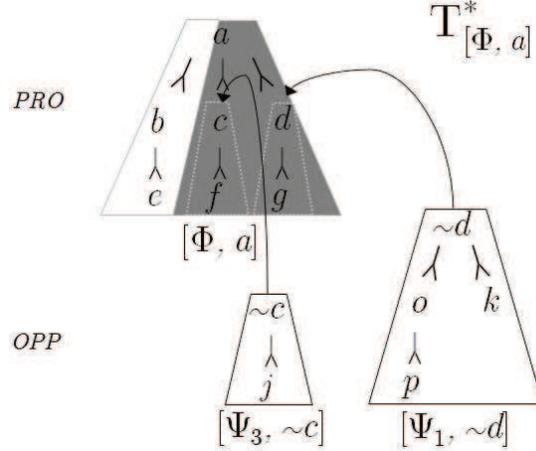


Figura 6.9: Calculando $IN([\Phi, a])$

Analicemos ahora la forma de calcular el narrowing OUT (y el *undecided*) de una a-estructura dada $[\Phi, h]$. Por un lado, el procedimiento dialéctico presentado nos permite calcular el narrowing IN de una a-estructura dada $[\Phi, h]$. Por otra parte, de acuerdo a la definición de asignación de status el narrowing OUT de una a-estructura dada $[\Phi, h]$ se obtiene a partir de los narrowings IN de sus atacantes $[\Psi_1, k_1], \dots, [\Psi_n, k_n]$ (conjunto Σ'). Concretamente, $OUT([\Phi, h])$ coincide con el D-narrowing de $[\Phi, h]$ resultado del ataque combinado de los $IN([\Psi_i, k_i])$. Este análisis sugiere el siguiente procedimiento simple para calcular el narrowing OUT de $[\Phi, h]$: obtener el narrowing IN de cada a-estructura $[\Psi_i, k_i]$ que ataca $[\Phi, h]$ empleando el procedimiento de prueba dialéctico presentado, y luego calcular el D-narrowing de $[\Phi, h]$ respecto al ataque combinado de los $IN([\Psi_i, k_i])$ (Fig. 6.7b).

A modo de ejemplo, calculemos el narrowing OUT de la a-estructura $[\Phi, a]$ de la Fig. 6.8. Las Figs. 6.10a y Figs. 6.10b muestran, respectivamente, los ADTs evaluados para las a-estructuras $[\Psi_3, \sim c]$ y $[\Psi_1, \sim d]$ atacando $[\Phi, a]$. Finalmente, la Fig. 6.11 muestra el ataque combinado considerado para el cálculo del narrowing OUT de $[\Phi, a]$, acorde a lo establecido por la Def. 36 (asignación de status). Nótese que $IN([\Psi_3, \sim c]) \notin \Sigma'$ dado que $IN([\Psi_3, \sim c]) = [\emptyset, \epsilon]$.

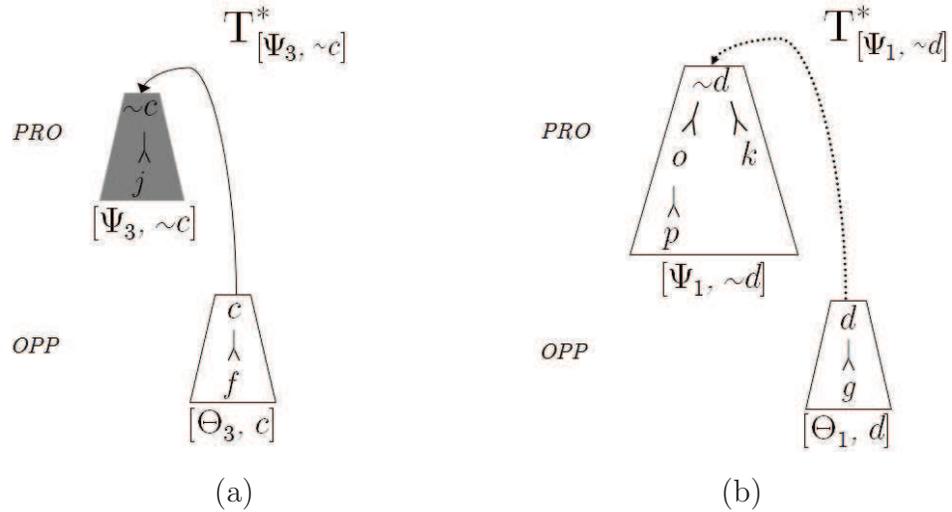


Figura 6.10: Calculando narrowings IN de a-estructuras que atacan $[\Phi, a]$

6.4. Conclusiones

En este capítulo se presentó un procedimiento de prueba dialéctico para la semántica a-grounded (E5 – enfoque procedural), basado en el procedimiento dialéctico para la semántica grounded presentado en el capítulo 3. En este caso el procedimiento puede verse como la formalización de una discusión entre dos jugadores ficticios, considerando estructuras agregadas en lugar de argumentos.

En primer lugar se presentó la noción de *línea dialéctica agregada*, que resulta de adaptar la definición de línea dialéctica para contemplar a-estructuras. En segundo lugar se introdujo la noción de *árbol dialéctico agregado*, análoga a la de árbol dialéctico pero involucrando a-estructuras en lugar de argumentos, y modelando una discusión exhaustiva contemplando agregación. Luego se formalizó la evaluación de árboles dialécticos agregados, que consiste en asociar un narrowing no derrotado (o *U-narrowing*) a cada nodo del árbol. El U-narrowing de un nodo se obtiene recursivamente partir de los U-narrowings asociados a sus hijos (ataque combinado) mediante la noción de degradación secuencial. Finalmente se estableció formalmente y analizó la conexión entre el procedimiento de prueba y la semántica a-grounded.

Cabe destacar que el procedimiento presentado hereda una de las ventajas más notables del enfoque dialéctico para argumentos: el árbol dialéctico construido constituye en sí mismo una explicación intuitiva de la respuesta obtenida.

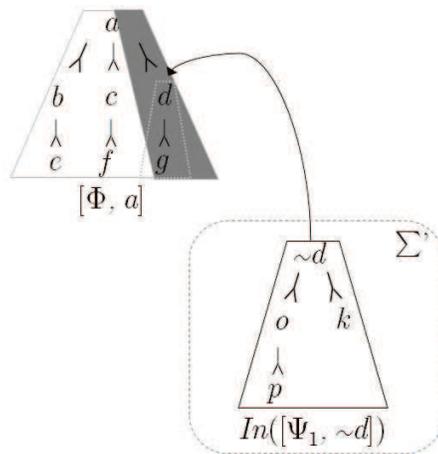


Figura 6.11: Calculando $OUT([\Phi, a])$ de acuerdo a la Def. 36

Capítulo 7

Generalización del Lenguaje de Representación y algunas instancias relevantes

La formalización de agregación presentada en los Caps. 4, 5 y 6 se basa en el lenguaje de representación y la noción de argumento provista por un formalismo de argumentación particular: la Programación en Lógica Rebatible (DeLP). La razón de comprometernos a DeLP en una primera instancia fue simplificar la presentación del enfoque. En este capítulo se presenta la noción de *Framework Argumentativo con Agregación*, definiendo un lenguaje de representación y noción de argumento más generales, en base a los cuales se aplica directamente toda la formalización de agregación presentada. Asimismo se definen distintos sistemas como instancias del framework, incorporando el modelamiento de diferentes características, como la regla deductiva modus tollens, el tipo de ataque undercutting, la discriminación de argumentos redundantes en una agregación y la incorporación explícita de incertidumbre posibilística a nivel del lenguaje objeto.

7.1. Framework Argumentativo con Agregación

Definición 48 (Lenguaje) *Un lenguaje es un conjunto \mathcal{L} de expresiones involucrando un símbolo distinguido ‘ \neg ’ denotando negación fuerte (conectivo lógico unario), y tal que:*

1. no existe elemento de \mathcal{L} involucrando una sub-expresión $\neg\neg x$ (i.e., se prohíbe la aplicación consecutiva de dos o más ' \neg '), y
2. \mathcal{L} es cerrado bajo complemento con respecto a ' \neg ', es decir, si $s \in \mathcal{L}$, entonces $\bar{s} \in \mathcal{L}$, donde \bar{s} (el complemento de s) es $\neg s$, si $s \neq \neg x$, o x , si $s \equiv \neg x$.

Los elementos del lenguaje se denominarán sentencias.

La razón para prohibir la aplicación consecutiva de dos o más ' \neg ' es mantener simple la definición de conflicto entre argumentos (y a-estructuras), ya que de permitirse debería asegurarse de alguna manera la detección de conflictos entre sentencias no complementarias como x y $\neg\neg\neg x$, por ejemplo. Sin embargo, nótese que bajo la suposición realizada de que el símbolo ' \neg ' tiene significado fijo de negación fuerte, esta simplificación no implica pérdida de poder expresivo o generalidad.

Finalmente, desde el punto de vista pragmático, el lenguaje \mathcal{L} se empleará para especificar el conocimiento acerca del dominio en consideración. Las inferencias que pueden realizarse a partir de conocimiento expresado en \mathcal{L} se especificarán mediante reglas de inferencia, introducidas a continuación. La sintaxis adoptada para las reglas de inferencia se basa en la empleada por Vreeswijk en [Vre92].

Definición 49 (Reglas de Inferencia) Sea \mathcal{L} un lenguaje.

- Una regla de inferencia estricta es una fórmula de la forma $p_1, \dots, p_n \longrightarrow p$, donde $n \geq 0$ y $p_1, \dots, p_n, p \in \mathcal{L}$.
- Una regla de inferencia rebatible es una fórmula de la forma $p_1, \dots, p_n \implies p$, donde $n \geq 0$ y $p_1, \dots, p_n, p \in \mathcal{L}$.

Las reglas de inferencia son expresiones meta-lingüísticas: a pesar de estar definidas en términos de \mathcal{L} , no constituyen elementos de \mathcal{L} . Las reglas de inferencia se emplearán para modelar patrones de inferencia independientes del dominio (de la misma forma que las reglas de inferencia de la Lógica Proposicional son independientes del dominio). De esta forma mediante las reglas de inferencia (estrictas y rebatibles), el presente enfoque es capaz de modelar tanto razonamiento de tipo *deductivo* (modus ponens, modus tollens, etc), como razonamiento de tipo *no deductivo* (modus ponens rebatible, abducción, etc), e

incluso los esquemas de argumentación propuestos por Walton [Wal96]. Para representar reglas específicas del dominio (como un “perro es un mamífero”, o “las aves generalmente vuelan”), el lenguaje \mathcal{L} debe definirse de manera tal de incluir conectivos como la implicación lógica y/o alguna clase de conectivo rebatible (al estilo de ‘ \leftarrow ’ utilizado en DeLP).

Definición 50 (Framework Argumentativo con Agregación) *Un framework argumentativo con agregación (AAF – argument accrual framework) en una 3-tupla $(\mathcal{L}, \mathcal{R}, \succcurlyeq)$, donde:*

- \mathcal{L} es un lenguaje,
- \mathcal{R} es un conjunto de reglas de inferencia, y
- \succcurlyeq es una relación de preferencia (Def. 31) entre estructuras agregadas (a-estructuras) construibles a partir de \mathcal{L} y \mathcal{R} .

Un AAF involucra un lenguaje de representación, pero no establece el conocimiento concreto del dominio sobre el que se hará la argumentación (a ser especificado por el ingeniero de conocimiento). En tal sentido, a continuación se introduce formalmente la noción de *base de conocimiento* (respecto a un AAF).

Definición 51 (Base de Conocimiento) *Sea $\Gamma = (\mathcal{L}, \mathcal{R}, \succcurlyeq)$ un AAF. Una base de conocimiento (o KB para abreviar) es un subconjunto \mathcal{K} de \mathcal{L} .*

Como resultado de instanciar el AAF con un lenguaje, reglas de inferencia y relación de preferencia concretos, diremos que se obtiene un *sistema argumentativo con agregación* (AAS – argument accrual system). El siguiente ejemplo presenta un AAS Γ_{DeLPMt} , que puede verse como una extensión del sistema de agregación presentado en los Caps. 4, 5 y 6 (basado en el lenguaje de DeLP), incorporando (una versión de) la regla de inferencia deductiva conocida como modus tollens.

Ejemplo 21 *Sea Γ_{DeLPMt} un AAS $(\mathcal{L}_{DeLP}, \mathcal{R}, \succcurlyeq)$, donde:*

- \mathcal{L}_{DeLP} , representando formalmente el lenguaje de DeLP, es el menor conjunto (con respecto a ‘ \subseteq ’) tal que:

- si q es un literal, entonces $q \in \mathcal{L}_{DeLP}$,
- si q_0, q_1, \dots, q_k son literales, entonces $q_0 <- q_1, \dots, q_k \in \mathcal{L}_{DeLP}$,
- si q_0, q_1, \dots, q_k son literales, entonces $q_0 \prec q_1, \dots, q_k \in \mathcal{L}_{DeLP}$,
- si $s \in \mathcal{L}_{DeLP}$ entonces $\bar{s} \in \mathcal{L}_{DeLP}$ (cerrado bajo complemento).

■ \mathcal{R} está formado por las siguientes reglas de inferencia:

- $(a <- b_1, \dots, b_n) , b_1 , \dots , b_n \longrightarrow a$ *Modus ponens generalizado*
- $(a <- b_1, \dots, b_n) , \bar{a} , b_1 , \dots , b_{i-1} , b_{i+1} , \dots , b_n \longrightarrow \bar{b}_i$,
cualquiera sea i . *Modus tollens generalizado*
- $(a \prec b_1, \dots, b_n) , b_1 , \dots , b_n \implies a$ *Modus ponens rebatible*

■ \succcurlyeq , la relación de preferencia, se irá definiendo al momento de comparar a -estructuras en ejemplos subsiguientes.

La última condición en la especificación de \mathcal{L}_{DeLP} impone la clausura por complemento del conjunto, con el propósito de cumplir con la definición formal de lenguaje (Def. 48). Esto provoca la inclusión del complemento de reglas (del estilo de $\neg(q_0 \prec q_1, \dots, q_k)$), que a pesar de no tener utilidad en el contexto de Γ_{DeLPMt} , resultarán clave para modelar el ataque *undercut* en una extensión subsiguiente (Sec. 7.2).

El siguiente ejemplo presenta una base de conocimiento modelando, en términos de \mathcal{L}_{DeLP} , información acerca del problema de alquilar un departamento (presentado en el Cap. 1).

Ejemplo 22 Sea $\mathcal{K}_1 (\subseteq \mathcal{L}_{DeLP})$ una KB asociada a Γ_{DeLPMt} , conteniendo los siguientes elementos:

$alq \prec b_ubic$	$inq_estud \prec cerca_uns$	b_ubic
$\neg alq \prec pequeño$	$inq_estud \prec \neg edif_cochera$	$pequeño$
$\neg alq \prec humedad$	$entrada_autos <- edif_cochera$	$humedad$
$\neg alq \prec disturb$	$\neg entrada_autos$	$cerca_uns$
$disturb \prec inq_estud$		$dice_no_dist(j)$
$\neg disturb \prec dice_no_dist(V)$		$dice_no_dist(p)$

La información representada corresponde al problema del alquiler presentado en el Cap. 1, con algunas modificaciones para ilustrar la aplicación de modus tollens. Concretamente, en este caso Alicia no conoce efectivamente que los inquilinos del edificio son estudiantes (nótese que inq_stud ya no es un hecho). Sin embargo, sabe que el edificio está cerca de la universidad ($cerca_uns$), y que esto generalmente implica que sus inquilinos sean estudiantes. Además sabe que los edificios que no poseen cochera ($\neg edif_cochera$) generalmente son alquilados por estudiantes, que un edificio con cochera necesariamente tiene entrada para autos ($entrada_autos$), y que en particular el edificio en consideración no posee entrada para autos.

Finalmente se introduce la noción de argumento con respecto a una KB dada (en el contexto de un AAF), que involucra un conjunto minimal de (aplicaciones de) reglas de inferencia soportando la conclusión.

Definición 52 (Argumento) Sea $\Gamma = (\mathcal{L}, \mathcal{R}, \gg)$ un AAF. Sea \mathcal{K} una KB.

- Si $h \in \mathcal{K}$, entonces $\langle \emptyset, h \rangle$ es un argumento.
- Si $h \notin \mathcal{K}$, $R = k_1, \dots, k_n \implies h$ [$/k_1, \dots, k_n \longrightarrow h$] es una regla de inferencia rebatible [$/estricta$], $\langle \mathcal{B}_1, k_1 \rangle, \dots, \langle \mathcal{B}_n, k_n \rangle$ son argumentos, y vale que no existen $R_1, R_2 \in \mathcal{B}_1 \cup \dots \cup \mathcal{B}_n \cup \{R\}$ tal que $R_1 \neq R_2$ y $Conc(R_1) = Conc(R_2)$, entonces $\langle \mathcal{B}_1 \cup \dots \cup \mathcal{B}_n \cup \{R\}, h \rangle$ es un argumento.

La condición $h \notin \mathcal{K}$ en el caso recursivo, forzando la exclusión mutua con el caso base, impide que se emplee una regla de inferencia para concluir h cuando esta última se encuentra en la base de conocimiento (en ese caso la única posibilidad es la capturada por el caso base, considerando como soporte el conjunto vacío). Por otro lado, la condición sobre el conjunto $\mathcal{B}_1 \cup \dots \cup \mathcal{B}_n \cup \{R\}$ en el caso recursivo de la definición evita que un argumento soporte de dos maneras distintas una misma conclusión intermedia. Estas dos condiciones juntas aseguran la minimalidad del soporte del argumento, como se establece formalmente a continuación.

Proposición 15 (minimalidad del soporte) Sea $\langle \mathcal{A}, h \rangle$ un argumento. Entonces no existe argumento $\langle \mathcal{A}', h \rangle$ tal que $\mathcal{A}' \subset \mathcal{A}$.

Prueba. Supongamos por el absurdo que existe tal argumento $\langle \mathcal{A}', h \rangle$. Si $\mathcal{A}' = \emptyset$, por def. de argumento (caso base), $h \in \mathcal{K}$, y de esta forma $\langle \mathcal{A}', h \rangle$ es el único argumento para h (i.e., $\mathcal{A}' = \mathcal{A} = \emptyset$), arribando a un absurdo. Consideremos el otro caso ($\mathcal{A}' \neq \emptyset$). Ambos argumentos corresponden al caso recursivo en la Def. 52. Es decir, por un lado $\mathcal{A} = \mathcal{B}_1 \cup \dots \cup \mathcal{B}_n \cup \{R\}$, donde $R = k_1, \dots, k_n \implies h$ [$/k_1, \dots, k_n \longrightarrow h$] y $\langle \mathcal{B}_i, k_i \rangle$ son argumentos, $1 \leq i \leq n$, y por otro lado $\mathcal{A}' = \mathcal{B}'_1 \cup \dots \cup \mathcal{B}'_m \cup \{R'\}$, donde $R' = k'_1, \dots, k'_m \implies h$ [$/k'_1, \dots, k'_m \longrightarrow h'$] y $\langle \mathcal{B}'_i, k'_i \rangle$ son argumentos, $1 \leq i \leq m$. Dado que $R' \in \mathcal{A}'$, y $\mathcal{A}' \subset \mathcal{A}$ (hipótesis), $R' \in \mathcal{A}$. Pero por ser $\langle \mathcal{A}, h \rangle$ un argumento, y $R \in \mathcal{A}$, entonces debe ser que $R = R'$ (de lo contrario se violaría la condición impuesta en el caso recursivo de la definición de argumento sobre el conjunto $\mathcal{A} = \mathcal{B}_1 \cup \dots \cup \mathcal{B}_n \cup \{R\}$). Luego $k'_i = k_i$, $1 \leq i \leq n = m$. Consideremos ahora uno de los argumento $\langle \mathcal{B}'_i, k'_i \rangle$ arbitrariamente. Nuevamente surgen dos casos. Si $\mathcal{B}'_i = \emptyset$, entonces por la misma razón que antes, $\mathcal{B}'_i = \mathcal{B}_i = \emptyset$. En el otro caso ($\mathcal{B}'_i \neq \emptyset$), ambos argumentos corresponden al caso recursivo en la Def. 52, y por lo tanto cada uno tienen una regla en el tope (concluyendo k_i), R'_i y R_i , respectivamente. Dado que $R'_i \in \mathcal{A}'$, y $\mathcal{A}' \subset \mathcal{A}$ (hipótesis), $R'_i \in \mathcal{A}$. Pero nuevamente por ser $\langle \mathcal{A}, h \rangle$ un argumento, y $R_i \in \mathcal{A}$, entonces debe ser que $R_i = R'_i$. El razonamiento continúa hasta agotar el argumento $\langle \mathcal{A}', h \rangle$ (que por ser argumento tiene profundidad finita), comprobando regla por regla que $\mathcal{A}' = \mathcal{A}$, y arribando así a un absurdo. Luego, no existe argumento $\langle \mathcal{A}', h \rangle$ tal que $\mathcal{A}' \subset \mathcal{A}$ \square

Observación 1 *Algunas formalizaciones de argumentación, entre ellas DeLP, contemplan solo reglas rebatibles como soporte de los argumentos, dejando implícitas las reglas estrictas involucradas en la derivación de la conclusión. Luego, al exigir minimalidad del soporte en estos casos, se está ignorando a las reglas estrictas, es decir, los argumentos serán minimales en cuanto al conjunto de reglas rebatibles involucradas. Al incluir reglas de inferencia estrictas en el soporte (el caso de la presente generalización), la minimalidad del conjunto soporte (en el sentido de la inclusión de conjuntos) no implica minimalidad rebatible. A modo de ejemplo, considere dos argumentos $\langle \mathcal{A}_1, x \rangle$ y $\langle \mathcal{A}_2, x \rangle$ (respecto a una KB dada $\mathcal{K} = \{a, c\}$, y en el contexto de un AAF dado), con $\mathcal{A}_1 = \{a \longrightarrow x\}$ y $\mathcal{A}_2 = \{(c \implies b), (a, b \longrightarrow x)\}$. Sea $defeasible(X)$ el conjunto de reglas rebatibles involucradas en un conjunto dado de reglas de inferencia X . A pesar de que $\mathcal{A}_1 \not\subseteq \mathcal{A}_2$ (involucran distintas reglas estrictas), $defeasible(\mathcal{A}_1) \subset defeasible(\mathcal{A}_2)$, y por lo tanto $\langle \mathcal{A}_2, x \rangle$ no es minimal en un sentido rebatible.*

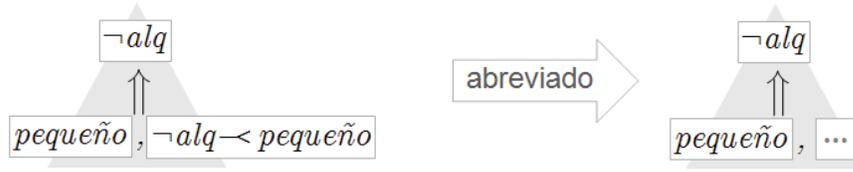


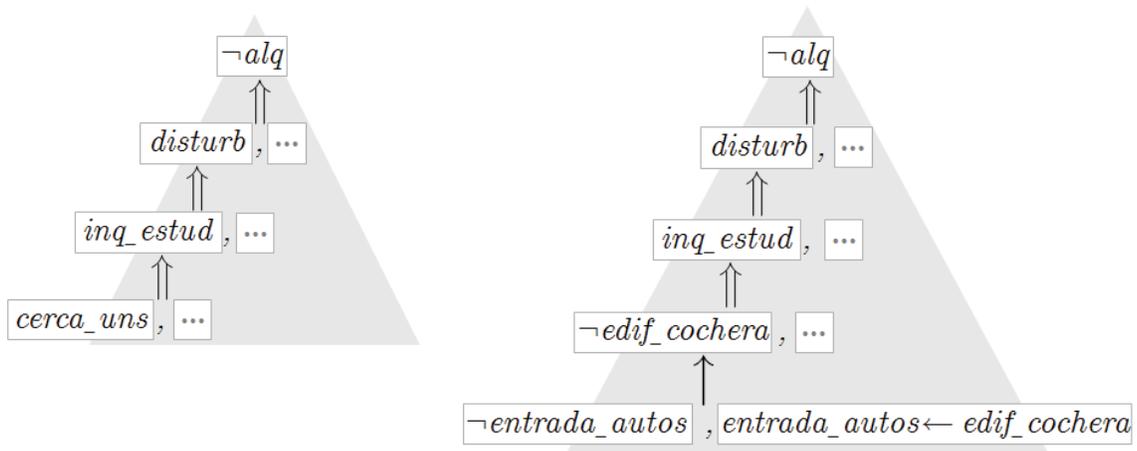
Figura 7.1: Argumento para $\neg alq$: notación gráfica completa y versión abreviada (ocultando la regla como premisa de modus ponens)

En caso de considerar deseable la minimalidad rebatible, debería incorporarse la siguiente condición al caso recursivo de la Def. 52: no debe ser el caso que exista regla $R' = k'_1, \dots, k'_n \implies h$ [$k'_1, \dots, k'_n \longrightarrow h$] y argumentos $\langle \mathcal{B}'_1, k'_1 \rangle, \dots, \langle \mathcal{B}'_n, k'_n \rangle$ tal que $defeasible(\mathcal{B}'_1 \cup \dots \cup \mathcal{B}'_n \cup \{R'\}) \subset defeasible(\mathcal{B}_1 \cup \dots \cup \mathcal{B}_n \cup \{R\})$. Finalmente debe notarse que la minimalidad del soporte como conjunto sigue siendo necesaria a pesar de imponer la minimalidad rebatible, ya que no queremos reglas estrictas redundantes en el soporte de los argumentos.

Ejemplo 23 La Fig. 7.1 muestra un argumento para $\neg alq$ respecto a \mathcal{K}_1 . En adelante, con el propósito de simplificar la representación gráfica para argumentos y a-estructuras, al notar la aplicación de modus ponens (tanto en su versión estricta como rebatible), se usarán puntos suspensivos (...) en lugar de la regla (de \mathcal{L}_{DeLP}) involucrada como premisa de la aplicación (Fig. 7.1). Nótese que la premisa (regla) omitida queda determinada unívocamente por el resto de los elementos de la aplicación de modus ponens, concretamente, su cuerpo está integrado por el resto de las premisas de la aplicación, y su cabeza coincide con la conclusión de la aplicación.

La Fig. 7.2 muestra dos argumentos para $\neg alq$. En particular, el de la derecha ilustra una aplicación de modus tollens.

La definición de a-estructura presentada en el Cap. 4 se aplica directamente al nuevo framework de argumentación con agregación, simplemente reemplazando la noción de argumento DeLP por la nueva noción de argumento respecto al AAF. Asimismo, las restantes definiciones introducidas en el Cap. 4 para a-estructuras (ataque parcial, derrota parcial, derrota combinada, etc.) se aplican directamente al nuevo framework, al igual que las tres semánticas de aceptabilidad para agregación introducidas en el Cap. 5 y el procedimiento de prueba dialéctico del Cap. 6.

Figura 7.2: Argumentos para $\neg alq$

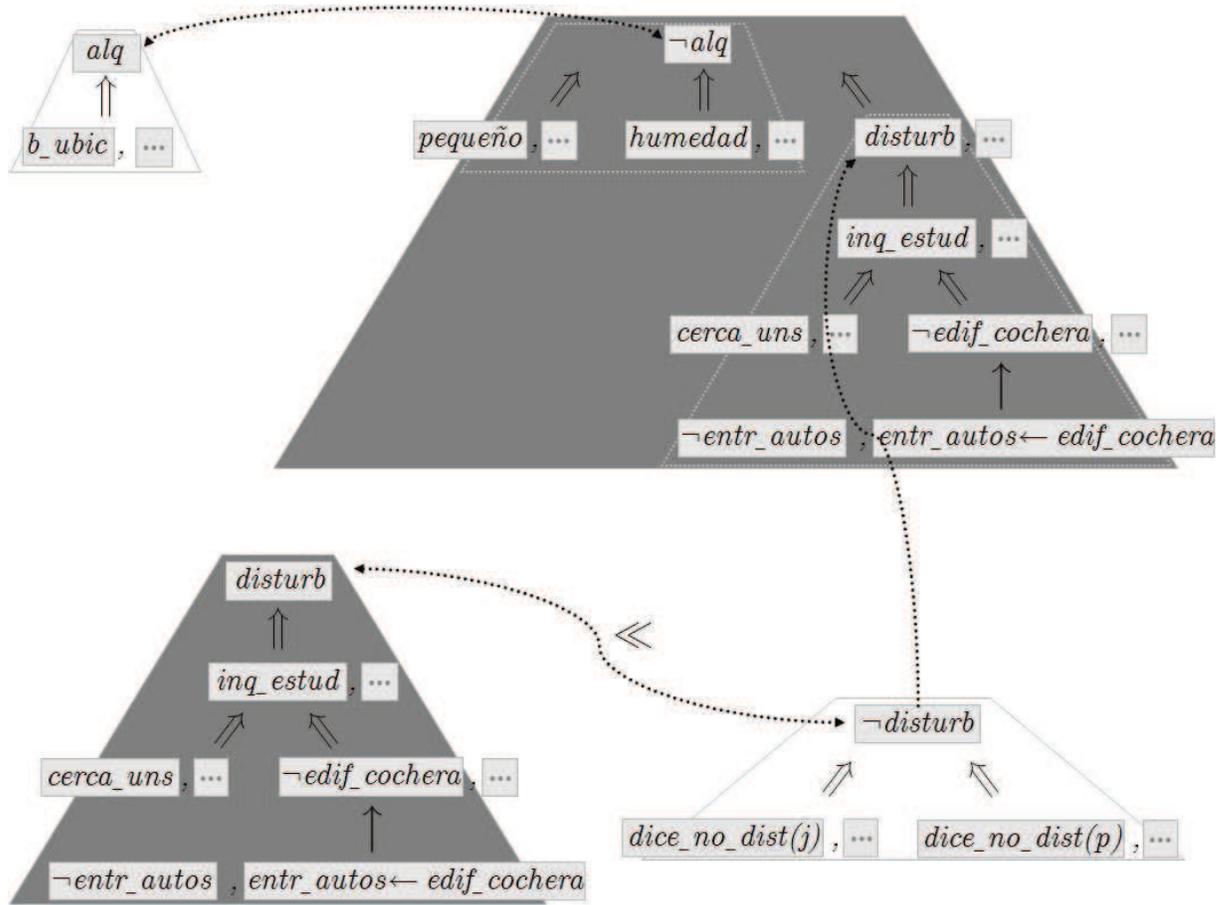


Figura 7.3: Asignación de Status para \mathcal{K}_1

La Fig. 7.3 muestra el grafo de ataques asociado a la KB \mathcal{K}_1 según el AAS Γ_{DeLPMT} , junto a la única asignación de status asociada, de acuerdo a la cual la a-estructura soportando alq se encuentra completamente IN. Se asumió que, de acuerdo al criterio de preferencia \succ , la a-estructura soportando $\neg disturb$ tiene más fuerza que aquella para $disturb$, causando así la derrota de esta última, y del narrowing de la a-estructura para $\neg alq$ que la involucra. Luego, a pesar de que la a-estructura soportando $\neg alq$ es preferida ante aquella soportando alq (de acuerdo a \succ), al sufrir la derrota del narrowing involucrando $disturb$, se debilita, resultando completamente derrotada, y prevaleciendo así la a-estructura para alq .

7.2. Modelando undercutting

El uso de razones *default* (también llamadas *rebatibles* o *prima facie*) es lo que hace a los argumentos rebatibles, ya que dichas razones pueden ser atacadas (y derrotadas) por otras razones [Vre92]. En la literatura sobre argumentación se distinguen principalmente dos formas de conflicto entre argumentos ampliamente aceptadas, identificadas originalmente por Pollock [Pol70]. Una de ellas, conocida como *rebutting attack*, se da cuando dos argumentos soportan conclusiones contrarias. Trayendo el ejemplo empleado por Pollock, si un cierto objeto se ve rojo para mí, esto me da una razón rebatible (argumento) para creer que es rojo. Sin embargo, si Jones, a quien considero confiable, insiste que no es rojo, cuento también con un argumento para creer en la conclusión contraria a la del primer argumento, y por lo tanto existe un conflicto entre estos dos argumentos (*rebutting attack*). La otra forma de ataque, conocida como *undercutting attack*, se da cuando un argumento cuestiona la conexión entre premisas y conclusión de una regla rebatible de otro argumento. Consideremos nuevamente el argumento concluyendo que un cierto objeto es rojo dado que a la vista parece rojo. Si además se sabe que el objeto está siendo iluminado por una luz roja, entonces tengo una razón para cuestionar la conexión entre premisa ('se ve rojo') y conclusión ('es rojo') del otro argumento, ya que la luz roja puede hacer que el objeto se vea rojo, aunque realmente no lo sea. Nótese que este último argumento no permite concluir que el objeto no es rojo, simplemente sostiene que no debería concluirse que es rojo por verse rojo.

Hasta ahora sólo hemos contemplado el *rebutting attack* en nuestro enfoque con agregación. A continuación presentaremos el AAS Γ_{Uc} , permitiendo modelar *undercutting attack*. El enfoque propuesto se basa en la estrategia para modelar *undercutting* planteada por Vreeswijk en [Vre92], de acuerdo a la cual tanto el *undercutting* como el *rebutting* se consiguen como casos particulares de la noción de conflicto adoptada en nuestra formalización, simplemente definiendo apropiadamente el lenguaje objeto \mathcal{L} . En otras palabras, el AAF no debe cambiarse en absoluto, simplemente instanciarse con un lenguaje más expresivo.

Consideremos un AAS $\Gamma_{Uc} = (\mathcal{L}_{Uc}, \mathcal{R}, \gg)$, donde:

- \mathcal{L}_{Uc} (lenguaje permitiendo expresar *undercut*) es el menor conjunto (con respecto a ' \subseteq ') tal que:

- si q es un literal, entonces $q \in \mathcal{L}_{Uc}$,
 - si $s \in \mathcal{L}_{Uc}$ tal que s no involucra ' \leftarrow ', y q_1, \dots, q_k son literales, entonces $s \leftarrow q_1, \dots, q_k \in \mathcal{L}_{Uc}$,
 - si $s \in \mathcal{L}_{Uc}$ tal que s no involucra ' \leftarrow ', y q_1, \dots, q_k son literales, entonces $s \prec q_1, \dots, q_k \in \mathcal{L}_{Uc}$,
 - si $s \in \mathcal{L}_{Uc}$ entonces $\bar{s} \in \mathcal{L}_{Uc}$ (cerrado bajo complemento).
- \mathcal{R} está formado por las reglas de inferencia Modus Ponens Generalizado y Modus Ponens Rebatible, presentadas al definir Γ_{DeLPMt} .
 - \succcurlyeq , la relación de preferencia, se irá definiendo al momento de comparar a-estructuras en ejemplos subsiguientes.

Nótese que \mathcal{L}_{Uc} admite no solo literales en la cabeza de las reglas, sino también reglas rebatibles (y la negación de reglas rebatibles). A continuación se ilustra cómo modelar el ejemplo de *undercutting* propuesto por Pollock empleando \mathcal{L}_{Uc} .

Ejemplo 24 *Considérese una KB \mathcal{K}_{Pol} respecto al AAS Γ_{Uc} , conteniendo los siguientes elementos:*

$$\begin{aligned}
 & (is_red \prec looks_red) \prec \\
 & \neg is_red \prec jones_says_not_red \\
 & \neg(is_red \prec looks_red) \prec ilum_red_light \\
 & looks_red \\
 & ilum_red_light
 \end{aligned}$$

El tercer elemento expresa que si un cierto objeto se encuentra iluminado por una luz roja, entonces el hecho de que aparente ser rojo ya no es una razón para creer que lo es. Nótese que se trata de una regla cuya cabeza es la negación de una regla rebatible, y es la forma en la que se modelará el *undercutting*. Antes de analizar el primer elemento, debemos presentar la noción de *presuposición*. Se denomina presuposición a una regla rebatible con cuerpo vacío, donde, siguiendo la semántica de las reglas rebatibles, una presuposición ' $h \prec$ ' expresa que "en general, vale h ". El propósito del primer elemento es modelar que lo que aparenta ser rojo, generalmente es rojo. Sin embargo, en lugar de modelarlo simplemente con una regla rebatible $is_red \prec looks_red$, se emplea una presuposición con dicha regla en la cabeza. De esta forma se está expresando que la regla

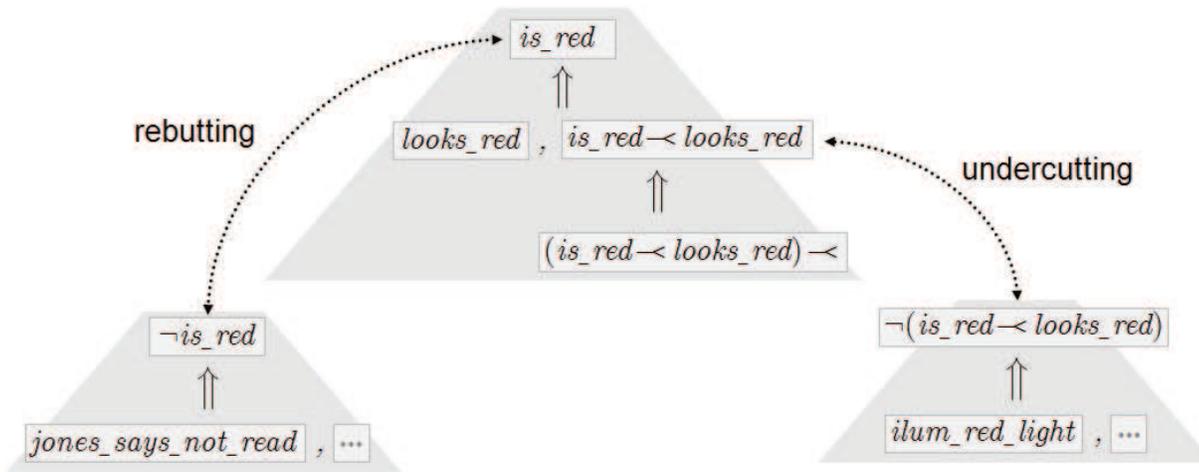


Figura 7.4: Ataque *undercutting* en nuestra formalización

$is_red \prec looks_red$ vale en general, pero podrían existir situaciones que la invaliden, o en otras palabras, la regla es susceptible a sufrir *undercutting*. Luego, como “regla del pulgar” para el ingeniero de conocimiento (encargado de expresar la información del dominio), las reglas rebatibles susceptibles a ser cuestionadas (*undercutting*) deben expresarse como cabezas de presuposiciones. La Fig. 7.4 muestra el grafo de ataques asociado a \mathcal{K}_{Pol} , ilustrando los dos tipos de ataques propuestos por Pollock como casos particulares de nuestra noción de ataque. A modo de simplificar el dibujo se omitió el nodo correspondiente a la a-subestructura de desacuerdo asociada al ataque undercutting (recordar que las a-subestructuras de desacuerdo son en sí mismas a-estructuras, y por lo tanto forman parte del grafo de ataques.)

A continuación se extiende el ejemplo original del alquiler del departamento para ilustrar el undercutting en presencia de agregación.

Ejemplo 25 Sea \mathcal{K}_2 una KB formada por los siguientes elementos:

$alq \prec b_ubic$	b_ubic
$\neg alq \prec pequeño$	$pequeño$
$\neg alq \prec humedad$	$humedad$
$\neg alq \prec disturbios$	inq_estud
$disturb \prec inq_estud$	$dice_no_dist(j)$
$(\neg disturb \prec dice_no_dist(V)) \prec$	$dice_no_dist(p)$
$\neg(\neg disturb \prec dice_no_dist(V)) \prec prob_audic(V)$	$prob_audic(j)$

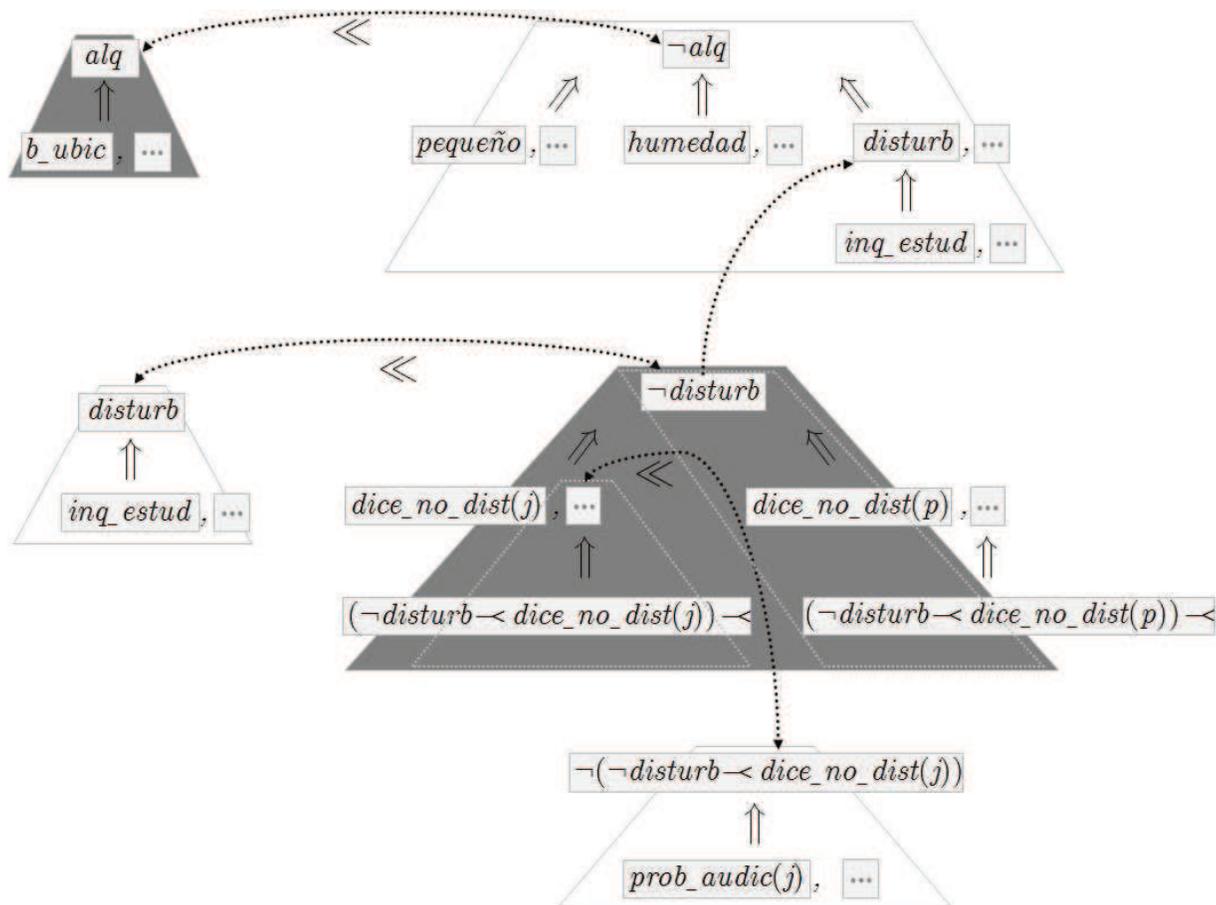
La regla $\neg(\neg\text{disturb} \prec \text{dice_no_dist}(V)) \prec \text{prob_audic}(V)$ expresa que si un vecino V tiene problemas de audición, entonces no debería considerarse su testimonio como una razón para creer que no hay disturbios, ya que aunque existieran, el seguramente no los percibiría.

La Fig. 7.5 muestra el grafo de ataques asociado a la KB \mathcal{K}_2 según el AAS Γ_{U_c} , junto a la única asignación de status asociada. La a -estructura para disturb sufre la derrota de un narrowing consecuencia de un ataque undercutting. Nótese que se está asumiendo que, de acuerdo a \succcurlyeq , la a -estructura soportando $\neg(\neg\text{disturb} \prec \text{dice_no_dist}(j))$ tiene más fuerza que aquella soportando $\neg\text{disturb} \prec \text{dice_no_dist}(j)$, y eso tiene sentido dado que esta última a -estructura se basa en una presuposición. Luego, a pesar de que la a -estructura soportando $\neg\text{disturb}$ es preferida ante aquella soportando disturb (de acuerdo a \succcurlyeq), esta primera se debilita al sufrir el undercutting antes mencionado. Como consecuencia resulta completamente derrotada, prevaleciendo la a -estructura para disturb . Finalmente, la a -estructura para $\neg\text{alq}$ no sufre derrota de narrowing alguno, prevaleciendo sobre aquella para alq .

7.3. Refinamiento para evitar la agregación de razones menos específicas

Luego de presentar su formalización de agregación, Prakken propone un refinamiento para evitar la agregación de razones menos específicas ([Pra05]). En otras palabras, este refinamiento evita la agregación de una razón dada soportando x , en presencia de una razón (“activa”) más específica para x . En esta sección se presenta una adaptación de dicho refinamiento para nuestro enfoque. Para comprender mejor la idea del refinamiento, considérese el siguiente ejemplo (adaptado de [Pra05]). Sea $\mathcal{K}_3 \subseteq \mathcal{L}_{U_c}$ una KB que contiene los siguientes elementos:

$$\begin{aligned} & \text{verídico}(X) \prec \text{testifíca}(P, X) \\ & \text{verídico}(X) \prec \text{testifíca}(P, X), \text{ conf_ant}(P) \\ & \text{testifíca}(p_1, x_1) \\ & \text{conf_ant}(p_1) \end{aligned}$$

Figura 7.5: Asignación de status para \mathcal{K}_2

La primer regla expresa que si una persona dada P testifica X , entonces generalmente X es verídico. La segunda regla expresa que si P testifica X , y además P ha probado ser confiable en el pasado, entonces seguramente X sea verídico. Nótese que la segunda regla modela una situación más específica, donde no solo se tiene en cuenta el hecho de que P haya testificado X , sino la confiabilidad que P ha demostrado tener en el pasado. Por supuesto, la primera regla sigue siendo necesaria a pesar de la más específica, dado que podría ocurrir que no se tenga información alguna acerca de la confiabilidad de un cierto testigo P , pero de todas maneras se considera que simplemente por haberse pronunciado como testigo debería creerse en su testimonio.

Puede verse que la KB \mathcal{K}_3 da lugar a dos argumentos para creer en la veracidad del testimonio x_1 de p_1 , uno más específico que el otro. Claramente, no queremos que estos dos argumentos se agreguen, dado que el más específico ya contempla el testimonio de p_1 , y además tiene en cuenta más información. En conclusión, la existencia de un argumento más específico, y no derrotado, para una conclusión dada, debería inhabilitar a uno menos específico para la misma conclusión, evitando así que se incorpore en la agregación.

Para modelar esta condición simplemente basta con incorporar la siguiente regla de inferencia estricta al AAS Γ_{Uc} :

$$(a \prec b_1, \dots, b_n) \ , \ b_1 \ , \ \dots \ , \ b_n \ \longrightarrow \ \neg(a \prec b'_1, \dots, b'_k)$$

dde. $\{b'_1, \dots, b'_k\} \subset \{b_1, \dots, b_n\}$

Esta regla de inferencia permite concluir la negación de una regla rebatible dada (*undercutting*) en caso de existir otra regla rebatible más específica aplicable. Denominaremos Γ_{Sp} al nuevo AAS. Finalmente, dado que las reglas rebatibles menos específicas son susceptibles a sufrir *undercutting*, deben ser expresadas como presuposiciones.

La Fig. 7.6 muestra el grafo de ataques para la KB \mathcal{K}_3' , junto a la única asignación de status asociada, donde \mathcal{K}_3' resulta de cambiar $verídico(X) \prec testifica(P, X)$ por la presuposición $(verídico(X) \prec testifica(P, X)) \prec$ en \mathcal{K}_3 .

Para terminar de entender por completo el efecto de la nueva regla de inferencia, consideremos la siguiente situación. El argumento más específico que sustenta la conclusión a podría sufrir una derrota en una de las sub-conclusiones que no comparte con el menos específico. Esto es, la situación más específica en realidad no aplica, con lo cual el argumento menos específico es el que debería considerarse en realidad. Puede verse que la nueva regla

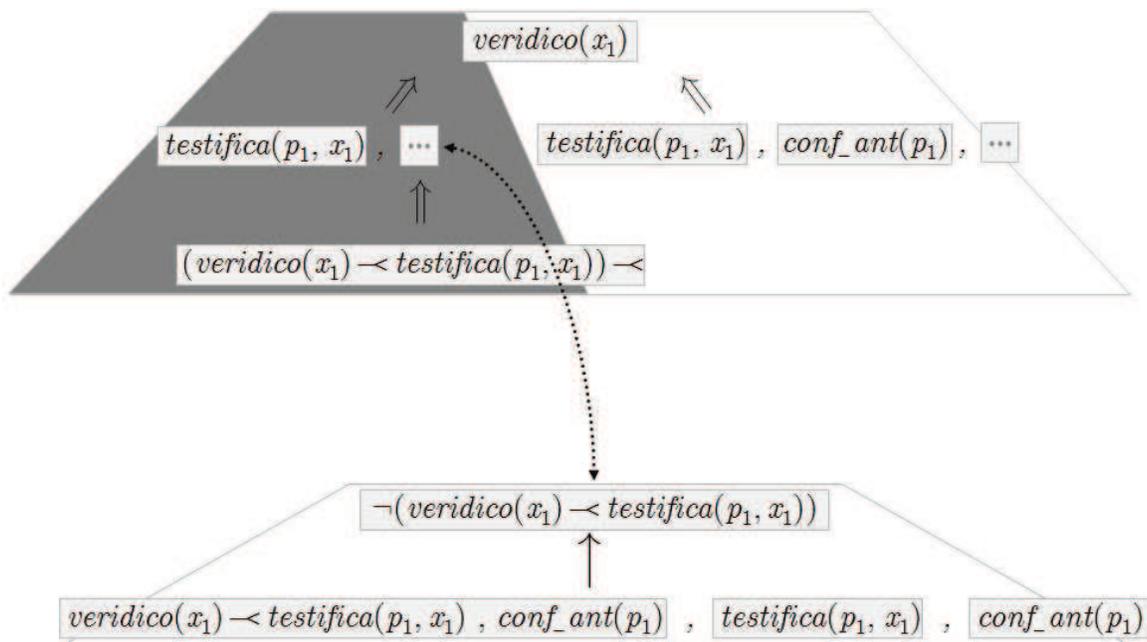


Figura 7.6: Asignación de status para \mathcal{K}_3'

de inferencia consigue el efecto deseado. Si el argumento más específico sufriera la derrota en una de sus sub-conclusiones, luego la a-estructura soportando a (e involucrando a ambos argumentos) sufrirá consecuentemente la derrota del narrowing involucrando a dicho argumento más específico. Adicionalmente, la a-estructura involucrando la nueva regla de inferencia y provocando el *undercut* del argumento menos específico también resultará derrotada (nótese que las premisas de la nueva regla de inferencia coinciden con las de argumento más específico), con lo cual el argumento menos específico resultará finalmente reinstaurado.

7.4. Incorporación de manejo explícito de incertidumbre posibilística

El propósito de esta sección es mostrar que el *AAF* presentado previamente puede ser instanciado para incorporar el manejo explícito de incertidumbre posibilística. Concretamente se definirá un *AAS* permitiendo asociar grados de necesidad o certeza (valores reales en el intervalo $[0, 1]$) a las sentencias a nivel del lenguaje objeto (\mathcal{L}), y que empleará dichos valores posibilísticos para definir la evaluación y comparación entre a-estructuras (relación de preferencia ‘ \gg ’). La instanciación que propondremos se basa parcialmente en el sistema argumentativo P-DeLP (Possibilistic DeLP [CSAG04, ACGS08, ACG⁺08]), que extiende las capacidades para razonamiento cualitativo de DeLP mediante la incorporación de incertidumbre posibilística. En la sección siguiente se describen brevemente los elementos de P-DeLP relevantes a nuestro enfoque. En la Sec. 7.4.2 se presenta finalmente el *AAS* mencionado. Cabe destacar que el enfoque adoptado para integrar agregación e incertidumbre posibilística se encuentra publicado en un artículo propio [LCS09a].

7.4.1. Argumentación con incertidumbre posibilística: P-DeLP

En esta sección se presentan los elementos E1-E4 (del esquema conceptual de argumentación presentado en el Cap. 1) para P-DeLP. El lenguaje lógico subyacente de P-DeLP (elemento E1) permite expresar incertidumbre posibilística en forma explícita mediante *cláusulas pesadas*. Una cláusula pesada es un par (φ, α) , donde φ es una regla $q \leftarrow p_1 \wedge \dots \wedge p_k$ o un hecho q , donde q, p_1, \dots, p_k son literales, y $\alpha \in [0, 1]$ expresa una

cota inferior para el grado de necesidad de φ . También diremos que la cláusula φ se encuentra etiquetada con el grado de necesidad α . Distinguiremos entre cláusulas *ciertas* e *inciertas*. Una cláusula (φ, α) se considera cierta si $\alpha = 1$ e incierta en caso contrario.

Como noción de consecuencia lógica asociada al lenguaje se adopta ‘ \vdash_P ’, representando deducción mediante la siguiente versión posibilística de modus ponens generalizado:

$$\frac{(q \prec -p_1 \wedge \dots \wedge p_k, \alpha) \quad (p_1, \beta_1), \dots, (p_k, \beta_k)}{(q, \min(\alpha, \beta_1, \dots, \beta_k))} [PMP]$$

A modo de ejemplo, sea $\Lambda = \{(a \prec b, c, 0.6), (b, 1), (c \prec d, 0.8), (d, 0.7)\}$. Entonces $\Lambda \vdash_P (d, 0.7)$, $\Lambda \vdash_P (c, 0.7)$ y $\Lambda \vdash_P (a, 0.6)$.

Un conjunto de cláusulas pesadas Λ será considerado *contradictorio*, denotado $\Lambda \vdash_P \perp$, si para algún átomo *avale* que $\Lambda \vdash_P (a, \alpha)$ y $\Lambda \vdash_P (\sim a, \beta)$. Finalmente un programa P-DeLP es un par (Π, Δ) , donde Π es un conjunto finito y no contradictorio de cláusulas ciertas, y Δ es un conjunto finito de cláusulas inciertas. A continuación de definen las nociones de argumento (y subargumento) respecto a un programa P-DeLP dado (elemento E2) y de ataque (elemento E3). Estas definiciones son muy similares a las correspondientes para DeLP.

Definición 53 (Argumento) Sea $\mathcal{P} = (\Pi, \Delta)$ un programa P-DeLP. Diremos que un conjunto $\mathcal{A} \subseteq \Delta$ es un argumento para un literal h con grado de necesidad $\alpha > 0$, denotado $\langle \mathcal{A}, h, \alpha \rangle$, sssi:

1. $\mathcal{A} \vdash_P (h, \alpha)$,
2. $\Pi \cup \mathcal{A}$ es no contradictorio,
3. \mathcal{A} es minimal (con respecto a ‘ \subseteq ’), i.e., no existe $\mathcal{A}' \subset \mathcal{A}$ tal que $\mathcal{A}' \vdash_P (h, \alpha)$.

Definición 54 (Subargumento) Un argumento $\langle \mathcal{S}, k, \gamma \rangle$ es un sub-argumento de $\langle \mathcal{A}, h, \alpha \rangle$ sssi $\mathcal{S} \subseteq \mathcal{A}$.

Para la siguiente definición, usaremos \bar{h} para denotar el complemento de h respecto a ‘ \sim ’, es decir, \bar{h} denota “ $\sim a$ ” si $h \equiv a$, y “ a ” si $h \equiv \sim a$.

Definition 5 (Ataque) Sea $\mathcal{P} = (\Pi, \Delta)$ un programa P-DeLP, y sean $\langle \mathcal{A}, h, \alpha \rangle$ y $\langle \mathcal{B}, k, \beta \rangle$ dos argumentos respecto a \mathcal{P} . Diremos que $\langle \mathcal{B}, k, \beta \rangle$ ataca $\langle \mathcal{A}, h, \alpha \rangle$ (en el literal \bar{k}) sssi existe un subargumento (denominado subargumento de desacuerdo) $\langle \mathcal{S}, \bar{k}, \gamma \rangle$ de $\langle \mathcal{A}, h, \alpha \rangle$.

Finalmente la evaluación y comparación de argumentos (elemento E4) se basa en los valores posibilísticos asociados a los argumentos en conflicto, dando lugar a la siguiente definición de derrota.

Definición 55 (Derrota) Sean $\langle \mathcal{A}, h, \alpha \rangle$ y $\langle \mathcal{B}, k, \beta \rangle$ dos argumentos. Diremos que $\langle \mathcal{B}, k, \beta \rangle$ derrota a $\langle \mathcal{A}, h, \alpha \rangle$ (o de manera equivalente que $\langle \mathcal{B}, k, \beta \rangle$ es un derrotador para $\langle \mathcal{A}, h, \alpha \rangle$) si 1) $\langle \mathcal{B}, k, \beta \rangle$ ataca $\langle \mathcal{A}, h, \alpha \rangle$, donde $\langle \mathcal{S}, \bar{k}, \gamma \rangle$ es el subargumento de desacuerdo, y 2) $\beta \geq \gamma$.

7.4.2. Agregación con incertidumbre posibilística

A continuación presentaremos el AAS $\Gamma_P = (\mathcal{L}_P, \mathcal{R}_P, \gg_P)$, permitiendo, al igual que P-DeLP, representar en forma explícita grados de necesidad asociados a las sentencias de la KB, y definir la preferencia entre a-estructuras en base a estos valores posibilísticos. En P-DeLP, el grado de necesidad (o fuerza posibilística) asociado a un argumento se obtiene propagando grados de necesidad mediante la regla de inferencia *PMP*. En nuestro sistema, la fuerza de una a-estructura (en que se basará la definición de \gg_P) se obtendrá combinando el efecto de *PMP* para propagar grados de necesidad a través de reglas de inferencia, con un mecanismo para agregar los grados de necesidad provenientes de diferentes reglas de inferencia para la misma conclusión.

Lenguaje: \mathcal{L}_P

El lenguaje \mathcal{L}_P se define como el menor conjunto (con respecto a ‘ \subseteq ’) tal que:

- si q es un literal, entonces $q \in \mathcal{L}_P$,
- si q_0, q_1, \dots, q_k son literales, entonces $q_0 \leftarrow q_1, \dots, q_k \in \mathcal{L}_P, k > 0$,
- si q_0, q_1, \dots, q_k son literales, entonces $q_0 \prec q_1, \dots, q_k \in \mathcal{L}_P, k \geq 0$,

- si q_0, q_1, \dots, q_k son literales y $\alpha \in [0, 1)$, entonces $(q_0 \prec q_1, \dots, q_k, \alpha) \in \mathcal{L}_P$, $k \geq 0$,
- si $s \in \mathcal{L}_{DeLP}$ entonces $\bar{s} \in \mathcal{L}_P$ (cerrado bajo complemento).

Observación: el lenguaje permite expresar presuposiciones ($q_0 \prec$ — considerando $k = 0$ en el tercer ítem) y presuposiciones etiquetadas ($(q_0 \prec, \alpha)$ — considerando $k = 0$ en el cuarto ítem).

Nótese que \mathcal{L}_P presenta algunas diferencias respecto al lenguaje de P-DeLP, aunque son equivalentes en cuanto a lo que permiten expresar. En lugar de emplear un único tipo de conectivo para las reglas (\leftarrow), y distinguir entre estrictas y rebatibles (o ciertas e inciertas) a partir del grado de necesidad asociado, se utilizan dos conectivos para distinguir (\prec para estrictas y \prec para rebatibles), al igual que en DeLP. A su vez, las reglas rebatibles se presentan en dos versiones: una con grado de necesidad asociado (o etiquetadas), con la restricción de que dicho grado de necesidad debe ser menor a 1, y otra sin grado de necesidad asociado. Solo los hechos, las reglas estrictas y las rebatibles etiquetadas (incluyendo presuposiciones etiquetadas) se emplearán para expresar información del problema, es decir, integrarán la KB. El propósito de las reglas rebatibles no etiquetadas se explicará luego de presentar las reglas de inferencia.

Ejemplo 26 (KB posibilística) Sea \mathcal{K}_4 una KB formada por los siguientes elementos:

$(a \prec b, c, 0.7)$	$(b \prec e, 0.6)$	c
$a \prec d$	$(b \prec f, 0.5)$	e
	$(d \prec g, 0.3)$	f
		g

Reglas de Inferencia: \mathcal{R}_P

El conjunto \mathcal{R}_P está formado por las siguientes reglas de inferencia:

- $(a \prec b_1, \dots, b_n, \alpha) \longrightarrow (a \prec b_1, \dots, b_n),$ Desetiquetado
- $(a \prec b_1, \dots, b_n), b_1, \dots, b_n \longrightarrow a$ Modus ponens generalizado

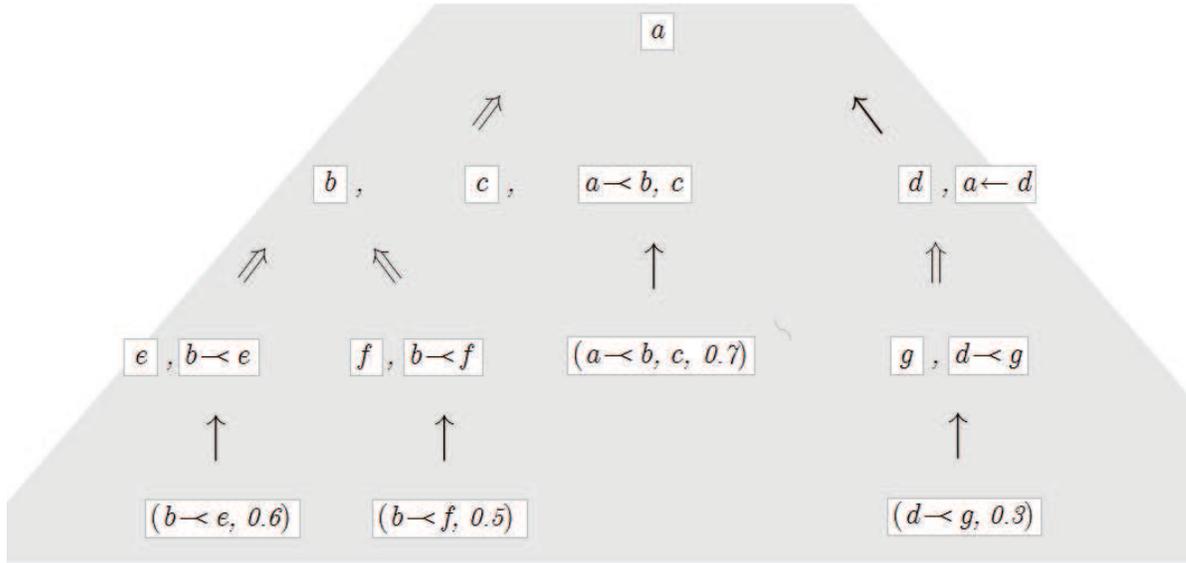


Figura 7.7: a-estructura $[\Phi, a]$ en Γ_P

$$- (a \prec b_1, \dots, b_n) , b_1 , \dots , b_n \implies a \quad \text{Modus ponens rebatible}$$

La primer regla de inferencia tiene como único propósito desetiquetar reglas rebatibles de la KB para que luego puedan aplicarse normalmente las restantes reglas de inferencia, es decir, sin la interferencia de las etiquetas. Luego, los argumentos y a-estructuras del sistema tendrán sentencias etiquetadas solo en su “base”, y se ahí hacia arriba serán análogos a los de los AAS introducidos previamente, permitiendo que la agregación y la detección de conflictos se dé normalmente ¹. La Fig. 7.7 muestra la a-estructura maximal $[\Phi, a]$ de acuerdo a \mathcal{K}_4 .

Relación de Preferencia: \gg_P

La relación \gg_P estará basada en la fuerza posibilística de las a-estructuras (un valor posibilístico, entre 0 y 1), que se obtendrá a partir de las sentencias etiquetadas que la a-estructura involucra. Para obtener la fuerza de una a-estructura $[\Phi, h]$ se emplearán dos funciones mutuamente recursivas: $str_{\Phi}^+(\cdot)$ (agregación de fuerza o función de agregación) y

¹Nótese que si las conclusiones de los argumentos fueran sentencias etiquetadas, dos argumentos soportando (h, α) y (h, β) no se agregarían, porque estrictamente están soportando conclusiones distintas. De la misma manera, dos argumentos soportando (h, α) y $(\neg h, \beta)$ no estarían en conflicto, dado que las conclusiones no son complementarias.

$str_{\Phi}^{inf}(\cdot)$ (fuerza de la inferencia – propagando grados de necesidad como *PMP*). Dado que no queremos comprometernos a una fórmula matemática específica para agregar grados de necesidad, asumiremos que $str_{\Phi}^{+}(\cdot)$ se encuentra parametrizada con respecto a una función *ACC* a ser especificada por el usuario.

Definición 56 ($str_{\Phi}^{+}(\cdot)$ y $str_{\Phi}^{inf}(\cdot)$) *Sea $[\Phi, h]$ una a-estructura. Sea $s \in \mathcal{L}_P$ tal que s aparece en Φ (como cabeza y/o como parte del cuerpo de alguna regla de inferencia en Φ). Sean R_1, \dots, R_n todas las reglas de inferencia en Φ con cabeza s . Entonces*

$$str_{\Phi}^{+}(s) =_{def} \begin{cases} \beta & \text{si } n = 0 \text{ y } s = (q, \beta); \\ 1 & \text{si } n = 0 \text{ y } s \text{ es una} \\ & \text{sentencia sin etiquetar } q; \\ ACC(str_{\Phi}^{inf}(R_1), \dots, str_{\Phi}^{inf}(R_n)) & \text{si } n > 0 \end{cases}$$

Sea $R \in \Phi$. Entonces

$$str_{\Phi}^{inf}(R) =_{def} \min(str_{\Phi}^{+}(p_1), \dots, str_{\Phi}^{+}(p_n)) \quad \text{dde. } R = p_1, \dots, p_n \longrightarrow p \\ \text{o } R = p_1, \dots, p_n \implies p$$

Intuitivamente, $str_{\Phi}^{+}(s)$ representa la fuerza o grado de necesidad resultante de agregar todas las razones soportando s en Φ . Si no existen reglas de inferencia soportando s (caso base) estamos en presencia de un elemento de la KB, y en ese caso $str_{\Phi}^{+}(s)$ retornará el grado de necesidad asociado a s , si se trata de una regla etiquetada, o 1 si se trata de una sentencia sin etiquetar (regla estricta o hecho). En caso de existir una o más reglas de inferencia soportando s , $str_{\Phi}^{+}(s)$ es el resultado de agregar (mediante *ACC*) las fuerzas “provenientes” de dichas reglas de inferencia ($str_{\Phi}^{inf}(R_i)$). La función $str_{\Phi}^{inf}(R)$ representa la fuerza con que la regla de inferencia R soporta su conclusión en Φ , y se calcula como el mínimo de las fuerzas de sus premisas (al igual que *PMP*).

Finalmente, dada una a-estructura $[\Phi, h]$, consideraremos que $str_{\Phi}^{+}(h)$ representa su fuerza, y luego la relación de preferencia \succcurlyeq_P se definirá como sigue. Sean $[\Phi, h]$ y $[\Psi, k]$ dos a-estructuras. Entonces

$$[\Phi, h] \succcurlyeq_P [\Psi, k] \quad \text{ssi} \quad str_{\Phi}^{+}(h) \geq str_{\Psi}^{+}(k)$$

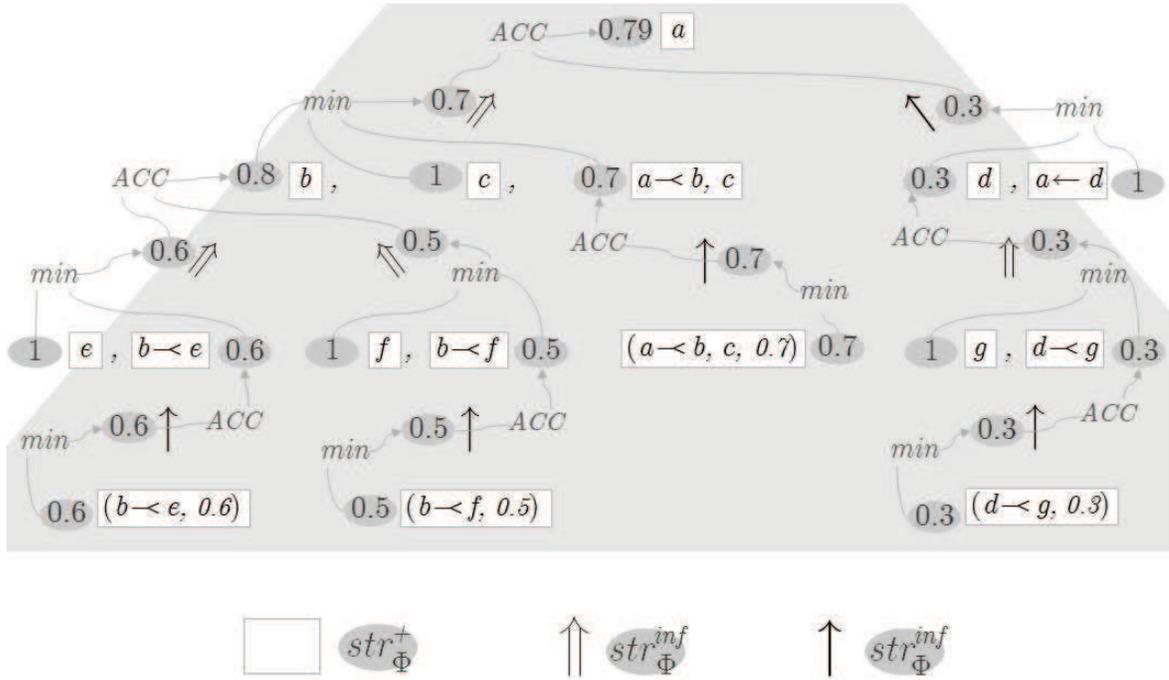


Figura 7.8: Fuerza de una a-estructura

Antes de mostrar un ejemplo del cálculo de la fuerza de una a-estructura debemos dar una instancia concreta para la función ACC . Adoptaremos la función $ACC_{1'}$ (*agregación 1-complemento*) presentada en el trabajo previo [LCS09a], y definida como sigue:

$$ACC_{1'}(\alpha_1, \dots, \alpha_n) =_{def} 1 - \prod_{i=1}^n (1 - \alpha_i)$$

La Fig. 7.8 muestra cómo se obtiene la fuerza de la a-estructura $[\Phi, a]$ ($str_{\Phi}^+(a)$) empleando las funciones $str_{\Phi}^+(\cdot)$ y $str_{\Phi}^{inf}(\cdot)$. Tal como indican las referencias en la parte inferior de la figura, los óvalos grises al costado de las sentencias (rectángulos blancos) representan los valores de $str_{\Phi}^+(\cdot)$ para dichas sentencias, mientras que los óvalos grises al costado de las reglas de inferencia representan los valores de $str_{\Phi}^{inf}(\cdot)$ asociados a las mismas.

Finalmente, resulta importante aclarar que no cualquier instanciación para ACC resulta adecuada, ya que debe procurarse que la relación \succcurlyeq_P resultante cumpla con el requerimiento establecido en la Def. 31 de relación de preferencia, esto es, que agregar nunca debe debilitar. Puede demostrarse que ACC resultará adecuada si y solo si satisface el siguiente requerimiento, identificado en [LCS09a] bajo el nombre de *no depreciación* (*non-depreciation*):

$$ACC(\alpha_1, \dots, \alpha_n) \geq \max(\alpha_1, \dots, \alpha_n) \quad \text{[No depreciación]}$$

Otro requerimiento para ACC identificado en [LCS09a], no fundamental para nuestra formalización, pero sensato, es *maximalidad*, estableciendo que solo puede obtenerse grado de certeza 1 como resultado de agregar cuando alguno de los argumentos agregados tiene grado de certeza 1, o en otras palabras, no puede lograrse certeza total a partir de agregar solo argumentos inciertos. Formalmente:

$$ACC(\alpha_1, \dots, \alpha_n) = 1 \text{ solo si } \alpha_i = 1 \text{ para algún } i, 1 \leq i \leq n. \quad [\mathbf{Maximalidad}]$$

7.5. Conclusiones

En una primera instancia, con el propósito de simplificar la presentación del enfoque de agregación (Caps. 4, 5 y 6), se adoptó el lenguaje de representación y la noción de argumento de DeLP. En este capítulo se presentó la noción de *Framework Argumentativo con Agregación (AAF)*, definiendo un lenguaje de representación y noción de argumento más generales, sobre los cuales se aplica directamente toda la formalización de agregación presentada (ver esquema de la Fig. 7.9).

Asimismo se definieron distintos sistemas (*AAS*) como instancias del framework, incorporando el modelamiento de diferentes características (ver Fig. 7.9). En primer lugar se presentó el *AAS* Γ_{DeLPMt} , adoptando como lenguaje de representación el de DeLP e incorporando (una versión de) la regla de inferencia deductiva conocida como *modus tollens*.

Luego se presentó el *AAS* Γ_{Uc} , que adopta un lenguaje de representación permitiendo modelar el tipo de ataque *undercutting* como caso particular de la noción de conflicto empleada en nuestra formalización.

Prakken propuso en [Pra05] que razones menos específicas no deberían agregarse en presencia de razones “activas” (*i.e.*, no derrotadas) más específicas, y presentó un refinamiento de su formalización para contemplar este principio. En este sentido, en el presente capítulo se definió el *AAS* Γ_{Sp} , que modela esta característica.

Finalmente se presentó el *AAS* $\Gamma_P = (\mathcal{L}_P, \mathcal{R}_P, \gg_P)$, permitiendo representar en forma explícita grados de necesidad asociados a las sentencias a nivel del lenguaje objeto, y definir la preferencia entre a-estructuras en base a estos valores posibilísticos. El enfoque adoptado para integrar agregación e incertidumbre posibilística se encuentra publicado en un artículo propio [LCS09a].

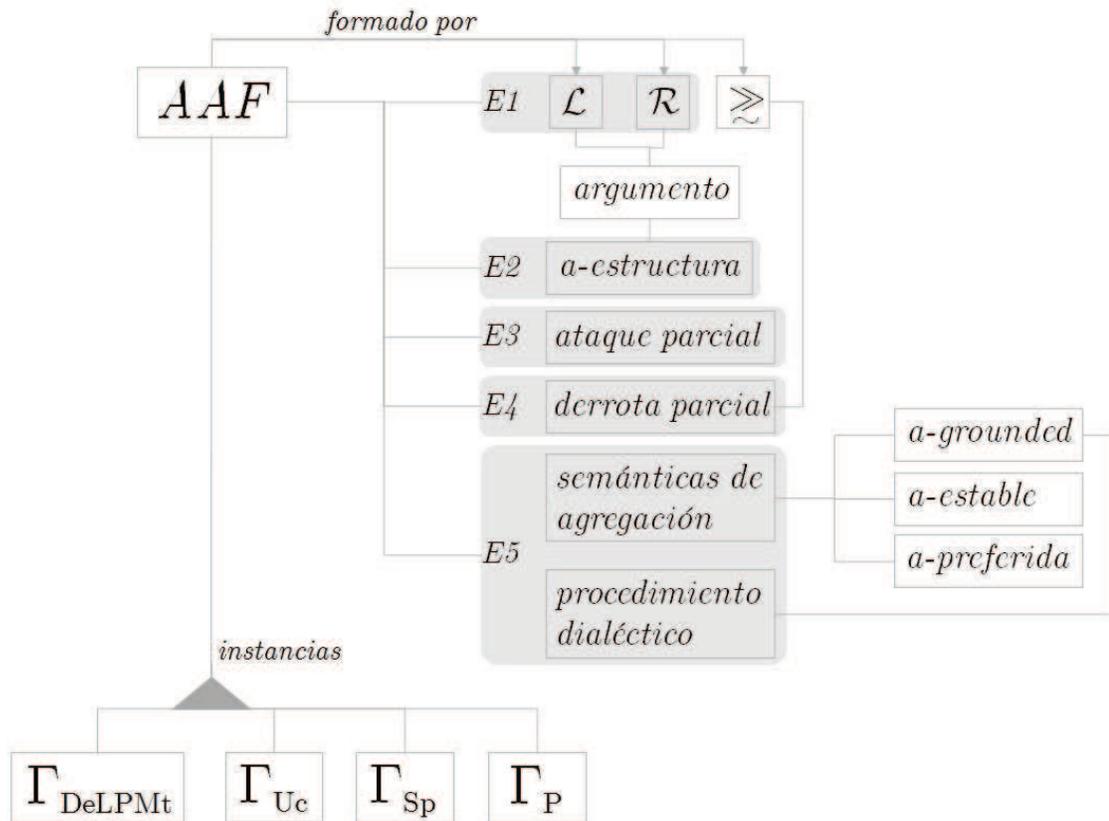


Figura 7.9: Red conceptual describiendo el AAF presentado

Es importante destacar que las distintas extensiones propuestas (AASs) son ortogonales, es decir, pueden combinarse según se requiera. Por ejemplo, puede definirse un sistema modelando incertidumbre posibilística y *undercutting* combinando Γ_P y Γ_{Uc} . Incluso podrían combinarse los cuatro AAS propuestos en este capítulo.

Capítulo 8

Trabajo Relacionado

En este capítulo se describen los dos principales enfoques para agregación existentes, concretamente las formalizaciones de Prakken [Pra05] y de Verheij [Ver95, Ver96]. Asimismo, estas dos formalizaciones se analizan comparativamente con nuestra formalización de agregación, destacando las características salientes de nuestro enfoque.

8.1. Enfoque de Prakken

En [Pra05], Prakken enuncia tres principios deseables que “todo tratamiento formal de agregación debería satisfacer”, y en base a dichos principios desarrolla una formalización para modelar agregación. Adicionalmente, Prakken estudia en [Pra05] la importancia de la agregación para modelar razonamiento evidencial y en [BCP06] aplica su formalización de agregación para el razonamiento acerca de acciones (*practical reasoning*).

8.1.1. Principios de la Agregación

El primer principio establece que “*las agregaciones son en ocasiones más débiles que sus elementos*” debido a la posibilidad de que las razones que se agregan no sean independientes. Prakken presenta el siguiente ejemplo para ilustrar el principio. Considérese las siguientes dos razones para no salir a correr: porque hace calor, y porque está lloviendo. Para un corredor en particular, la combinación de calor y lluvia puede resultar menos desagradable que el calor o la lluvia por separado, de manera que en este caso la agregación

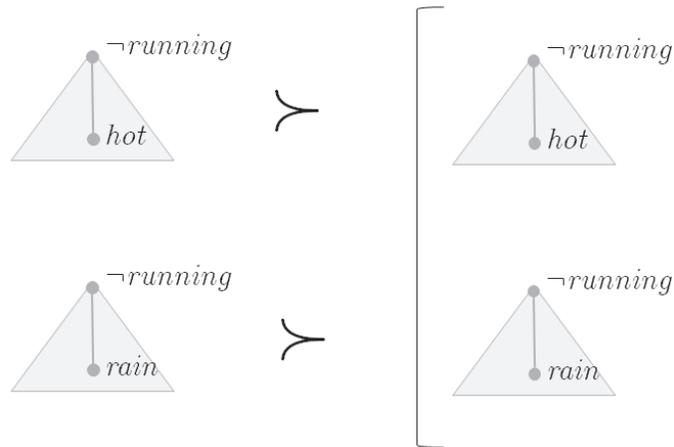


Figura 8.1: Situación que ilustra el primer principio de Prakken

constituye una razón más débil para no salir a correr que las razones individuales (ver Fig. 8.1).

El segundo principio establece que “*si aplica una agregación ‘mayor’, entonces todas sus versiones ‘menores’ (narrowings) se consideran inaplicables*”, y solo tiene sentido si se modela el primer principio. Intuitivamente, si se considerara una agregación menor para una conclusión dada x , a pesar de existir una agregación mayor para x , si esta agregación menor fuera más fuerte (lo que solo puede ocurrir si se modela el primer principio), se corre el riesgo de aceptar x por haber ignorado ciertas razones, es decir, por haber considerado una situación incompleta. Retomando el ejemplo del corredor, supongamos que ahora se tiene en cuenta una razón para salir a correr, por ejemplo, dado que desde hace dos días no se hace ejercicio. Supongamos además que si bien esta razón no llega a superar a las razones individuales *hot* y *rain* para \neg *running*, es más fuerte que la agregación de estas dos últimas (y esto solo puede ocurrir con razones como *hot* y *rain*, que al agregarse conforman una razón más débil). Esta situación se ilustra en la Fig. 8.2. Luego, si se consideraran las razones individuales para \neg *running*, estas prevalecerán sobre la razón para *running*, que quedará derrotada. Este resultado es intuitivamente incorrecto, ya que se está aceptando \neg *running* basado en las razones individuales, que consideran la información por separado, en lugar de resolver el conflicto de acuerdo a la agregación, que captura la interacción entre las razones individuales. Por lo tanto, la existencia de una agregación mayor debe inhabilitar a las menores cuando estas pueden ser más fuertes.

El tercer principio establece que “*las razones o argumentos derrotados (flawed) no deberían formar parte de la agregación*”. Este principio induce un proceso de argumentación

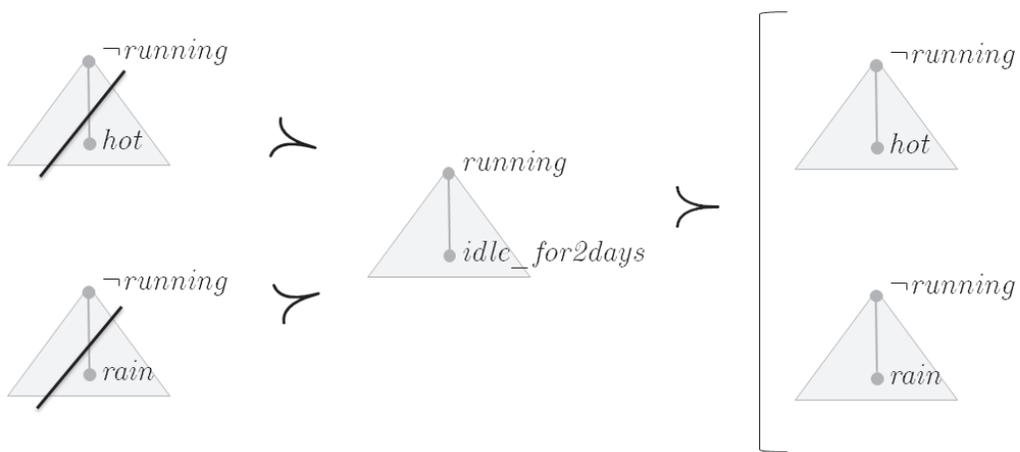


Figura 8.2: Situación que ilustra el segundo principio de Prakken

en dos etapas. Primero todas las razones individuales para una conclusión dada son testeadas para ver si pueden entrar en la agregación, y luego aquellas razones que pasan el test son agregadas y comparadas con todas las razones que efectivamente se agregan para la conclusión contraria.

8.1.2. Formalización de Prakken

La formalización de Prakken adapta la forma de modelar agregación de razones en Reason-Based Logic [Hag96] a un enfoque basado en argumentación. Esta formalización se basa en una combinación de dos sistemas de argumentación ampliamente reconocidos: el framework argumentativo abstracto de Dung [Dun95] instanciado con el enfoque de Pollock para la estructura de los argumentos [Pol94]. Prakken define argumentos agregados (o simplemente agregaciones) como una clase especial de derivación rebatible involucrando etiquetas. Dada una base de conocimiento (KB), consistiendo de un conjunto de reglas rebatibles, una derivación rebatible puede estructurarse como se describe a continuación. Las reglas rebatibles en la base de conocimiento solo pueden aplicarse a premisas no etiquetadas, y la conclusión asociada a una aplicación se etiqueta con las premisas de la aplicación. Además existe una regla de inferencia especial para agregación, que a partir de cualquier conjunto de versiones etiquetadas de una fórmula dada produce una versión no etiquetada. Luego, esta última puede a su vez emplearse como premisa de otra regla de inferencia rebatible en la base de conocimiento, y así siguiendo. Considere la siguiente KB:

$$\begin{array}{llll} r_1 : b \Rightarrow a & r_3 : c \Rightarrow b & c & f \\ r_2 : d \Rightarrow a & r_4 : f \Rightarrow \sim b & d & \end{array}$$

La Fig. 8.3 muestra dos derivaciones para a . La de la izquierda representa la agregación de dos razones para a , mientras que la otra representa la agregación de una única razón para a .

$$\begin{array}{c} \frac{c \quad r_3}{b^{(c \quad r_3)}} \\ \frac{b \quad r_1}{a^{(b \quad r_1)}} \quad \frac{d \quad r_2}{a^{(d \quad r_2)}} \\ \hline a \end{array} \qquad \begin{array}{c} \frac{c \quad r_3}{b^{(c \quad r_3)}} \\ \frac{b \quad r_1}{a^{(b \quad r_1)}} \\ \hline a \end{array}$$

Figura 8.3: Agregaciones de Prakken como derivaciones etiquetadas

La relación de ataque entre argumentos se encuentra asociada a la siguiente noción particular de conflicto entre fórmulas: dos fórmulas están en conflicto solo si son complementarias y no se encuentran etiquetadas. Al requerir que las formulas se encuentren sin etiquetar para estar en conflicto hace que la relación de ataque se defina solo entre agregaciones (que poseen conclusiones no etiquetadas) y no entre razones individuales (cuyas conclusiones se encuentran etiquetadas).

El framework de Prakken se abstrae de la evaluación (y comparación) de agregaciones, necesaria para determinar si un ataque dado realmente tiene éxito, constituyendo una derrota. Luego de determinar qué ataques constituyen derrotas de acuerdo al criterio de evaluación adoptado, se construye un *grafo de derrotas* representando todas las agregaciones y la relación de derrota, que luego es analizado bajo la semántica de Dung seleccionada para determinar el status de las agregaciones.

A diferencia de la formalización presentada en esta tesis, el enfoque de Prakken considera no solo agregaciones maximales en el análisis de status, sino todas las posibles agregaciones para cada conclusión. En otras palabras, el grafo de derrotas contendrá un nodo por cada posible agregación derivable de la KB. Luego, para satisfacer el segundo principio de agregación, se introduce una ‘construcción’ especial denominada *undercutter para agregación* (*accrual undercutter*). Las instancias de esta construcción se representan también como nodos en el grafo de derrotas, y se emplean para establecer que cuando un conjunto dado de razones para la misma conclusión se agregan, ningún subconjunto propio de esas razones debería agregarse. La Fig. 8.4 muestra dos derivaciones A_1 y A_2 , representando dos razones individuales diferentes para a , una derivación B , representando una razón para b , y una derivación D representando una razón para $\sim b$, todas a partir de la KB presentada previamente. Además muestra una porción del grafo de derrotas generado por el sistema a partir de la KB (la porción relevante a la conclusión a). En el grafo existen tres agregaciones para a , aquella agregando solo A_1 , aquella agregando solo A_2 y la agregación involucrando A_1 y A_2 . Aparece también una instancia de un *undercutter* para agregación (representado gráficamente como una agregación con ‘*’ como conclusión) estableciendo que si A_1 y A_2 se agregan con éxito (*i.e.*, ni A_1 ni A_2 se encuentran derrotados) entonces todas las agregaciones menores (en esta caso aquella involucrando solo A_1 y la agregación involucrando solo A_2) quedan ‘inhabilitadas’ (*undercut*). Nótese las flechas entre el *undercutter* para agregación y las agregaciones menores. Finalmente existen agregaciones para b y $\sim b$, involucrando las derivaciones B y D , respectivamente, las

cuales se atacan mutuamente. Dado que la derivación A_1 contiene a b como conclusión intermedia, la agregación para $\sim b$ también ataca a cada agregación involucrando a A_1 . Más aún, asumamos que, de acuerdo al criterio de evaluación y comparación adoptado, la agregación para $\sim b$ es preferida a la agregación para b , y entonces los ataques del primero constituyen derrotas. Para este grafo, la extensión grounded de Dung coincide con la única extensión preferida, y es el conjunto conteniendo la agregación para $\sim b$ y la agregación para a involucrando solo A_2 . Nótese que esta última es la ‘mayor’ agregación para a que no contiene razones derrotadas.

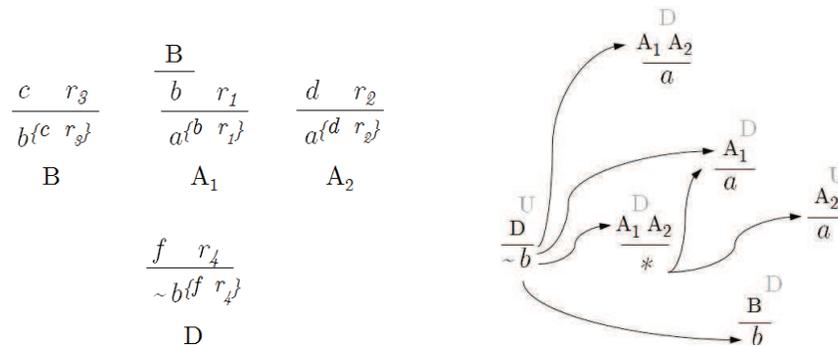


Figura 8.4: Grafo de derrotas de la formalización de Prakken

En la Fig. 8.4 se ilustró el caso donde una agregación para a sufre la derrota de uno de sus narrowings (el correspondiente a A_1 , involucrando b). La Fig. 8.5 muestra otras dos situaciones de acuerdo a la formalización de Prakken. Por un lado (Fig. 8.5a), dos razones para a que individualmente resultan derrotadas por una razón para $\sim a$, prevalecen sobre esta última al agregarse. Por otro lado (Fig. 8.5b), se combinan las dos situaciones anteriores: dos razones para a se agregan, prevaleciendo sobre $\sim a$, pero la agregación para a sufre la derrota de uno de sus narrowings (el correspondiente a A_1 , involucrando b), provocando que finalmente prevalezca la razón para $\sim a$. Finalmente, a modo de comparación, la Fig. 8.6 muestra las mismas tres situaciones modeladas por nuestra formalización.

A continuación se analiza como la formalización de Prakken satisface los tres principios de la agregación. En primer lugar, no se realiza ninguna suposición acerca de la fuerza relativa de las agregaciones, de manera que el primer principio es satisfecho. Además, el segundo principio es satisfecho gracias al *undercutter* para agregación. Finalmente, respecto al tercer principio nótese que la regla de inferencia para agregación permite la agregación de cualquier conjunto de argumentos para una conclusión dada, no sólo del maximal, de

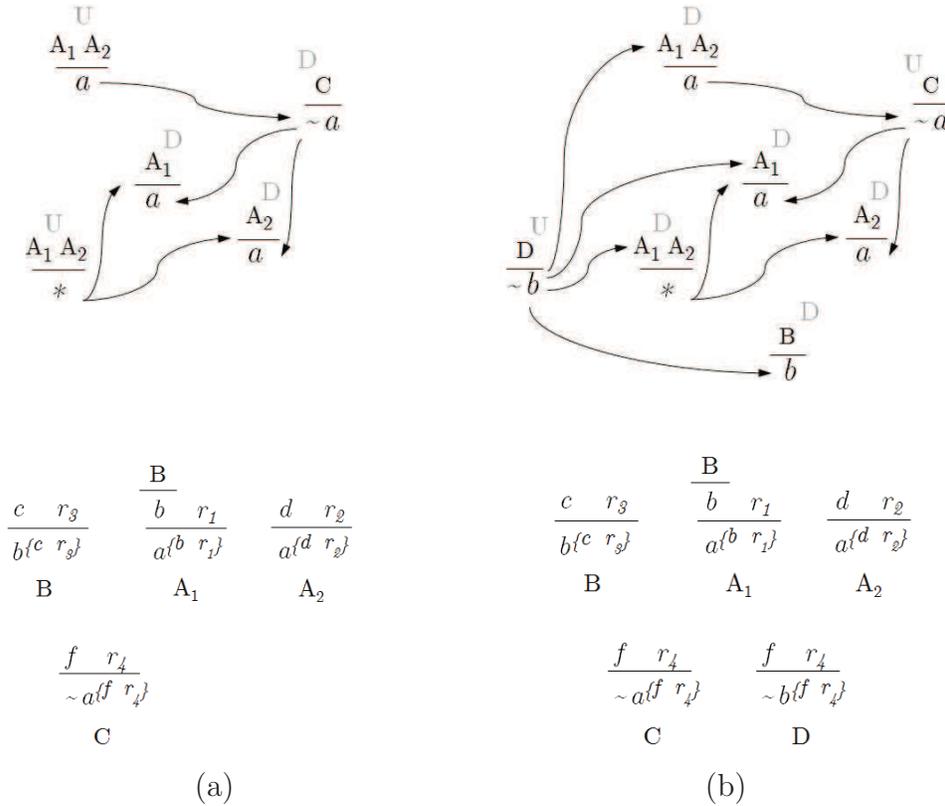


Figura 8.5: Otras grafos del enfoque de Prakken

manera que si alguna razón en el antecedente del *undercutter* para agregación es derrotada (*flawed*) el *undercutter* será derrotado en un subargumento y una de las agregaciones ‘menores’ que este pretendía inhabilitar (concretamente aquella no involucrando la razón derrotada), será reinstaurada.

8.2. Enfoque de Verheij (Cumula)

En el sistema Cumula de Verheij [Ver96, Ver95], los argumentos se representan mediante estructuras recursivas denotando árboles, similar al enfoque de Pollock, pero permitiendo la representación de coordinación (agregación) de razones. Los argumentos se construyen a partir de reglas y sentencias individuales definidas sobre un lenguaje no estructurado. Los conflictos entre argumentos son modelados a través de la noción de *derrota compuesta* (*compound defeat*), que en su forma más general permite establecer que un cierto conjunto de argumentos derrota a otro conjunto de argumentos. La relación

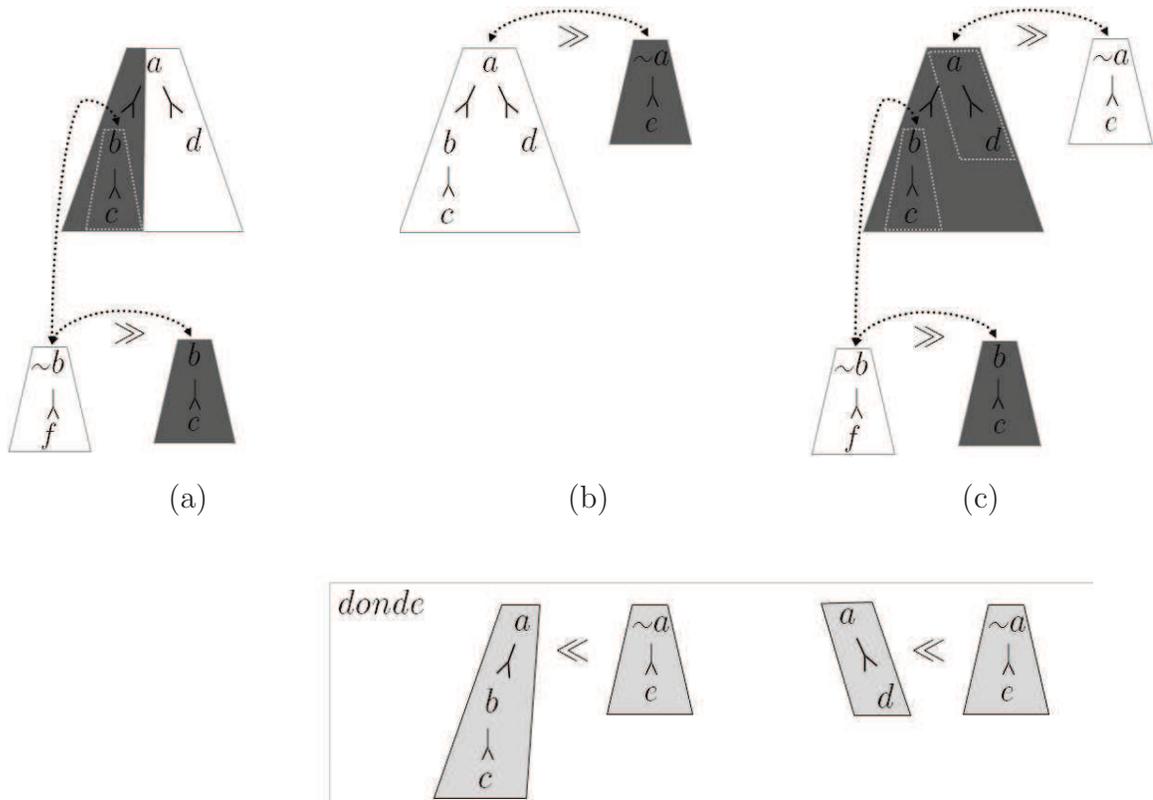


Figura 8.6: Situaciones en nuestra formalización

de derrota se especifica explícitamente a través de una *construcción de derrota combinada* con la siguiente sintaxis:

$$A_1, \dots, A_n[B_1, \dots, B_m],$$

representando que si los argumentos A_1, \dots, A_n (argumentos atacantes) están no derrotados, causan que los argumentos B_1, \dots, B_m (argumentos atacados) sean colectivamente derrotados (esto es, se tornarán todos derrotados, siempre y cuando ninguno de los argumentos en el grupo estuviera por otros motivos derrotado). Finalmente, una teoría argumentativa se define como una 3-tupla (*Lenguaje, Reglas, Derrotadores*), donde *Lenguaje* es un conjunto de sentencias representando el lenguaje, *Reglas* es un conjunto de reglas (rebatibles) definidas sobre *Lenguaje*, y que determina los argumentos construibles, y *Derrotadores* es un conjunto de derrotadores explícitamente especificando qué argumentos derrotan qué otros argumentos.

Cumula define el status de los argumentos a través de la noción de asignación de status. Dado un conjunto de argumentos $Args$, cerrado bajo la relación de narrowing y subargumento, una asignación de status para $Args$ es una partición $UArgs(DArgs)$ de

$Args$, donde $UArgs$ y $DArgs$ son los conjuntos de argumentos considerados no derrotados y derrotados, respectivamente, de acuerdo a la asignación de status. Una asignación de status $UArgs(DArgs)$ para un conjunto de argumentos $Args$ se determina considerando los derrotadores de la teoría argumentativa y las relaciones de subargumento y narrowing entre argumentos. Considérese un derrotador $A_1, \dots, A_n[B_1, \dots, B_m]$ de la teoría, tal que 1) $A_1, \dots, A_n \in UArgs$ (derrotador habilitado) y 2) no existe otro derrotador $A'_1, \dots, A'_n[B'_1, \dots, B'_m]$, con $A'_1, \dots, A'_n \in UArgs$ y $B'_1, \dots, B'_m \in DArgs$ (derrotador respetado) tal que algunos pero no todos los B_i son elementos de, o tienen un subargumento o *ampliación* (broadening) en $\{B'_1, \dots, B'_m\}$ (derrotador activo). Luego $B_1, \dots, B_m \in DArgs$ (derrota directa), y además todos los narrowings de cada B_i y todos los argumentos que tienen a B_i como subargumento, son elementos de $DArgs$ (derrota indirecta). Intuitivamente, la condición 2 establece que B_1, \dots, B_m son colectivamente derrotados solo si ningún subconjunto propio de los B_i es (directa o indirectamente) derrotado por otro derrotador.

Debe resaltarse que la derrota combinada no es equivalente a la agregación (coordinación) de argumentos. Considérese, por ejemplo, argumentos individuales $A_1 \rightarrow a, A_2 \rightarrow a, \dots, A_n \rightarrow a$, todos soportando la misma conclusión a , y considérese el derrotador compuesto (a izquierda) $A_1 \rightarrow a, A_2 \rightarrow a, \dots, A_n \rightarrow a[B]$. Si se derrotara al menos uno de los argumentos activadores $A_i \rightarrow a$ esto causará que el derrotador no resulte habilitado. Considérese ahora el argumento $A_1; A_2; \dots; A_n \rightarrow a$, representando la agregación de todos los argumentos individuales $A_i \rightarrow a, 1 \leq i \leq n$, y considérese el derrotador simple (no compuesto) $A_1; A_2; \dots; A_n \rightarrow a[B]$. Si se derrotara uno de los $A_i \rightarrow a$, esto no afectará la agregación (dado que derrotar un narrowing de una agregación no implica derrotar a la agregación), y luego el derrotador simple puede aún ser habilitado.

La Fig. 8.7 ilustra, para CumulA, las mismas tres situaciones analizadas en la Sec. 8.1 para la formalización de Prakken (y la nuestra). Para cada situación se describe el conjunto de derrotadores correspondiente y la asignación de status asociada, indicando en cada caso los argumentos derrotados y no derrotados mediante marcas 'D' y 'U', respectivamente. La Fig. 8.7a ilustra el caso donde dos razones individuales para a se agregan ($A_1 \rightarrow a$ y $A_2 \rightarrow a$), conformando el argumento agregado $A_1; A_2 \rightarrow a$, pero uno de los argumentos individuales ($A_1 \rightarrow a$) resulta derrotado por ($D \rightarrow no.b$). Dado que $D \rightarrow no.b$ no tiene derrotadores, luego se encuentra trivialmente no derrotado, causando la derrota *directa* de $B \rightarrow b$, e *indirecta* de $A_1 \rightarrow a$ y $A_1; A_2 \rightarrow a$ (que tienen a $B \rightarrow b$ como

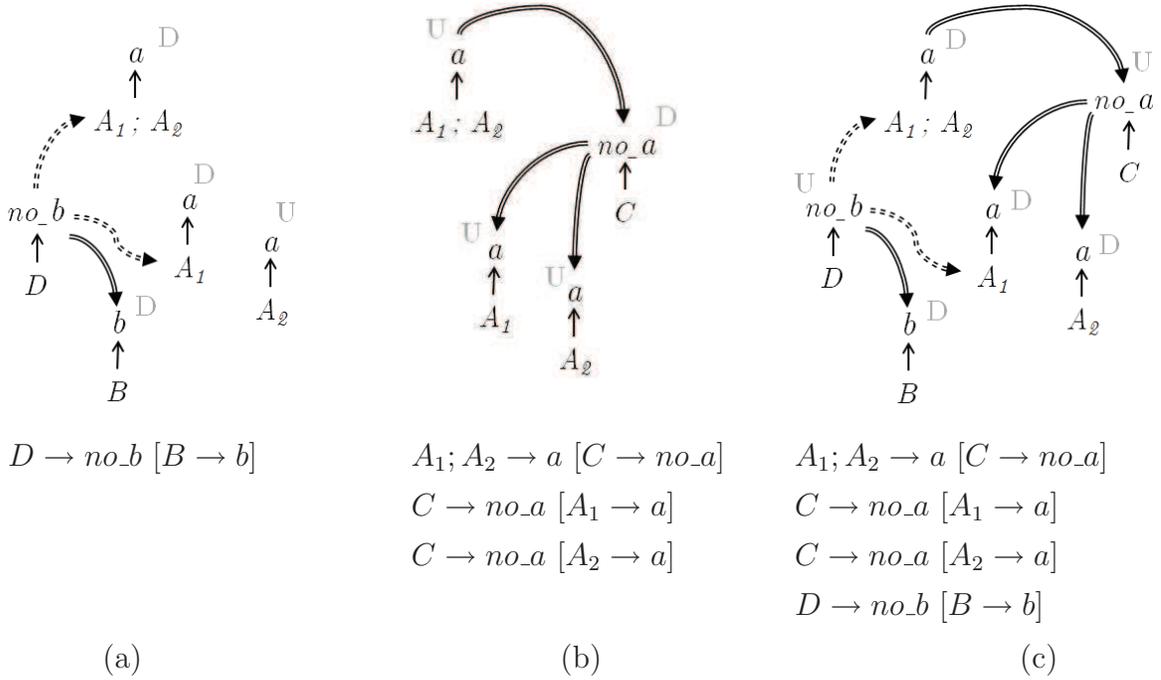


Figura 8.7: Asignaciones de Status en CumulA

subargumento), a través del derrotador $D \rightarrow no_b[B \rightarrow b]$ (habilitado y activo). El caso de la Fig. 8.7b modela la situación donde, a pesar de que cada uno de los argumentos individuales $A_1 \rightarrow a$ y $A_2 \rightarrow a$ se encuentran individualmente derrotados por $C \rightarrow no_a$, la agregación $A_1; A_2 \rightarrow a$ derrota a $C \rightarrow no_a$. Dado que $A_1; A_2 \rightarrow a$ no tiene derrotadores, luego se encuentra trivialmente no derrotado, causando la derrota de $C \rightarrow no_a$ a través del derrotador $A_1; A_2 \rightarrow a[C \rightarrow no_a]$. Dado que $C \rightarrow no_a$ resulta derrotado, luego los derrotadores involucrando a $C \rightarrow no_a$ en sus lados izquierdos no son habilitados, causando que $A_1 \rightarrow a$ y $A_2 \rightarrow a$ permanezcan no derrotados. Finalmente, el caso de la Fig. 8.7c resulta de combinar los dos anteriores, donde a pesar de que la agregación de A_1 y A_2 ($A_1; A_2 \rightarrow a$) en principio derrota a $C \rightarrow no_a$, especificado mediante el derrotador $A_1; A_2 \rightarrow a[C \rightarrow no_a]$, éste derrotador no se encuentra habilitado dado que la agregación en su lado izquierdo se encuentra derrotada por $D \rightarrow no_b$, mediante el derrotador habilitado y activo $D \rightarrow no_b[B \rightarrow b]$.

A continuación analizaremos los tres principios de la agregación en el contexto de CumulA. El primer principio no es satisfecho. La regla estableciendo la derrota (indirecta) de todos los narrowings de un argumento dado como consecuencia de la derrota (directa) del argumento claramente tiene sentido únicamente bajo la suposición de que los narrowings nunca pueden ser más fuertes que el argumento, lo que viola el primer principio. El segun-

do principio tampoco es satisfecho. Esto es ilustrado por el ejemplo en la Fig. 8.7b, donde a pesar de que la agregación para a involucrando A_1 y A_2 se encuentra no derrotada, sus narrowings también se encuentran no derrotados. Finalmente, CumulA satisface el tercer principio de la agregación.

8.3. Análisis de los principios de Prakken para nuestro enfoque

A continuación se analizan los tres principios de agregación propuestos por Prakken en el contexto de nuestra formalización. A primera vista pareciera que el primer principio no es satisfecho. Nótese que de acuerdo a la definición de la relación de preferencia ‘ \succcurlyeq ’ se requiere explícitamente que una a-estructura sea siempre al menos tan fuerte como sus narrowings. Esta condición refleja la siguiente suposición implícita asociada a nuestra noción de agregación: razones individuales para una misma conclusión se asumen por defecto independientes, y por lo tanto susceptibles de ser agregadas. Luego, situaciones como la propuesta por Prakken involucrando razones individuales no independientes, son consideradas excepciones a la suposición de independencia que habilita la agregación. Concretamente, en vez de considerar que $\neg\text{running} \prec \text{hot}$ y $\neg\text{running} \prec \text{rain}$ deberían agregarse para conformar una a-estructura más débil, creemos que, por no ser independientes, debería modelarse explícitamente una razón adicional (más débil) $\neg\text{running} \prec \text{hot}, \text{rain}$, combinando a las individuales, y que, en caso de estar activa, inhabilitará la agregación de las razones individuales. Nótese que la regla combinada puede verse como modelando una situación más específica que las individuales, y de esta forma, el mecanismo de inhabilitar la agregación de las razones individuales frente a la activación de la combinada se logra naturalmente mediante la regla de inferencia introducida en la Sec. 7.3 para nuestra formalización, que evita la agregación de razones menos específicas en presencia de más específicas. Finalmente, nuestra formalización permite modelar adecuadamente situaciones (excepcionales) donde combinar las razones produce una razón más débil, y de esta forma satisface el primer principio.

A modo de ejemplo, consideremos la KB \mathcal{K}_5 , modelando el ejemplo propuesto por Prakken en el contexto del AAS Γ_{Sp} presentado en la Sec. 7.3, y conteniendo los siguientes elementos:

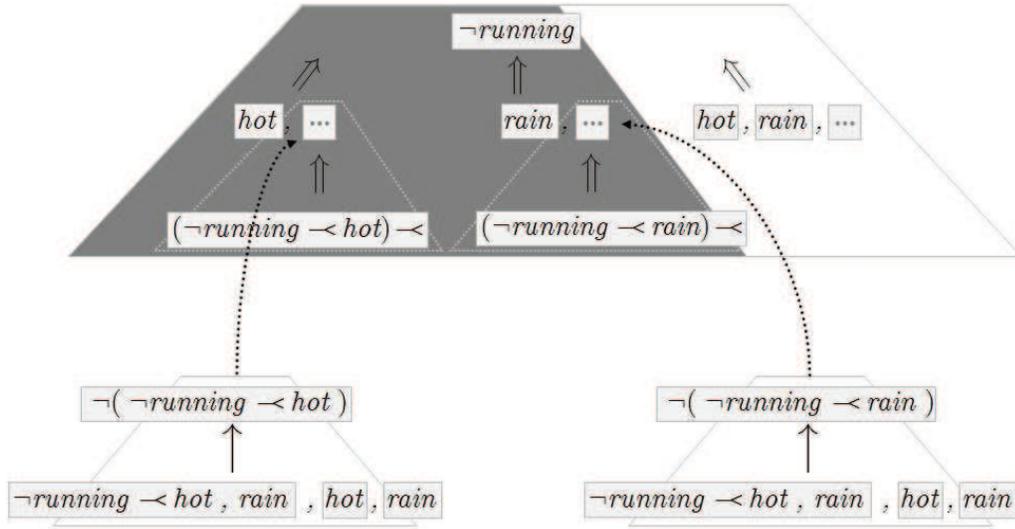


Figura 8.8: Modelando los principios 1 y 2 de la agregación

$$\begin{array}{ll}
 (\neg running \prec hot) \prec & hot \\
 (\neg running \prec rain) \prec & rain \\
 \neg running \prec hot, rain &
 \end{array}$$

Recordar que las razones individuales para $\neg running$ deben modelarse como presuposiciones para ser susceptibles a sufrir undercutting. La Fig. 8.8 muestra el grafo de ataques asociado a \mathcal{K}_5 , junto a la única asignación de status asociada. Dado que vale hot y $rain$, la razón combinada (más específica) se encuentra *activa*, provocando el undercutting de las individuales (menos específicas). Finalmente solo $\neg running \prec hot, rain$ forma parte del narrowing garantizado, quién determina efectivamente la fuerza de la agregación (en este caso más débil que cada una de las razones individuales por separado).

El segundo principio establece que la existencia de una agregación *mayor* (*i.e.*, involucrando más razones) debe inhabilitar a las *menores* (narrowings) cuando estas últimas pueden ser más fuertes que la agregación mayor. En términos del ejemplo propuesto por Prakken, la agregación de hot y $rain$ soportando $running$ debería inhabilitar las razones individuales (más fuertes). Como se explicó al analizar el primer principio, en nuestra formalización, una agregación que debilita se modela explícitamente mediante una regla adicional combinando las razones individuales en el cuerpo (en este caso, $running \prec hot, rain$). Luego, como ilustra el ejemplo de la Fig. 8.8, el segundo principio es satisfecho mediante la regla de inferencia introducida en la Sec. 7.3, que ante una razón más específica activa ($running \prec hot, rain$), inhabilita las razones menos es-

pecíficas (*running* \prec *hot* y *running* \prec *rain*). Puede verse que nuestra regla de inferencia actúa de manera similar al *undercutter para agregación* de la formalización de Prakken, cuyo propósito es justamente satisfacer el segundo principio.

Finalmente, el tercer principio también es satisfecho. La idea detrás de este principio es que al evaluar dos agregaciones soportando conclusiones contrarias para determinar cuál prevalece, las agregaciones en conflicto evaluadas no deberían contener razones derrotadas (es decir, las razones derrotadas deben descubrirse y dejarse de lado antes de la evaluación). En nuestra formalización, esta evaluación y comparación entre agregaciones en conflicto ocurre al aplicar la noción de degradación secuencial en la definición de asignación de status. Concretamente, la degradación secuencial se emplea para determinar los narrowings IN y OUT asociados a una a-estructura dada $[\Phi, h]$, a partir de los narrowings IN y OUT de sus atacantes. En primer lugar, las a-estructuras atacando $[\Phi, h]$ consideradas en la degradación secuencial no contienen razones derrotadas. Nótese que los OUT narrowings de los atacantes siempre están excluidos. En segundo lugar, para determinar en una degradación secuencial si un ataque dado constituye una derrota, la a-estructura que ataca se compara con la a-subestructura de desacuerdo asociada, y esta última es la “agregación soportando la conclusión contraria” mencionada al explicar la idea del principio. Finalmente, a pesar de que los derrotadores se aplican en cualquier momento de la degradación secuencial, incluso si la a-subestructura de desacuerdo asociada involucra razones que eventualmente serían derrotadas por otras a-estructuras del ataque combinado, podemos asegurar que estas aplicaciones son seguras. Si una a-estructura dada del ataque combinado constituye un derrotador, significa que es tanto o más fuerte que la a-subestructura de desacuerdo, y dado que en nuestra formalización la agregación no debilita, también será más fuerte que cualquier narrowing de dicha a-subestructura (en particular aquel no conteniendo razones derrotadas). De hecho, este último razonamiento se encuentra capturado formalmente mediante la teorema 1 de convergencia para degradaciones secuenciales, estableciendo que el orden en que se aplican los derrotadores no altera el resultado. Luego, nuestra formalización satisface el propósito del tercer principio.

8.4. Características salientes del enfoque propuesto: discusión

A continuación resumiremos los aspectos más salientes de nuestro enfoque, contrastándolo con los enfoques de Prakken y Verheij.

8.4.1. Aceptabilidad: comparación de respuestas

Nuestro enfoque propone tres semánticas de aceptabilidad alternativas, la *a-grounded*, la *a-estable* y la *a-preferida*, que pueden verse como extensiones naturales para contemplar agregación de las semánticas *grounded*, *estable* y *preferida*, propuestas por Dung en [Dun95] para el framework de argumentación abstracto *AF* (y presentadas en la Sec. 3.1).

El enfoque de Prakken es una instanciación del framework abstracto *AF* de Dung. Como ventaja, todas las semánticas definidas para el *AF* (entre ellas la *grounded*, *estable* y *preferida*) se aplican directamente a su formalización de agregación.

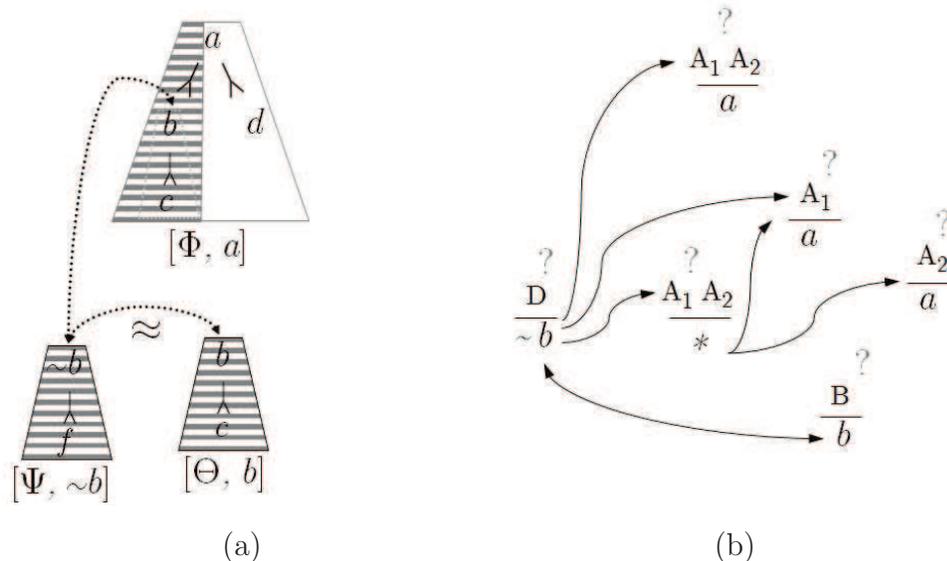


Figura 8.9: Respuesta de semántica a-grounded más completa

A pesar de que el enfoque de Prakken y el del presente trabajo coinciden en las respuestas para las situaciones estándares mostradas en la Sec. 8.1, nuestro enfoque brinda respuestas más completas frente a ciertas situaciones de indecisión. La Fig. 8.9a muestra

un grafo de ataques involucrando una situación de bloqueo, junto a la asignación de status a-grounded asociada. Claramente, el bloqueo entre $[\Psi, \sim b]$ y $[\Theta, b]$ provoca indecisión acerca del narrowing de $[\Phi, a]$ involucrando a $[\Theta, b]$. No obstante, el narrowing $\{(a \prec d)\}, a]$ de $[\Phi, a]$ se asigna completamente **IN**, y esto es intuitivamente correcto dado que dicho narrowing no resulta afectado por el bloqueo. Frente a la misma situación, la formalización de Prakken empleando la semántica grounded considera a todas las agregaciones como *undecided* (ver Fig. 8.9b), lo que constituye, intuitivamente, una respuesta más escéptica de lo necesario en este caso. Esta ventaja de nuestro enfoque también se manifiesta al considerar la semántica a-preferida (vs. la preferida en la formalización de Prakken), en casos de indecisión por ciclos de longitud impar.

Finalmente, CumulA de Verheij define una única semántica de aceptabilidad, similar a la semántica de admisibilidad de Dung. Respecto a la definición de esta semántica, puede verse que el efecto indirecto de los derrotadores es demasiado débil (o incompleto) en algunas situaciones. Un argumento dado B es indirectamente derrotado por un derrotador activo y habilitado $A_1, \dots, A_n[B_1, \dots, B_m]$ si B es un narrowing de alguno de los B_i o B tiene algún B_i como subargumento. Sin embargo, B debería también ser derrotado cuando tiene como subargumento a un narrowing de algún B_i . La asignación de status de la Fig. 8.10a ilustra este problema de la noción de derrota de CumulA. Recordar que las asignaciones de status en CumulA consideran conjuntos de argumentos cerrados bajo la relación de narrowings, y es por eso que aparecen tres argumentos soportando x . Luego, a pesar de que $u \rightarrow \text{no}_y$ derrota directamente a $z; t \rightarrow y$ (y por lo tanto derrota indirectamente a $z; t \rightarrow y \rightarrow x$), no derrota los argumentos $z \rightarrow y \rightarrow x$ y $t \rightarrow y \rightarrow x$, cada uno soportando y con un narrowing de $z; t \rightarrow y$.

8.4.2. Evaluación de agregaciones: determinando la relación de derrota

La formalización de Prakken se abstrae de la evaluación de las agregaciones, asumiendo que éstas son comparadas de alguna forma para determinar si un conflicto dado constituye una derrota. El enfoque CumulA de Verheij también se abstrae de la evaluación de las agregaciones, requiriendo que la relación de derrota sea explícitamente especificada, en lugar de definirla en términos de una noción de conflicto junto a una fase de evaluación. Nuestra formalización de agregación avanza en este sentido, proponiendo una extensión

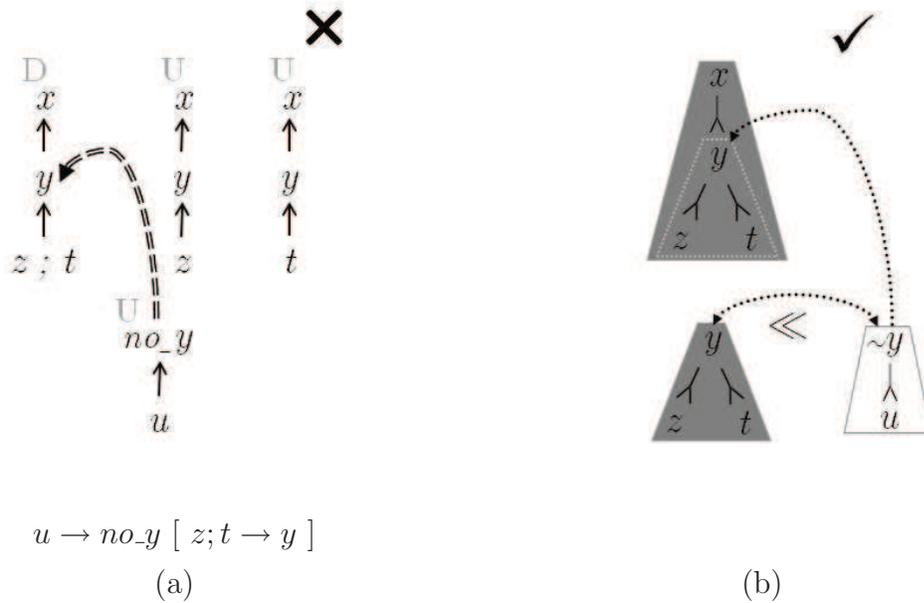


Figura 8.10: Efecto indirecto de derrotadores en Cumula demasiado débil

(AAS Γ_P presentado en la Sec. 7.4.2) que toma en cuenta información posibilística para definir explícitamente la evaluación y comparación de las agregaciones.

8.4.3. Complejidad del análisis argumentativo

En el análisis de aceptabilidad propuesto por nuestro enfoque solo se consideran agregaciones (a-estructuras) maximales. Por otro lado, en el enfoque de Prakken, no solo se consideran agregaciones maximales, sino además todas las ‘menores’. Luego, si existen n razones individuales diferentes para una conclusión dada x , el sistema de Prakken construirá una agregación soportando x por cada subconjunto no vacío de esas n razones, lo que implica un número de agregaciones para x exponencial en n (precisamente, $2^n - 1$ agregaciones). Por la misma razón, el número de *undercutters* para agregación (*accru- al undercutters*) también es exponencial en el número de razones individuales para la conclusión asociada. Finalmente, el número de derrotas (flechas) también crece considerablemente. Por un lado, cada *undercutter* para agregación involucrando k razones derrota a todas las agregaciones involucrando un subconjunto propio (no vacío) de esas k razones, que implica un número de derrotas exponencial en k (concretamente, $2^k - 2$ derrotas). Además, un conflicto con una razón individual implica una derrota contra cada agregación involucrando esa razón. La Fig. 8.11a modela, de acuerdo al enfoque de Prakken,

una situación donde existen tres razones individuales para a (A_1 , A_2 y A_3), donde una de ellas (involucrando una conclusión intermedia b) resulta derrotada por una razón para $\sim b$. La Fig. 8.11b muestra la misma situación de acuerdo a nuestra formalización, que da lugar a una representación y análisis más sucintos.

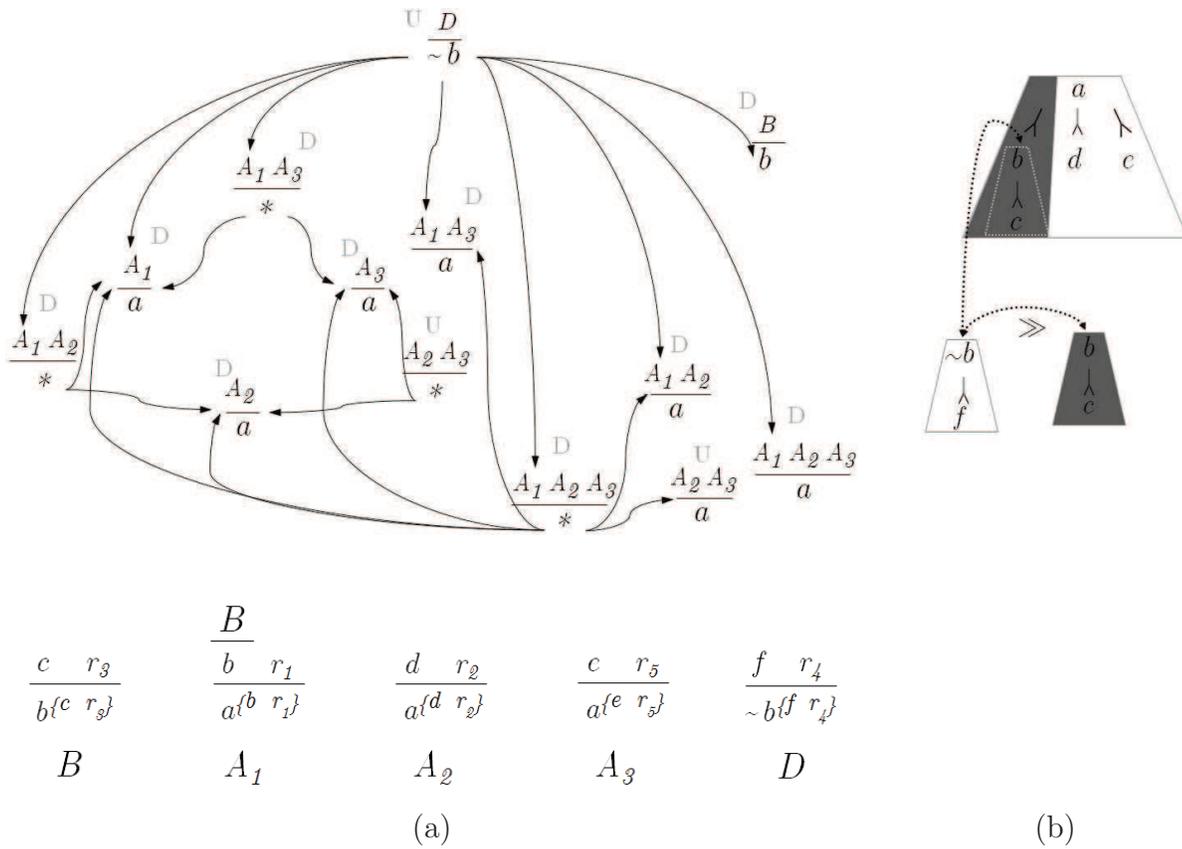


Figura 8.11: Complejidad del análisis argumentativo

Análogamente a lo que ocurre con el enfoque de Prakken, en CumulA todas las posibles agregaciones para cada conclusión (y no solo las maximales) necesitan ser consideradas. Recordar que una asignación de status se define como una partición de un conjunto de argumentos $Args$, que es cerrado bajo la relación de narrowing. Finalmente, como consecuencia de considerar todas las posibles agregaciones para cada conclusión, el número de derrotas (ya sean explícitas o implícitas) que tienen que tenerse en cuenta al analizar el status de argumentos también resulta considerable.

8.4.4. Explicación de respuestas

Una ventaja importante de los formalismos basados en argumentación es la idea de explicación de las respuestas obtenidas. La mayoría de los sistemas basados en argumentación generan estructuras o definen representaciones con el propósito de formalizar el análisis argumentativo que determina el status de los argumentos (los más populares son los grafos de derrotas y los árboles dialécticos). Dado que los humanos generalmente estamos bastante familiarizados con la noción de argumentación, esas representaciones construidas, reflejando de alguna manera el análisis argumentativo efectuado, pueden considerarse en sí mismas como explicaciones (o justificaciones), entendibles para el ser humano, de las respuestas obtenidas.

A pesar de que tanto el enfoque de Prakken como el nuestro exhiben la característica previamente mencionada, la complejidad del grafo de derrotas definido por Prakken limita su uso como una explicación humanamente legible de la respuesta obtenida. Nótese que los grafos de Prakken no solo consideran todas las agregaciones para cada conclusión (lo que incrementa el número de derrotas), sino que además incluye un número considerable de argumentos artificiales: los undercutters para agregación. En el caso de CumulA, no se define estructura (al estilo de un grafo o árbol dialéctico) al formalizar el status de argumentos. A pesar de que en este capítulo el análisis realizado por CumulA se representó gráficamente mediante grafos, dicha notación no resulta suficiente para representar derrotas compuestas.

8.4.5. Computación

Dado que la formalización de Prakken es una instancia del framework abstracto AF de Dung, cualquier procedimiento dialéctico de los definidos en [TDH09] para las semánticas de Dung puede ser usado. Los procedimientos de prueba dialécticos permiten determinar si un argumento dado A (nodo en el grafo de derrotas) es aceptado de acuerdo a una semántica dada a través de la construcción de un árbol dialéctico de prueba modelando una discusión acerca de A . Sin embargo, a pesar de que estos procedimientos de prueba dialécticos están pensados para ser aplicados a cualquier grafo de derrotas, en el caso del definido por la formalización de Prakken, dichos procedimientos deben ser usados de una forma un tanto ineficiente para obtener una respuesta útil, y esto resulta más evidente al adoptar una semántica escéptica (o de única extensión, como la grounded). De acuerdo

a la formalización de Prakken, todas las agregaciones para una conclusión dada a son construidas, y son todas consideradas como justificaciones potenciales alternativas para a , esto es, a lo sumo una de las agregaciones para a puede estar presente en una extensión de aceptabilidad dada (la ‘mayor’ agregación no conteniendo razones derrotadas (flawed) de acuerdo a la extensión). Por lo tanto, a pesar de que a lo sumo una de las agregaciones para a será aceptada (si se emplea una semántica escéptica), en el peor caso se tendrá que aplicar el procedimiento de prueba dialéctico para cada agregación soportando a para descubrir cuál es la agregación aceptada (si es que alguna).

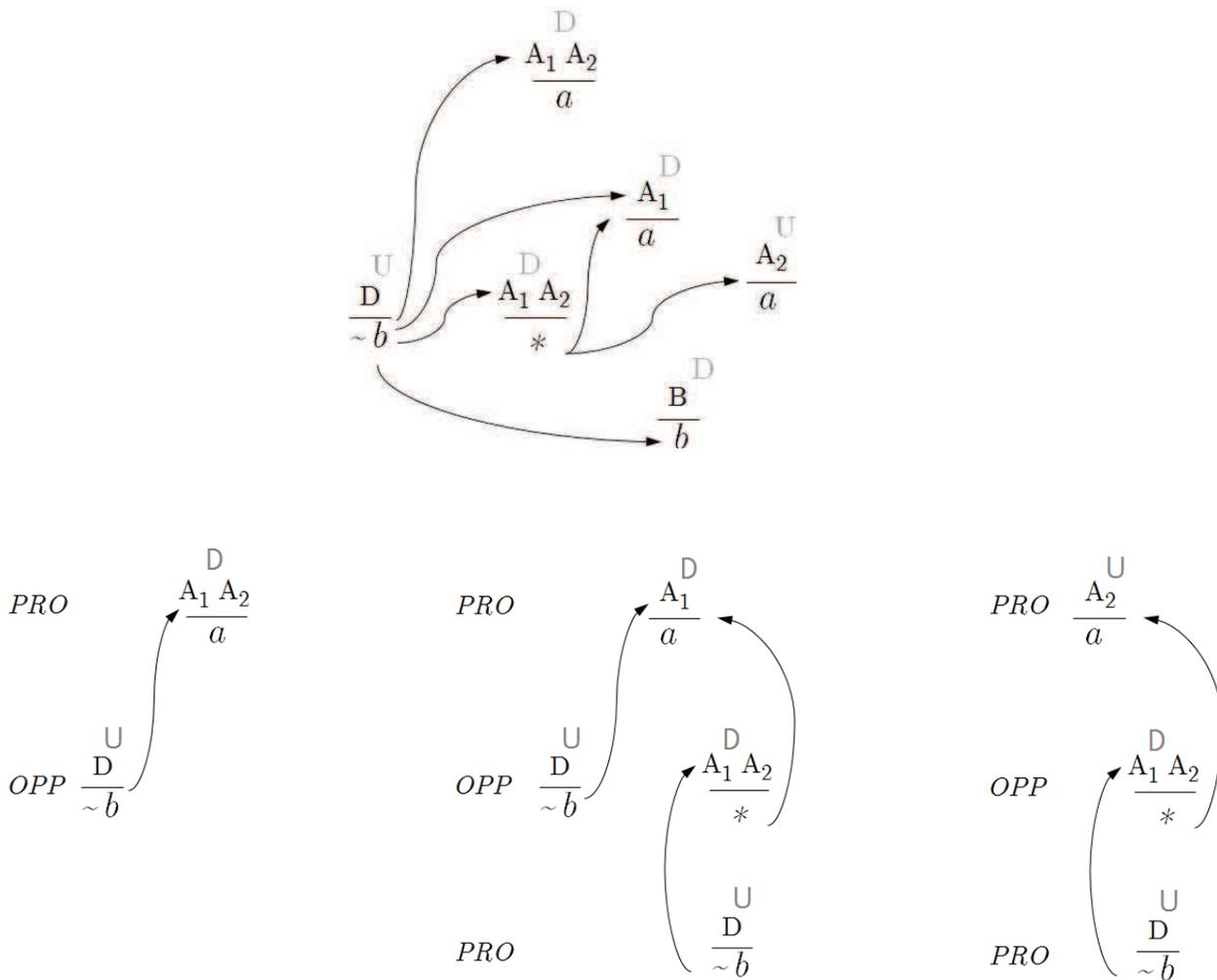


Figura 8.12: Procedimiento de prueba dialéctico *grounded* aplicado al enfoque de Prakken

Consideremos nuevamente uno de los grafos de derrota de la formalización de Prakken analizados en la Sec. 8.1.2 (repetido en la parte superior de la Fig. 8.12). Puede verse que para determinar la agregación finalmente aceptada de acuerdo a la semántica *grounded*,

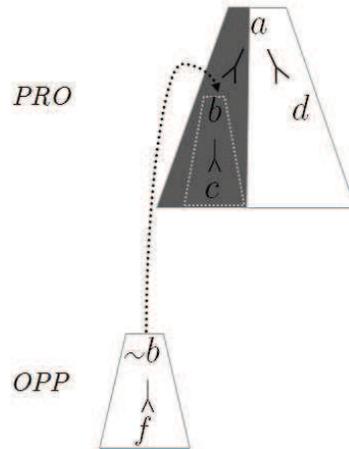


Figura 8.13: Árbol dialéctico agregado

por ejemplo, deben construirse en el peor de los casos tres árboles dialécticos, uno por cada agregación para a (parte inferior de la Fig. 8.12¹.)

En esta tesis se presentó un procedimiento de prueba dialéctico para la semántica a -grounded, como una extensión para contemplar agregación del procedimiento para la semántica grounded. A diferencia del enfoque de Prakken, solo se consideran agregaciones maximales en el análisis dialéctico, y por lo tanto solo un árbol debe ser construido para determinar la agregación garantizada (si es que alguna) para una conclusión dada a . A modo de comparación, la Fig. 8.13 muestra el árbol dialéctico agregado (*ADT*) construido por nuestro enfoque para la misma situación considerada para Prakken.

Finalmente, no se presentó ningún procedimiento de prueba asociado con el sistema CumulA.

¹Se empleó el procedimiento dialéctico para la semántica grounded presentado en la Sec. 3.3, y ampliamente aceptado en la literatura.

Capítulo 9

Conclusiones

La *argumentación* ha demostrado ser un paradigma poderoso para formalizar y automatizar el *razonamiento de sentido común*. En las últimas décadas varios formalismos surgieron basados en la idea de argumentación, encontrando aplicaciones en diversos dominios tales como razonamiento automatizado, toma de decisiones, disputas legales, negociación automatizada, etc. Sin embargo, la mayoría de estos formalismos no permiten modelar la *agregación de argumentos*, identificada inicialmente por Pollock, y cuya importancia ha sido reconocida en incremento en los últimos años. En esta tesis se presentó una formalización novedosa de la agregación de argumentos, involucrando una caracterización declarativa de esta noción, y una caracterización operacional asociada, abordando su computación. La formalización desarrollada presenta ventajas significativas respecto a los enfoques de agregación existentes, concretamente el sistema CumulA de Verheij [Ver96, Ver95] y la formalización de Prakken [Pra05].

Como se mostró en el capítulo 1, la agregación de argumentos constituye una característica natural de la argumentación en general, afectando el resultado final del proceso argumentativo. En la introducción se describió un modelo conceptual propuesto por Prakken [PV02] para la argumentación, dentro del cual pueden enmarcarse la mayoría de los sistemas de argumentación existentes. Dicho modelo identifica cinco elementos presentes al formalizar la argumentación: E1) *lenguaje lógico*, E2) definición de *argumento*, E3) definición de *conflicto* (o *ataque*) entre argumentos, E4) *evaluación* de argumentos y definición de *derrota*, y E5) definición de *aceptabilidad* de argumentos. Adicionalmente, para cada elemento se describieron distintas alternativas adoptadas por diferentes formalismos argumentativos existentes, y a partir de esta descripción se evidencia la falta de

atención hacia la noción de agregación de los enfoques existentes. Por último, el modelo conceptual de argumentación fue empleado para organizar la presentación de los conceptos introducidos en esta tesis, logrando de esta manera destacar cómo la formalización de agregación propuesta extiende los enfoques tradicionales en cada elemento del modelo.

En el capítulo 2 se presentaron el *lenguaje lógico*, y las nociones de argumento, conflicto y derrota (elementos E1-E4) del formalismo de argumentación DeLP (*Programación en Lógica Rebatible*), a ser empleados como base para los mismos cuatro elementos de la formalización de agregación desarrollada.

En el capítulo 3 se presentaron y analizaron comparativamente las tres *semánticas de aceptabilidad* propuestas por Dung en [Dun95] (*grounded*, *estable* y *preferida*), ampliamente adoptadas por las formalizaciones de argumentación existentes para definir la aceptabilidad de argumentos (elemento E5 – enfoque declarativo.) Se presentó además un *procedimiento de prueba* para la semántica *grounded* basado en el enfoque *dialéctico*, o de juegos argumentales (*argument games approach*), donde la aceptabilidad de un argumento se determina mediante la simulación de una discusión entre dos personas o jugadores ficticios (elemento E5 – enfoque procedural.) Una ventaja de este enfoque es que la propia representación formal del análisis dialéctico realizado constituye en sí misma una explicación intuitiva del resultado obtenido. Luego, versiones contemplando agregación de estas tres semánticas y del procedimiento de prueba dialéctico fueron introducidas en el capítulo 5.

A lo largo del capítulo 4 se definieron los elementos E1-E4 del marco conceptual para nuestro sistema con agregación. Adoptando como lenguaje lógico subyacente el de DeLP (E1), se introdujo la noción de *estructura agregada* o *a-estructura* (E2), denotando la agregación de diferentes argumentos DeLP soportando la misma conclusión. Adicionalmente se adoptó la noción de *narrowing*, inicialmente propuesta por [Ver95, Ver96], asociada a una a-estructura dada, consistiendo básicamente de una a-estructura representando un subconjunto de las razones presentes en la primera. Se introdujeron además las nociones de *ataque parcial* (E3) y *derrota parcial* (E4), como versiones para agregación de las nociones de ataque y derrota en DeLP. El adjetivo ‘parcial’ se debe a que, a diferencia de las nociones tradicionales de ataque y derrota, es posible que afecten solo a una parte (*narrowing*) de la a-estructura víctima. Finalmente se introdujeron las nociones de *ataque combinado* y *derrota combinada*, que constituyen un primer paso hacia la caracterización del análisis de aceptabilidad (E5). Estas nociones capturan que el ataque de varias a-

estructuras, consideradas conjuntamente, pueden provocar un ‘daño mayor’ sobre una a-estructura víctima, que si fueran consideradas individualmente.

Se presentaron las semánticas *a-grounded*, *a-estable* y *a-preferida* para la agregación (E5 – enfoque declarativo), basadas en las correspondientes semánticas de aceptabilidad propuestas por Dung, y se analizaron comparativamente a la luz de diferentes situaciones. La caracterización de estas semánticas contempla las interacciones (conflictos) entre a-estructuras y la noción de derrota combinada, sancionando para cada a-estructura tres narrowings complementarios representando su “status de aceptabilidad”: *narrowing garantizado* (denotando las razones definitivamente aceptadas), *narrowing rechazado* (denotando las razones definitivamente no aceptadas) y *narrowing discutible* (representando las razones que presentan indecisión). La distinción de estos tres narrowings significa una gran ventaja a la hora de interpretar las respuestas brindadas por el sistema. Finalmente se enunciaron y demostraron propiedades deseables de consistencia satisfechas por la caracterización propuesta.

En el capítulo 6 se presentó un procedimiento de prueba dialéctico para la semántica *a-grounded* (E5 – enfoque procedural), basado en el procedimiento dialéctico para la semántica *grounded* presentado en el capítulo 3. En este caso el procedimiento puede verse como la formalización de una discusión entre dos jugadores ficticios, considerando estructuras agregadas en lugar de argumentos. Finalmente, el procedimiento presentado hereda la ventaja del enfoque dialéctico antes mencionada, es decir, el análisis dialéctico realizado constituye en sí mismo una explicación intuitiva de la respuesta obtenida.

Con el propósito de simplificar la presentación del enfoque de agregación, se adoptó en una primera instancia (Caps. 4, 5 y 6) el lenguaje de representación y la noción de argumento de DeLP. En el capítulo 7 se presentó la noción de *Framework Argumentativo con Agregación*, definiendo un lenguaje de representación y noción de argumento más generales, en base a los cuales se aplica directamente toda la formalización de agregación desarrollada en los capítulos previos. Asimismo se definen distintos sistemas como instancias del framework, incorporando el modelamiento de diferentes características, como la regla deductiva *modus tollens*, el tipo de ataque *undercutting*, la discriminación de argumentos redundantes en una agregación y la incorporación explícita de *incertidumbre possibilística* a nivel del lenguaje objeto.

En el capítulo 8 se describieron los dos principales formalismos de agregación existentes, concretamente [Pra05, Ver95], analizándolos comparativamente con el enfoque

desarrollado en esta tesis. Como parte de la comparación se identifican características salientes de nuestro enfoque (respecto a los otros) en relación a las respuestas brindadas, la explicación de respuestas, la evaluación y comparación de agregaciones (para lo cual nuestra formalización permite el uso explícito de incertidumbre posibilística), la complejidad del análisis argumentativo, y la eficiencia de la computación.

Como trabajo futuro se planea desarrollar las siguientes líneas de investigación:

- Desarrollar una implementación de la formalización operacional propuesta, con especial énfasis en la eficiencia computacional. Particularmente deberán definirse procedimientos eficientes para la construcción de a-estructuras y el cómputo de degradaciones secuenciales.
- analizar la aplicación del formalismo de agregación desarrollado a distintos dominios concretos de inteligencia artificial. Particularmente se vislumbra la aplicación en el contexto de *toma de decisiones* (*decision making*) en agentes inteligentes, dado que resulta natural agregar las distintas razones en favor de una decisión dada, para enfrentarlas con la agregación de las razones en contra de dicha decisión o en favor de decisiones alternativas.
- Desarrollar procedimientos dialécticos asociados a las semánticas a-estable y a-preferida presentadas en la tesis, y formalizar versiones para agregación de otras semánticas de argumentación, como la *semi-estable* [Cam06b], la *ideal* [DMT06] y la *prudente* [CMDM05].

Apéndice A

Pruebas de Lemas y Teoremas

Lemma 1 Sea $[\Phi, h]$ una a -estructura y sea Σ un conjunto de a -estructuras atacando $[\Phi, h]$. Sea $\Upsilon = [\Phi_1, h], \dots, [\Phi_{m+1}, h]$ una degradación secuencial de $[\Phi, h]$ asociada con el ataque combinado de las a -estructuras en Σ , y sea $[\Psi_1, k_1], [\Psi_2, k_2], \dots, [\Psi_m, k_m]$ la secuencia de derrotadores asociados con Υ en la definición de degradación secuencial. Sea $[\Phi', h] \sqsubseteq [\Phi, h]$ tal que $[\Phi', h] \not\sqsubseteq [\Phi_{m+1}, h]$. Entonces $[\Psi_j, k_j]$ es un derrotador parcial de $[\Phi', h]$ para algún j , $1 \leq j \leq m$.

Prueba. Primero probaremos que $[\Psi_i, k_i]$ ataca a $[\Phi', h]$ para algún i , $1 \leq i \leq n$. Dado que $[\Phi', h] \sqsubseteq [\Phi, h]$ y $[\Phi', h] \not\sqsubseteq [\Phi_{m+1}, h]$ (hipótesis), luego existe $\langle \mathcal{A}, h \rangle \in \text{Args}([\Phi, h])$ tal que $\langle \mathcal{A}, h \rangle \in \text{Args}([\Phi', h])$ y $\langle \mathcal{A}, h \rangle \notin \text{Args}([\Phi_{m+1}, h])$. Dado que $[\Phi, h] = [\Phi_1, h] \supseteq \dots \supseteq [\Phi_{m+1}, h]$, entonces existe i , $1 \leq i \leq m$, tal que $\langle \mathcal{A}, h \rangle \in \text{Args}([\Phi_i, h])$ y $\langle \mathcal{A}, h \rangle \notin \text{Args}([\Phi_{i+1}, h])$ (esto es, $[\Phi_i, h]$ es el último narrowing de $[\Phi, h]$ en Υ conteniendo $\langle \mathcal{A}, h \rangle$). Considérese la a -estructura (derrotador) $[\Psi_i, k_i]$ ($[\Psi_i, k_i]$ es el derrotador responsable por la derrota definitiva de $\langle \mathcal{A}, h \rangle$ en Υ). Luego puede asegurarse que \bar{k}_i es una conclusión intermedia en $\langle \mathcal{A}, h \rangle$. Más aún, dado que $\langle \mathcal{A}, h \rangle \in \text{Args}([\Phi', h])$, \bar{k}_i también será una conclusión intermedia de $[\Phi', h]$, y por lo tanto, $[\Phi', h]$ será atacado por $[\Psi_i, k_i]$.

Considérese ahora el menor j , $1 \leq j \leq m$, tal que $[\Psi_j, k_j]$ ataca a $[\Phi', h]$. Probaremos que $[\Psi_j, k_j]$ es un derrotador propio para $[\Phi', h]$. Sea $[\Lambda, \bar{k}_j] [[\Lambda', \bar{k}_j]]$ la a -subestructura de desacuerdo asociada con el ataque de $[\Psi_j, k_j]$ contra $[\Phi_j, h] [[\Phi', h]]$. Probemos que $[\Lambda', \bar{k}_j] \sqsubseteq [\Lambda, \bar{k}_j]$.

Supongamos por el absurdo que $[\Lambda', \bar{k}_j] \not\sqsubseteq [\Lambda, \bar{k}_j]$. Entonces dado que $[\Lambda', \bar{k}_j]$ es a -subestructura de $[\Phi', h, \alpha']$ y $[\Lambda, \bar{k}_j]$ es a -subestructura de $[\Phi_j, h]$, vale que $[\Phi', h] \not\sqsubseteq [\Phi_j, h]$. Luego existe $\langle \mathcal{B}, h \rangle \in \text{Args}([\Phi, h])$ tal que $\langle \mathcal{B}, h \rangle \in \text{Args}([\Phi', h])$ y $\langle \mathcal{B}, h \rangle \notin \text{Args}([\Phi_j, h])$. Mediante un razonamiento análogo al efectuado para $\langle \mathcal{A}, h \rangle$ e i , podemos asegurar que existe r , $1 \leq r \leq j$, tal

que $\langle \mathcal{B}, h \rangle \in \text{Args}([\Phi_r, h])$ y $\langle \mathcal{B}, h \rangle \notin \text{Args}([\Phi_{r+1}, h])$. Más aún, vale que el derrotador $[\Psi_r, k_r]$, responsable por la derrota definitiva de $\langle \mathcal{B}, h \rangle$ en Υ , ataca a $[\Phi', h]$ (involucrando $\langle \mathcal{B}, h \rangle$). Se arribó a un absurdo, dado que $r < j$ y previamente se había considerado a j como el menor subíndice tal que $[\Psi_j, k_j]$ ataca a $[\Phi', h]$. Luego $[\Lambda', \overline{k_j}] \sqsubseteq [\Lambda, \overline{k_j}]$.

Dado que el ataque de $[\Psi_j, k_j]$ contra $[\Phi_j, h]$ con a-subestructura de desacuerdo $[\Lambda, \overline{k_j}]$ constituye una derrota defeat, entonces $[\Psi_j, k_j] \ggg [\Lambda, \overline{k_j}]$. Dado que la a-subestructura de desacuerdo $[\Lambda', \overline{k_j}]$ asociada con el ataque de $[\Psi_j, k_j]$ contra $[\Phi', h]$ es un narrowing de $[\Lambda, \overline{k_j}]$, y por condición en Def. 31 de relación de preferencia, vale que $[\Lambda, \overline{k_j}] \ggg [\Lambda', \overline{k_j}]$. Finalmente $[\Psi_j, k_j] \ggg [\Lambda', \overline{k_j}]$, y por lo tanto $[\Psi_j, k_j]$ es un derrotador de $[\Phi', h]$ \square

Teorema 1 (Convergencia). *Sea $[\Phi, h]$ una a-estructura y sea Σ un conjunto de a-estructuras atacando $[\Phi, h]$. Sean $\Upsilon = [\Phi_1, h], \dots, [\Phi_m, h]$ y $\Upsilon' = [\Phi'_1, h], \dots, [\Phi'_m, h]$ dos degradaciones secuenciales de $[\Phi, h]$ asociadas con el ataque combinado de las a-estructuras en Σ . Entonces $[\Phi_m, h] = [\Phi'_m, h]$.*

Prueba. Supongamos por el absurdo que $[\Phi_m, h] \neq [\Phi'_m, h]$. Luego, o bien $[\Phi_m, h] \not\sqsubseteq [\Phi'_m, h]$, o $[\Phi'_m, h] \not\sqsubseteq [\Phi_m, h]$, o ambas cosas. Supongamos, sin pérdida de generalidad, que $[\Phi'_m, h] \not\sqsubseteq [\Phi_m, h]$. Sea $[\Psi_1, k_1], [\Psi_2, k_2], \dots, [\Psi_m, k_m]$ la secuencia de derrotadores asociados con Υ en la definición de degradación secuencial. Luego, por lema 1, $[\Psi_j, k_j]$ es un derrotador parcial de $[\Phi'_m, h]$ para algún j , $1 \leq j \leq m$. Pero por definición de degradación secuencial, $[\Phi'_m, h]$ no tiene derrotadores en Σ , arribando a una contradicción. Luego $[\Phi_m, h] = [\Phi'_m, h]$ \square

Lemma 2 *Sea $[\Theta, q]$ una a-estructura. Sea $\Sigma \setminus [\Sigma']$ un conjunto de a-estructuras atacando $[\Theta, q]$, tal que para cada $[\Psi', k] \in \Sigma'$, existe $[\Psi, k] \in \Sigma$ tal que $[\Psi', k] \sqsubseteq [\Psi, k]$. Entonces $UNwg([\Theta, q], \Sigma) \sqsubseteq UNwg([\Theta, q], \Sigma')$.*

Prueba. Supongamos por el absurdo que $UNwg([\Theta, q], \Sigma) \not\sqsubseteq UNwg([\Theta, q], \Sigma')$. Sea $\Upsilon' = [\Theta'_1, q], \dots, [\Theta'_{m+1}, q]$ ($= UNwg([\Theta, q], \Sigma')$) una degradación secuencial de $[\Theta, q]$ asociada con el ataque combinado de las a-estructuras en Σ' , y sea $[\Psi'_1, k_1], [\Psi'_2, k_2], \dots, [\Psi'_m, k_m]$ la secuencia de derrotadores asociados con Υ' en la definición de degradación secuencial. Por definición de U-narrowing vale que $UNwg([\Theta, q], \Sigma) \sqsubseteq [\Theta, q]$, y por hipótesis del absurdo

$UNwg([\Theta, q], \Sigma) \not\sqsubseteq UNwg([\Theta, q], \Sigma')$. Luego, aplicando lema 1 vale que $[\Psi'_j, k_j]$ es un derrotador parcial de $UNwg([\Theta, q, \delta], \Sigma)$ para algún j , $1 \leq j \leq m$ (adoptando $[\Theta, q]$ como $[\Phi, h]$ y $UNwg([\Theta, q, \delta], \Sigma)$ como $[\Phi', h]$ en el lema 1).

Luego, por hipótesis, existe $[\Psi, k_j] \in \Sigma$ tal que $[\Psi'_j, k_j] \sqsubseteq [\Psi, k_j]$, y por condición en Def. 31 de relación de preferencia, vale que $[\Psi'_j, k_j] \lesssim [\Psi, k_j]$. Por lo tanto, $[\Psi, k_j]$ también es un derrotador parcial de $UNwg([\Theta, q, \delta], \Sigma)$, arribando a una contradicción, dado que de acuerdo a la definición de degradación secuencial de una a-estructura $[\Theta, q]$ asociada con el ataque de a-estructuras en Σ , el último elemento de la secuencia $(UNwg([\Theta, q], \Sigma))$ no tiene derrotadores en Σ . Luego $UNwg([\Theta, q], \Sigma) \sqsubseteq UNwg([\Theta, q], \Sigma')$ \square

Proposición 16 (monotonía de $F(\cdot)$) Sean \mathbb{S}_1 y \mathbb{S}_2 dos asignaciones de status arbitrarias para un programa dado \mathcal{P} . Si $\mathbb{S}_1 \preceq \mathbb{S}_2$ entonces $F(\mathbb{S}_1) \preceq F(\mathbb{S}_2)$.

Prueba. Asumamos que $\mathbb{S}_1 \preceq \mathbb{S}_2$. Sea $[\Phi, h]$ una a-estructura maximal de \mathcal{P} . Quiero probar que 1) $\text{IN}_{F(\mathbb{S}_1)}([\Phi, h]) \sqsubseteq \text{IN}_{F(\mathbb{S}_2)}([\Phi, h])$ y que 2) $\text{OUT}_{F(\mathbb{S}_1)}([\Phi, h]) \sqsubseteq \text{OUT}_{F(\mathbb{S}_2)}([\Phi, h])$. Probemos 1 en primer lugar. Por definición del operador $F(\cdot)$, $\text{IN}_{F(\mathbb{S}_1)}([\Phi, h]) = UNwg([\Phi, h], \Sigma)$, donde $\Sigma = \{\neg\text{OUT}_{\mathbb{S}_1}([\Psi, k]) \mid [\Psi, k] \text{ es a-estructura maximal que ataca parcialmente } [\Phi, h], \text{ y } \neg\text{OUT}_{\mathbb{S}_1}([\Psi, k]) \neq [\emptyset, \epsilon]\}$. Asimismo $\text{IN}_{F(\mathbb{S}_2)}([\Phi, h]) = UNwg([\Phi, h], \Sigma')$, donde $\Sigma' = \{\neg\text{OUT}_{\mathbb{S}_2}([\Psi, k]) \mid [\Psi, k] \text{ es a-estructura maximal que ataca parcialmente } [\Phi, h], \text{ y } \neg\text{OUT}_{\mathbb{S}_2}([\Psi, k]) \neq [\emptyset, \epsilon]\}$. Consideremos $[\Lambda', k] \in \Sigma'$. Entonces existe a-estructura maximal $[\Psi, k]$ de \mathcal{P} tal que $[\Lambda', k] = \neg\text{OUT}_{\mathbb{S}_2}([\Psi, k])$ y $\neg\text{OUT}_{\mathbb{S}_2}([\Psi, k]) \neq [\emptyset, \epsilon]$. Dado que $\mathbb{S}_1 \preceq \mathbb{S}_2$ (hipótesis), $\text{OUT}_{\mathbb{S}_1}([\Psi, k]) \sqsubseteq \text{OUT}_{\mathbb{S}_2}([\Psi, k])$ Por lo tanto, $\neg\text{OUT}_{\mathbb{S}_1}([\Psi, k]) \neq [\emptyset, \epsilon]$, y entonces existe $[\Lambda, k] \in \Sigma$ tal que $[\Lambda, k] = \neg\text{OUT}_{\mathbb{S}_1}([\Psi, k])$. Además $[\Lambda', k] \sqsubseteq [\Lambda, k]$. Aplicando el lemma 2, obtenemos que $UNwg([\Phi, h], \Sigma) \sqsubseteq UNwg([\Phi, h], \Sigma')$. Finalmente, $\text{IN}_{F(\mathbb{S}_1)}([\Phi, h]) \sqsubseteq \text{IN}_{F(\mathbb{S}_2)}([\Phi, h])$. Ahora probemos 2. Por definición del operador $F(\cdot)$, $\text{OUT}_{F(\mathbb{S}_1)}([\Phi, h]) = DNwg([\Phi, h], \Sigma)$, donde $\Sigma = \{\text{IN}_{\mathbb{S}_1}([\Psi, k]) \mid [\Psi, k] \text{ es a-estructura maximal que ataca parcialmente } [\Phi, h], \text{ e } \text{IN}_{\mathbb{S}_1}([\Psi, k]) \neq [\emptyset, \epsilon]\}$. Asimismo $\text{OUT}_{F(\mathbb{S}_2)}([\Phi, h]) = DNwg([\Phi, h], \Sigma')$, donde $\Sigma' = \{\text{IN}_{\mathbb{S}_2}([\Psi, k]) \mid [\Psi, k] \text{ es a-estructura maximal que ataca parcialmente } [\Phi, h], \text{ e } \text{IN}_{\mathbb{S}_2}([\Psi, k]) \neq [\emptyset, \epsilon]\}$. Análogamente a lo efectuado en 1, aplicamos el lemma 2, pero donde esta vez Σ' es el ataque más fuerte, obteniendo que $UNwg([\Phi, h], \Sigma') \sqsubseteq UNwg([\Phi, h], \Sigma)$. Por definición de $DNwg$ vale que $DNwg([\Phi, h], \Sigma) \sqsubseteq DNwg([\Phi, h], \Sigma')$, y entonces $\text{OUT}_{F(\mathbb{S}_1)}([\Phi, h]) \sqsubseteq \text{OUT}_{F(\mathbb{S}_2)}([\Phi, h])$ \square

Proposición 12. *Sea \mathcal{P} un programa DeLP. Sea \circlearrowleft una asignación de status arbitraria vacía para \mathcal{P} . Considere la siguiente secuencia de asignaciones de status arbitrarias para \mathcal{P} :*

$$F^0(\circlearrowleft), F^1(\circlearrowleft), F^2(\circlearrowleft), \dots$$

donde

- $F^0(\circlearrowleft) = \circlearrowleft$, y
- $F^{i+1}(\circlearrowleft) = F(F^i(\circlearrowleft))$.

Entonces vale que, para todo $i \in \mathbb{N} \cup \{0\}$:

- $F^i(\circlearrowleft) \preceq F^{i+1}(\circlearrowleft)$
- $F^i(\circlearrowleft) \preceq \mathbb{S}_f$, para cualquier punto fijo \mathbb{S}_f de $F(\cdot)$.

Prueba. Probaremos por inducción sobre i .

CB ($i = 0$): trivialmente vale que $F^0(\circlearrowleft) = \circlearrowleft \preceq F^1(\circlearrowleft)$ y $F^0(\circlearrowleft) = \circlearrowleft \preceq \mathbb{S}_f$ (por def. de \circlearrowleft y \preceq).
 CI ($i > 0$): asumamos que vale para $i - 1$, es decir $F^{i-1}(\circlearrowleft) \preceq F^i(\circlearrowleft)$ y $F^{i-1}(\circlearrowleft) \preceq \mathbb{S}_f$ (hipótesis inductiva — HI). Por HI y monotonía de $F(\cdot)$ vale que $F(F^{i-1}(\circlearrowleft)) \preceq F(F^i(\circlearrowleft))$, y por lo tanto $F^i(\circlearrowleft) \preceq F^{i+1}(\circlearrowleft)$. Nuevamente por HI y monotonía $F(\cdot)$ vale que $F(F^{i-1}(\circlearrowleft)) \preceq F(\mathbb{S}_f)$, y por ser \mathbb{S}_f un punto fijo de $F(\cdot)$, $F^i(\circlearrowleft) \preceq \mathbb{S}_f$ \square

Proposición 13. *Sea \mathcal{P} un programa, y sea $F^0(\circlearrowleft), F^1(\circlearrowleft), F^2(\circlearrowleft), \dots$ una secuencia de asignaciones de status arbitrarias para \mathcal{P} tal como fue definida en la proposición 12. Si existe solo un número finito de a-estructuras construibles a partir de \mathcal{P} , entonces existe n tal que $F^n(\circlearrowleft) = F^{n+1}(\circlearrowleft)$, y $F^n(\circlearrowleft)$ coincide con el menor punto fijo del operador $F(\cdot)$.*

Prueba. Por proposición 12, $F^i(\circlearrowleft) \preceq F^{i+1}(\circlearrowleft)$, para todo i . Supongamos por el absurdo que no existe $n \in \mathbb{N} \cup \{0\}$ tal que $F^n(\circlearrowleft) = F^{n+1}(\circlearrowleft)$. Entonces debe valer que $F^i(\circlearrowleft) \prec F^{i+1}(\circlearrowleft)$ para todo $i \in \mathbb{N}$. Consideremos el conjunto Γ de todos los IN – narrowings y OUT – narrowings de cada a-estructura maximal $[\Phi, h]$, de acuerdo a cada asignación de status $F^i(\circlearrowleft)$ de la secuencia. Formalmente $\Gamma = \{\text{IN}_{F^i(\circlearrowleft)}([\Phi, h]), \text{OUT}_{F^i(\circlearrowleft)}([\Phi, h]) \mid [\Phi, h] \text{ es a-estructura maximal de } \mathcal{P}, i \in \mathbb{N}\}$. Dado que $F^i(\circlearrowleft) \prec F^{i+1}(\circlearrowleft)$ para todo $i \in \mathbb{N}$, podemos asegurar que para cada asignación de status $F^i(\circlearrowleft)$, $\text{IN}_{F^i(\circlearrowleft)}([\Phi, h]) \sqsubset \text{IN}_{F^{i+1}(\circlearrowleft)}([\Phi, h])$ o $\text{OUT}_{F^i(\circlearrowleft)}([\Phi, h]) \sqsubset \text{OUT}_{F^{i+1}(\circlearrowleft)}([\Phi, h])$ para al menos una a-estructura maximal $[\Phi, h]$. En consecuencia, cada conjunto $F^i(\circlearrowleft)$ contribuye

con al menos un elemento nuevo al conjunto Γ , respecto a los contribuidos por las asignaciones de status previas ($F^0(\emptyset)$ a $F^{i-1}(\emptyset)$). Consecuencia de esto último, y por ser la secuencia de $F^i(\emptyset)$ infinita, Γ será un conjunto infinito de a-estructuras. Pero se asumió como hipótesis que la cantidad de a-estructuras era finita. Absurdo, que provino de suponer que no existe $n \in \mathbb{N} \cup \{0\}$ tal que $F^n(\emptyset) = F^{n+1}(\emptyset)$. Además, $F^n(\emptyset)$ es el menor punto fijo de $F(\cdot)$, ya que de suponer que existe un punto fijo \mathbb{S} tal que $\mathbb{S} \prec F^n(\emptyset)$, estaríamos contradiciendo la propiedad 12 previamente demostrada, que establece que $F^i(\emptyset) \preceq \mathbb{S}_f$, para cualquier punto fijo \mathbb{S}_f de $F(\cdot)$, arribando a un absurdo \square

Lema 1 *Sea \mathbb{S} una asignación de status para un programa dado \mathcal{P} , y sean $[\Phi, h]$ y $[\Phi', \bar{h}]$ dos a-estructuras maximales de \mathcal{P} . Entonces vale que si $\text{IN}_{\mathbb{S}}([\Phi, h]) \neq [\emptyset, \epsilon]$ entonces $\text{OUT}_{\mathbb{S}}([\Phi', \bar{h}]) = [\Phi', \bar{h}]$.*

Prueba. Supongamos por el absurdo que $\text{IN}_{\mathbb{S}}([\Phi, h]) \neq [\emptyset, \epsilon]$ y $\text{OUT}_{\mathbb{S}}([\Phi', \bar{h}]) \neq [\Phi', \bar{h}]$. Consideremos $\text{IN}_{\mathbb{S}}([\Phi, h]) = \text{UNwg}([\Phi, h], \Sigma)$, donde $\Sigma = \{\neg\text{OUT}_{\mathbb{S}}([\Psi, k]) \mid [\Psi, k] \text{ es a-estructura maximal que ataca parcialmente } [\Phi, h], \text{ y } \neg\text{OUT}_{\mathbb{S}}([\Psi, k]) \neq [\emptyset, \epsilon]\}$. Dado que $[\Phi', \bar{h}]$ ataca a $[\Phi, h]$ y $\neg\text{OUT}_{\mathbb{S}}([\Phi', \bar{h}]) \neq [\emptyset, \epsilon]$ (hipótesis), entonces $\neg\text{OUT}_{\mathbb{S}}([\Phi', \bar{h}]) \in \Sigma$. Además, siendo $\text{UNwg}([\Phi, h], \Sigma) (= \text{IN}_{\mathbb{S}}([\Phi, h])) \neq [\emptyset, \epsilon]$ (hipótesis), entonces $\neg\text{OUT}_{\mathbb{S}}([\Phi', \bar{h}])$ ataca parcialmente $\text{UNwg}([\Phi, h], \Sigma)$, y por definición de degradación secuencial $\neg\text{OUT}_{\mathbb{S}}([\Phi', \bar{h}]) \ll \text{UNwg}([\Phi, h], \Sigma) = \text{IN}_{\mathbb{S}}([\Phi, h])$ (si no fuera así, $\neg\text{OUT}_{\mathbb{S}}([\Phi', \bar{h}])$ debería haber sido aplicado en la degradación secuencial, y $\text{UNwg}([\Phi, h], \Sigma)$ sería $[\emptyset, \epsilon]$). Consideremos ahora $\neg\text{OUT}_{\mathbb{S}}([\Phi', \bar{h}]) = \text{UNwg}([\Phi', \bar{h}], \Sigma')$, donde $\Sigma' = \{\text{IN}_{\mathbb{S}}([\Psi', k]) \mid [\Psi', k] \text{ es a-estructura maximal que ataca parcialmente } [\Phi', \bar{h}], \text{ e } \text{IN}_{\mathbb{S}}([\Psi', k]) \neq [\emptyset, \epsilon]\}$. Dado que $[\Phi, h]$ ataca a $[\Phi', \bar{h}]$ e $\text{IN}_{\mathbb{S}}([\Phi, h]) \neq [\emptyset, \epsilon]$ (hipótesis), entonces $\text{IN}_{\mathbb{S}}([\Phi, h]) \in \Sigma'$. Además, siendo $\text{UNwg}([\Phi', \bar{h}], \Sigma') (= \neg\text{OUT}_{\mathbb{S}}([\Phi', \bar{h}])) \neq [\emptyset, \epsilon]$ (hipótesis), entonces $\text{IN}_{\mathbb{S}}([\Phi, h])$ ataca parcialmente $\text{UNwg}([\Phi', \bar{h}], \Sigma')$, y por definición de degradación secuencial $\text{IN}_{\mathbb{S}}([\Phi, h]) \ll \text{UNwg}([\Phi', \bar{h}], \Sigma') = \neg\text{OUT}_{\mathbb{S}}([\Phi', \bar{h}])$ (si no fuera así, $\text{IN}_{\mathbb{S}}([\Phi, h])$ debería haber sido aplicado en la degradación secuencial, y $\text{UNwg}([\Phi', \bar{h}], \Sigma')$ sería $[\emptyset, \epsilon]$). Finalmente arribamos a un absurdo, y luego $\text{IN}_{\mathbb{S}}([\Phi, h]) \neq [\emptyset, \epsilon]$ implica $\text{OUT}_{\mathbb{S}}([\Phi', \bar{h}]) = [\Phi', \bar{h}]$. \square

Lema 2 (consistencia de IN-narrowings a nivel de conclusión final) *Sea \mathbb{S} una asignación de status para un programa dado \mathcal{P} . Podemos asegurar que no existen a-estructuras maximales $[\Phi, h]$ y $[\Phi', \bar{h}]$ tal que $\text{IN}_{\mathbb{S}}([\Phi, h]) \neq [\emptyset, \epsilon]$ e $\text{IN}_{\mathbb{S}}([\Phi', \bar{h}]) \neq [\emptyset, \epsilon]$.*

Prueba. Asumiendo por el absurdo que no vale y empleando el lema 1 se arriba directamente a una contradicción \square

Lema 3 *Sea \mathbb{S} una asignación de status para un programa dado \mathcal{P} . Sean $[\Phi, h]$ una a-estructura maximal de \mathcal{P} y sea $[\Theta, q]$ una a-subestructura de $[\Phi, h]$. Sea $[\Theta', q] (\sqsubseteq [\Theta, q])$ la a-subestructura de $\text{IN}_{\mathbb{S}}([\Phi, h])$. Entonces $[\Theta', q] \sqsubseteq \text{IN}_{\mathbb{S}}([\Theta, q])$.*

Prueba. Supongamos por el absurdo que $[\Theta', q] \not\sqsubseteq \text{IN}_{\mathbb{S}}([\Theta, q])$. Por definición de asignación de status $\text{IN}_{\mathbb{S}}([\Theta, q]) = \text{UNwg}([\Theta, q], \Sigma)$, donde $\Sigma = \{\neg\text{OUT}_{\mathbb{S}}([\Psi, k]) \mid [\Psi, k] \text{ es a-estructura maximal que ataca parcialmente } [\Theta, q], \text{ y } \neg\text{OUT}_{\mathbb{S}}([\Psi, k]) \neq [\emptyset, \epsilon]\}$.

Sea $\Upsilon = [\Theta_1, q], \dots, [\Theta_{m+1}, q] (= \text{IN}_{\mathbb{S}}([\Theta, q]))$ una degradación secuencial de $[\Theta, q]$ asociada con el ataque combinado de las a-estructuras en Σ , y sea $[\Psi_1, k_1] \dots, [\Psi_m, k_m]$ la secuencia de derrotadores asociada a Υ por la definición de degradación secuencial. Dado que $[\Theta', q] \sqsubseteq [\Theta, q]$ pero $[\Theta', q] \not\sqsubseteq \text{IN}_{\mathbb{S}}([\Theta, q])$ (hipótesis), por lema 1 podemos afirmar que $[\Psi_j, k_j]$ es un derrotador parcial de $[\Theta', q]$ para algún j , $1 \leq j \leq m$ (tomando $[\Theta, q]$ como $[\Phi, h]$ en el lema 1 y $[\Theta', q]$ como $[\Phi', h]$ en el lema 1). Además, al ser $[\Theta', q]$ a-subestructura completa de $\text{IN}_{\mathbb{S}}([\Phi, h])$ (hipótesis), $[\Psi_j, k_j]$ también es un derrotador para $\text{IN}_{\mathbb{S}}([\Phi, h])$. Finalmente, dado que $[\Psi_j, k_j]$ un elemento de Σ , $[\Psi_j, k_j] = \neg\text{OUT}_{\mathbb{S}}([\Psi'_j, k_j])$ donde $[\Psi'_j, k_j]$ es una a-estructura maximal atacando parcialmente $[\Phi, h]$. De esta forma arribamos a un absurdo, dado que el IN-narrowing de una a-estructura maximal dada (en este caso $\text{IN}_{\mathbb{S}}([\Phi, h])$) no puede tener derrotadores entre los $\neg\text{OUT}$ -narrowings de las a-estructuras maximales que la atacan (por definición de asignación de status y de degradación secuencial) \square

Teorema 2 (consistencia dos a dos de IN-narrowings). *Sea \mathbb{S} una asignación de status para un programa dado \mathcal{P} . Sean $[\Phi_1, h_1]$ y $[\Phi_2, h_2]$ a-estructuras maximales de \mathcal{P} . Entonces podemos asegurar que $\text{IN}_{\mathbb{S}}([\Phi_1, h_1])$ e $\text{IN}_{\mathbb{S}}([\Phi_2, h_2])$ son consistentes, es decir, no es el caso que existan a-subestructuras $[\Theta_1, q] \neq [\emptyset, \epsilon]$ y $[\Theta_2, \bar{q}] \neq [\emptyset, \epsilon]$ de $\text{IN}_{\mathbb{S}}([\Phi_1, h_1])$ e $\text{IN}_{\mathbb{S}}([\Phi_2, h_2])$, respectivamente.*

Prueba. Supongamos por el absurdo que existan a-subestructuras $[\Theta_1, q] \neq [\emptyset, \epsilon]$ y $[\Theta_2, \bar{q}] \neq [\emptyset, \epsilon]$ de $\text{IN}_{\mathbb{S}}([\Phi_1, h_1])$ e $\text{IN}_{\mathbb{S}}([\Phi_2, h_2])$, respectivamente. Sea $[\Theta'_1, q]$ la a-subestructura completa de $[\Phi_1, h_1]$, y sea $[\Theta'_2, \bar{q}]$ a-subestructura completa de $[\Phi_2, h_2]$. Por lema 3, $[\Theta_1, q] \sqsubseteq \text{IN}_{\mathbb{S}}([\Theta'_1, q])$,

y entonces, $\text{IN}_{\mathbb{S}}([\Theta'_1, q]) \neq [\emptyset, \epsilon]$. Análogamente podemos afirmar que $\text{IN}_{\mathbb{S}}([\Theta'_2, \bar{q}]) \neq [\emptyset, \epsilon]$. Finalmente, por lema 2, asegurando la consistencia de IN-narrowings a nivel de conclusión final, arribamos a un absurdo \square

Teorema 3 (consistencia dos a dos de narrowings garantizados). *Sean $[\Phi_1, h_1]$ y $[\Phi_2, h_2]$ a-estructuras maximales de \mathcal{P} , y sean $[\Phi'_1, h_1]$ y $[\Phi'_2, h_2]$ los narrowings garantizados de $[\Phi_1, h_1]$ y $[\Phi_2, h_2]$, respectivamente, de acuerdo a la semántica a-grounded [/a-estable / a-preferida]. Entonces podemos asegurar que $[\Phi'_1, h_1]$ y $[\Phi'_2, h_2]$ son consistentes, es decir, no es el caso que existan a-subestructuras $[\Theta_1, q] \neq [\emptyset, \epsilon]$ y $[\Theta_2, \bar{q}] \neq [\emptyset, \epsilon]$ de $[\Phi'_1, h_1]$ y $[\Phi'_2, h_2]$, respectivamente.*

Prueba. El caso de la semántica a-grounded es trivial a partir del teorema 2. Consideremos ahora el caso de la semántica a-estable (el razonamiento sirve también para la a-preferida). Supongamos por el absurdo que existen tales a-estructuras $[\Theta_1, q]$ y $[\Theta_2, \bar{q}]$. Luego, de acuerdo a la definición de narrowing garantizado de acuerdo a la semántica estable $[\Phi'_1, h_1] =_{def} \bigsqcap_{\mathbb{S} \in \Omega} \text{IN}_{\mathbb{S}}([\Phi_1, h_1])$, donde Ω es el conjunto de todas las asignaciones de status a-estables asociadas a \mathcal{P} . Entonces, por definición de ' \bigsqcap ', $[\Phi'_1, h_1] \sqsubseteq \text{IN}_{\mathbb{S}}([\Phi_1, h_1])$ para toda asignación $\mathbb{S} \in \Omega$. Consideremos en particular una asignación $\mathbb{S}_i \in \Omega$. Por hipótesis, $[\Theta_1, q] \neq [\emptyset, \epsilon]$ es a-subestructura de $[\Phi'_1, h_1] \sqsubseteq \text{IN}_{\mathbb{S}_i}([\Phi_1, h_1])$, y por lo tanto existe una a-subestructura $[\Theta'_1, q] \neq [\emptyset, \epsilon]$ de $\text{IN}_{\mathbb{S}_i}([\Phi_1, h_1])$ (más aún, $[\Theta_1, q] \sqsubseteq [\Theta'_1, q]$). Haciendo el mismo razonamiento para $[\Theta_2, \bar{q}]$, concluimos que existe una a-subestructura $[\Theta'_2, \bar{q}] \neq [\emptyset, \epsilon]$ de $\text{IN}_{\mathbb{S}_i}([\Phi_2, h_2])$. Finalmente la asignación \mathbb{S}_i viola el teorema 2, arribando a un absurdo. \square

Teorema 4. *Sean $[\Phi_1, h]$ y $[\Phi_2, \bar{h}]$ dos a-estructuras maximales. De acuerdo a la semántica a-grounded [/a-estable / a-preferida], si el narrowing garantizado de $[\Phi_1, h]$ es distinto de $[\emptyset, \epsilon]$, entonces el narrowing rechazado de $[\Phi_2, \bar{h}]$ es $[\Phi_2, \bar{h}]$.*

Prueba. El caso de la semántica a-grounded es trivial a partir del lema 1. Consideremos ahora el caso de la semántica a-estable (el razonamiento sirve también para la a-preferida). Sea $[\Phi'_1, h]$ el narrowing garantizado de $[\Phi_1, h]$ y $[\Phi''_2, \bar{h}]$ el narrowing rechazado de $[\Phi_2, \bar{h}]$. Sea $[\Phi'_1, h] \neq [\emptyset, \epsilon]$ (hipótesis). De acuerdo a la definición de narrowing garantizado para la semántica a-estable $[\Phi'_1, h] =_{def} \bigsqcap_{\mathbb{S} \in \Omega} \text{IN}_{\mathbb{S}}([\Phi_1, h])$, donde Ω es el conjunto de todas las asignaciones de

status a-estables asociadas a \mathcal{P} . Entonces, por definición de ‘ \sqcap ’, $[\Phi'_1, h] \sqsubseteq \text{IN}_{\mathbb{S}}([\Phi_1, h])$ para toda asignación $\mathbb{S} \in \Omega$, y dado que $[\Phi'_1, h] \neq [\emptyset, \epsilon]$ (hipótesis), luego $\text{IN}_{\mathbb{S}}([\Phi_1, h]) \neq [\emptyset, \epsilon]$, cualquiera sea $\mathbb{S} \in \Omega$. Aplicando el lema 1 para \mathbb{S} , $[\Phi_1, h]$ y $[\Phi_2, \bar{h}]$, concluimos que $\text{OUT}_{\mathbb{S}}([\Phi_2, \bar{h}]) = [\Phi_2, \bar{h}]$. Finalmente $[\Phi'_2, \bar{h}] =_{def} \bigsqcap_{\mathbb{S} \in \Omega} \text{OUT}_{\mathbb{S}}([\Phi_2, \bar{h}]) = [\Phi_2, \bar{h}] \square$

Bibliografía

- [AC02] AMGOUD, L., AND CAYROL, C. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence* 34 (March 2002), 197–215.
- [ACG⁺08] ALSINET, T., CHESÑEVAR, C., GODO, L., SANDRI, S., AND SIMARI, G. Formalizing argumentative reasoning in a possibilistic logic programming setting with fuzzy unification. *International Journal of Approximate Reasoning* 48, 3 (2008), 711–729.
- [ACGS08] ALSINET, T., CHESÑEVAR, C. I., GODO, L., AND SIMARI, G. R. A logic programming framework for possibilistic argumentation: Formalization and logical properties. *Fuzzy Sets and Systems* 159, 10 (2008), 1208–1228.
- [BCP06] BENCH-CAPON, T. J. M., AND PRAKKEN, H. Justifying actions by accruing arguments. In *COMMA* (2006), P. E. Dunne and T. J. M. Bench-Capon, Eds., vol. 144 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, pp. 247–258.
- [BG09] BARONI, P., AND GIACOMIN, M. *Semantics of abstract argument systems*. Springer Verlag, 2009, ch. 2, pp. 25–44.
- [BGG05] BARONI, P., GIACOMIN, M., AND GUIDA, G. Scc-recursiveness: a general schema for argumentation semantics. *Artif. Intell.* 168 (October 2005), 162–210.
- [BTK93] BONDARENKO, A., TONI, F., AND KOWALSKI, R. An assumption-based framework for non-monotonic reasoning. *Proc. 2nd. International Workshop on Logic Programming and Non-monotonic Reasoning* (1993), 171–189.

Se presenta un sistema basado en suposiciones que permite modelar distintos formalismos para razonamiento no monótono. A través del mismo se capturan distintos tipos de semánticas, particularmente aquellas de la programación en lógica.

- [Cam06a] CAMINADA, M. On the issue of reinstatement in argumentation. In *JELIA (2006)*, M. Fisher, W. van der Hoek, B. Konev, and A. Lisitsa, Eds., vol. 4160 of *Lecture Notes in Computer Science*, Springer, pp. 111–123.
- [Cam06b] CAMINADA, M. Semi-stable semantics. In *Proceeding of the 2006 conference on Computational Models of Argument: Proceedings of COMMA 2006* (Amsterdam, The Netherlands, The Netherlands, 2006), IOS Press, pp. 121–130.
- [Cam07] CAMINADA, M. An algorithm for computing semi-stable semantics. In *Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (Berlin, Heidelberg, 2007), ECSQARU '07, Springer-Verlag, pp. 222–234.
- [CDM03] CAYROL, C., DOUTRE, S., AND MENGIN, J. On decision problems related to the preferred semantics for argumentation frameworks. *J. Log. Comput.* 13, 3 (2003), 377–403.
- [CMDM05] COSTE-MARQUIS, S., DEVRED, C., AND MARQUIS, P. Prudent semantics for argumentation frameworks. In *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence* (Washington, DC, USA, 2005), ICTAI '05, IEEE Computer Society, pp. 568–572.
- [CML00] CHESÑEVAR, C., MAGUITMAN, A., AND LOUI, R. Logical models of argument. *ACM Computing Surveys* 32, 4 (2000), 337–383.
- [CMS06] CHESÑEVAR, C. I., MAGUITMAN, A. G., AND SIMARI, G. R. Argument-Based Critics and Recommenders: A Qualitative Perspective on User Support Systems. *Journal of Data and Knowledge Engineering* 59, 2 (2006), 293–319.
- [CSAG04] CHESÑEVAR, C., SIMARI, G., ALSINET, T., AND GODO, L. A Logic Programming Framework for Possibilistic Argumentation with Vague Knowledge. In *Proc. of the Intl. Conf. in Uncertainty in Artificial Intelligence (UAI 2004). Canada* (2004), pp. 76–84.
- [CSG05] CHESÑEVAR, C. I., SIMARI, G. R., AND GODO, L. Computing dialectical trees efficiently in possibilistic defeasible logic programming. In *Lecture Notes in Computer Science Vol. 3662 (Proc. of the LPNMR Conference)* (2005), C. Baral, G. Greco, N. Leone, and G. Terracina, Eds., pp. 158–171.
- [Dav89] DAVIS, R. E. *Truth, Deduction and Computation: Logic and Semantics for Computer Science*. Computer Science Press, 1989.

Se presentan los aspectos fundamentales referidos al uso de la lógica en Ciencias de la Computación. Se aborda la lógica proposicional, la lógica de predicados, el cálculo lambda y la teoría de funciones recursivas parciales, indicándose para cada una de ellas la relación entre la noción de verdad (modelos), deducción (prueba) y computación.

- [DBC03] DUNNE, P. E., AND BENCH-CAPON, T. J. M. Two party immediate response disputes: properties and efficiency. *Artif. Intell.* 149 (October 2003), 221–250.
- [DKT06] DUNG, P. M., KOWALSKI, R. A., AND TONI, F. Dialectic proof procedures for assumption-based, admissible argumentation. *Artif. Intell.* 170, 2 (2006), 114–159.
- [DMT06] DUNG, P. M., MANCARELLA, P., AND TONI, F. A dialectic procedure for sceptical, assumption-based argumentation. In *Proceeding of the 2006 conference on Computational Models of Argument: Proceedings of COMMA 2006* (Amsterdam, The Netherlands, The Netherlands, 2006), IOS Press, pp. 145–156.
- [DMT07] DUNG, P. M., MANCARELLA, P., AND TONI, F. Computing ideal sceptical argumentation. *Artif. Intell.* 171, 10-15 (2007), 642–674.
- [Dun93a] DUNG, P. M. An argumentation semantics for logic programming with explicit negation. In *Proc. ICLP'93* (1993), MIT Press, pp. 616–630.
- [Dun93b] DUNG, P. M. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning and Logic Programming. In *Proc. of the 13th. International Joint Conference in Artificial Intelligence (IJCAI), Chambéry, Francia* (1993), pp. 321–357.
- Se presenta una teoría para argumentación en la cual se define con precisión un criterio de aceptabilidad de argumentos, demostrándose que la programación en lógica y el razonamiento no monótono son distintas formas de argumentación.
- [Dun95] DUNG, P. M. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77, 2 (1995), 321–358.
- [FKEG93] FOX, J., KRAUSE, P., AND ELVANG-GORANSSON, M. Argumentation as a general framework for uncertain reasoning. In *Proceedings 9th. Conf. on Uncertainty in AI* (1993), David Heckerman and Abe Mamdani.

Se analiza el uso de argumentación en un contexto general de aplicación para razonamiento con incertidumbre. Se estudia en particular problemas de dominio médico.

- [FP97] FOX, J., AND PARSONS, S. On using arguments for reasoning about action and values. In *Proceedings of the AAAI Spring Symposium on Qualitative (1997)*, Stanford.
- En este trabajo se considera el uso de argumentación a partir del sistema definido originalmente en [FKEG93] para modelar acciones y asociar valores numéricos para evaluar argumentos.
- [FP98] FOX, J., AND PARSONS, S. Arguing about beliefs and actions. *Lecture Notes in Artificial Intelligence 1445* (1998).
- Versión para revista de [FP97], donde se expanden y amplían los resultados allí presentados.
- [Gar00] GARCÍA, A. J. *Programación en Lógica Rebatible: Lenguaje, Semántica Operacional y Paralelismo*. PhD thesis, Departamento de Ciencias de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina, December 2000.
- [GRS07] GARCÍA, A. J., ROTSTEIN, N. D., AND SIMARI, G. R. Dialectical explanations in defeasible argumentation. In *ECSQARU (2007)*, pp. 295–307.
- [GS04] GARCÍA, A., AND SIMARI, G. Defeasible logic programming: An argumentative approach. *Theory Practice of Logic Programming 4*, 1 (2004), 95–138.
- [Hag96] HAGE, J. A theory of legal reasoning and a logic to match. *Artificial Intelligence and Law 4*, 3–4 (1996), 157–368.
- [Hun94] HUNTER, A. Defeasible Reasoning with structured information. In *Proceedings of KR 1994* (1994).
- [JV99] JAKOBOVITS, H., AND VERMEIR, D. Dialectic semantics for argumentation frameworks. In *Proceedings of the 7th international conference on Artificial intelligence and law* (New York, NY, USA, 1999), ICAIL '99, ACM, pp. 53–62.
- [Kon88] KONOLIGE, K. Defeasible argumentation in reasoning about events. In *Methodologies for Intelligent Systems*, Z. W. Ras and L. Saitta, Eds. North-Holland, New York, NY, 1988, pp. 380–390.
- [KT99] KAKAS, A., AND TONI, F. Computing argumentation in logic programming. *Journal of Logic Programming (to appear)* (1999).
- Este artículo presenta un marco abstracto que permite computar distintas semánticas argumentativas usando variaciones paramétricas de una simple teoría de prueba.

- [LCS08] LUCERO, M. J. G., CHESÑEVAR, C. I., AND SIMARI, G. R. Formalizing accrual in defeasible logic programming. In *Proc. of the 12th Intl. Workshop on Nonmonotonic Reasoning (NMR 2008)* (2008), pp. 122–130.
- [LCS09a] LUCERO, M. J. G., CHESÑEVAR, C. I., AND SIMARI, G. R. Modelling argument accrual in possibilistic defeasible logic programming. In *ECSQARU* (2009), pp. 131–143.
- [LCS09b] LUCERO, M. J. G., CHESÑEVAR, C. I., AND SIMARI, G. R. On the accrual of arguments in defeasible logic programming. In *IJCAI* (2009), pp. 804–809.
- [LS89] LIN, F., AND SHOHAM, Y. Argument systems: A uniform basis for nonmonotonic reasoning. In *KR* (1989), pp. 245–255.
- [MC09] MODGIL, S., AND CAMINADA, M. Proof theories and algorithms for abstract argumentation frameworks. *Argumentation in AI, I. Rahwan and G. Simari. eds. Springer* (2009).
- [Pol70] POLLOCK, J. The Structure of Epistemic Justification. *American Philosophical Quarterly, monograph series 4* (1970), 62–78.
- [Pol91] POLLOCK, J. L. Self-defeating arguments. *Minds and Machines (Special issue: defeasible reasoning) 1, 4* (November 1991).
Pollock se concentra aquí en analizar el rol de argumentos auto-derrotantes en el sistema OSCAR en particular, y en razonamiento rebatible en general.
- [Pol94] POLLOCK, J. L. Justification and defeat. *Artificial Intelligence 67, 2* (1994), 377–407.
- [Pol95] POLLOCK, J. L. *Cognitive Carpentry: A Blueprint for How to Build a Person*. The MIT Press, Cambridge, Massachusetts, 1995.
- [Pra93] PRAKKEN, H. *Logical Tools for Modelling Legal Arguments*. PhD thesis, Vrije University, Amsterdam (Holanda), January 1993.
Se estudian los aspectos lógicos de aplicar técnicas de Inteligencia Artificial al dominio legal, con énfasis en el razonamiento rebatible y el razonamiento con información incompleta. Se comparan varios sistemas argumentativos existentes aplicados a problemas de carácter legal.
- [Pra05] PRAKKEN, H. A study of accrual of arguments, with applications to evidential reasoning. In *ICAAIL '05: Proceedings of the 10th international conference on Artificial intelligence and law* (New York, NY, USA, 2005), ACM, pp. 85–94.

- [PS96] PRAKKEN, H., AND SARTOR, G. A dialectical model of assessing conflicting arguments in legal reasoning. *Artificial Intelligence and Law* 4 (1996), 331–368.
- [PS97a] PRAKKEN, H., AND SARTOR, G. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logics* 7 (1997), 25–75.
Se define una variante de la programación en lógica extendida para el tratamiento de argumentos. El sistema se presenta en dos variantes: una de ellas con prioridades fijas, y la otra con prioridades rebatibles.
- [PS97b] PRAKKEN, H., AND SARTOR, G. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics* 7, 1 (1997), 25–27.
- [PS98] PRAKKEN, H., AND SARTOR, G. Modelling reasoning with precedents in a formal dialogue game. *Artificial Intelligence and Law* 6 (1998), 231–287.
Se presenta un acercamiento basado en diálogos para modelar el razonamiento en un contexto legal.
- [PS99] PRAKKEN, H., AND SARTOR, G. A system for defeasible argumentation, with defeasible priorities. In *Artificial Intelligence Today*. 1999, pp. 365–379.
- [PV02] PRAKKEN, H., AND VREESWIJK, G. Logical systems for defeasible argumentation. In *Handbook of Phil. Logic* (Kluwer, 2002), D. Gabbay and F. Guenther, Eds., pp. 219–318.
- [RS09] RAHWAN, I., AND SIMARI, G. *Argumentation in Artificial Intelligence*. Springer Verlag, 2009.
- [Sar93] SARTOR, G. A simple computational model for nonmonotonic and adversarial legal reasoning. In *Proceedings of the 4th International Conference on Artificial Intelligence and Law* (1993), ACM Press, pp. 192–201.
- [SL92] SIMARI, G. R., AND LOUI, R. P. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence* 53, 1–2 (1992), 125–157.
- [TDH09] THANG, P. M., DUNG, P. M., AND HUNG, N. D. Towards a common framework for dialectical proof procedures in abstract argumentation. *J. Log. and Comput.* 19, 6 (2009), 1071–1109.
- [Ver95] VERHEIJ, B. Accrual of arguments in defeasible argumentation. In *Proceedings of the 2nd Dutch/German Workshop on Nonmonotonic Reasoning* (Utrecht, 1995), pp. 217–224.

- [Ver96] VERHEIJ, B. *Rules, Reasons, Arguments: Formal studies of argumentation and defeat*. Doctoral dissertation, University of Maastricht, 1996.
- [Ver07] VERHEIJ, B. A labeling approach to the computation of credulous acceptance in argumentation. In *Proceedings of the 20th international joint conference on Artificial intelligence* (San Francisco, CA, USA, 2007), Morgan Kaufmann Publishers Inc., pp. 623–628.
- [VP00] VREESWIJK, G., AND PRAKKEN, H. Credulous and sceptical argument games for preferred semantics. In *JELIA* (2000), pp. 239–253.
- [Vre91] VREESWIJK, G. The feasibility of defeat in defeasible reasoning. In *In Proceedings of the 1st International Conference on Knowledge Representation and Reasoning* (1991), Morgan Kaufmann, pp. 526–534.
- [Vre92] VREESWIJK, G. Abstract Argumentation Systems. *Artificial Intelligence* 2, 3 (June 1992), 259–310.
- Este trabajo sintetiza los principales conceptos de la Tesis de G.Vreeswijk [Vre93b].
- [Vre93a] VREESWIJK, G. Defeasible dialectics: A controversy-oriented approach towards defeasible argumentation. *J. Log. Comput.* 3, 3 (1993), 317–334.
- [Vre93b] VREESWIJK, G. A. *Studies in Defeasible Argumentation*. PhD thesis, Vrije University, Amsterdam (Holanda), 1993.
- Se presentan en esta Tesis distintos estudios sobre argumentación rebatibles. Se define un sistema formal denominado *sistema argumentativo abstracto*, analizándose a partir de él distintos elementos del razonamiento rebatible.
- [Vre06] VREESWIJK, G. A. W. An algorithm to compute minimally grounded and admissible defence sets in argument systems. In *Proceeding of the 2006 conference on Computational Models of Argument: Proceedings of COMMA 2006* (Amsterdam, The Netherlands, The Netherlands, 2006), IOS Press, pp. 109–120.
- [VTS09] VIGLIZZO, I., TOHME, F., AND SIMARI, G. The foundations of delp: defeating relations, games and truth values. *Annals of Mathematics and Artificial Intelligence* (2009), 181–204.
- [Wal96] WALTON, D.Ñ. *Argumentation Schemes for Presumptive Reasoning*. Lawrence Erlbaum Associates, Mahwah, NJ, USA, 1996.