Revisión de Intenciones desde el Aprendizaje BDI

TESIS

que para obtener el grado de

Maestro en Inteligencia Artificial

presenta

José Martín Castro Manzano

Director de tesis: Dr. Alejandro Guerra-Hernández

Departamento de Inteligencia Artificial

Universidad Veracruzana

Xalapa, Ver.

2008

Agradecimientos

A mi familia. Por apoyarme a largo de este reto. Por permitirme comenzar este estudio y terminar este proyecto.

A mis profesores. A todos y cada uno de ellos, por soportar mis formaciones y deformaciones. Muy en especial al Dr. Alejandro Guerra por permitirme colaborar con él... y por tenerme paciencia. Al Dr. Manuel Martínez, por su amistad y por los libros. Al Dr. Nicandro Cruz, pues sin su aviso yo no estaría aquí en este momento. Al Dr. Héctor Acosta, por aceptarme en esta maestría. Al Dr. José Negrete: sus clases son, sin duda, magistrales.

A mis compañeros. Juan Carlos, Daniel y Luis: todos ellos fueron una gran ayuda en las noches de desvelo y los días de sueño. Sólo nosotros conocemos el significado de las frases: *pero lo hice yo* y *hay que hacerlo general*. A Gustavo, por supuesto, por su esencial ayuda en los experimentos de este trabajo.

A los administrativos. Por soportar mis preguntas y mis tardanzas.

Al soporte CONACYT de ciencia básica CB-2007-1 para el proyecto 78910 y por la beca CONACYT 214783. También al programa Promep PIFI de fortalecimiento de Cuerpos Académicos.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1.1. Presentación

Entre los múltiples propósitos de la Inteligencia Artificial se encuentra, fundamentalmente, el estudio sistemático del comportamiento inteligente. En este sentido, este tipo de estudio es, *prima facie*, doble: científico e ingenieril. La inteligencia artificial, de este modo, tiene como uno de sus objetivos principales lograr tanto el entendimiento como la reproducción del comportamiento inteligente. Tal meta supone una clásica dicotomía como la que otrora algunos filósofos establecieron: aquella entre la teoría y la práctica. Y dicha dicotomía, no está de más decirlo, no es falsa ni falaz; sin embargo, tampoco es evidente o trivial. Schopenhauer argumentaba, y no sin razón, que dicha bifurcación puede ser falaz cuando se afirma que cierta proposición es cierta en la teoría pero falsa en la práctica. A lo largo de este trabajo enfrentamos tal problema y nos remitimos a la tesis schopenhaueriana. Es nuestro interés revisitar las interconexiones de la teoría y la práctica: entre fundamentos filosóficos de racionalidad práctica, métodos formales y lenguages de programación.

Por parte de la filosofía, la teoría de razonamiento práctico propuesta por Bratman expone los fundamentos filosóficos del modelo BDI (*belief, desire, intention*) de agencia racional. Esta teoría es interesante porque, como veremos, no reduce las intenciones a una combinación o suma de creencias y deseos, sino que asume que las intenciones son componentes propios de la planeación y que consisten en planes parciales jerárquicos. Tal suposición explica los aspectos temporales del razonamiento práctico en su relación con las intenciones futuras, la persistencia de una intención, la reconsideración y el compromiso.

Por parte de los métodos formales, existen diversos sistemas lógicos que han sido propuestos para especificar y verificar propiedades racionales de sistemas agentes, al-

gunos de ellos son revisados en este trabajo. Desafortunadamente, la verificación de dichas propiedades en agentes implementados ha sido una tarea difícil y no evidente. Así, tenemos diferentes lógicas desarrolladas para caracterizar la conducta racional de los agentes. Las lógicas más utilizadas para hacer tal cosa son las lógicas BDI. En estas lógicas la conducta de un agente es especificada en términos de los cambios temporales en las actitudes mentales del agente (e.d., en sus creencias, deseos e intenciones). Por ejemplo, es racional intentar deseos que se creen como posibles, y los cambios temporales en estas actitudes mentales definen cuándo es racional para un agente abandonar sus intenciones.

Las lógicas BDI capturan la racionalidad de los agente a través de axiomas y reglas de inferencia que capturan las propidedades racionales de los agentes. Entre tales propiedades encontramos las estrategias de compromiso, las cuales están definidas mediante axiomas que usan operadores temporales y epistémicos, y dictan bajo qué condiciones un agente debería abandonar una intención. Así, estas lógicas son usadas para razonar acerca de los agentes, pero no para programarlos. Por otro lado, tenemos lenguages de programación, como $AgentSpeak(L)$, que han sido propuestos y usados para reducir la laguna entre la teoría (la especificación lógica) y la práctica (la implementación). En este caso, $AgentSpeak(L)$ tiene una semántica operacional bien definida, pero la verificación de propiedades racionales de los agentes programados no es evidente, pues dicha semántica excluyó, por mor de la eficiencia, el uso de las modalidades que hacen a las lógicas BDI lenguajes altamente expresivos.

Así que, para razonar acerca de tales propiedades proponemos tres vías:

- La revisión de los fundamentos filosóficos, tanto descriptivos como normativos, de la reconsideración y el compromiso según Bratman.

- La lógica $CTL_{AgentSpeak(L)}$ como un lenguage formal para la especificación y verificación de agentes programados en $AgentSpeak(L)$. Nuestra principal contribución es la definición de la semántica de los operadores temporales CTL en términos de una estructura de Kripke inducida por el sistema de transición de la semántica operacional de $AgentSpeak(L)$; y la verificación de que cualquier agente programado en $AgentSpeak(L)$ satisface ciertas propiedades expresadas en la especificación lógica.

- La relación de tales resultados para discutir las bondades del aprendizaje intencional en relación con el compromiso y la reconsideración.

De este modo, nuestro trabajo se perfila como una triangulación afortunada entre fundamentos filosóficos, métodos formales y lenguajes de programación para agencia racional.

1.2. Planteamiento del problema

El problema central que este trabajo pretende solventar consiste en determinar qué tipo de estrategia de compromiso utilizan los agentes $AgentSpeak(L)$.

1.3. Hipótesis

La hipótesis sugerida es que los agentes $AgentSpeak(L)$ siguen una forma limitada de compromiso flexible (*single-minded*).

1.4. Justificación

En el contexto de una teoría computacional sobre razonamiento práctico, basada en $AgentSpeak(L)$, el aprendizaje intencional es necesario para poder modelar e implementar las estrategias de compromiso flexible y abierto (*single* y *open-minded* respectivamente). Esto es relevante porque una estrategia de compromiso flexible o abierto permite una relación adaptativa con el medio ambiente.

1.5. Objetivos

Son cuatro los objetivos principales de nuestro trabajo, de los cuales los tres últimos constituyen el núcleo de esta investigación:

- Mostrar los conceptos básicos que fundamentan y soportan este trabajo:
  - El concepto de agencia: capítulo 2.
  - El modelo formal BDI: capítulo 4.
  - El lenguage $AgentSpeak(L)$ junto con su intérprete $Jason$: capítulo 5.
- Revisar los fundamentos filosóficos de los conceptos de compromiso y reconsideración en las teorías computacionales de razonamiento práctico para el lenguage $AgentSpeak(L)$: capítulo 3.
- Proponer la lógica $CTL_{AgentSpeak(L)}$ como un lenguage formal para la especificación y verificación de agentes programados en $AgentSpeak(L)$:

. Formalizar los conceptos de compromiso de intenciones en una teoría basada en $AgentSpeak(L)$ a través de $CTL_{AgentSpeak(L)}$: capítulo 6.

. Verificar formalmente las propiedades sobre compromiso que cumplen los agentes $AgentSpeak(L)$ a través de $CTL_{AgentSpeak(L)}$: capítulo 6.

- Discutir las bondades del aprendizaje intencional en relación con el compromiso y la reconsideración:

  . Introducir el aprendizaje intencional como una forma de aproximar una estrategia de compromiso flexible: capítulo 7.

  . Experimentar con el uso del aprendizaje intencional como una aproximación a la formación de políticas de abandono intencional en $AgentSpeak(L)$: capítulo 7.

1.6. Trasfondo

Nuestro trabajo descansa sobre la base de cuatro líneas de investigación principales: la teoría de agencia racional de Michael Bratman [8], el lenguage $AgentSpeak(L)$ [41], la metodología de Bordini *et al* [6] y la propuesta de Guerra *et al* [25].

1.7. Antecedentes del problema: estado del arte

Hay tres antecedentes claros: el problema del aprendizaje en el modelo BDI, el problema de la revisión de intenciones y el problema de la laguna entre la teoría y la práctica.

1.7.1. Aprendizaje en el modelo BDI

El primer antecedente consiste en que el modelo de agencia BDI presenta un problema: carece de un modelo de aprendizaje. A partir de esta carencia el problema del aprendizaje en Sistemas Multi-Agente bajo una arquitectura BDI ha sido estudiado individual y socialmente [24]. Hay un protocolo de aprendizaje social basado en la adopción cooperativa de metas donde los agentes deciden cooperar intencionalmente para actualizar el contexto de los planes cuya ejecución ha fallado. Cada agente en el sistema es capaz de recordar, hasta cierto punto, las creencias que soportan la adopción de planes como intenciones, así como el resultado de su ejecución: éxito o fallo.

De este modo cada agente dispone de un conjunto de ejemplos de entrenamiento para aprender el contexto de éxito o fracaso al ejecutar planes. Un agente debe tener razones prácticas que aseguran la ejecución exitosa de sus intenciones. De esta manera los contextos de los planes se pueden interpretar como un conocimiento común que los agentes deben mantener actualizado y consistente.

El protocolo de aprendizaje social se describe, *grosso modo*, así: cada vez que un agente falla en la ejecución de un plan intenta aprender el contexto del plan individualmente. Si esta intención falla, busca ayuda con otros agentes que comparten el mismo plan fallido. Estos agentes también comparten la intención de aprender un nuevo contexto para sus planes fallidos, de tal modo que envían de vuelta al agente que quiere aprender sus ejemplos de entrenamiento relevantes. Una vez que un nuevo y consistente contexto es aprendido el resultado es compartido por todos los agentes que participaron en este proceso de aprendizaje social.

1.7.2. Revisión de intenciones

El segundo antecedente consiste en que, si bien el cambio de creencias sobre la base de nueva información ha sido ampliamente estudiado con cierto éxito [1], la revisión de otros estados mentales ha recibido poca atención, y en particular las intenciones. Empero existen algunas teorías formales de la intención [12], la lógica de la revisión de intenciones ha sido escasamente considerada. Así, tenemos la propuesta de una teoría de revisión de intenciones [53] y un experimento con mentalidades de agentes programados en un ambiente estructurado como el TileWorld [32].

1.7.3. El problema entre la teoría y la práctica

La teoría de razonamiento práctico propuesta por Bratman [8] expone los fundamentos filosóficos de los enfoques computacionales de agencia racional conocido como BDI. Esta teoría es inovadora porque no reduce las intenciones a una combinación de creencias y deseos [12]; antes bien, asume que las intenciones son elementos particulares compuestos de planes parciales y jerárquicos. Bajo esta teoría, diferentes lógicas BDI [39] han sido propuestas para caracterizar formalmente la conducta racional de los agentes en términos de las propiedades de las actitudes intencionales y sus mutuas relaciones . Debido a su expresividad, estas lógicas han sido usadas para razonar acerca de las propiedades racionales de los agentes; pero, dado su costo computacional, no son usadas para programar agentes. Inversamente, lenguages de programación de agentes, tales como $AgentSpeak(L)$ [41], han sido propuestos para reducir la laguna entre la teoría (especificación lógica) y la práctica (implementación) de agentes

racionales. Y aún cuando este lenguaje de programación, como veremos, tiene una semántica operacional bien definida, la verificación de propiedades racionales no es evidente, pues las modalidades intencionales y temporales son abandonadas por mor de la eficiencia computacional.

1.8. Exposición y método

El trabajo está dividido en tres partes. La primera parte presenta los antecedentes filosóficos y computacionales sobre los que se basa nuestro trabajo: capítulos 2, 3, 4 y 5. A lo largo de tales capítulos la exposición es más bien descriptiva, y el método es narrativo. Hacemos esto con el propósito de presentar la información que requeriremos en la siguiente parte. La segunda parte muestra nuestros resultados, tanto en la dimensión formal como en la dimensión experimental: capítulos 6 y 7. De este modo, durante estos capítulos la exposición es demostrativa y descriptiva, con el fin de probar nuestra hipótesis y mostrar nuestros resultados. La tercera parte, el apéndice, muestra nuestros dos artículos publicados.

En la primera parte, el capítulo 2 muestra los conceptos de agencia, intencionalidad y agencia BDI. El capítulo 3 pasa revista a los argumentos filosóficos de Bratman que soportan la teoría de agencia racional: revisamos su teoría y, principalmente, los conceptos de compromiso y revisión. El capítulo 4 expone la especificación formal de la agencia BDI. El capítulo 5 expone la sintaxis y la semántica de $AgentSpeak(L)$ así como la sintaxis y la semántica de su intérprete $Jason$. En la segunda parte, donde mostramos nuestros resultados, el capítulo 6 formaliza y verifica los conceptos de compromiso en una teoría basada en $AgentSpeak(L)$ a través de $CTL_{AgentSpeak(L)}$. En el capítulo 7 discutimos nuestros resultados formales con los resultados experimentales en relación con el compromiso y la reconsideración: introducimos el aprendizaje intencional como una forma de aproximar una estrategia completa de compromiso flexible. Finalmente, en el capítulo 8, desplegamos nuestras conclusiones y apreciaciones finales. Y en la tercera parte añadimos, a manera de apéndice, los artículos publicados en MICAI 08 y LANMR 08.

Parte I

Antecedentes

Capítulo 2

Agencia BDI

2.1. Introducción

Se expondrán el concepto de agencia y de intencionalidad. Para el primero haremos dos aproximaciones: una informal, enfatizando los detalles generales de un programa agente; y una formal, definiendo los términos de una arquitectura agente. Posteriormente revisamos el concepto de intencionalidad a través de tres conceptos: lenguaje, conducta y razonamiento. Finalmente relacionamos los conceptos de agencia e intencionalidad para demarcar el modelo de agencia BDI de modo general.

2.2. Agencia

2.2.1. Aproximación informal

Entre los propósitos de la Inteligencia Artificial (IA) se encuentra, fundamentalmente, el estudio sistemático del comportamiento inteligente. En este sentido, este tipo de estudio es, *prima facie*, doble: científico e ingenieril [21].

- **Científico.** Porque se encarga del estudio de la inteligencia de manera general.

- **Ingenieril.** Porque se encarga también de la construcción de sistemas inteligentes.

Así, la IA tiene como uno de sus objetivos principales lograr el entendimiento y la reproducción del comportamiento inteligente. Este es el enfoque o paradigma de la IA propuesto principalmente por Russell y Norvig [44] y Nilsson [37]. Con base en este

enfoque, el concepto IA se define como el estudio de agentes inteligentes, haciendo que el término *agente* aparezca como propio de la IA en su doble modo de estudio.

Intuitiva y generalmente, un agente es cualquier cosa capaz de percibir y actuar. Este es el sentido de agencia en la filosofía clásica de Aristóteles, en *De anima*, como *nous poietikós*, es decir, como entendimiento que ejecuta acciones o entendimiento que hace algo [2]. Esta definición intuitiva tiene una traducción a una definición especializada como la que dan Wooldridge y Jennings [52]:

> *Un agente es un sistema de computadora que está situado en algún ambiente, y que es capaz de actuar autónomamente sobre él para alcanzar sus objetivos.*

Mientras que el ambiente, a su vez, se entiende como un entorno físico o virtual [17] donde se enfatizan las propiedades del agente: su percepción, su autonomía y su acción sobre él: lo cual caracteriza la noción débil de agencia. Aquí, sin embargo, asumiremos la noción fuerte de agencia, la cual incluye las propiedades de la noción débil más las posturas mentalistas como las que se especificarán más adelante.

Esta definición especializada, si bien sigue siendo general, provee un nivel de abstracción que presenta las siguientes ventajas:

- Permite observar las facultades cognitivas de los agentes al realizar sus acciones.
- Permite considerar diferentes tipos de agente, incluyendo aquellos que no se supone tengan tales facultades cognitivas.
- Permite considerar diferentes especificaciones sobre los sub-sistemas que componen los agentes.
- Muestra que un agente no es tal sin un ambiente correspondiente.

El ambiente, por su parte, también puede caracterizarse. Brooks considera que el medio ambiente por antonomasia es el mundo real, que el mundo es el mejor modelo del mundo, por lo que un agente debe tener una implementación robótica [11]. Por otra parte, Etzioni argumenta que no es necesario que los agentes tengan implementaciones robóticas dado que los ambientes virtuales, como los sistemas operativos y la web, son tan reales como el mundo real [17]. A lo largo de este trabajo asumimos la postura de Etzioni, sin rechazar la de Brooks, haciendo énfasis en que lo importante es que la interacción del agente con su ambiente se dé en los términos de la definición especializada: de forma autónoma.

Russell y Norvig [44] señalan que, independientemente de la postura anterior, los ambientes pueden clasificarse del siguiente modo:

- **Accesible o inaccesible.** Si un agente puede percibir a través de sus sensores los estados completos del ambiente donde se encuentra, se dice que el ambiente es accesible. Esta propiedad depende no sólo del ambiente, sino de las capacidades de percepción del agente. Como puede notarse, mientras más accesible sea el ambiente, más sencillo será de construir.

- **Determinista o no-determinista.** Si el siguiente estado del ambiente está determinado por la acción del agente, se dice que el ambiente es determinista. Si otros factores influyen en el próximo estado del ambiente, se dice que éste es no-determinista. El no-determinismo implica dos nociones importantes: *i)* que los agentes tienen un control parcial sobre el ambiente, y *ii)* que las acciones del agente pueden fallar.

- **Episódico o no-episódico.** Si la experiencia del agente puede evaluarse a través de episodios o rondas, decimos que el ambiente es episódico. Las acciones se evalúan en cada episodio. Dada la persistencia temporal de los agentes, estos tienen que hacer continuamente decisiones locales que tienen consecuencias globales. Los episodios reducen el impacto de estas consecuencias, y por lo tanto es más sencillo construir agentes en ambientes episódicos.

- **Estático o dinámico.** Si el ambiente puede cambiar mientras el agente se encuentra deliberando, se dice que el ambiente es dinámico; de otro modo es estático. Si el ambiente no cambia con el paso del tiempo, pero la evaluación de las acciones del agente si lo hace, se dice que el ambiente es semi-dinámico.

- **Discreto o continuo.** Si hay un número limitado de posibles estados del ambiente, distintos y claramente definidos, se dice que el ambiente es discreto; de otro modo se dice que es continuo. Como también podrá notarse, es más sencillo construir agentes en ambientes discretos, porque las computadoras también son sistemas discretos.

Esta categorización sugiere que es posible encontrar diferentes clases de ambientes. Russell y Norvig [44] presentan algunos ejemplos de ambientes bien estudiados en IA y sus propiedades (cuadro 2.1). Cada ambiente, o clase de ambientes, requiere de alguna forma agentes diferentes para que estos tengan éxito. La clase más compleja de ambientes corresponde a aquellos que son inaccesibles, no-episódicos, dinámicos y continuos. Por ejemplo, es discutible concebir a un *daemon* de sistema operativo, como *xbiff*, como un agente. Pero tal sistema cumple con la definición de agente: se las arregla para identificar a su usuario, encontrar su buzón electrónico en la red (su ambiente), buscar mensajes nuevos y comunicar al usuario la presencia de estos. El resultado es que podemos aproximar la definición de *xbiff* de una manera más

Ambiente	Accesible	Deter.	Episódico	Estático	Discreto
Ajedrez sin reloj	sí	sí	no	sí	sí
Ajedrez con reloj	sí	sí	no	semi	sí
Análisis imágenes	sí	sí	sí	semi	no
Backgammon	sí	no	no	sí	sí
Poker	no	no	no	sí	sí
Tutor inglés	no	no	no	no	sí
Robot toma piezas	no	no	sí	no	no
Controlador refinería	no	no	no	no	no
Robot navegador	no	no	no	no	no
Conductor de autos	no	no	no	no	no
Diagnóstico médico	no	no	no	no	no

Cuadro 2.1: Ejemplos de ambientes estudiados en IA y sus propiedades

comprensible para el usuario: el agente *xbiff*, un *daemon* del sistema X Windows situado en un ambiente UNIX, vigila constantemente el buzón de su usuario para avisarle cuándo llegan mensajes nuevos a través de una interfaz gráfica.

2.2.2. Aproximación formal

Tanto la definición intuitiva como la especializada tienen una afortunda formalización, de tal modo que el concepto de agente y ambiente quedan bien definidos en una arquitectura abstracta [52]. En primer lugar introducimos:

Definición 1 *(Estados) Un conjunto finito de estados discretos* $E = \{e_1, ..., e_n\}$

Según la clasificación de ambientes que proponen Russell y Norvig [44], los ambientes no necesariamente han de ser discretos, pero en esta aproximiación se asumirá que E es discreto. Posteriormente se define:

Definición 2 *(Acciones) Un conjunto de acciones que un agente puede ejecutar:* $A = \{\alpha_1, ..., \alpha_n\}$

Y así:

Definición 3 *(Corrida) Dado E y A, una corrida se define como una secuencia c:*

$$c = e_0 \xrightarrow{\alpha_0} e_1, ..., \xrightarrow{\alpha_{n-1}} e_n$$

Diremos que $C = \bigcup_i^n c_i$. Así, C_{A_i} será el subconjunto de las corridas que terminan en una acción i y C_{E_n} el subconjunto C que terminan en un n estado del ambiente. Y con esto podemos definir:

Definición 4 *(Acción en el ambiente) Dado C_{A_i} y E, la acción en un ambiente es una función f que va de las corridas que terminan en una acción i a todos los estados posibles: $f : C_{A_i} \rightarrow \wp(E)$*

De este modo, el ambiente es sensible a su historia, por lo que las acciones ejecutadas por el agente en el pasado también afectan la transición a estados futuros. Y con esto estamos ya en condiciones de definir formalmente las definiciones intuitiva y especializada de ambiente y agente:

Definición 5 *(Ambiente) Un ambiente es una 3-tupla $Env = \langle E, e_0, f \rangle$*

Definición 6 *(Agente) Un agente es una función $a : C_{E_n} \rightarrow A$*

De este modo un:

Definición 7 *(Sistema agente) Es una tupla $\langle Env, a \rangle$*

Esta arquitectura abstracta de un sistema agente nos muestra la relación que hay entre el agente y su ambiente (figura 2.1). El algoritmo 1 muestra la función que im-

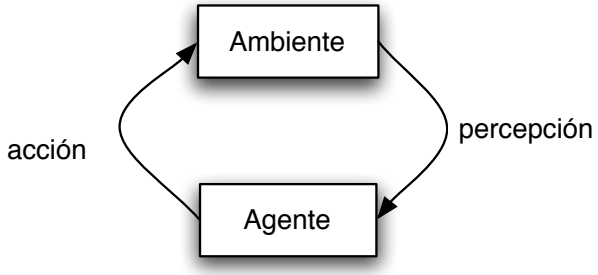

Figura 2.1: Abstracción de un agente a partir de su interacción con el medio ambiente.

plementa un agente de este tipo. Podemos también implementar un programa básico de ambiente que ilustre la relación entre éste y los agentes situados en él. El algoritmo 2 muestra el programa *Ambiente*. La historicidad del ambiente queda oculta en las percepciones del agente.

Algoritmo 1 Agente basado en mapeo ideal

```
1: function AGENTE-MAPEO-IDEAL(p)          ▷ p es una percepción del ambiente.
2:     percepciones ← percepciones ∪ p
3:     acción ← busca(percepciones, mapeo)          ▷ mapeo predefinido.
4:     return acción
5: end function
```

Algoritmo 2 Ambiente

```
1: procedure AMBIENTE(e, τ, ags, fin)          ▷ e Estado incial del ambiente.
2:     repeat
3:         for all ag ∈ ags do                  ▷ ags Conjunto de agentes.
4:             p(ag) ← percibir(ag, e)
5:         end for
6:         for all ag ∈ ags do
7:             acción(ag) ← ag(p(ag))
8:         end for
9:         e ← τ(⋃_{ag∈ags} acción(ag))        ▷ τ Función de transición del ambiente.
10:    until fin(e)                            ▷ fin Predicado de fin de corrida.
11: end procedure
```

2.2.3. Agencia débil

Bajo la anterior definición, un sistema agente en sentido débil tiene las siguientes propiedades:

- **Autonomía.** Una vez activo, un agente opera en su ambiente sin intervención directa externa, y tiene cierto control sobre sus acciones y su estado interno.

- **Reactividad.** Un agente percibe su ambiente y responde a él.

- **Proactividad.** Un agente exhibe una conducta orientada a metas.

- **Habilidad social.** Un agente puede interactuar con otros sistemas a través de ciertos lenguages [22].

2.2.4. Agencia fuerte

El concepto de agencia fuerte, por otro lado, incluye las propiedades definidas para el concepto de agencia débil, más una serie de elementos que son usualmente

aplicados o asociados con humanos. Así, es común caracterizar a los agentes usando nociones o supuestos mentalistas como *conocimento*, *creencia*, *intención* y *obligación* [49]. Más aún, algunos investigadores han llegado a considerar el desarrollo de agentes emocionales [4]. Es en el marco conceptual de la agencia fuerte en el que el concepto de agencia intencional adquiere significado teórico y práctico.

2.2.5. Otros atributos

Finalmente, algunos autores sugieren otras propiedades que pueden o no incluirse en la especificación formal del agente:

- **Movilidad.** Un agente tendrá la habilidad para moverse un su entorno [50].

- **Veracidad.** Un agente no comunicará información falsa [20].

- **Benevolencia.** Un agente tratará de alcanzar sus metas [43].

- **Racionalidad.** Un agente actuará de tal modo que alcance sus objetivos [20].

Una discusión sobre varios atributos de agencia aparece en [23].

2.3. Agentes como sistemas intencionales

Teniendo ya una noción de agencia (informal y formal), procedemos a mostrar una noción de intencionalidad que nos permita aproximarnos a nuestro marco teórico de agencia BDI.

Del latín *intentio*, (acción de tender hacia un objetivo), la intencionalidad inicialmente se entendió en la escolástica como la doctrina que afirma que todos los hechos de conciencia poseen y manifiestan una dirección u orientación hacia un objeto. Esta orientación, que se afirma de todo pensamiento, volición, deseo o representación consciente en general consiste, por un lado, en la presencia o existencia mental del objeto que se conoce, quiere o desea y, por el otro, en la referencia de este hecho mental a un objeto en principio real [19]. Pero fue Franz Brentano quien desarrolló la idea de que la intencionalidad es la característica propia de todos los fenómenos mentales y sus estudios influyeron particularmente en el desarrollo de la fenomenología de Husserl [10]. Aquí, sin embargo, manejaremos *intencionalidad* en tres sentidos: uno relacionado con el lenguaje, otro con la interpretación de la conducta y otro relacionado con el razonamiento.

2.3.1. Intencionalidad y lenguaje

El uso de *intencionalidad* como aquí será entendido tiene sus raíces en la filosofía analítica iniciada por Frege y Russell, quienes sacan la intencionalidad de la conciencia y la sitúan en el significado de las palabras, en las actitudes proposicionales [36]. A esta postura se añaden los supuestos generales del conductismo clásico, resultando así un concepto de intencionalidad que se define como un comportamiento linguístico, y que tiene su mejor explicación en la teoría de los actos de habla de Austin [3] y Searle [45].

Aquí el término *actitud proposicional* es usado para denotar un estado o evento mental intencional. De alguna forma, este término enfatiza el hecho de que no todos los estados mentales son intencionales. El problema semántico radica en que el término *estado*, particularmente en informática, es usado para expresar ciertas características globales de un sistema o que el sistema se encuentra en una situación bien identificada. Nos referimos a los estados mentales, no como estados en ese sentido, sino como procesos y estructuras computacionales modelando actitudes proposicionales.

Tradicionalmente, tres actitudes proposicionales son consideradas para modelar agentes racionales: creencias, deseos e intenciones (actitudes que le dan el nombre al modelo BDI). Jacques Ferber [18] presenta una clasificación más detallada de ellas y su uso en el modelado de los agentes:

- **Uso interactivo.** Percepción, información, comando, petición, norma.

- **Uso representacional.** Creencias, hipótesis.

- **Uso conativo.** Deseos, metas, impulsos, demandas, intenciones, compromisos.

- **Uso organizacional.** Métodos, tareas.

2.3.2. Intencionalidad y conducta

Si bien McCarthy es uno de los primeros en argumentar a favor de la adscripción de estados intencionales a máquinas [33], es el enfoque de Dennett el que ha sido utilizado como el fundamento filosófico para describir a los agentes como entidades a las que se les puede predicar creencias, deseos y otras actitudes proposicionales, de tal modo que los agentes se entienden como sistemas intencionales: en esto consiste la postura intencional [15].

De acuerdo a Daniel Dennett, los sistemas intencionales son, por definición, todas y sólo aquellas entidades cuyo comportamiento puede ser explicado o predicho desde una posición intencional, la cual consiste en la interpretación del comportamiento de

tal entidad (persona, animal, artefacto o lo que sea) como si fuera un agente racional que gobierna su selección de acción considerando sus actitudes proposicionales como creencias y deseos. A modo de ejemplo, veamos el caso del apagador de luz, propuesto por Yoav Shoham [49]: es perfectamente coherente tratar a un apagador como un agente con la capacidad de transmitir corriente eléctrica cuando quiero el cuarto iluminado, y de no hacerlo en cualquier otra circunstancia. Oprimir el botón del apagador es una forma simple de comunicarle nuestros deseos. La descripción intencional es, en efecto, simplista, porque no es necesaria en este caso. Pero hay casos más complejos, por ejemplo, al explicar conductas en enunciados como: *Juan leyó el* Ulises *de Joyce porque* creía *que era una buena obra.*

Dennett propone otras dos aproximaciones posibles para interpretar la conducta: la física, donde usamos nuestra intuición de la física; y la de diseño, que se adapta perfectamente al caso del apagador. El punto es que para sistemas más complejos, como los agentes inteligentes, las aproximaciones física y de diseño no están siempre disponibles, o no es práctico usarlas: piénsese, por ejemplo, en interpretar un procesador de palabras en términos físicos. Tal cosa es posible, pero no es conveniente para fines prácticos. Por lo general, cuanto más sabemos del sistema que queremos describir, menos necesitamos del enfoque intencional.

Las explicaciones de tipo intencional hacen uso de términos de la psicología *folk*, por lo que la conducta humana es explicada a través de la atribución de actitudes proposicionales tales como *creer*, *querer*, *esperar*, *desear*, etc. Este tipo de psicología *folk* está bien establecida y sus términos son descriptores de tipo intencional. Dennett propuso el término *sistema intencional* para describir a las entidades cuya conducta pudiera ser explicada y predicha a través de la atribución de tales actitudes proposicionales pertenecientes a la psicología *folk* [15] y observó que hay grados de intencionalidad:

- **Sistemas intencionales de primer orden.** Sistemas intencionales con creencias, deseos y otras actitudes proposicionales pero sin creencias ni deseos acerca de sus propias creencias y deseos (e.d., sin actitudes proposicionales anidadas).

- **Sistemas intencionales de segundo orden.** Sistemas intencionales con creencias, deseos y otras actitudes proposicionales, más creencias y deseos acerca de sus propias creencias y deseos (e.d., con actitudes proposicionales anidadas).

- **Sistemas intencionales de orden** $n > 2$. La jerarquía de intencionalidad puede extenderse tanto como sea necesario.

2.3.3. Intencionalidad y razonamiento

El tercer sentido de intencionalidad tiene su origen en la filosofía de la mente y en la propuesta de razonamiento práctico propuesto por Bratman. Michael Bratman propone en su teoría la idea de que nuestras intenciones están ligadas a la planeación, de tal modo que una intención consiste, entre otras cosas, en un plan adoptado para su ejecución [8]. Así, su teoría define un marco psicológico de sentido común para entender a los demás y a nosotros mismos, con base en las intenciones. La idea central es que nuestra concepción de las intenciones, de acuerdo al sentido común, está ligada al fenómeno de planeación y a los planes. Esto provee las bases para nuestros intentos cotidianos por predecir lo que los demás harán, explicar porqué lo han hecho, y coordinar nuestras acciones con las suyas. Este tipo de razonamiento comprende dos actividades:

- **Deliberación.** Decidir qué metas el agente debe lograr.

- **Análisis medios-fines.** Decidir cómo es que el agente va a satisfacer sus metas.

Ambas actividades pueden verse como procesos computacionales ejecutados por agentes racionales acotados. La racionalidad acotada consiste en una racionalidad con ciertas limitaciones:

- **Dinamicidad del ambiente**. En ambientes dinámicos, un agente debe controlar su razonamiento eficientemente para tener un buen desempeño.

- **No-deliberación infinita**. Los agentes no pueden deliberar indefinidamente, y deben eventualmente elegir las metas a atender, y comprometerse a satisfacerlas.

Con estas tres nociones de intencionalidad agregamos al modelo de agencia inicial un componente intencional que tiene una modelación perspicua en el modelo de agencia BDI. Algunos argumentos computacionales para el uso de la postura intencional han sido formulados por Singh [47]:

- Las actitudes proposicionales nos son familiares a todos: diseñadores, analistas de sistemas, programadores y usuarios.

- La postura provee descripciones sucintas del comportamiento de los sistemas complejos, por lo que ayuda a entenderlos y explicarlos.

- Provee de ciertas regularidades y patrones de acción que son independientes de la implementación física de los agentes.

- Un agente puede razonar sobre sí mismo y sobre otros agentes adoptando la postura intencional.

La postura intencional se basa en el hecho de que asumimos que un agente actuará racionalmente, dada su perspectiva limitada. Esta perspectiva limitada se describe atribuyendo al agente creencias y deseos particulares, con base en su percepción y sus metas. Este supuesto es el que nos permite hacer predicciones, de forma que esta capacidad es extremadamente sensible a la forma en que las creencias y los deseos son expresados por nosotros o representados por el agente.

2.3.4. IRMA: una arquitectura intencional

Para ejemplificar los conceptos de la agencia intencional mostramos la arquitectura IRMA [9]. Esta arquitectura tiene cinco componentes:

- **Creencias.** Denotadas por B. Representan el estado del mundo. Y se representan simbólicamente, como hechos en Prolog, es decir, son literales cerradas de la lógica de primer orden. Para efectos de esta arquitectura abstracta no es necesario especificar la representación exacta de las creencias.

- **Deseos.** Denotados por D. Es el conjunto de los deseos actuales del agente. Al igual que con las creencias, los detalles de representación serán por el momento omitidos.

- **Biblioteca de planes.** Denotada por P. Es el conjunto de planes π que el agente tiene como recetas o procedimientos. La función $execute(\pi)$ toma un plan como argumento y lo ejecuta. La biblioteca de planes está incluida en B, puesto que los agentes creen sus planes como procedimientos, mientras que las intenciones no están incluidas en B.

- **Intenciones.** Denotadas por I. Es el conjunto de las intenciones actuales del agente. Estas son planes como estados mentales, esto es, planes tomados de la biblioteca e instanciados por contexto.

- **Percepción.** Denotada por ρ. Adicionalmente, un agente necesita procesar su percepción del ambiente. La percepción es normalmente empaquetada en conjuntos discretos llamados perceptos. La función $nextPercept$ regresa la siguiente percepción disponible para el agente. Por simplicidad, sólo se muestra en la figura 2.2 como una entrada al agente ligada directamente a las creencias.

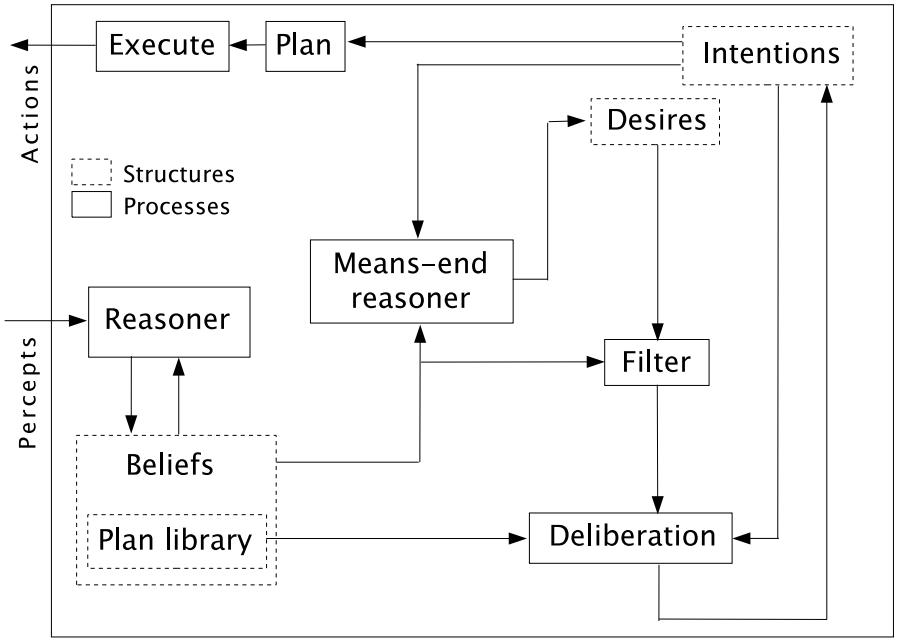

Figura 2.2: Una arquitectura para agentes racionales acotados basada en IRMA.

El agente comienza su percepción con una función

$$nextPercept : \wp(\rho) \longrightarrow \rho$$

y actualiza sus creencias con una función de revisión de creencias basada en su percepción y sus creencias actuales. Algunas veces esta función se conoce como razonador (*reasoner*) y tiene la siguiente forma:

$$reasoner : \wp(B) \times \rho \longrightarrow \wp(B)$$

El agente selecciona los planes relevantes usando un razonador de medios-fines (*meansEndsReasoner*) basado en las creencias y en las intenciones. Los planes seleccionados por esta función son los deseos del agente:

$$meansEndsReasoner : \wp(B) \times \wp(I) \longrightarrow \wp(D)$$

Los planes generados por el razonador de medios-fines son filtrados por el agente basado en sus creencias actuales, sus deseos y sus intenciones previas. La función filtro (*filter*) mantiene la consistencia y restringe las opciones que serán consideradas en la deliberación. El filtrado debe ser computacionalmente acotado, de forma que un proceso para detener el filtrado es necesario para equilibrar inercia y reconsideración:

$$filter : \wp(B) \times \wp(D) \times \wp(I) \longrightarrow \wp(D)$$

Finalmente una función de deliberación (*deliberation*) selecciona las opciones que serán incorporadas como intenciones, basada en razones creencia-deseo y la biblioteca de planes.

$$deliberation : \wp(B) \times \wp(D) \times Planes \longrightarrow \wp(I)$$

Y dado que las intenciones están estructuradas como planes, una función *plan* es usada para seleccionar la función ejecutada:

$$plan : \wp(I) \longrightarrow P$$

Esta arquitectura fue la primera implementación de un sistema computacional de razonamiento práctico: algoritmo 3.

Algoritmo 3 Agente Intencional

procedure AGENTE-INTENCIONAL($creencias, intenciones, planes$)
  while *true* **do**
    $\rho \leftarrow sensado()$
    $creencias \leftarrow percepción(creencias, \rho)$
    $deseos \leftarrow análisis - medios - fines(creencias, intenciones)$
    $deseos \leftarrow filtro(creencias, deseos, intenciones)$
    $intenciones \leftarrow deliberación(creencias, deseos, planes)$
    $\pi \leftarrow plan(intenciones)$
    $execute(\pi)$
  end while
end procedure

2.4. Agencia BDI

Entre los modelos de agencia racional propuestos en IA, y dentro del marco de la agencia fuerte, el modelo de agencia BDI ha resultado de gran relevancia. Y esto no ha sido baladí: por un lado, como hemos visto, cuenta con una serie de presupuestos filosóficos arduamente argumentados basados en las posturas de Frege, Austin, Searle, Dennett y Bratman, proveyendo las herramientas para describir a los agentes como sistemas intencionales (de razonamiento práctico y epistémico; por otro lado cuenta con una semántica operacional sumamente perspicua y refinada como lo es la lógica multimodal BDI [39].

En general, una arquitectura BDI consiste en la modelación de un conjunto de creencias (*Beliefs*), deseos (*Desires*) e intenciones (*Intentions*) y sus relaciones, que se definen informalmente del siguiente modo:

- **Creencias.** Representan información sobre el medio ambiente del agente. Constituyen la parte informativa del agente. Cada creencia se representa como una literal instanciada de la lógica de primer orden. Las literales no instanciadas se conocen como fórmulas de creencia y son usadas en la definición de planes, aunque no son consideradas creencias del agente. Las creencias se actualizan por la percepción del agente y la ejecución de sus intenciones, que producen la acción del mismo.

- **Deseos.** Representan las metas o tareas asignadas al agente. Constituyen la parte motivacional del agente. Normalmente se consideran como lógicamente consistentes entre sí. Los deseos incluyen lograr (*achieve*) que una creencia se vuelva verdadera y verificar (*test*) si una situación, representada como una conjunción o disyunción de fórmulas de creencia, es verdadera o falsa.

- **Eventos.** Las percepción del agente se mapea a eventos discretos almacenados temporalmente en una cola de eventos. Los eventos incluyen la adquisición o eliminación de una creencia; la recepción de mensajes en el caso multiagente y la adquisición de una nueva meta.

- **Planes.** Todo agente BDI cuenta con una librería de planes. Un plan está constituido por un evento disparador (*trigger event*) que especifica cuándo un plan debe ejecutarse, un contexto que determina si el plan puede ejecutarse y un cuerpo que determina los posibles cursos de acción a ejecutar. El cuerpo toma la forma de un árbol donde los nodos se consideran estados del ambiente y los arcos son acciones o metas del agente.

- **Intenciones.** Representan los cursos de acción que el agente se ha comprometido a cumplir. Constituyen la parte deliberativa del agente.

Un intérprete BDI (algoritmo 4) ejecuta las operaciones descritas en el orden adecuado, sobre las estructuras de datos de la arquitectura (figura 2.3). Es importante notar que los agentes BDI llevan a cabo dos tipos de razonamiento: el razonamiento práctico que los lleva a formar intenciones y ejecutarlas; y el razonamiento epistémico, típico de la IA, al momento de validar si el contexto de un plan es consecuencia lógica de sus creencias.

2.5. Resumen

Hemos presentado el concepto de agencia (como agencia fuerte) y una noción de intencionalidad que nos ha permitido mostrar, informalmente, el modelo de agencia

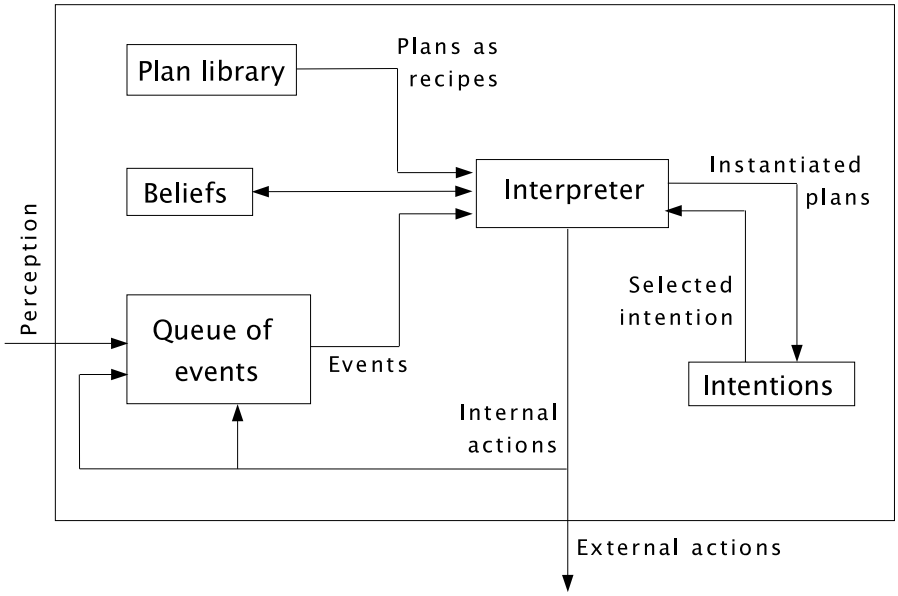

Figura 2.3: Arquitectura genérica BDI.

racional BDI, así como su intérprete y su arquitectura. Este modelo ha resultado de gran relevancia pues, como vimos, cuenta con una serie de presupuestos filosóficos arduamente argumentados basados en las posturas Austin, Searle, Dennett y Bratman. Así, hemos mostrado dos conceptos básicos que fundamentan y soportan este trabajo: agencia e intencionalidad.

Algoritmo 4 Agente BDI

procedure AGENTE-BDI($planes, creencias, deseos$)
  while $true$ **do**
    $eventos \leftarrow percepción()$
    while $eventos \neq \emptyset$ **do**
      $ev \leftarrow pop(eventos)$
      $deseos \leftarrow relevantes(aplicables(planes, creencias, ev)))$
      $int \leftarrow \sigma_{des}(deseos)$
      $intenciones \leftarrow pushInt(int, intenciones)$
    end while
    $ejecuta(top(\sigma_{int}(intenciones)))$
  end while
end procedure

Capítulo 3

La postura de Bratman

3.1. Introducción

Siguiendo con el paradigma que pretende tanto el entendimiento como la reproducción del comportamiento inteligente, pasamos a revisar los fundamentos filosóficos, tanto descriptivos como normativos, de la reconsideración y el compromiso según Bratman. Como veremos, esta teoría expone los fundamentos filosóficos del modelo BDI de agencia racional; sin embargo, no todos los postulados de su teoría han sido explorados o utilizados en la agencia BDI.

3.2. Supuestos bratmanianos

Para proceder a dar una descripción detallada de la postura que usaremos, primero revisaremos brevemente los supuestos básicos de la postura de Bratman tal y como quedan expuestos en [8].

- **Supuesto 1.** Las intenciones están ligadas a los planes. En primer lugar debemos notar que uno de los elementos que permite entender nuestra conducta y la de otros es la noción de intención. Y Bratman sugiere que nuestra concepción natural de intención está íntimamente ligada al fenómeno de la planeación y a los planes mismos.

- **Supuesto 2.** Somos agentes que planean. Frecuentemente formamos planes a futuro.

- **Supuesto 3.** Somos agentes racionales. Nuestros planes y su ejecución dependen de cierta deliberación.

- **Supuesto 4.** Una intención no es igual a un deseo. Tanto la intención como el deseo tienen roles motivacionales (ambas son pro-actitudes), pero la intención implica compromiso.

3.3. El trilema de Bratman

Sin embargo, dado el supuesto 2, las intenciones futuras y el compromiso nos llevan a un trilema difícil de resolver. Esto lo podemos apreciar con el siguiente ejemplo. Asumamos que el agente α tiene la intención, I, de ir de compras el próximo fin de semana. Entonces α tiene una intención futura. Pero ésta no es una suposición inocente. Una vez formada esta intención, surgen tres problemas:

- **Objeción metafísica.** Cuando I se forma, no controla todas las acciones futuras, pues de ser así, I implicaría acción a distancia: pero una cosa es compromiso y otra cosa es acción a distancia.

- **Objeción racional.** Una vez que I se forma, no es preciso ni se sigue que I no sea irrevocable: el mundo es dinámico y los agentes no siempre anticipan el futuro del mundo.

- **Objeción pragmática.** Dadas las dos objeciones anteriories, parece que I debería formarse sólo si es racional para α formar I, pero eso es inútil: si ese fuera el caso, no tendría porque haber planes a futuro, pero los hay.

3.4. Intenciones

La principal tradición en la filosofía de la mente ha propuesto un enfoque intencional para resolver el trilema a través de cuatro tesis principales:

- **La tesis metodológica**. La prioridad metodológica de la intención de actuar: comenzar siempre por la intención presente sin considerar la intención futura.

- **La tesis creencia-deseo.** Las acciones intencionales se definen como las compatibles con las creencias y los deseos del agente.

- **La tesis de la extensión.** Asumiendo que si las tesis 1 y 2 son suficientes para explicar la intención presente, también lo serán para explicar el caso de las intenciones futuras.

- **La tesis de la reducción.** Una reducción de la intención futura a un conjunto apropiado de deseos y creencias, es decir, que una intención se reduce a una combinación de creencias y deseos.

Bratman argumenta, como veremos a continuación, que estas tesis son no son suficientes para tratar con las intenciones futuras, y por lo tanto no resuelven las objeciones del trilema.

3.5. El modelo BD

Para resolver este trilema, el modelo BD -B por creencias y D por deseos- tiene dos aspectos: uno normativo para encontrar qué hace que una acción sea racional; y uno descriptivo, que concierne a cómo es que una acción es racional. Normativamente, el modelo BD parace adecuado para explicar el papel de las creencias y los deseos en la formación de intenciones. Sin embargo, como apunta Bratman, uno de los factores que nos permite entender las intenciones es el caso de las intenciones futuras y su diferencia con los deseos.

- **El problema de las intenciones futuras.** El problema del modelo BD es que, si bien normativamente provee argumentos para justificar la formación de intenciones, descriptivamente no permite tratar con las intenciones futuras, las cuales tienen un papel fundamental en las nociones de plan y compromiso.

- **La intención no es reducible al deseo.** La teoría BD reduce, descriptivamente, las intenciones a una relación entre creencias y deseos. Sin embargo, esta reducción no puede ser correcta, dado el supuesto 4: los deseos como las intenciones son pro-actitudes, pero los deseos son potenciales influencias de la conducta, mientras que las intenciones son conductores de la conducta. En otros términos, los deseos no implican compromiso ni explican la planeación a futuro: las intenciones sí.

Tomando esto en cuenta, el trilema no queda resuelto, por lo que Bratman procede a extender la teoría BD con miras a resolver el trilema.

3.6. Extensión modesta del modelo BD

La extensión de la teoría BD consiste en la extensión del sentido descriptivo de la teoría BD, preservando el sentido normativo de la misma.

Dado el supuesto bratmaniano de que los agentes son agentes que planean, el comportamiento de estos está delineado por un proceso de razonamiento práctico. Este tipo de razonamiento está enfocado a realizar acciones basadas en lo que el agente cree y desea, y tiene dos características importantes: la deliberación y razonamiento medios-fines:

- El proceso de deliberación consiste en la adopción de intenciones.

- El razonamiento-medios fines consiste en la determinación de los medios para cumplir las metas de las intenciones.

Bajo estos dos procesos, las intenciones pueden adaptarse con sus características:

- **Pro-actividad.** Las intenciones motivan a alcanzar una meta: son controladores de conducta.

- **Inercia**. Las intenciones persisten, es decir, una vez tomada, una intención se resiste a ser revocada: si una intención fuera tomada e inmediatamente revocada, tendríamos que decir que la intención jamás fue tomada. Sin embargo, apunta Bratman, si la razón por la cual se creó la intención desaparece, entonces es racional abandonar la intención.

- **Intenciones futuras.** Una vez adoptada una intención, esta restringirá los futuros razonamientos prácticos: mientras se mantiene una intención particular, el agente no considerará opciones contradictorias con dicha intención. Por ende, las intenciones proveen un filtro de admisibilidad para las posibles intenciones que un agente puede considerar.

Antes de proceder con el siguiente punto, es conveniente recalcar un punto sobre la inercia: dado un curso de acción, la intención resiste a la reconsideración, pero esta inercia o resistencia no es absoluta, es decir, una intención no es irrevocable (lo cual está garantizado por el segundo cuerno del trilema). Como diría Austin, cada intención que implica alguna acción futura es ambulatoria y revocable.

3.7. Planes

Los planes en tanto que cursos de acción, son intenciones y, en ese sentido, comparten las propiedades de estas: poseen inercia, son controladores de la conducta del agente y sirven como futuras entradas para próximos razonamientos prácticos. Sin embargo, los planes también tienen ciertas características distintivas:

- **Los planes son parciales.** Los planes no son estructuras completas y estáticas.

- **Los planes tienen una estructura jerárquica**. Los planes contienen razones medios-fines, y estas razones tienen un procedimiento ordenado.

Pero los planes también tienen ciertas exigencias para poder ser planes:

- **Consistencia interna.** El plan debe ser ejecutable.

- **Consistencia fuerte.** Un plan debe ser consistente con las creencias del agente.

- **Coherencia medios-fines.** Las razones medios-fines del plan, que típicamente son sub-planes, deben ser coherentes con los fines del plan.

Dos cosas que es importante dejar como antecedentes antes de proceder con la descripción de la postura de Bratman, es que tanto la coherencia medios-fines, y por lo tanto los planes, son derrotables (*defeasible*), es decir, que hay posibles circunstancias en las cuales el agente puede violar esta coherencia.

3.8. Las tesis de asimetría

Una de las exigencias de los planes, la consistencia fuerte, nos muestra que las creencias y las intenciones mantienen ciertas relaciones. Bratman considera estas relaciones como principios de racionalidad cuando un agente se enfrenta con razonamientos prácticos. Estas relaciones son las conocidas tesis de asimetría:

- **Inconsistencia intención-creencia.** Es irracional para un agente intentar ϕ y creer al mismo tiempo que no hará ϕ.

- **Incompletud intención-creencia.** Es racional para un agente intentar ϕ pero no creer que logrará ϕ.

Hasta aquí tenemos moldeada la postura de Bratman en lo que concierne ya al modelo BDI y a su funcionamiento. Procedemos a continuación a revisar los conceptos de reconsideración de una intención y compromiso, tomando en cuenta que:

- **Antecedente 1.** Dado un curso de acción o plan, la intención resiste a la reconsideración, pero esta inercia o resistencia no es absoluta, sino revocable.

- **Antecedente 2.** Los planes son derrotables.

3.9. El principio de intención-acción

Las intenciones, decíamos, son controladores de la conducta: en el curso normal de acciones, si un agente racional intenta ϕ al menos tratará de cumplir ϕ. Esto sugiere el siguiente principio: si es racional para un agente intentar ϕ, y el agente ejecuta con éxito esta intención y por ende hace ϕ intencionalmente, entonces es racional para el agente intentar ϕ.

La relevancia de este principio es doble. Por un lado, muestra que la intención presente y el resultado de esa intención están conectados: y esto es así porque la intención y la acción no están separadas por el agente. Al contrario, el agente tiene control de sus acciones a través de las intenciones. Por otro lado, este principio no es algo interno a los razonamientos prácticos del agente, sino más bien una norma externa para definir la racionalidad del agente.

3.10. Reconsideración

Para las intenciones hay ciertas normas de racionalidad que conciernen a la revisión (reconsideración o no-reconsideración) de las intenciones. Y aquí tenemos dos temas a tratar: uno descriptivo (los tipos de reconsideración y los aspectos de la reconsideración) y uno normativo (cuándo es racional reconsiderar intenciones).

Descriptivamente, Bratman distingue tres tipos de reconsideración.

- **Reconsideración no-reflexiva.** Reconsideración no-deliberativa.

- **Reconsideración deliberativa.** Se evalúan las creencias y los deseos sobre la propia reconsideración.

- **Reconsideración basada en políticas.** Se apela a una regla general sobre cuándo reconsiderar y cuándo no reconsiderar.

Bratman también sugiere que hay tres tipos de no-reconsideración. Y sugiere algunas tesis de asimetría: una reconsideración no-reflexiva es intencional, pero una no-reconsideración no-reflexiva es no intencional, sino, al contrario, una conducta por *default*.

Una reconsideración de una intención no consiste en meramente entretener la posibilidad de cambiar dicha intención, sino en considerar si la intención realmente se ha de continuar. Entonces surgen dos aspectos de la reconsideración:

- **Espera.** Una reconsideración implica poner en espera a la intención.
  - . Ocurren procesos de cambios de razones (*reason-changing*).
  - . Ocurren procesos de preservación de razones (*reason-preserving*).
- **Costo.** Una reconsideración implica una revisión de los planes mismos, y esto implica ciertos costos de acuerdo a la jerarquía de los planes y a su alcance en razonamientos futuros.

Normativamente, la cuestión de la reconsideración tiene que ver con cuándo es racional para un agente reconsiderar una intención.

- **Reconsideración no-reflexiva.** Un plan es estable (lo cual está garantizado por la inercia del mismo), pero tampoco es irrevocable. Entonces hay que notar en qué se basa esta estabilidad. Esta estabilidad o firmeza de la intención está relacionada con las atenciones del agente, y estas atenciones pueden ser sólo a ciertas cosas. La racionalidad de la reconsideración no-reflexiva se basa en una manifestación de los hábitos generales que un agente tiene para su reconsideración. Pero entonces surge la pregunta: qué hábitos son razonables. Respecto a esta pregunta hay dos puntos generales:
  - . **Problemas para los planes.** Un problema para un plan consiste en un problema de tipo interno: *i)* el mundo esperado puede ser diferente al mundo actual; *ii)* los deseos pueden cambiar y *iii)* las propias intenciones pueden cambiar.
  - . **Reconsideración ocasional.** Si hay oportunidad y recursos suficientes para llevar a cabo una reconsideración, es razonable reconsiderar la intención.
- **Reconsideración deliberativa y basada en políticas.** Cuando el agente delibera acerca de la reconsideración está, implícitamente, reconsiderando. Pero esta reconsideración implícita es no-reflexiva, por lo que los puntos generales de la reconsideración no-reflexiva valen para esta última.

3.11. Compromiso

Como se ha mencionado, las intenciones implican un compromiso. Y siguiendo la exposición, el compromiso tiene dos aspectos: uno normativo y uno descriptivo. Sin embargo, por el supuesto 2, aún nos enfrentamos al trilema. Pero ahora contamos con los elementos descriptivos para salir de él: las intenciones son elementos en planes más grandes que son parciales y jerárquicos. Estos planes tienen un papel crucial en la extensión de la deliberación. Como elementos de los planes, también funcionan como entradas para futuros razonamientos prácticos destinados a cumplir o a modificar estos planes. Y el funcionamiento normal de tales intenciones implica aspectos de agencia racional diferentes al razonamiento por cálculo: hábitos de reconsideración de intenciones. De este modo, se evita el trilema, pues los procesos de razonamiento práctico y retención de intenciones permiten una descripción que escapa a las objeciones del trilema.

- **Descriptivamente.** Este aspecto consiste en el papel de las intenciones para conectarse con futuras deliberaciones: hay dos niveles de compromiso:

  . **Volitivo.** Consiste en el papel característico de la intención presente. En términos ejemplares, en la dimensión volitiva, si el agente intenta ϕ ahora, normalmente hará ϕ ahora.

  . **Centrado-en-razones.** Consiste en el papel característico de las intenciones futuras. Para las intenciones futuras el aspecto centrado-en-razones tiene un papel fundamental en el ínterin entre la formación de la intención y su posterior ejecución. También, las intenciones futuras implican deliberación y análisis medios-fines.

- **Normativamente.** Los papeles de las intenciones tienen una dimensión que concierne a las normas o reglas de racionalidad, y se distinguen dos normas:

  . **Normas internas.** Son normas dentro de las razones prácticas del agente. Y éstas exigen una consistencia interna, consistencia fuerte y coherencia medios-fines.

  . **Normas externas.** Son normas fuera de las razones prácticas del agente. Y éstas consisten en la racionalidad de la reconsideración no-reflexiva y el principio de intención-acción.

3.12. Resumen

Hemos revisado la postura filosófica que sostiene a la agencia BDI: la teoría de Bratman. Esto es importante para lograr los objetivos de este trabajo: una aproximación a la revisión de intenciones y al compromiso desde el aprendizaje.
Así, podemos concluir provisionalmente:

- Las intenciones y los planes son revisables/derrotables bajo ciertas circunstancias, y por lo tanto el compromiso también lo es.

- La política de reconsieración/revisión de intenciones es diferente a la semántica de una intención: no es lo mismo el modo como una intención se revisa a como se forma o se abandona.

- En la reconsideración/revisión se apela a una regla general sobre cuándo reconsiderar y cuándo no reconsiderar.

- El compromiso centrado-en-razones provee de un marco teórico adecuado para la cuestión de cuándo reconsiderar una intención.

Con esto, hemos analizado los fundamentos filosóficos de los conceptos de compromiso y política de abandono para las teorías computacionales de razonamiento práctico.

Capítulo 4

Especificación formal

4.1. Introducción

Por parte de los métodos formales, existen diversos sistemas lógicos que han sido propuestos para especificar y verificar propiedades racionales de sistemas agentes: nosotros revisamos el trabajo fundacional de Cohen y Levesque, así como la propuesta de Rao. Desafortunadamente, la verificación de dichas propiedades en agentes implementados ha sido una tarea difícil y no evidente. Así, tenemos diferentes lógicas desarrolladas para caracterizar la conducta racional de los agentes. Las lógicas más utilizadas para hacer tal cosa son las lógicas BDI. En estas lógicas la conducta de un agente es especificada en términos de los cambios temporales en las actitudes mentales del agente (e.d., en sus creencias, deseos e intenciones).

4.2. La teoría intencional de Cohen y Levesque

La teoría de Cohen y Levesque (CL de ahora en adelante) es compleja [12], [28]. CL comienza con la noción de meta ($GOAL$) como una primitiva de lenguaje. Las metas de un agente especifican las alternativas que implícitamente escoge. Como uno de los postulados de CL está la consistencia mutua de las metas y la consistencia de éstas con las creencias del agente. CL define una meta persistente ($P-GOAL$) como una meta en la que el agente persiste hasta que cree que se ha cumplido o cree que es imposible. De este modo, en CL, las intenciones son entendidas como metas persistentes.

Los componentes esenciales de CL son: un modelo, M, que incluye un conjunto de secuencias lineales de mundos posibles (e.d., una función que va de los enteros

a los mundos posibles), un dominio de discurso; una función v que une variables a objetos del dominio, Φ que interpreta los predicados de diferentes cursos de eventos en diferentes índices temporales. La semántica, entonces, se define dado un modelo M, una secuencia de mundos posible σ, una función de valuación v y un índice temporal n tales que $M, \sigma, v, n \models \phi$ indica que ϕ es satisfecha en tal punto en tal modelo bajo la función de valuación v. Como es usual, $\models \phi$ significa que ϕ es lógicamente válida. A continuación mostramos la semántica de CL:

CL 1 $M, \sigma, v, n \models P(t_1; \ldots, t_n) \Leftrightarrow (v(t_1), \ldots, v(t_n)) \in \Phi[P, \sigma, v]$

CL 2 Las definiciones de $\neg$, $\vee$, $\exists$ y $=$ se definen de manera estándar.

CL 3 $M, \sigma, v, n \models (HAPPENS\ \alpha) \Leftrightarrow \exists m, m \geq n$ t.q. $M, \sigma, v, n\,[[\alpha]]\,m$ e.d., α describe una secuencia de eventos que ocurre al siguiente momento.

CL 4 $M, \sigma, v, n \models (DONE\ \alpha) \Leftrightarrow \exists m, m \leq n$ t.q. $M, \sigma, v, m\,[[\alpha]]\,|n$ e.d., α describe una secuencia de eventos que acaba de ocurrir.

$DONE\ x\ \alpha$ y $HAPPENS\ x\ \alpha$ significan que x es el agente que efectua $DONE\ \alpha$ y $HAPPENS\ \alpha$. La semántica de la relación de creencia (B) es euclidiana, transitiva y serial, mientras que la relación de meta (G) es serial. Se asume que $G \subseteq B$.

CL 5 $M, \sigma, v, n \models (BELIEF\ x\ \alpha) \Leftrightarrow \forall \sigma*$ t.q. $\langle \sigma, n \rangle\, B[v(x)]\sigma*, M, \sigma*, v, n \models \alpha$. Es decir, que α es verdadera en todos los mundos accesibles con B en σ y n.

CL 6 $M, \sigma, v, n \models (GOAL\ x\ \alpha) \Leftrightarrow \forall \sigma*$ t.q. $\langle \sigma, n \rangle\, G[v(x)]\sigma*, M, \sigma*, v, n \models \alpha$. Es decir, que α es verdadera en todos los mundos accesibles con G en σ y n.

La semántica de las acciones está dada en términos de [[]], que denotan que las acciones dadas toman lugar en tal intervalo.

CL 7 $M, \sigma, v, n\,[[e]]\,n + m \Leftrightarrow v(e) = e_1, \ldots, e_m$ y $\sigma(n + i) = e_i, 1 \leq i \leq m$. La secuencia de eventos denotados por e ocurren de n a m en σ.

CL 8 $M, \sigma, v, n\,[[\alpha; \beta]]\,n + m \Leftrightarrow \exists k, n \leq k \leq m$ t.q. $M, \sigma, v, n\,[[\alpha]]\,k$ y $M, \sigma, v, k\,[[\beta]]\,m$. Primero ocurre α después de n y posteriormente ocurre β terminando en m.

CL 9 $M, \sigma, v, n\,[[\alpha?]] \Leftrightarrow M, \sigma, v, n \models \alpha$ $\alpha?$ ocurre si α es verdadera, y falla si es falsa.

CL 10 $(BEFORE\ p\ q) =_{def} \forall c(HAPPENS\ c; q?) \Rightarrow \exists a(a \leq c) \wedge (HAPPENS\ \alpha; p?)$. Si q ocurre en el futuro, p ocurre antes que q, e.d., o bien p se da en el futuro y $\neg q$ se mantiene hasta entonces, o bien ni p ni q se dan.

CL 11 $(KNOW\ p) =_{def} p \wedge (BELIEF\ p)$. El conocimiento es una creencia verdadera.

CL 12 $(COMPETENT\ p) =_{def} ((BELIEF\ p) \Rightarrow (KNOW\ p))$. Un agente es competente acerca de p si y sólo si cree que p y sabe que p.

CL 13 $\diamond\alpha =_{def} \exists x(HAPPENS\ x; \alpha?)$. α es eventualmente verdadera si es verdadera después de una secuencia de eventos.

CL 14 $L\alpha =_{def} \neg \diamond \neg\alpha$

CL 15 $(LATER\ p) =_{def} \neg p \wedge \diamond p$. Esto es que $(LATER\ p)$ es verdadera si y sólo si p es verdadera en el futuro estricto.

CL 16 $(P - GOAL\ x\ p) =_{def} (GOAL\ x\ (LATER\ p)) \wedge (BELIEF\ x\ \neg p) \wedge [BEFORE((BELIEF\ x\ p) \vee (BELIEF\ x\ L\neg p))\ \neg(GOAL\ x\ (LATER\ p))]$. Un meta persistente es una proposición p t.q. *i)* es una meta del agente que p sea verdadera en el futuro estricto, *ii)* el agente no cree p ahora y *iii)* la primera condición se mantendrá a menos que el agente llegue a creer que p es verdadera o que p es imposible.

CL 17 $(INTEND_1\ x\ \alpha) =_{def}$
$(P{-}GOAL\ x[DONE\ x(BELIEF\ x\ (HAPPENS\ \alpha))?; \alpha])$. Un agente intenta una acción α si y sólo si tiene la meta persistente de hacer α inmediatamente después de creer que iba a suceder. Las intenciones, por tanto, son un tipo de metas persistentes.

Con todo, un análisis crítico [46] de esta postura muestra algunos problemas. El análisis de las intenciones según Bratman [8] es útil para la IA, y en particular, para los sistemas BDI. De acuerdo con Bratman, el compromiso de los agentes es meramente una condición que se mantiene en circunstancias *normales*. Esto significa que los agentes normalmente persisten en sus intenciones. En efecto, como decíamos renglones arriba, las intenciones tienen una inercia, e.d., resisten, hasta cierto punto, la revisión y la reconsideración mientras implican procesos de reconsideración. Esto, por supuesto, sólo es razonable si se toma en cuenta que las intenciones tienen un papel que no se reduce a una combinación de creencias y deseos. Sin embargo, como hemos descrito, CL mezcla la persistencia de los agentes en la semántica de las intenciones. Y de este modo la distinción entre *semántica* de la intención y *política* de la revisión de intenciones se pierde.

La semántica de la intención se aplica a todos los agentes en todas las circunstancias. Caracteriza los estados de los agentes y sus relaciones con diferentes mundos posibles. En contraste, diferentes políticas de revisión de intenciones aplican sólo en

algunos agentes y sólo en ciertas circunstancias. Así pues, CL no captura la distinción esencial entre la semántica de la intención y la política de la intención que consisten en saber cuándo actualizar una intención o cuándo no actualizar o persistir en dicha intención.

Debido a este problema recurrimos al formalismo BDI_{CTL} para posteriormente mostrar cómo aproximarnos a una teoría intencional que incorpore las nociones de semántica y política de la intención.

4.3. BDI_{CTL} y BDI_{CTL*}

El componente BDI para representar creencias, deseos e intenciones, se basa en operadores modales. El trabajo fundacional sobre este tipo de formalismos se debe a Cohen y Levesque (como vimos en la sección anterior); no obstante, ellos optan por una teoría de la intención reducida a una combinación de creencias y deseos (en términos bratmanianos, su tesis es la tesis de la reducción). Un componente temporal nos permitirá representar y razonar sobre los aspectos dinámicos del sistema y cómo estos cambian sobre el tiempo. Las lógicas computacionales arborescentes CTL y $CTL*$ [16] han sido usadas con este propósito. Finalmente, las lógicas BDI pueden incluir un componente de acción para representar los eventos registrados por los agentes y sus acciones. Este componente se basa normalmente en la lógica dinámica o se define usando fórmulas de estado para expresar la ocurrencia de eventos. El último enfoque es el utilizado aquí. En el resto del capítulo se definen la sintaxis y la semántica de las lógicas BDI, siguiendo el estilo adoptado en la sección anterior. Posteriormente se utilizan estas lógicas para formular una teoría BDI.

4.4. Sintaxis de BDI_{CTL} y BDI_{CTL*}

Como los mundos posibles de estas lógicas son estructuras temporales, la sintaxis de ambas es muy similar a la sintaxis de las lógicas temporales arborescentes. La diferencia con éstas es que ahora consideramos los operadores modales para las actitudes proposicionales:

- Conjunto de variables proposicionales: Var.

- Operadores unarios : $\neg$, $\bigcirc$, BEL, DES, INTEND.

- Operadores binarios: $\vee$, U .

- Cuantificadores de camino: E

- Signos de agrupación: (,).

Al igual que en las lógicas temporales arborescentes, hay dos tipos de fórmulas bien formadas: las fórmulas de estado, que son evaluadas con respecto a un mundo posible en particular; y las fórmulas de camino, que son evaluadas con respecto al camino formado por una serie de transiciones entre mundos posibles. Para la lógica BDI_{CTL*} las fórmulas de estado se definen inductivamente del siguiente modo:

BDI_{CTL*} **1** Si $\phi \in Var$, ϕ es una fórmula de estado.

BDI_{CTL*} **2** Si ϕ y ψ son fórmulas de estado, $\neg\phi$ y $\phi \wedge \psi$ son fórmulas de estado.

BDI_{CTL*} **3** Si ϕ es una fórmula de estado, $\mathsf{BEL}(\phi)$, $\mathsf{DES}(\phi)$ e $\mathsf{INTEND}(\phi)$ son fórmulas de estado.

BDI_{CTL*} **4** Si ϕ es una fórmula de camino, $\mathsf{E}(\phi)$ es una fórmulas de estado.

BDI_{CTL*} **5** Toda fórmula de estado es una fórmula de camino.

BDI_{CTL*} **6** Si ϕ y ψ son fórmulas de camino, $\neg\phi$ y $\phi \wedge \psi$ son fórmulas de camino.

BDI_{CTL*} **7** Si ϕ y ψ son fórmulas de camino, $\bigcirc\phi$ y $\phi\ \mathsf{U}\ \psi$ son fórmulas de camino.

La lógica restringida BDI_{CTL} se obtiene al prohibir combinaciones booleanas y anidamiento de operadores temporales en las fórmulas de camino. Formalmente, las reglas de fórmula de camino se sustituyen por la siguiente regla:

BDI_{CTL*} Si ϕ y ψ son fórmulas de estado, $\bigcirc\phi$ y $\phi\ \mathsf{U}\ \psi$ son fórmulas de camino.

Los conectivos lógicos de disyunción ($\vee$), implicación ($\Rightarrow$), eventualmente ($\Diamond$) y siempre (A) son abreviaturas. El uso de cuantificadores de camino introduce una nueva clasificación de las fbf's en estas lógicas: una fórmula es opcional (e.d., una O-fórmula) cuando no contiene ocurrencias de A (o de la negación de $\Diamond$) fuera del alcance de los operadores BEL, DES e INTEND. Por otro lado, una fórmula inevitable (e.d., una I-fórmula) es una fórmula que no contiene ocurrencias de $\Diamond$ (o de la negación de A) fuera del alcance de los operadores BEL, DES e INTEND.

4.5. Semántica de BDI_{CTL} y BDI_{CTL*}

En estas lógicas cada mundo posible se define como una estructura de árbol con un pasado único y un futuro arborescente. Cada estructura de árbol denota los cursos opcionales de eventos que pueden ser elegidos por el agente a partir de un mundo particular. Por tanto, los estados funcionan como índices en estas estructuras de árbol. Una relación de accesibilidad para las creencias establece qué mundos son creíbles a partir de cierto mundo y cierto estado. Las relaciones de accesibilidad para creencias y deseos funcionan de modo similar.

Para definir la semántica de las lógicas BDI_{CTL} y BDI_{CTL*} es necesaria una estructura de Kripke:

Definición 8 $K = \langle W, Var, \{S_w : w \in W\}, \{R_w : w \in W\}, L, \mathsf{BEL}, \mathsf{DES}, \mathsf{INTEND}\rangle$ *donde:*

- W *es el conjunto de mundos posibles.*
- Var *es el conjunto de variables.*
- S_w *es el conjunto de estados para cada mundo* $w \in W$*.*
- R_w *es una relación binaria serial sobre los estados del mundo* $w \in W$*.*
- L *es una función de asignación de valores de verdad para las proposiciones en cada mundo* $w \in W$ *y en cada estado* $s \in S_W$*.*
- BEL, DES *e* INTEND *son relaciones sobre los mundos y sus estados.*

Un mundo es sub-mundo de otro si contiene menos ramas que el otro pero es idéntico en todo lo demás. Formalmente, un mundo w' es un sub-mundo de un mundo w, $w' \sqsubseteq w$, si y sólo si *i)* $S_w \subseteq S_{w'}$; *ii)* $R_w \subseteq R_{w'}$; *iii)* $\forall s \in S_w, L(w,s) = L(w',s)$; y *iv)* $\forall s \in S_w, (w,s,v) \in \mathsf{BEL}$ si y sólo si $(w',s,v) \in \mathsf{BEL}$. Para DES e INTEND las definiciones son similares. También es importante notar que un mundo w' es un sub-mundo estricto de w si y sólo si $\forall x \in w' \Rightarrow x \in w \wedge \neg(\forall x \in w \Rightarrow x \in w')$.

Denotamos la satisfacción de una fbf por $\models$, y se define con respecto a una estructura de Kripke, un mundo w y un estado s. La expresión $K, w_s \models \phi$ se entiende como: la estructura K en el mundo w en el estado s satisface a ϕ. Estamos en condiciones de mostrar la semántica de las lógicas BDI_{CTL} y BDI_{CTL*}:

BDI_{CTL*} **1** $K, w_s \models \phi \Leftrightarrow \phi \in L(w,s)$

BDI_{CTL*} **2** $K, w_s \models \neg\phi \Leftrightarrow \neg(K, w_s \models \phi)$

BDI_{CTL*} **3** $K, w_s \models \phi \vee \psi \Leftrightarrow K, w_s \models \phi$ ó $K, w_s \models \psi$

BDI_{CTL*} **4** $K, w_s \models \mathsf{E}\phi \Leftrightarrow \exists c = (w_{s0}, \ldots) : K, c \models \phi$

BDI_{CTL*} **5** $K, w_s \models \mathsf{A}\phi \Leftrightarrow \forall c = (w_{s0}, \ldots) : K, c \models \phi$

BDI_{CTL*} **6** $K, w_s \models \Pi\phi \Leftrightarrow \exists v | (w, s, v) \in \Pi, K, V_S \models \phi$ donde $\mathsf{BEL}, \mathsf{DES}, \mathsf{INTEND} \in \Pi$.

BDI_{CTL*} **7** $K, (w_{s0,\ldots}) \models \phi \Leftrightarrow K, w_0 \models \phi$.

BDI_{CTL*} **8** $K, (w_{s0,\ldots}) \models \neg\phi \Leftrightarrow \neg(K, w_0 \models \phi)$.

BDI_{CTL*} **9** $K, (w_{s0,\ldots}) \models \phi \vee \psi \Leftrightarrow K, (w_{s0,\ldots}) \models \phi$ ó $K, (w_{s0,\ldots}) \models \psi$.

BDI_{CTL*} **10** $K, (w_{s0,\ldots}) \models \bigcirc\phi \Leftrightarrow \neg(K, w_1 \models \phi)$

BDI_{CTL*} **11** $K, (w_{s0,\ldots}) \models \phi \ \mathsf{U} \ \psi \Leftrightarrow$ *i)* $\exists k, k \geq 0 : K, (w_{sk}, \ldots) \models \psi$ y $\forall 0 \leq j < k, K, (w_{sj}, \ldots) \models \phi$; ó *ii)* $\forall j \geq 0, K, (w_{sj}, \ldots) \models \phi$

La validez y la satisfacción, como veremos más adelante, se definen de manera estándar. Informalmente, una fbf es válida si y sólo si es verdadera en todo estado, en todo mundo de toda estructura. La validez y la satisfacción con respecto a una familia de estruturas también puede definirse. Rao y Georgeff consideran dos clases de estructuras con respecto a las cuales evaluar validez y satisfacción: *i)* M que requiere que R sea total sin imponer ninguna restricción sobre los operadores intencionales; y *ii)* R^{est} que requiere que R sea total, BEL serial, transitiva y euclidiana; y DES e INTEND seriales. Este modelo subyace en la lógica denominada como $B^{KD45}D^{KD}I^{KD}_{CTL}$. En lo que sigue consideraremos sólo la lógica BDI_{CTL}.

4.6. Axiomatización de los componentes BDI

Dado que los componentes BDI se comportan como sistemas modales normales, el axioma K se adopta para Π:

Ax1. B-K $\mathsf{BEL}(\phi) \wedge \mathsf{BEL}(\phi \Rightarrow \psi) \Rightarrow \mathsf{BEL}(\psi)$

Ax2. D-K $\mathsf{DES}(\phi) \wedge \mathsf{DES}(\phi \Rightarrow \psi) \Rightarrow \mathsf{DES}(\psi)$

Ax3. I-K $\mathsf{INTEND}(\phi) \wedge \mathsf{INTEND}(\phi \Rightarrow \psi) \Rightarrow \mathsf{INTEND}(\psi)$

La regla de generalización se adopta para Π y establece que toda fórmula válida es creída, deseada e intentada. A la lógica resultate se le denomina BDI^{K}_{CTL}:

Ax4. B-gen $\vdash \phi \Rightarrow \mathsf{BEL}(\phi)$

Ax5. D-gen $\vdash \phi \Rightarrow \mathsf{DES}(\phi)$

Ax6. I-gen $\vdash \phi \Rightarrow \mathsf{INTEND}(\phi)$

El sistema modal $KD45$, o S5-débil, es adoptado para las creencias. El axioma D expresa la consistencia de las creencias, y los axiomas 4 y 5 expresan introspección positiva y negativa, respectivamente:

Ax7. B-D $\mathsf{BEL}(\phi) \Rightarrow \neg\mathsf{BEL}(\neg\phi)$

Ax8. B-S4 $\mathsf{BEL}(\phi) \Rightarrow \mathsf{BEL}(\mathsf{BEL}(\phi))$

Ax9. B-S5 $\neg\mathsf{BEL}\phi \Rightarrow \mathsf{BEL}(\neg\mathsf{BEL}(\phi))$

Para los deseos y las intenciones se adopta además el axioma D para expresar consistencia entre los deseos y las intenciones:

Ax10. D-D $\mathsf{DES}(\phi) \Rightarrow \neg\mathsf{DES}(\neg\phi)$

Ax11. I-D $\mathsf{INTEND}(\phi) \Rightarrow \neg\mathsf{INTEND}(\neg\phi)$

Como decíamos, la lógica resultate en $B^{KD45}D^{KD}I^{KD}_{CTL}$ es consistente y completa con respecto a la familia de estructuras M^{est} [39].

4.7. Realismos

El conjunto de relaciones estructurales BDI puede combinarse para obtener una variedad de estructuras de mundos posibles diferentes. Es posible obtener nueve relaciones entre los mundos creídos y los mundos deseados. De manera similar, hay nueve relaciones posibles entre los mundos deseados e intentados, y entre los mundos creídos e intentados. Tres de estas relaciones han sido consideradas en la literatura bajo los términos de realismo [12], realismo fuerte [39], y realismo débil [40].

4.7.1. Realismo fuerte

El realismo fuerte indica que el conjunto de mundos accesibles por las creencias es un subconjunto de los mundos accesibles por los deseos; y cada mundo accesible por creencias es un subconjunto de los mundos deseados. Como resultado, si el agente desea opcionalmente lograr una proposición, entonces cree que la proposición es una opción que, si es elegida, se logra. El realismo fuerte también puede aplicarse a los deseos e intenciones. Como resultado, si el agente intenta opcionalmente lograr una proposición, entonces también desea opcionalmente lograr esa proposición.

Los diferentes mundos accesibles por creencias, deseos e intenciones, representan diferentes posibles escenarios para el agente. Intuitivamente, el agente cree que el mundo actual es uno de sus mundos accesibles por creencias; si sucede que estuviera en el mundo accesible por creencias b_1, entonces sus deseos (con respecto a b_1) serían un mundo accesible por deseos, por ejemplo, d_1; y sus intenciones un mundo accesible por intenciones, por ejemplo., i_1. Los mundos d_1 e i_1 representan incrementalmente opciones selectivas desde b_1 acerca de los deseos por una opción y opciones de posibles cursos de acción futuros. Si ϕ es una fórmula-O, las condiciones anteriores se expresan con los axiomas de realismo fuerte:

Ax12. BD-Realismo fuerte $\mathsf{DES}(\phi) \Rightarrow \mathsf{BEL}(\phi)$

Ax13. DI-Realismo fuerte $\mathsf{INTEND}(\phi) \Rightarrow \mathsf{DES}(\phi)$

Los axiomas anteriores expresan que si el agente tiene la intención hacía ϕ, también desea que ϕ, esto es, existe al menos un camino en el que en todos los mundos accesibles por deseo ϕ es verdadera. Esto se asegura porque las relaciones BEL y DES son seriales.

Las condiciones semánticas para el realismo fuerte se expresan como:

DB-Realismo fuerte $\forall w \forall s \forall v$ si $(w, s, v) \in \mathsf{BEL}$ entonces $\exists v', (w, s, v') \in \mathsf{DES}$ y $v \sqsubseteq v'$ (ó $\mathsf{BEL} \subseteq_{sub} \mathsf{DES}$)

DI-Realismo fuerte $\forall w \forall s \forall v$ si $(w, s, v) \in \mathsf{DES}$ entonces $\exists v', (w, s, v') \in \mathsf{INTEND}$ y $v \sqsupseteq v'$ (ó $\mathsf{DES} \subseteq_{sup} \mathsf{INTEND}$)

4.7.2. Realismo

Phil Cohen y Hector Levesque [12] consideran una estructura donde el conjunto de mundos accesibles por intención es un subconjunto del conjunto de mundos accesibles por creencias y las estructuras de creencia e intención son idénticas (una línea de

tiempo). Esta restricción se conoce como realismo y tiene como efecto que si un agente cree una proposición también tendrá una intención con respecto a esa proposición. Los axiomas del realismo son:

Ax14. BD-Realismo $\mathsf{BEL}(\phi) \Rightarrow \mathsf{DES}(\phi)$

Ax15. DI-Realismo $\mathsf{DES}(\phi) \Rightarrow \mathsf{INTEND}(\phi)$

Las condiciones semánticas del realismo:

DB-Realismo $\forall w \forall s \forall v$ si $(w, s, v) \in \mathsf{DES}$ entonces $(w, s, v) \in \mathsf{BEL}$ (ó $\mathsf{DES} \subseteq \mathsf{BEL}$)

ID-Realismo $\forall w \forall s \forall v$ si $(w, s, v) \in \mathsf{INTEND}$ entonces $(w, s, v) \in \mathsf{DES}$ (ó $\mathsf{INTEND} \subseteq \mathsf{DES}$)

4.7.3. Realismo débil

Es posible obtener un balance entre los dos enfoques anteriores si los agentes no desean aquellas proposiciones cuya negación es creída; no intentan proposiciones cuya negación es deseada; y no intentan proposiciones cuya negación es creda. A esta propiedad se le conoce como realismo débil y se especifica con los siguientes axiomas:

Ax16. DB-Realismo débil $\mathsf{DES}(\phi) \Rightarrow \neg\mathsf{BEL}(\neg\phi)$

Ax17. IB-Realismo débil $\mathsf{INTEND}(\phi) \Rightarrow \neg\mathsf{BEL}(\neg\phi)$

Ax18. ID-Realismo débil $\mathsf{INTEND}(\phi) \Rightarrow \neg\mathsf{DES}(\neg\phi)$

A estos axiomas les corresponde la versión multi-modal de la condición serial:

DB-Realismo débil $\forall w \forall s \exists v (w, s, v) \in \mathsf{DES}$ si y sólo si $(w, s, v) \in \mathsf{BEL}$ (ó $\mathsf{BEL} \cap \mathsf{DES} \neq \{\}$)

IB-Realismo débil $\forall w \forall s \exists v (w, s, v) \in \mathsf{INTEND}$ si y sólo si $(w, s, v) \in \mathsf{BEL}$ (ó $\mathsf{BEL} \cap \mathsf{INTEND} \neq \{\}$)

ID-Realismo débil $\forall w \forall s \exists v (w, s, v) \in \mathsf{INTEND}$ si y sólo si $(w, s, v) \in \mathsf{DES}$ (ó $\mathsf{DES} \cap \mathsf{INTEND} \neq \{\}$)

4.7.4. Otras relaciones

Si un agente tiene una intención entonces cree que tiene tal intención:

Ax19. I-BI $\mathsf{INTEND}(\phi) \Rightarrow \mathsf{BEL}(\mathsf{INTEND}(\phi))$

I-BI $\forall w \forall s \forall w' \forall w''$ si $(w, s, w') \in \mathsf{BEL}$ y $(w, s, w'') \in \mathsf{INTEND}$ entonces $(w', s, w'') \in \mathsf{BEL}$

Si un agente tiene un deseo, entonces cree que tiene tal deseo:

Ax20. D-BD $\mathsf{DES}(\phi) \Rightarrow \mathsf{BEL}(\mathsf{DES}(\phi))$

D-BD $\forall w \forall s \forall w' \forall w''$ si $(w, s, w') \in \mathsf{BEL}$ y $(w, s, w'') \in \mathsf{DES}$ entonces $(w', s, w'') \in \mathsf{BEL}$

Si un agente tiene una intención, debe desear tal intención:

Ax21. I-DI $\mathsf{INTEND}(\phi) \Rightarrow \mathsf{DES}(\mathsf{INTEND}(\phi))$

I-DI $\forall w \forall s \forall w' \forall w''$ si $(w, s, w') \in \mathsf{DES}$ y $(w, s, w'') \in \mathsf{INTEND}$ entonces $(w', s, w'') \in \mathsf{DES}$

Si la relación de equivalencia ($\equiv$) es usada en este axioma en lugar de la implicación $\supset$, las modalidades anidadas se colapsan. Finalmente, si un agente forma una intención, entonces en algún momento futuro la abandona. Esto se conoce como no-compromiso infinito (*no-infinite deferral*):

Ax22. $\mathsf{INTEND}(\phi) \Rightarrow \mathsf{A}\Diamond(\neg\mathsf{INTEND}(\phi))$

4.7.5. Eventos

Para poder describir la conducta de un agente, es necesario describir la ocurrencia de acciones, aquí llamados eventos. Las extensiones necesarias en este sentido incluyen permitir que las fbf expresen el éxito y fracaso de los eventos. Si e es un evento primitivo, $succeeded(e)$ denota la ocurrencia exitosa de e en el pasado inmediato; $failed(e)$ denota el fracaso de e en el pasado inmediato; $done(e)$ denota la ocurrencia de e en el pasado inmediato (con éxito o fracaso). De manera similar, $succeeds(e)$, $fails(e)$, y $does(e)$ se usan para denotar ocurrencias futuras de e.

Sintaxis de los eventos

Primero, necesitamos un conjunto de símbolos para identificar a los eventos primitivos, por ejemplo, E. Ahora, la definición de fórmula de estado debe extenderse con la inclusión de la siguiente fórmula:

- Si $e \in E$, entonces $succeeds(e)$, $fails(e)$, $does(e)$, $succeeded(e)$, $failed(e)$, y $done(e)$ son fórmulas de estado.

Semántica de los eventos

E es un conjunto de tipos de evento primitivos; $SE_w : S_w \times S_w \mapsto E$ y $FE_w : S_w \times S_w \mapsto E$ que representan ocurrencias con éxito y fracaso de los eventos. Nótese que SE_w y FE_w son disjuntos. La semántica de los eventos, en un modelo M, se define que sigue:

- $M, w_{s_1} \models succeeded(e)$ si y sólo si $SE_w(s_0, s_1) = e$
- $M, w_{s_1} \models failed(e)$ si y sélo si $FE_w(s_0, s_1) = e$

El resto de los eventos se define como sigue: $done(e)$ se define como $succeeded(e) \vee failed(e)$, esto es, la ocurrencia del evento e independientemente de si se realizó con éxito o fracaso; $succeeds(e)$ se define como $\mathsf{A} \bigcirc (succeeded(e))$; $fails(e)$ se define como $\mathsf{A} \bigcirc (failed(e))$, esto es, el evento e se realiza con éxito o fracaso en todas las ramas a partir del estado actual; y $does(e)$ se define como $\mathsf{A} \bigcirc (done(e))$.

De este modo podemos axiomatizar los eventos que capturen el carácter volitivo del compromiso subyacente en las intenciones. Este axioma debe expresar que un agente actuará si tiene una intención dirigida hacía un tipo de evento primitivo:

Ax23. $\mathsf{INTEND}(does(e)) \Rightarrow does(e)$

Un agente debe ser consciente (debe creer) de todos los tipos de eventos primitivos que ocurren en su medio ambiente:

Ax24. $done(e) \Rightarrow \mathsf{BEL}(done(e))$

Al elegir uno de los realismos y adoptar el resto de los axiomas descritos, configuramos lo que Rao y Georgeff [40] llaman *Basic I system*.

4.8. Compromiso como axioma de cambio

Con este formalismo estamos en condiciones de especificar cómo las intenciones actuales se relacionan con las intenciones futuras. Una alternativa consiste en pensar en esta relación como un proceso de mantenimiento y revisión de intenciones o una estrategia de compromiso. Tres estrategias son bien conocidas en la literatura multi-agentes [39]: la estrategia ciega, la flexible y la abierta (*blind*, *single minded* y *open minded* respectivamente). Eligiendo una de estas tres estrategias y asumiendo el *Basic I System*, se configuran tres diferentes agentes básicos.

- **Blind commitment.** Un agente se compromete ciegamente si mantiene sus intenciones hasta que cree que las ha logrado satisfacer:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ \mathsf{BEL}(\phi))$$

  Es importante notar que este axioma se define para I-fórmulas. Nada se dice sobre la intención de un agente por lograr opcionalmente algún medio o fin particular. Esta estrategia es demasiado fuerte, pues para un agente comprometido ciegamente resulta inevitable creer que ha logrado sus fines. Esto se debe a que este tipo de compromiso sólo permite caminos futuros en los cuales o bien el objeto de la intención es creído o bien la intención se mantiene para siempre. Sin embargo, debido a la propiedad de no-deliberación infinita (*no-infinite deferral*), tenemos que $(\mathsf{INTEND}(\phi) \Rightarrow \mathsf{A}\Diamond(\neg\mathsf{INTEND}(\phi)))$, por lo que tal clase de caminos no está permitida, y en consecuencia obtenemos agentes que creen que eventualmente han logrado sus intenciones:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \Rightarrow \mathsf{A}\Diamond\mathsf{BEL}(\phi)$$

- **Single-minded commitment.** Al relajar la estrategia anterior de modo que el agente mantenga sus intenciones en tanto considere que siguen siendo una opción viable obtenemos una estrategia flexible. Formalmente

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi) \vee \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi))$$

  En tanto el agente crea que sus intenciones se pueden lograr, un agente de este tipo no abandonará sus intenciones y seguirá comprometido. Así, un agente flexible de manera inevitable, eventualmente creerá que ha logrado satisfacer sus fines sólo si continua creyendo que el objeto de sus intenciones sigue siendo una opción.

- **Open-minded commitment.** Un agente abierto mantiene sus intenciones mientras éstas sigan siendo deseadas. Formalmente:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi) \vee \neg\mathsf{DES}(\mathsf{A}\Diamond\phi)))$$

4.9. Resumen

Hemos revisado el modelo formal BDI. Revisamos el trabajo fundacional de Cohen y Levesque, las lógicas computacionales arborescentes CTL y $CTL*$ y, finalmente, las lógicas BDI. Con esto hemos definido cómo las intenciones actuales se relacionan con las intenciones futuras a través de estrategias de compromiso (un proceso de mantenimiento y revisión de intenciones). Mostramos tres estrategias que son bien conocidas en la literatura multi-agentes: la estrategia ciega, la flexible y la abierta (*blind*, *single minded* y *open minded* respectivamente).

Capítulo 5

AgentSpeak(L)/Jason

5.1. Introducción

Los métodos formales nos permiten especificar, diseñar y verificar agentes racionales. Sin embargo, Rao observó que una gran parte de las implementaciones exitosas de la arquitectura BDI asumían una serie de simplificaciones: modelaban las actitudes proposicionales como estructuras de datos, y mejoraban su desempeño con base en los planes programados por el usuario, dando como resultado una aparente falta de fundamentos teóricos. El problema de fondo es que las lógicas multimodales BDI, empleadas en la especificación de estos sistemas, poco ofrecían en relación con los problemas prácticos de su programación. Los primeros intentos para resolver este problema se concentraron en proponer una arquitectura abstracta BDI como una idealización de la implementación de estos sistemas y como un vehículo para continuar investigando las propiedades teóricas de la agencia intencional. Estos trabajos culminaron en dMARS [31]. El problema con este enfoque es que, debido a su nivel de abstracción no fue posible establecer una correspondencia uno-a-uno entre la teoría de modelos, la teoría de prueba y el intérprete abstracto.

Con *AgentSpeak*(L), una versión simplificada de dMARS, A. Rao propone un camino diferente hacía tal lenguaje [39], [41]. *AgentSpeak*(L) es un lenguaje de programación basado en una lógica restringida de primer orden con eventos y acciones. Las actitudes proposicionales no están representados directamente como expresiones modales. El estado actual de un agente, que es modelo de él mismo, su ambiente y otros agentes, puede considerarse como el conjunto de creencias presentes del agente. Los estados que el agente quiere lograr con base en sus estímulos internos y externos, constituyen sus deseos. Y la adopción de programas para satisfacer estos deseos constituyen las intenciones del agente. Esta apuesta por adscribir intencionalidad a

un modelo ejecutable del agente constituyó un cambio de paradigma que acercaba la teoría a la praxis de la agencia racional. En lo que sigue discutiremos la sintaxis y la semántica de $AgentSpeak(L)$ así como su teoría de prueba. Posteriormente revisamos la sintaxis y la semántica de *Jason*, el intérprete de $AgentSpeak(L)$.

5.2. Sintaxis de $AgentSpeak(L)$

En esta sección abordaremos el lenguaje para escribir programas de agente en $AgentSpeak(L)$. El alfabeto de este lenguaje formal consiste en variables, constantes, símbolos funcionales, símbolos de predicado, símbolos de acciones, conectivas, cuantificadores y signos de puntuación. Además de las conectivas de primer orden, se usan ! y ? para identificar ciertas metas, ; para secuencias y $\leftarrow$ para la implicación. Las definiciones estándar para término, fórmula bien formada, fórmula cerrada, ocurrencia libre y acotada de variables son adoptadas.

Definición 9 *(Creencias) Sea b un símbolo de predicado y $t_1, \ldots, t_n$ una secuencia de términos, entonces $b(t_1, \ldots, t_n)$ ó $b(t)$ es una creencia atómica. Si $b(t_1)$ y $b(t_2)$ son creencias atómicas, entonces $b(t_1) \wedge (t_2)$ y $\neg b(t_1)$ son creencias. Una creencia atómica o su negación será identificada como literal de creencia. Una creencia atómica sin variables libres será llamada creencia básica.*

Tómese como ejemplo una simulación de tráfico en una autopista de cuatro carriles. Los autos pueden aparecer en cualquier carril moviéndose de norte a sur. Es posible que aparezca basura en los carriles y nuestro contratista ha comprado un robot que recoge la basura y la pone en un depósito. Necesitamos programar al robot de tal manera que haga su trabajo sin ponerse en peligro (e.d., sin quedarse en un carril donde ha aparecido un auto). A lo largo de este capítulo construiremos el programa de agente para este robot. Las creencias de este agente representan la configuración de la autopista, agentes y objetos en ella incluidos. Las creencias lucen como $adyacente(X, Y)$ ó $pos(Robot, X)$, etc. Las creencias básicas del agente son casos sin variables libres de estas creencias atómicas, por ejemplo: $adyacente(c1, c2)$, $pos(r1, c2)$, etc.

Definición 10 *(Metas) Si g es un símbolo de predicado y $t_1, \ldots, t_n$ es una secuencia de términos, entonces $g(t_1, \ldots, t_n)$, $!g(t)$, $?g(t_1, \ldots, t_n)$ ó $?g(t)$ son metas.*

Una meta es un estado del sistema que el agente desearía ver logrado. Los agentes en $AgentSpeak(L)$ consideran dos tipos de meta: las metas que propiamente el agente quiere lograr (*achievement goals*, $!g(t)$) y las metas que el agente quiere verificar (*test goals*, $?g(t)$). En el ejemplo que estamos siguiendo, limpiar el carril 2 es una meta a

lograr $!limpiar(c2)$ y preguntarse si hay un auto en el carril 1 es una meta a verificar $?pos(auto, c1)$.

Definición 11 *(Eventos disparadores) Si $b(t)$ es una creencia atómica y $!g(t)$ y $?g(t)$ son metas, entonces $+b(t)$, $-b(t)$, $+!g(t)$, $+?g(t)$, $-!g(t)$ y $-?g(t)$ son eventos disparadores.*

Los cambios en el ambiente del agente y en su estado interno generan eventos disparadores (*te* por *trigger events*). Estos eventos incluyen agregar (+) y borrar (−) metas o creencias al estado del agente. Por ejemplo, detectar basura en un carril cualquiera toma la forma del evento disparador $+pos(basura, X)$ y adquirir la meta de limpiar un carril $+limpiar(X)$.

Definición 12 *(Acciones) Si a es un símbolo de acción y $t_1, \ldots, t_n$ es una secuencia de términos de primer orden, entonces $a(t_1, \ldots, t_n)$ ó $a(t)$ es una acción.*

El agente debe ejecutar acciones para lograr el cumplimiento de sus metas. Las acciones pueden verse como procedimientos a ejecutar. Normalmente, estas acciones son implementadas en el mismo lenguaje de programación en el que $AgentSpeak(L)$ ha sido implementado, pero éste no es necesariamente el caso, pues puede hacerse uso de interfaces a lenguajes foráneos en el caso de *Java* y *Lisp*. En todo caso, una acción como $ir(X, Y)$ debería tener como resultado que el agente se halle en el carril Y y no en X.

Definición 13 *(Planes) Si te es un evento disparador, $b_1, \ldots, b_n$ es una secuencia de literales de creencia y $g_1, \ldots, g_n$ es una secuencia de metas o acciones, entonces $te : b_1 \wedge \ldots \wedge b_n \leftarrow g_1, \ldots, g_n$ es un plan.*

La expresión a la izquierda de la flecha se conoce como la cabeza del plan. La expresión a la derecha de la flecha se conoce como cuerpo del plan. La expresión a la derecha de los dos puntos (:) en la cabeza del plan se conoce como contexto. Por conveniencia un cuerpo vacío se denota como *true*. Como el resto de los agentes BDI, los agentes $AgentSpeak(L)$ poseen una biblioteca de planes. A continuación tenemos un plan para responder a la aparición de basura en algún carril. Si el agente está en el mismo carril que la basura, ejecutará la acción de levantarla, y se planteará la meta de ir al depósito para finalmente ejecutar la acción de tirar ahí la basura. El comportamiento del agente se completa con los planes para ir a algún sitio:

```
p0
+pos(basura,X)
```

```
   : (pos(r1,X) & pos(deposito,Y))
   <- levantar(basura);
   !pos(r1,Y);
   tirar(basura).

p1
+!pos(r1,X)
   : pos(r1,X)
   <- true.

p2
+!pos(r1,X)
   : (pos(r1,Y) & ((not(X=Y)) & adyacente(Y,Z)))
<- ir(Y,Z);
!pos(r1,X).
```

Esta forma de especificar un agente es similar a lo que hacemos en programación lógica al especificar hechos y reglas. Sin embargo, existen diferencias importantes entre un programa lógico y uno de agente:

- En un programa lógico puro, no hay diferencias entre una meta en el cuerpo de una regla y en su cabeza. En un programa de agente, la meta en la cabeza es un evento disparador, no una meta en sí. Esto permite más expresividad para invocar planes posibilitando procesos dirigidos por datos (al agregar y eliminar creencias) y dirigidos por metas (al agregar y eliminar metas).

- Las reglas en la programación lógica pura no son sensibles al contexto, como es en el caso de los planes.

- La ejecución exitosa de una regla en la programación lógica regresa una substitución; la ejecución de un plan genera secuencias de acciones que modifican el medio ambiente donde está situado el agente.

- Cuando computamos una meta en programación lógica el proceso no puede ser detenido (*querying*); los planes de un agente, al contrario, pueden ser interrumpidos.

5.3. Semántica de $AgentSpeak(L)$

Primero definimos los conceptos de agente, intención y evento:

Definición 14 *Un agente es una tupla* $\langle E, B, P, I, A, \mathcal{S}_\mathcal{E}, \mathcal{S}_\mathcal{O}, \mathcal{S}_\mathcal{I} \rangle$ *donde*

- E *es un conjunto de eventos.*

- B *es un conjunto de creencias.*

- P *es un conjunto de planes.*

- A *es un conjunto de acciones.*

- $\mathcal{S}_\mathcal{E}$ *es una función de selección eventos.*

- $\mathcal{S}_\mathcal{O}$ *es una función de selección de planes aplicables.*

- $\mathcal{S}_\mathcal{I}$ *es una función de selección de intenciones.*

Definición 15 *(Intenciones) El conjunto I se compone de las intenciones del agente. Una intención es una pila de planes cerrados parcialmente (planes que pueden incluir algunas variables libres y otras con valores asignados). Una intención se denota por $[p_1 \ddagger \ldots \ddagger p_z]$, donde p_1 representa el fondo de la pila y p_z el tope de la misma. Por conveniencia, la intención $[+!true : true \leftarrow true]$ será denotada por $T = true$.*

Definición 16 *(Eventos) El conjunto E se compone de eventos. Cada evento es una tupla $\langle e, i \rangle$ donde e es un evento disparador e i es una intención. Si la intención $i = true$, al evento se le identifica como un evento externo; en cualquier otro caso es un evento interno.*

Ahora podemos definir formalmente los conceptos de plan relevante y aplicable. Para ello necesitamos recordar el concepto de unificador más general (MGU):

Definición 17 *(Unificador) Sean α y β términos. Una substitución θ tal que α y β sean idénticos ($\alpha\Theta = \beta\Theta$) es llamada unificador de α y β.*

Definición 18 *(Generalidad entre substituciones) Una substitución θ se dice más general que una substitución σ, si y sólo si existe una substitución γ tal que $\sigma = \theta\gamma$.*

Definición 19 (MGU) *Un unificador θ se dice el unificador más general (MGU) de dos términos, si y sólo si θ es más general que cualquier otro unificador entre esos términos.*

Definición 20 *(Plan relevante) Sea* $\mathcal{S}_{\mathcal{E}}(E) = \epsilon = \langle d, i \rangle$ *y sea el plan* $p = e : b_1 \wedge \cdots \wedge b_n \leftarrow h_1; \ldots; h_m$*. El plan p es relevante con respeto al evento* ϵ *si y sólo si existe un unificador más general (MGU)* σ *tal que* $d\sigma = e\sigma$*. A* σ *se le llama el unificador relevante para* ϵ*.*

Definición 21 *(Plan aplicable) Un plan p denotado por* $e : b_1 \wedge \cdots \wedge b_n \leftarrow h_1; \ldots; h_m$ *es un plan aplicable con respecto a un evento* ϵ *si y sólo si existe un unificador relevante* σ *para* ϵ *y existe una substitución* θ *tal que* $\forall(b_1 \wedge \cdots \wedge b_n)\sigma\theta$ *es una consecuencia lógica de B (creencias del agente). La composición* $\sigma\theta$ *se conoce como el unificador aplicable para* ϵ*; y* θ *se conoce como la substitución de respuesta correcta.*

Siguiendo con el ejemplo, si asumimos que las creencias básicas del agente son las siguientes:

```
adyacente(c1,c2).
adyacente(c2,c3).
adyacente(c3,c4).
pos(r1,c1).
pos(basura,c2).
pos(deposito,c4).
```

tenemos que el unificador aplicable es $\{X/c2, Y/c1, Z/c2\}$ y por lo tanto el único plan aplicable es $p2$. Dependiendo del tipo de evento (interno o externo) la intención será diferente. En el caso de los eventos externos, los medios se obtienen seleccionando un plan aplicable para el evento y entonces se aplica el unificador aplicable al cuerpo del plan. Este medio es utilizado para crear una nueva intención que se agrega a I. En el ejemplo, el plan $p2$ formará una nueva intención de la forma:

```
+!pos(r1,c2)
   : (pos(r1,c1) & ((not(c2=c1)) & adyacente(c1,c2)))
   <- ir(c1,c2);
   !pos(r1,c2).
```

Definición 22 *(Intención evento externo) Sea* $\mathcal{S}_{\mathcal{O}}(O_\epsilon) = p = e : b_1 \wedge \cdots \wedge b_n \leftarrow h_1; \ldots; h_m$ *donde* O_ϵ *es el conjunto de todos los planes aplicables u opciones para el evento* $\langle d, i \rangle$*. El plan p es intentado con respecto al evento* ϵ*, donde la intención* $i = T$*, si y sólo si existe un unificador aplicable* σ *tal que* $[+!true : true \leftarrow true \ddagger (e : b_1 \wedge \cdots \wedge b_n \leftarrow h_1; \ldots; h_m)\sigma] \in I$*.*

Definición 23 *(Intención evento interno) Sea* $\mathcal{S}_{\mathcal{O}}(O_\epsilon) = p = +!g(s) : b_1 \wedge \cdots \wedge b_j \leftarrow h_1; \ldots; h_k$ *donde* O_ϵ *es el conjunto de todos los planes aplicables u opciones para el evento* $\epsilon = \langle d, [p_1 \ddagger \ldots \ddagger f : c_1 \wedge \cdots \wedge c_m \leftarrow !g(t); k_2; \ldots; k_n] \rangle$*. El plan* p *es intentado con respecto al evento* ϵ*, si y sólo si existe un unificador aplicable* σ *tal que* $[p_1 \ddagger \ldots \ddagger f : c_1 \wedge \cdots \wedge c_m \leftarrow !g(t); k_2; \ldots; k_n \ddagger (+!g(s) : b_1 \wedge \cdots \wedge b_j)\sigma \leftarrow (h_1; \ldots; h_k)\sigma; (k_2; \ldots k_n)\sigma] \in I$.

Definición 24 *(Ejecución achieve) Sea* $\mathcal{S}_{\mathcal{I}}(I) = i = [p_1 \ddagger \ldots \ddagger f : c_1 \wedge \cdots \wedge c_m \leftarrow !g(t); h_2; \ldots; h_n]$*. Se dice que la intención* i *ha sido ejecutada, si y sólo si* $\langle +!g(t), i \rangle \in E$.

Definición 25 *(Ejecucin test) Sea* $\mathcal{S}_{\mathcal{I}}(I) = i = [p_1 \ddagger \ldots \ddagger f : c_1 \wedge \cdots \wedge c_m \leftarrow ?g(t); h_2; \ldots; h_n]$*. Se dice que la intención* i *ha sido ejecutada, si y sólo si existe una substitución* θ *tal que* $\forall g(t)\theta$ *es una consecuencia lógica de* B *e* i *es remplazada por* $[p_1 \ddagger \ldots \ddagger (f : c_1; \ldots; c_m)\sigma \leftarrow (h_2; \ldots h_n)\sigma]$.

Definición 26 *(Ejecución acción) Sea* $\mathcal{S}_{\mathcal{I}}(I) = i = [p_1 \ddagger \ldots \ddagger f : c_1 \wedge \cdots \wedge c_m \leftarrow a(t); h_2; \ldots; h_n]$*. Se dice que la intención* i *ha sido ejecutada, si y sólo si* $a(t) \in A$ *e* i *es remplazada por* $[p_1 \ddagger \ldots \ddagger f : c_1 \wedge \cdots \wedge c_m \leftarrow \ h_2; \ldots; h_n]$

Definición 27 *(Ejecución submeta) Sea* $\mathcal{S}_{\mathcal{I}}(I) = i = [p_1 \ddagger \ldots \ddagger p_{z-1} \ddagger !g(t) : c_1 \wedge \cdots \wedge c_m \leftarrow \ true]$*, donde* $p_{z-1} = e : b_1 \wedge \cdots \wedge b_x \leftarrow !g(s); h_2; \ldots; h_y$*. Se dice que la intención* i *ha sido ejecutada, si y sólo si existe una substitución* θ *tal que* $g(t)\theta = g(s)\theta$ *e* i *es remplazada por* $[p_1 \ddagger \ldots \ddagger p_{z-1} \ddagger (e : b_1 \wedge \cdots \wedge b_x)\theta \leftarrow \ h_2; \ldots; h_y)\theta]$.

Ahora es posible definir un intérprete para *AgentSpeak*(L) (Algoritmo 5). Las funciones *top*, *push*, *pop*, *head*, $first$ y *rest* tienen una semántica clara.

5.4. Teoría de prueba de *AgentSpeak*(L)

Para formular la teoría de prueba de *AgentSpeak*(L) recurrimos a un sistema de transición, como los propuestos por Plotkin [38].

Definición 28 *(Sistema transición BDI) Un sistema de transición BDI es un par* $\langle \Gamma, \vdash \rangle$ *que consiste en:*

- *Un conjunto* Γ *de configuraciones.*

Algoritmo 5 El algoritmo del intérprete *AgentSpeak*(L)

procedure AGENTSPEAK(L)()
  while $E \neq \emptyset$ **do**
    $\epsilon \leftarrow \langle d, i\rangle \leftarrow \mathcal{S}_{\mathcal{E}}(E)$;
    $E \leftarrow E \backslash \epsilon$;
    $O_\epsilon \leftarrow \{p\theta | \theta$ es un unificador aplicable para ϵ y $p\}$;
    if $externo(\epsilon)$ **then**
      $I \leftarrow I \cup [\mathcal{S}_{\mathcal{O}}(O_\epsilon)]$;
    else
      $push(\mathcal{S}_{\mathcal{O}}(O_\epsilon)\sigma, i)$ donde σ es un unificador aplicable para ϵ;
    end if
    if $first(body(top(\mathcal{S}_{\mathcal{I}}(I)))) = true$ **then**
      $x \leftarrow pop(\mathcal{S}_{\mathcal{I}}(I))$;
      $push(head(top(\mathcal{S}_{\mathcal{I}}(I)))\theta \leftarrow rest(body(top(\mathcal{S}_{\mathcal{I}}(I))))\theta, \mathcal{S}_{\mathcal{I}}(I))$
      donde θ es un mgu t.q. $x\theta = head(top(\mathcal{S}_{\mathcal{I}}(I)))\theta$;
    else if $first(body(top(\mathcal{S}_{\mathcal{I}}(I)))) = !g(t)$ **then**
      $E = E \cup \langle +!g(t), \mathcal{S}_{\mathcal{I}}(I)\rangle$
    else if $first(body(top(\mathcal{S}_{\mathcal{I}}(I)))) = ?g(t)$ **then**
      $pop(\mathcal{S}_{\mathcal{I}}(I))$;
      $push(head(top(\mathcal{S}_{\mathcal{I}}(I))))\theta \leftarrow rest(body(top(\mathcal{S}_{\mathcal{I}}(I))))\theta, \mathcal{S}_{\mathcal{I}}(I))$
      donde θ es la substitución de respuesta correcta.
    else if $first(body(top(\mathcal{S}_{\mathcal{I}}(I)))) = a(t)$ **then**
      $pop(\mathcal{S}_{\mathcal{I}}(I))$;
      $push(head(top(\mathcal{S}_{\mathcal{I}}(I)))) \leftarrow rest(body(top(\mathcal{S}_{\mathcal{I}}(I)))), \mathcal{S}_{\mathcal{I}}(I))$;
      $A = A \cup \{a(t)\}$;
    end if
  end while
end procedure

- *Una relación binaria de transición* $\vdash \subseteq \Gamma \times \Gamma$.

Definición 29 *(Configuración BDI) Una tupla* $\langle E_i, B_i, I_i, A_i, i\rangle$*, donde* $E_i \subseteq E$*,* $B_i \subseteq B$*,* $I_i \subseteq I$*,* $A_i \subseteq A$*, e* i *es la etiqueta de la transición es una configuración BDI.*

El conjunto de planes no forma parte de las configuraciones, pues se asume que permanece constante (aunque, como veremos, este no es el caso si el agente puede modificar sus planes originales, por ejemplo, mediante aprendizaje). Tampoco se lleva un registro explícito de las metas, pues se asume que estas aparecen como intenciones cuando son adoptadas por los agentes. Ahora es posible escribir reglas de transición que lleven al agente de una configuración BDI a otra.

La primera regla define la transición al intentar un plan al nivel más alto (un fin, en términos de razonamiento medios-fines). La regla especifica cómo el agente modifica sus intenciones en respuesta a un evento externo:

$$\textbf{(IntendEnd)}\frac{\langle\{\ldots,\langle+!g(t),T\rangle,\ldots\},B_iI_i,A_i,i\rangle}{\langle\{\ldots\},B_i,I_i\cup\{[p\sigma\theta]\},A_i,i+1\rangle}$$

donde: $p = +!g(s) : b_1\wedge\cdots\wedge b_m \leftarrow h_1;\ldots;h_n \in P$, $\mathcal{S}_{\mathcal{E}}(E) = \langle+!g(t),T\rangle$, $g(t)\sigma = g(s)\sigma$ y $\forall(b_1\wedge\cdots\wedge b_m)\theta$ es consecuencia lógica de B_i.

La regla para intentar un medio es similar a la regla para intentar un fin, sólo que el plan aplicable es colocado sobre la pila cuyo tope es la intención dada como segundo argumento del evento elegido:

$$\textbf{(IntendMeans)}\frac{\langle\{\ldots,\langle+!g(t),j\rangle,\ldots\},B_i\{\ldots,[p_1\ddagger\ \ldots\ \ddagger p_z],\ldots\},A_i,i\rangle}{\langle\{\ldots\},B_i,\{\ldots,[p_1\ddagger\ \ldots\ \ddagger p_z\ \ddagger p\sigma\theta],\ldots\},A_i,i+1\rangle}$$

donde $p_z = f : c_1\wedge\cdots\wedge c_y \leftarrow !g(t);h_2;\ldots;h_n$, $p = +!g(s) : b_1\wedge\cdots\wedge b_m \leftarrow k_1;\ldots;k : x$, $\mathcal{S}_{\mathcal{E}}(E) = \langle+!g(t),j\rangle$ es $[p_1\ddagger\ldots\ddagger p_n]$, $g(t)\sigma = g(s)\sigma$ y $\forall(c_1\wedge\cdots\wedge c_y)\theta$ es una consecuencia lógica de B_i.

Rao define una regla más para la adopción de metas y propone que el lector puede elaborar reglas parecidas para el resto de las transiciones en el sistema. De esta forma, es posible definir derivaciones y refutaciones, usando las reglas de prueba.

Definición 30 *(Derivación) Una derivación BDI es una secuencia finita o infinita de configuraciones* $\gamma_0,\ldots,\gamma_i,\ldots$.

La noción de refutación en *AgentSpeak*(L) se da con respecto a una intención particular. La refutación de una intención inicia cuando ésta es adoptada y termina cuando su pila queda vacía. Por lo tanto, usando las reglas anteriores es posible verificar seguridad y viabilidad del sistema. Además hay una correspondencia uno-a-uno entre las reglas de prueba y la semántica operacional del sistema. Dentro de las extensiones posibles se encuentran operadores más interesantes para el cuerpo de los planes (aquellos de la lógica dinámica) y post-condiciones diferenciadas para los casos de éxito y fracaso, como se especifica en dMARS [31].

5.5. *Jason*

Jason [34] es un intérprete que implementa una semántica operacional extendida de *AgentSpeak*(L). Fue desarrollado en el lenguaje de propósito general *Java*, y su IDE soporta el desarrollo y la ejecución de sistemas multi-agentes distribuidos. Algunas de sus características son:

- Comunicación entre agentes basada en actos de habla, incluyendo anotaciones en las creencias con información de las fuentes [35].

- Anotaciones sobre las etiquetas de los planes, las cuales pueden ser empleadas para diseñar funciones de selección basadas en teoría de decisión [5].

- La posibilidad de correr un sistema multi-agente distibuido sobre una red, utilizando SACI o algún *middleware* [29].

- Funciones de selección totalmente configurables mediante *Java*.

- La posibilidad de extender el repertorio de acciones internas, programándolas directamente en codigo *Java*.

- Una clara noción de medio ambiente, que permite simular la situacionalidad de los agentes en cualquier ambiente implementado en *Java*.

- Incorpora un editor gráfico, *jEdit*, que facilita el desarrollo de sistemas en *Jason*.

Existen diversos sistemas que implementan agentes basados en el modelo *BDI*; sin embargo, uno de los principales puntos a favor de *Jason* es el fundamento teórico que emplea. Con la implementación de *Jason* se busca probar ciertas características de los agentes *BDI* que sirvan para trabajar con la verificación formal de los agentes programados utilizando *AgentSpeak*(L). En la práctica, otra característica que tiene

el intérprete *Jason* a su favor, y que lo hace muy versátil, es el hecho de que está implementado en el lenguaje de propósito general *Java*, lo que le atribuye propiedades tales como ser ejecutado en diferentes plataformas y una fácil expansión. Es importante mencionar que esta plataforma y todos sus componentes están distribuidos bajo licencia libre, en GNU LGPL.

En estricta comparación, por ejemplo con *Jade*, tenemos que este último requiere ser programado directamente en *Java*, siendo más bien un paquete de clases y funciones de utilería para el desarrollo de sistemas basados en agentes BDI empleando el lenguaje abstracto $AgentSpeak(L)$, mientras que *Jason* es una plataforma completa que permite interpretar directamente dicho lenguaje abstracto. A diferencia de los diversos sistemas que implementan agentes BDI, *Jason* es un lenguaje de más alto nivel.

5.5.1. Sintaxis de *Jason*

El cuadro 5.1 muestra una gramática detallada de *Jason*. $\langle ATOM \rangle$ es un identificador que comienza con una letra minúscula o el caracter punto (.); $\langle VAR \rangle$ es un identificador que comienza con letra mayúscula; $\langle NUMERO \rangle$ es cualquier entero o número de punto flotante; y $\langle CADENA \rangle$ es una cadena de caracteres delimitada por comillas.

Algunas de las principales diferencias de la sintaxis de *Jason* con la propuesta originalmente por Rao, para $AgentSpeak(L)$ son:

- En lugar de átomos, *Jason* acepta literales.

- La sintaxis de *Jason*, a diferencia de $AgentSpeak(L)$, permite el uso de anotaciones para las literales. Así, por ejemplo, es posible indicar la fuente de las percepciones, etc. El costo es una unificación más complicada.

- Es posible etiquetar los planes empleando @ $\langle STRING \rangle$.

- Las anotaciones pueden ser usadas en la definición de funciones de selección más sofisticadas.

Existen otras extensiones prácticas, como por ejemplo, el operador + que agrega una creencia inmediatamente después de remover la primera ocurrencia existente de esa creencia en la base de creencias del agente.

$$
\begin{array}{rcl}
agente & \rightarrow & (creencias_0|metas_0)^*\ planes \\
creencias_0 & \rightarrow & creencias\ reglas \\
creencias & \rightarrow & (literal\ .)^* \\
reglas & \rightarrow & (literal\ :-\ exprLog\ .)^* \\
metas_0 & \rightarrow & (!\ literal\ .)^* \\
planes & \rightarrow & (plan)^* \\
plan & \rightarrow & [@atomo]\ eventoDisp\ [:\ contexto]\ [<-cuerpo]\ . \\
eventoDisp & \rightarrow & (+|-)[!|?]literal \\
literal & \rightarrow & [\ ]atomo \\
contexto & \rightarrow & exprLog|true \\
exprLog & \rightarrow & exprLogSimple \\
 & & |\ not\ exprLog \\
 & & |\ exprLog\ \&\ exprLog \\
 & & |\ exprLog\ |\ exprLog \\
 & & |\ (exprLog) \\
exprLogSimple & \rightarrow & (literal|relExpr|\langle VAR\rangle) \\
cuerpo & \rightarrow & fbfCuerpo(;\ fbfCuerpo)^* \\
 & & |\ true \\
bfbCuerpo & \rightarrow & (!|?|+|-|-+)\ literal \\
 & & |\ atomo \\
 & & |\ \langle VAR\rangle \\
 & & |\ exprLog \\
atomo & \rightarrow & (\langle ATOMO\rangle|\langle VAR\rangle)[(listaTerms)][[listaTerms]] \\
listaTerms & \rightarrow & termino(,\ termino)^* \\
termino & \rightarrow & literal \\
 & & |\ lista \\
 & & |\ exprAritm \\
 & & |\ \langle VAR\rangle \\
 & & |\ \langle CADENA\rangle \\
lista & \rightarrow & [[termino(,\ termino)^*[|\ (lista|\langle VAR\rangle)]] \\
exprRel & \rightarrow & termRel(<|<=|>|>=|==|\backslash\backslash==|=)termRel \\
exprTerm & \rightarrow & literal|exprAritm \\
exprAritm & \rightarrow & termAritm(+|-)termAritm \\
termAritm & \rightarrow & factorAritm(*|/|div|mod)factorAritm \\
factorAritm & \rightarrow & Aritm[++factorAritm] \\
Aritm & \rightarrow & \langle NUMERO\rangle \\
 & & |\ \langle VAR\rangle \\
 & & |\ -Aritm \\
 & & |\ (exprAritm)
\end{array}
$$

Cuadro 5.1: Sintaxis de *Jason*

5.5.2. Semántica de $Jason/AgentSpeak(L)$

La semántica operacional, también definida para $Jason$ [7], está dada por un sistema de transición entre configuraciones que están definidas por una tupla $\langle ag, C, M, T, s\rangle$ donde:

- ag es un programa agente formado por bs y ps.

- Una circunstancia C del agente es una tupla $\langle I, E, A\rangle$ donde I es el conjunto de intenciones $\{i, i', \ldots, n\}$ t.q. $i \in I$ es una pila de planes parcialmente instanciados $p \in ps$; E es un conjunto de eventos $\{\langle te, i\rangle, \langle te', i'\rangle, \ldots, n\}$, t.q. te es un $triggerEvent$ y cada i es una intención (evento interno) o una intención vacía $\top$ (evento externo); y A es un conjunto de acciones a efectuarse por el agente en el ambiente.

- M es una tupla $\langle In, Out, SI\rangle$ que trabaja como un $mailbox$, donde In es la bandeja de entrada del agente, Out es una lista de mensajes que el agente entregará y SI es un registro de las intenciones suspendidas (intenciones que esperan por un mensaje de respuesta).

- T es una tupla $\langle R, Ap, \iota, \epsilon, \rho\rangle$ que registra la información temporal: R es el conjunto de planes relevantes dado cierto $triggerEvent$; Ap es el conjunto de planes aplicables (el subconjunto de R t.q. $bs \models ctx$); ι, ϵ y ρ registran, respectivamente, la intención, el evento y plan actuales durante la ejecución del agente.

- La etiqueta s indica el paso actual en el ciclo del agente.

El intérprete para $AgentSpeak(L)$ se muestra como un sistema de transición en la figura 5.1. El estado inicial es $s = ProcMsg$. Las reglas de la semántica operacional [7] definen el sistema de transición.

En lo que sigue, adoptamos de $AgentSpeak(L)$ las definiciones de plan relevante, plan aplicable, unificador relevante, unificador aplicable, etc. Para efectos de describir las reglas semánticas para se adopta la siguiente notación:

- C denota una configuración de $Jason$; para hacer referencia al componente E (eventos) de C escribimos C_E. De manera similar accesamos a los demás componentes de C.

- Para indicar que no hay ninguna intención siendo considerada en la ejecución del agente, se emplea $C_I = \emptyset$. De forma similar para C_P y C_ϵ.

- Se usa $i[p]$ para denotar a la intención i que tiene al plan p como tope.

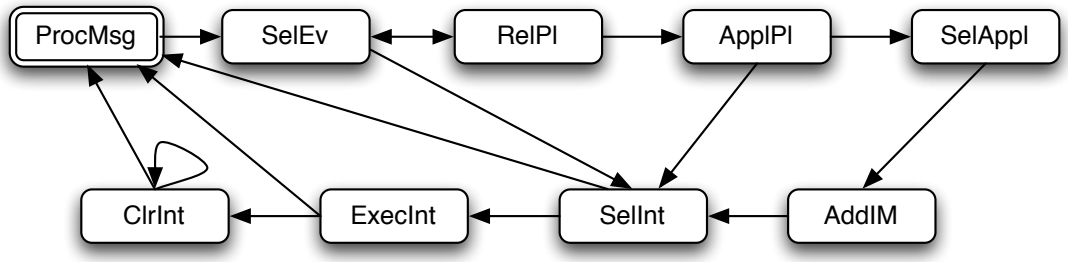

Figura 5.1: El intérprete de $AgentSpeak(L)$ como un sistema de transición.

Si asuminos que p es un plan de la forma $t : \phi \leftarrow h$, entonces: $TrEv(p) = t$ y $Ctxt(p) = \phi$. A continuación revisaremos las reglas de transición propuestas por A. Moreira y R. Bordini [35] en su extensión a $AgentSpeak(L)$. Primero tenemos algunas definiciones auxiliares:

Definición 31 (Planes relevantes) *El conjunto de planes relevantes con respecto a un evento disparador te está dado por:*

$$RelPlanes(ps, te) = \{p\theta | p \in ps \wedge \theta = mgu(te, TrEv(p))\}$$

Definición 32 (Planes aplicables) *Dado un conjunto R de planes relevantes, el conjunto de planes aplicables está dado por:*

$$AppPlanes(creencias, R) = \{p\theta | p \in R \wedge \theta t.q. creencias \models Ctxt(p)\theta\}$$

Definición 33 (Test) *Dadas las creencias de un agente y una fbf at el conjunto de substituciones para probar la fbf contra las creencias está dado por:*

$$Test(creencias, at) = \{\theta | creencias \models at\theta\}$$

Las funciones de selección de $AgentSpeak(L)$ son denotadas aquí por S_E, S_{Ap} y S_I. En estos términos, la selección de un evento se computa como:

$$\textbf{SelEnv} \quad \frac{S_E(C_E) = (te, i)}{C, creencias \rightarrow C', creencias} \quad C_\epsilon = \_, C_{Ap} = C_R = \emptyset$$
$$donde: \quad C'_E = C_E \backslash (te, i), C'_\epsilon = (te, i)$$

La regla Rel_1 asigna a R el conjunto de planes relevantes. Si no existe ningún plan relevante, el evento es descartado de ε por la regla Rel_2.

$$\textbf{Rel}_\textbf{1} \quad \frac{RelPlans(plans, te) \neq \emptyset}{C, creencias \rightarrow C', creencias} \quad C_\epsilon = (te, i), C_{Ap} = C_R = \emptyset$$
$$donde: \quad C'_R = RelPlans(plans, te)$$

$$\textbf{Rel}_\textbf{2} \quad \frac{RelPlans(plans, te) = \emptyset}{C, creencias \rightarrow C', creencias} \quad C_\epsilon = (te, i), C_{Ap} = C_R = \emptyset$$
$$donde: \quad C'_R = \emptyset$$

El caso de los planes aplicables es parecido:

$$\textbf{Appl}_\textbf{1} \quad \frac{AppPLans(C_R, creencias) \neq \emptyset}{C, creencias \rightarrow C', creencias} \quad C_\epsilon \neq \emptyset, C_{Ap} = \emptyset, C_R \neq \emptyset$$
$$donde: \quad C'_R = \emptyset$$
$$C'_{Ap} = AppPlanes(C_R, creencias)$$

$$\textbf{Appl}_\textbf{2} \quad \frac{AppPLans(C_R, creencias) = \emptyset}{C, creencias \rightarrow C', creencias} \quad C_\epsilon \neq \emptyset, C_{Ap} = \emptyset, C_R \neq \emptyset$$
$$donde: \quad C'_R = \emptyset, C'_\epsilon = \emptyset$$

La siguiente regla asume la existencia de una función de selección S_{Ap}, la cual selecciona un plan a partir del conjunto Ap de planes aplicables.

$$\textbf{SelAppl} \quad \frac{S_{Ap}(C_{Ap}) = p}{C, beliefs \rightarrow C', beliefs} \quad C_\epsilon \neq \emptyset, C_{Ap} \neq \emptyset$$
$$donde: \quad C'_\rho = p, C'_{Ap} = \emptyset$$

Recordemos que en *Jason* se distinguen dos tipos de eventos, internos y externos. La regla $ExtEv$ procesa los eventos externos:

$$\textbf{ExtEv} \quad \frac{}{C, creencias \rightarrow C', creeencias} \quad C_\epsilon = (te, T), C_p = p$$
$$donde: \quad C'_I = C_I \cup \{[p]\}$$
$$C'_\epsilon = \emptyset, C'_p = \emptyset$$

Si el evento es interno, la regla $IntEv$ lo procesará:

$$\textbf{IntEv} \quad \frac{}{C, creencias \rightarrow C', creencias} \quad C_\epsilon = (te, i), C_p = p$$
$$donde: \quad C'_I = C_I \cup \{i[p]\}$$
$$C'_\epsilon = \emptyset, C'_p = \emptyset$$

La regla para seleccionar una intención a ser ejecutada es como sigue:

$$\textbf{SelInt} \quad \frac{S_I(C_I) = i)}{C, creencias \rightarrow C', creencias} \quad C_\iota = \emptyset$$
$$donde: \quad C'_\iota = i$$

El grupo de reglas que describiremos a continuación, expresa el efecto de la ejecución de los planes. El plan siendo ejecutado es siempre aquel que se encuentra en el tope de la intención que ha sido previamente seleccionada. Todas las reglas en este grupo terminan descartando i, por lo que otra intención puede ser seleccionada eventualmente. Las reglas se ejecutan dependiendo del componente del cuerpo del plan que se ha seleccionado:

$$\textbf{Action} \quad \frac{}{C, creencias \rightarrow C', creencias} \quad C_\iota = i[head \leftarrow a; h]$$
$$donde: \quad C'_\iota = \_, C'_A = C_A \cup \{a\}$$
$$C'_I = (C_I \backslash \{C_\iota\}) \cup \{i[head \leftarrow h]\}$$

La siguiente regla registra una nueva meta de tipo *achieve*, la cual también podrá ser seleccionada dada la regla $SelEv$:

$$\textbf{Achieve} \quad \frac{}{C, creencias \rightarrow C', creencias} \quad C_\iota = i[head \leftarrow !at; h]$$
$$donde: \quad C'_\iota = \_, C'_E = C_E \cup \{(+!at, C_\iota)\}$$
$$C'_I = C_I \backslash \{C_\iota\}$$

Nótese que la intención que generó el evento interno es removida del conjunto de intenciones C_I. Esto implementa la suspención de una intención. Sólo cuando el curso de acción definida ha terminado se puede continuar con la ejecución de la intención que había sido suspendida, a partir de la siguiente fórmula del cuerpo de un plan dado.

Las metas de tipo *test*, se procesan mediante las siguientes dos reglas:

$$\textbf{Test}_\textbf{1} \quad \frac{Test(creencias, \phi) = \emptyset}{C, creencias \rightarrow C', creencias} \quad C_\iota = i[head \leftarrow ?at; h]$$
$$donde: \quad C'_\iota = \emptyset$$
$$C'_I = C_I\{C_\iota\} \cup \{i[head \leftarrow h]\}$$

$$\textbf{Test}_\textbf{2} \quad \frac{Test(creencias, \phi) \neq \emptyset}{C, creencias \rightarrow C', creencias} \quad C_\iota = i[head \leftarrow ?at; h]$$
$$donde: \quad C'_\iota = \emptyset, C'_I = C_I\{C_\iota\} \cup \{i[(head \leftarrow h)\theta]\}$$
$$theta \in Test(creencias, \phi)$$

Al igual que en dMARS [31], los agentes en *Jason* pueden agregar o eliminar creencias durante la ejecución de sus planes. Las siguientes reglas se encargan de ello:

$$\textbf{AddBel}\quad \frac{}{C, creencias \rightarrow C', creencias'}\quad C_\iota = i[head \leftarrow +at; h]$$
$$donde: \quad C'_\iota = \emptyset, creencias \models at$$
$$C_E \cup \{(+at, C_\iota)\}$$
$$C'_I = C_I\{C_\iota\} \cup \{i[head \leftarrow h']\}$$

$$\textbf{DelBel}\quad \frac{}{C, creencias \rightarrow C', creencias'}\quad C_\iota = i[head \leftarrow -at; h]$$
$$donde: \quad C'_\iota = \emptyset, creencias \not\models at$$
$$C_E \cup \{(-at, C_\iota)\}$$
$$C'_I = C_I\{C_\iota\} \cup \{i[head \leftarrow h']\}$$

Para concluir con la semántica operacional de *Jason* se definen dos reglas más, las llamadas *clearing house rules*. $ClearInt_1$ simplemente remueve una intención del conjunto de intenciones de un agente cuando no hay más que hacer al respecto, es decir, cuando ya no quedan más fórmulas (acciones o metas) a ejecutar dentro del cuerpo del plan.

$$\textbf{ClearInt}_\textbf{1}\quad \frac{}{C, creencias \rightarrow C', creencias'}\quad C_\iota = i[head \leftarrow]$$
$$donde: \quad C'_\iota = \emptyset, C'_I = C_I\{C_\iota\}$$

La segunda regla de limpieza procesa las intenciones que han sido suspendidas:

$$\textbf{ClearInt}_\textbf{2}\quad \frac{}{C, creencias \rightarrow C', creencias'}\quad C_\iota = i'[head' \leftarrow !at; h1 \ddagger head \leftarrow]$$
$$donde: \quad C'_\iota = \emptyset$$
$$C'_I = C_I\{C_\iota\} \cup \{i'[head' \leftarrow h']\}$$

5.6. Resumen

Hemos expuesto la sintaxis y la semántica del lenguage $AgentSpeak(L)$ junto su intérprete *Jason*. Como hemos visto, este lenguage ha sido propuestos y usado

para reducir la laguna entre la teoría (la especificación lógica BDI) y la práctica (la implementación). En este caso $AgentSpeak(L)$ tiene una semántica operacional bien definida, pero la verificación de propiedades racionales de los agentes programados no es evidente, pues dicha semántica excluyó, por mor de la eficiencia, el uso de las modalidades que hacen a las lógicas BDI lenguajes altamente expresivos.

Parte II

Resultados

Capítulo 6

$CTL_{AgentSpeak(L)}$

6.1. Introducción

Como hemos venido repitiendo, la teoría de razonamiento práctico propuesta por Bratman expone los fundamentos filosóficos del modelo BDI de agencia racional. Esta teoría es interesante e innovadora porque no reduce las intenciones a una combinación o suma de creencias y deseos, sino que asume que las intenciones son componentes propios de la planeación y que consisten en planes parciales jerárquicos. Tal suposición explica los aspectos temporales del razonamiento práctico en relación con las intenciones futuras, la persistencia de una intención, la reconsideración y el compromiso.

Diferentes lógicas han sido propuestas para caracterizar la conducta racional de los agentes. Las lógicas más utilizadas para hacer tal cosa son las lógicas BDI [42, 48, 51]. En estas lógicas la conducta de un agente es especificada en términos de los cambios temporales en las actitudes mentales del agente (e.d., en sus creencias, deseos e intenciones). Por ejemplo, es racional intentar deseos que se creen como posibles, y los cambios temporales en estas actitudes mentales definen cuándo es racional para un agente abandonar sus intenciones.

Las lógicas BDI capturan la racionalidad de los agente a través de axiomas y reglas de inferencia que capturan las propidedades racionales de los agentes. Entre tales propiedades encontramos las estrategias de compromiso, las cuales están definida mediante axiomas que usan operadores temporales y epistémicos, y dictan bajo qué condiciones un agente debería abandonar una intención. Así, estas lógicas son usadas para razonar acerca de los agentes, pero no para programarlos. Por otro lado, tenemos lenguages de programación, como $AgentSpeak(L)$ [41], que han sido propuestos y usados para reducir la laguna entre la teoría (la especificación lógica) y

la práctica (la implementación). En este caso, como vimos en el capítulo anterior, $AgentSpeak(L)$ tiene una semántica operacional bien definida, pero la verificación de propiedades racionales de los agentes programados no es evidente, pues dicha semántica excluyó, por mor de la eficiencia, el uso de las modalidades que hacen a las lógicas BDI lenguajes altamente expresivos.

Así que, para razonar acerca de tales propiedades, proponemos a $CTL_{AgentSpeak(L)}$ como una lógica para la especificación y verificación de agentes programados en $AgentSpeak(L)$. El enfoque es similar a la tradicional lógica BDI_{CTL} definida como el sistema modal B^{KD45} D^{KD} I^{KD}, con los operadores temporales: siguiente ($\bigcirc$), eventualmente ($\Diamond$), hasta (U), inevitable (A), etc., definidos de acuerdo a la lógica CTL

Nuestra principal contribución es la definición de la semántica de los operadores temporales CTL en términos de una estrutuctura de Kripke [30] inducida por el sistema de transición de la semántica operacional de $AgentSpeak(L)$. La semántica de los operadores intencionales es adoptada de [6]. Como resultado, la semántica de $CTL_{AgentSpeak(L)}$ está basada en la semántica operacional del lenguage de programación que usamos. De este modo podemos probar si cualquier agente programado en $AgentSpeak(L)$ satisface ciertas propiedades expresadas en la especificación lógica.

Es importante tener en cuenta que nuestro problema es diferente del *model checking* [13] en el siguiente sentido: en el *model checking* el problema consiste en verificar si cierta propiedad se cumple en cierto estado para cierto agente, mientras nuestro trabajo trata con la verficación de propiedades generales para cualquier agente.

Este capítulo está organizado del siguiente modo: la sección 2 presenta las estrategias de compromiso tal como y son especificadas en la lógica BDI. Las secciones 3 y 4 presentan, respectivamente, la sintaxis y la semántica de $AgentSpeak(L)$. La sección 5 muestra los resultados obtenidos, los cuales consisten, principalmente, en una conexión entre la especificación lógica y el lenguaje de programación. Finalmente, la sección 6 muestra un resumen y conclusiones.

6.2. Estrategias de compromiso

Diferentes teorías computacionales han sido propuestas para capturar las ideas de Bratman [8]. El trabajo fundacional de Cohen y Levesque [12], por ejemplo, definió las intenciones como una combinación de creencias y deseos basados en el concepto de meta persistente. Un análisis crítico de dicha teoría [46] mostró que la teoría de Cohen y Levesque no logra capturar ciertos aspectos importantes del compromiso. Alternativamente, el compromiso ha sido tratado como un proceso de mantenimiento y revisión de intenciones tomando en cuenta intenciones presentes y futuras.

Diferentes tipos de compromiso definen diferentes agentes. Tres estrategias han sido ampliamente estudiadas en el contexto de BDI_{CTL} [42], donde CTL [16] es la bien conocida lógica temporal:

- **Blind commitment.** Un agente que intenta que inevitablemente (A) eventualmente ($\Diamond$) es el caso que ϕ, inevitablemente mantiene su intención hasta (U) que realmente cree que ϕ:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ \mathsf{BEL}(\phi)) \tag{6.1}$$

- **Single-minded commitment.** Un agente mantiene su intención mientras crea que ha sido alcanzada o mientras sea opcionalmente (E) eventualmente alcanzable:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi) \vee \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi))) \tag{6.2}$$

- **Open-minded commitment.** Un agente mantiene su intención mientras haya sido lograda o mientras sea deseada:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi) \vee \neg\mathsf{DES}(\mathsf{A}\Diamond\phi))) \tag{6.3}$$

Estas tres estrategias definen, respectivamente, tres tipos de agentes: ciegos, flexibles y abiertos. Así, por ejemplo, un agente ciego que intenta eventualmente ir a París, mantendrá tal intención, para cualquier curso de acción, hasta que crea que está rumbo a París. Un agente flexible, por otro lado, puede abandonar su intención de ir a París si llega a creer que ya no es posible ir a París. Finalmente, un agente abierto puede abandonar la intención de ir a París si de pronto deja de desear ir a París. La pregunta que ha dirigido nuestra investigación es pues: qué estrategia de compromiso sigue un agente $AgentSpeak(L)$ cualquiera. Más aún, cómo podemos relacionar el compromiso con la reconsideración basada en políticas. Como revisamos en el capítulo 3, Bratman sugiere que hay tres tipos de reconsideración: no-reflexiva, deliberativa y la basada en políticas. Es claro, no obstante, que la primera tiene efectos cortos, la segunda es muy costosa, mientras la tercera implica un compromiso entre impacto y costo. Evidentemente, si un agente es ciego, no podemos hablar de reconsideración de ningún tipo. Pero si el agente es flexible o abierto, podemos entonces aproximar una reconsideración basada en políticas a través del aprendizaje intencional. De este modo podemos reconciliar aspectos relevantes del modelo de agencia BDI (compromiso en nuestro caso) con aspectos relevantes del modelo filosófico de Bratman (reconsideración).

ag	::=	$bs \quad ps$		at	::=	$P(t_1, \ldots, t_n)$ $(n \geq 0)$	
bs	::=	$b_1 \ldots b_n$	$(n \geq 0)$			$\mid P(t_1, \ldots, t_n)[s_1, \ldots, s_m]$ $(n \geq 0,$ $m \geq 0)$	
ps	::=	$p_1 \ldots p_n$	$(n \geq 1)$	s	::=	`percept` $\mid$ `self` $\mid id$	
p	::=	$te : ct \leftarrow h$		a	::=	$A(t_1, \ldots, t_n)$ $(n \geq 0)$	
te	::=	$+at \mid -at \mid +g \mid -g$		g	::=	$!at \mid ?at$	
ct	::=	$ct_1 \mid \top$		u	::=	$+b \mid -b$	
ct_1	::=	$at \mid \neg at \mid ct_1 \wedge ct_1$					
h	::=	$h_1; \top \mid \top$					
h_1	::=	$a \mid g \mid u \mid h_1; h_1$					

Cuadro 6.1: Sintaxis de *AgentSpeak*(L)

6.3. *AgentSpeak*(L)

6.3.1. Sintaxis de *AgentSpeak*(L)

La sintaxis de *AgentSpeak*(L) [41], tal y como es definida para su intérprete *Jason* [7], se muestra en el cuadro 6.1. Como es usual, un agente ag está formado por un conjunto de planes ps y creencias bs (*grounded literals*). Cada plan tiene la forma $triggerEvent : context \leftarrow body$. El contexto de un plan es un átomo, la negación de un átomo o la conjunción de ambos. El cuerpo de un plan no-vacío es una secuencia de acciones, metas (*achieve* o *test*), o actualizaciones de creencias (adición o borrado). $\top$ denota los elementos vacíos, los cuales pueden ser cuerpos, contextos o intenciones. Los eventos disparadores ($triggerEvent$)son actualizaciones de creencias o metas.

6.3.2. Semántica de *AgentSpeak*(L)

A continuación revisitamos la semántica operacional de *AgentSpeak*(L) para su intérprete *Jason* en términos de un sistema de transición, pero de un modo más compacto: mostrando sólo las reglas relevantes para los propósitos de este capítulo. Un sistema de transición es un conjunto de reglas de transformación que van de un estado a otro [38]. Cada regla de transformación tiene la forma:

$$\frac{cond}{C \rightarrow C'}$$

donde C es una configuración o estado que puede ser transformado al estado C' si la condición *cond* se cumple. Presentamos las reglas de *AgentSpeak*(L) y mencionamos su significado intuitivo. Asumimos las definiciones auxiliares de plan relevante y plan

aplicable, tal como quedaron definidas en el capítuo anterior. Asumimos también el sistema de transición de $AgentSpeak(L)$ (figura 6.1). En el cuadro 6.2 recordamos sólo las reglas relevantes para este capítulo. Así pues, aunque la semántica define

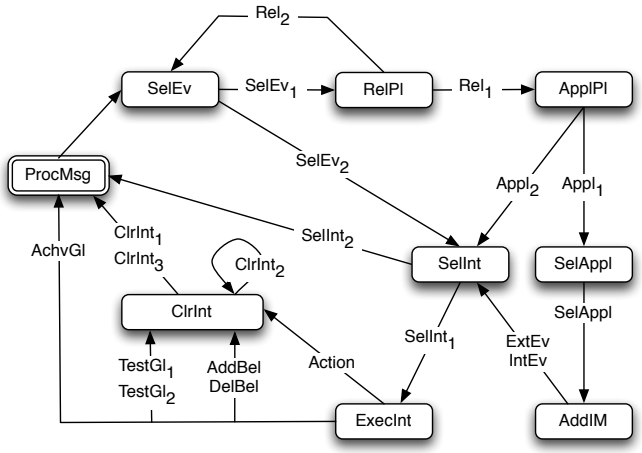

Figura 6.1: Sistema de transición $AgentSpeak(L)$.

claramente el razonamiento realizado por el agente, es difícil probar propiedades BDI, tales como las estrategias de compromiso, para cualquier agente. Esto, como hemos venido repitiendo, se debe al abandono de las modalidades temporales e intencionales en $AgentSpeak(L)$. No obstante, tal semántica operacional nos permite definir una ejecución en $AgentSpeak(L)$ del siguiente modo:

Definición 34 *(Ejecución) Una ejecución en AgentSpeak(L) es un conjunto*

$$P = \{(c_i, c_j) | \Gamma \vdash c_i \rightarrow c_j \wedge c_i, c_j \in C\}$$

donde Γ *es el sistema de transición definido por la semántica operacional y* C *es un conjunto de configuraciones de agente.*

como veremos, tal definición de ejecución nos permitirá inducir una estructura de Kripke.

6.4. $CTL_{AgentSpeak(L)}$

$CTL_{AgentSpeak(L)}$ pueder ser visto como una instancia de BDI_{CTL}. Aproximaciones similares han sido explorados para otros lenguages orientados a agentes, como la versión simplificada de $3APL$ [14]. La idea es definir la semántica de los operadores temporales e intencionales en términos de los operadores semánticos de $AgentSpeak(L)$.

Una vez hecho esto, podemos usar tal lógica para razonar acerca de las propiedades de los agentes $AgentSpeak(L)$.

Por tanto, lo que necesitamos es una lógica multimodal que use operadores temporales para lidiar con la ejecución del agente en el tiempo; y que use operadores epistémicos para especificar los estados mentales de un agente durante su ejecución. A continuación mostramos tal lógica.

6.4.1. Sintaxis de $CTL_{AgentSpeak(L)}$

Requerimos, pues, para nuestro lenguaje de especificación, fórmulas para operadores intencionales y temporales.

Definición 35 *(Sintaxis BDI) Si ϕ es una fórmula atómica $AgentSpeak(L)$, entonces $\mathsf{BEL}(\phi)$, $\mathsf{DES}(\phi)$ e $\mathsf{INTEND}(\phi)$ son fórmulas bien formadas de $CTL_{AgentSpeak(L)}$.*

La conducta de un agente es una estructura temporal sobre sus estados mentales. Para especificar tal conducta usamos la lógica $CTL*$ del siguiente modo:

Definición 36 *(Sintaxis temporal) Las fórmulas de estado y camino se definen, respectivamente, con las etiquetas s (por state) y p (por path):*

s1 *Toda fórmula en $AgentSpeak(L)$ es una fórmula de estado.*

s2 *Si ϕ y ψ son fórmulas de estado, $\phi \wedge \psi$ y $\neg\phi$ son fórmulas de estado.*

s3 *Si ϕ es una fórmula de camino, entonces $\mathsf{E}\phi$ y $\mathsf{A}\phi$ son fórmulas de estado.*

p1 *Toda fórmula de estado es una fórmula de camino.*

p2 *Si ϕ y ψ son fórmulas de camino, $\neg\phi$, $\phi \wedge \psi$, $\bigcirc\phi$, $\Diamond\phi$ y ϕ U ψ son fórmulas de camino.*

Así, por ejemplo,

$$\mathsf{INTEND}(\mathsf{A}\Diamond go(Paris))\ \mathsf{U}\ \neg\mathsf{BEL}(go(Paris, Summer))$$

es una fórmula bien formada de $CTL_{AgentSpeak(L)}$.

6.4.2. Semántica de $CTL_{AgentSpeak(L)}$

La semántica de los operadores BEL, DES e INTEND es adoptada de Bordini *et al* [6]. Pero antes necesitamos la ayuda de la función auxiliar $agoals(i)$ [6]:

$$\begin{array}{lcl} agoals(\top) & = & \{\}, \\ agoals(i[p]) & = & \begin{cases} \{at\} \cup agoals(i) & \text{si } p = +!at : ct \leftarrow h, \\ agoals(i) & \text{de otro modo} \end{cases} \end{array}$$

la cual regresa el conjunto de fórmulas atómicas (at) sujetas a una adición de una meta tipo *achieve* ($+!$) en los eventos disparadores de los planes que componen cierta intención.

Definición 37 *(Semántica de los operadores BDI) Los operadores* BEL, DES *e* INTEND *se definen en términos de un agente ag y su circunstancia C:*

$$\mathsf{BEL}(\phi) =_{def} \mathsf{BEL}_{\langle ag,C,M,T,s\rangle}(\phi) \equiv bs \models \phi$$

$$\mathsf{INTEND}(\phi) =_{def} \mathsf{INTEND}_{\langle ag,C,M,T,s\rangle}(\phi) \equiv \phi \in \bigcup_{i \in C_I} agoals(i) \vee \bigcup_{\langle te,i\rangle \in C_E} agoals(i)$$

$$\mathsf{DES}(\phi) =_{def} \mathsf{DES}_{\langle ag,C,M,T,s\rangle}(\phi) \equiv \langle +!\phi, i\rangle \in C_E \vee \mathsf{INTEND}(\phi)$$

Si un agente ag y su circunstancia C son explícitos, simplemente escribimos $\mathsf{BEL}(\phi)$, $\mathsf{DES}(\phi)$ e $\mathsf{INTEND}(\phi)$. De este modo, decimos que un agente cree la fórmula atómica ϕ si ϕ es consecuencia lógica de las creencias del agente. Decimos que un agente intenta ϕ si, o bien ϕ está sujeta a una meta de tipo *achieve* en el conjunto de intenciones activas (C_I) o bien ϕ pertenece al conjunto de intenciones suspedidas (C_E). Y decimos que un agente desea ϕ si esta fbf tiene asociada una intención en las intenciones suspendidas o el agente intenta ϕ.

La semántica de los operadores temporales requieren una estructura de Kripke $K = \langle S, R, V\rangle$ donde S es un conjunto de estados o configuraciones de agente, R es una relación de accesibilidad definida por el sistema de transición Γ y V es una función de valuación.

Definición 38 *Sea $K = \langle S, R, V\rangle$ una estructura de Kripke especificada por el sistema de transición Γ definido por la semántica operacional de $AgentSpeak(L)$:*

- *S es un conjunto de estados o configuraciones de agente $\langle ag, C, M, T, s\rangle$.*
- *$R \subseteq S^2$ es una relación serial t.q. para todo $(c_i, c_j) \in R$, $(c_i, c_j) \in \Gamma$ ó $c_i = c_j$*

- *V es un conjunto de valuaciones sobre los operadores intencionales y temporales:*

  . $V_{\mathsf{BEL}}(c, \phi) = \mathsf{BEL}(\phi)$ *donde* $c = \langle ag, C, M, T, s \rangle$.

  . $V_{\mathsf{DES}}(c, \phi) = \mathsf{DES}(\phi)$

  . $V_{\mathsf{INTEND}}(c, \phi) = \mathsf{INTEND}(\phi)$

- *Los caminos, como es usual, son secuencias de estados o configuraciones $c_0, \ldots, c_n$ t.q. para todo i, $(c_i, c_{i+1}) \in R$. Usamos x^i para indicar el i-ésimo estado del camino x. Entonces:*

S1 $K, c \models \mathsf{BEL}(\phi) \Leftrightarrow \phi \in V_{\mathsf{BEL}}(c)$

S2 $K, c \models \mathsf{DES}(\phi) \Leftrightarrow \phi \in V_{\mathsf{DES}}(c)$

S3 $K, c \models \mathsf{INTEND}(\phi) \Leftrightarrow \phi \in V_{\mathsf{INTEND}}(c)$

S4 $K, c \models \mathsf{E}\phi \Leftrightarrow \exists x = c_1, \ldots \in K | K, x \models \phi$

S5 $K, c \models \mathsf{A}\phi \Leftrightarrow \forall x = c_1, \ldots \in K | K, x \models \phi$

P1 $K, c \models \phi \Leftrightarrow K, x^0 \models \phi$ *donde* ϕ *es una fórmula de estado.*

P2 $K, c \models \bigcirc\phi \Leftrightarrow K, x^1 \models \phi$.

P3 $K, c \models \Diamond\phi \Leftrightarrow K, x^n \models \phi$ *para* $n \geq 0$

P4 $K, c \models \phi \ \mathsf{U} \ \psi \Leftrightarrow \exists k \geq 0$ *t.q.* $K, x^k \models \psi$ *y para todo* j, k, $0 \leq j < k | K, c^j \models \phi$ *ó* $\forall j \geq 0 : K, x^j \models \phi$

Es importante notar que la semántica de *until* corresponde a *weak until* (e.d., ψ puede nunca ocurrir).
Así pues, por ejemplo, la fórmula bien formada

$$\mathsf{INTEND}(\mathsf{A}\Diamond go(Paris)) \ \mathsf{U} \ \neg\mathsf{BEL}(go(Paris, Summer))$$

expresa que un agente intentará inevitablemente, e.d., para todo curso de acción, eventualmente ir a París en verano hasta que no crea que va a París en verano.

En la definición previa las fórmulas de estado son evaluadas en el modelo de Kripke K con la configuración c. Como es nuestro interés expresar que ciertas propiedades definidas en el lenguaje de especificación son satisfechas por cualquier agente programado en $AgentSpeak(L)$ necesitamos definir una noción de satisfacción:

Definición 39 *(Satisfacción) Una fórmula ϕ es satisfecha en el modelo K si y sólo si ϕ es una consecuencia lógica de todas las configuraciones c en K. Es decir, $K \models \phi \Leftrightarrow K, c \models \phi$ para todo $c \in C$.*

Definición 40 *(Ejecución de un agente en un modelo) Dada una configuración inicial β, un sistema de transición Γ y una valuación V, $K_\Gamma^\beta = \left\langle S_\Gamma^\beta, R_\Gamma^\beta, V \right\rangle$ es una ejecución de un agente en un modelo.*

Como puede notarse, el modelo es definido en términos de las definiciones previas y la relación serial de accesibilidad está dada por el sistema de transición del ciclo de $AgentSpeak(L)$.

Definición 41 *(Modelo agente) Dado un agente y una ejecución P, el modelo agente es definido como $K_\Gamma^\beta = \left\langle S_\Gamma^\beta, R_\Gamma^\beta, V \right\rangle$ donde:*

- $R_\Gamma^\beta = P \cup \{(c_{n-1}, c_n) | \exists (c_{n-1}, c_n) \in P \wedge \neg \exists (c_n, c_{n+1}) \in P\}$
- $S_\Gamma^\beta = \left\{(c, c') | (c, c') \in R_\Gamma^\beta\right\}$

Dado que nuestro interés es hallar que una propiedad general se cumple para cualquier agente, es necesario extender la noción de modelo para todos los programas agentes:

Definición 42 *(Validez) Una propiedad $\phi \in CTL_{AgentSpeak(L)}$ se cumple para cualquier ejecución de agente Ag en Γ si y sólo si $\phi \in K_\Gamma^\beta$ para todos los modelos agentes. Es decir, $\models_\Gamma \phi \Leftrightarrow \forall \beta, K_\Gamma^\beta \models \phi$*

6.5. Resultados sobre compromiso

Dada la semántica operacional de $AgentSpeak(L)$ y la especificación lógica del lenguaje $CTL_{AgentSpeak(L)}$, procedemos a mostrar algunas propiedades que expresadas en la especificación lógica se cumplen para cualquier agente programado en $AgentSpeak(L)$. Hemos decidido probar las estrategias de compromiso ciego y flexible. Pero primero verificamos el axioma de no-compromiso infinito o no-deliberación infinita (*no-infinite deferral*).

Proposición 1 *Los agentes AgentSpeak(L) satisfacen el axioma de no-compromiso infinito:* $\mathsf{INTEND}(\phi) \Rightarrow \mathsf{A}\Diamond(\neg\mathsf{INTEND}(\phi))$.

Prueba. El axioma de *no-infinite deferral* dice que si un plan p con contexto $+!\phi$ es adoptado para formar una intención, entonces es eventualmente removido de C_I (intenciones activas) y C_E (intenciones suspendidas). Asumamos que $\mathsf{INTEND}(\phi)$, lo cual implica por definición que $\phi \in C_E \vee \phi \in C_I$. Entonces tenemos que probar que $\forall x = c_1 \ldots . \in K, x \models \Diamond \neg \mathsf{INTEND}(\phi)$. Dado que los planes son finitos por definición, y dado el sistema de transición de $AgentSpeak(L)$, $K, x^{n \geq 0} \models \neg \mathsf{INTEND}(\phi)$ implica que tenemos tres alternativas: *i)* una ejecución exitosa del plan nos llevaría a $ClrInt_2$, y por tanto la intención i será removida de $i\,[p] \in C_I$; *ii)* una ejecución exitosa nos llevaría a una aplicación de $ClrInt_1$, y por tanto, la intención será removida; *iii)* la ejecución de una intención continua, y cuando cuerpo del plan en el tope de la intención se vacía, $ClrInt_3$ se aplica y entonces el ciclo agente comienza de nuevo. Si algo sale mal, un mecanismo de falla es activado por un evento de la forma $\langle -!\phi, i\,[p] \rangle$. Por tanto, o bien la ejecución del plan es exitosa o bien no lo es, se sigue que, inevitablemente, eventualmente la intención es abandonada. Aunque en [7] sólo se formaliza fallos al encontrar planes relevantes (Rel_1), lo cual descarta las intenciones en C_E, otras formas de detección de fallas han sido consideradas en el contexto del aprendizaje intencional [24, 25]. Por una ejecución exitosa o fallida de los planes cada intención adoptada es eventualmente abandonada. □

Proposición 2 *Los agentes AgentSpeak(L) no satisfacen el axioma de compromiso ciego (blind commitment).*

Prueba. El axioma de *blind commitment*, dado el axioma de *no-infinite deferral*, expresado en BDI_{CTL} es como sigue:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \Rightarrow \mathsf{A}\Diamond\mathsf{BEL}(\phi)$$

Lo traducimos a $CTL_{AgentSpeak(L)}$ en el siguiente modo:

$$\models_\Gamma \mathsf{INTEND}(\mathsf{A}\Diamond\phi) \Rightarrow \mathsf{A}\Diamond\mathsf{BEL}(\phi)$$

Por tanto, tenemos que derivar $K^\beta_\Gamma, c \models \mathsf{INTEND}(\mathsf{A}\Diamond\phi) \Rightarrow K^\beta_\Gamma, c \models \mathsf{A}\Diamond\mathsf{BEL}(\phi)$. Asumimos $K^\beta_\Gamma, c \models \mathsf{INTEND}(\mathsf{A}\Diamond(\phi))$. Por tanto tenemos que probar que $\forall x = c_1, \ldots \in K^\beta_\Gamma : K^\beta_\Gamma, x \models \Diamond\mathsf{BEL}(\phi)$. Tenemos que probar entonces que, para todo camino y para algún $n \geq 0$, $K^\beta_\Gamma : K, x^n \models \mathsf{BEL}(\phi)$.
Sea Γ el sistema de transición definido por la semántica operacional de $AgentSpeak(L)$. Por tanto tenemos que hallar $K, x^{n \geq 0} \models_\Gamma \mathsf{BEL}(\phi)$. Sin embargo, si encontramos que $\neg(K, x^{n \geq 0} \models_\Gamma \mathsf{BEL}(\phi))$, habremos mostrado, por la existencia de un contra-ejemplo, que el axioma de *blind commitment* no se satiface.
Sea $\beta = \langle ag, C, M, T, s \rangle$ con

$$ag = \langle bs = \{\}, ps = \{+b(t_1) : \top \leftarrow p(t_2).+!p(t_2) : \top + b(t_3).\} \rangle$$

Asúmase que $b(t_1)$ se añade a ag_{bs}. Por Γ, un evento $C_E = \{+b(t_1), \top\}$ es generado. Entonces la ejecución P inicia con las reglas $SelEv_1$, Rel_1, $ApplPl_1$ y entonces obtenemos una nueva configuración $C_I = \{+b(t_1) : \top \leftarrow !p(t_2)\}$ and $C_E = \{\}$. Entonces, siguiendo el sistema de transición, aplicamos $SelAppl$, $ExtEv$, $SelInt_1$, $AchvGl$ y obtenemos la configuración $C_E = \langle +!p(t_2), +b(t_1) : \top \leftarrow \top\rangle$, $C_I = \{\}$. Al aplicar de nuevo $SelEv_1$, Rel_1, $AppPl_1$, $SelAppl$ obtenemos una nueva configuración donde $C_I = \{+!p(t_2 : \top \leftarrow +b(t_3))\}$ y $C_E = \{\}$, es decir, que el agente tratará $\mathsf{INTEND}(p(t_2))$. Y entonces, siguiendo la ejecución, aplicamos $IntEv$, $SelInt_1$, $Addbel$ y por tanto $C_E = \langle +b(t_3)\,[self]\,, \top\rangle$, $ag_{bs} = \{b(t_1)\}$ y $C_I = \{+b(t_1) : \top \rightarrow \top; +!p(t_2) : \top \leftarrow \top\}$ y $ag_{bs} = \{b(t_1), b(t_3)\}$. Por tanto, la intención $p(t_2)$ se mantiene, pero dado que los cuerpos de los planes son vacíos (e.d., $\top$), las reglas $ClrInt$ eventualmente descartarán la intención completa y por ende es falso que para todo camino y para todo estado $x^{n\geq 0}$ a partir de β y Γ, $\mathsf{BEL}(\phi)$ sea derivado, e.d., $\neg(K, x^{n\geq 0} \models_\Gamma \mathsf{BEL}(\phi))$. Por lo tanto, $\models_\Gamma \mathsf{INTEND}(\mathsf{A}\Diamond\phi) \Rightarrow \mathsf{A}\Diamond\mathsf{BEL}(\phi)$ no es satisfecha. □

Proposición 3 *Los agentes AgentSpeak(L) satisfacen una forma limitada de compromiso flexible (single-minded):* $\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi))$.

Prueba. Tenemos que mostrar:

$$\models_\Gamma \mathsf{INTEND}(\mathsf{A}\Diamond\phi) \Rightarrow \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \cup (\mathsf{BEL}\phi \vee \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)))$$

es decir, $K^\beta_\Gamma, c \models \mathsf{INTEND}(\mathsf{A}\Diamond\phi) \Rightarrow K^\beta_\Gamma, c \models \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \cup (\mathsf{BEL}(\phi) \vee \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)))$.
Asumimos $K^\beta_\Gamma, c \models \mathsf{INTEND}(\mathsf{A}\Diamond\phi)$. Tenemos que mostrar, entonces, que $K^\beta_\Gamma, c \models \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \cup (\mathsf{BEL}(\phi) \vee \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)))$, e.d., que $\forall x = c_1, \ldots \in K^\beta_\Gamma : K^\beta_\Gamma, x \models \mathsf{INTEND}(\Diamond\phi) \cup (\mathsf{BEL}(\phi) \vee \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi))$.
Primero probamos que $\mathsf{INTEND}(\Diamond\phi)$ se cumple hasta $(\mathsf{BEL}(\phi) \vee \neg\mathsf{BEL}(\mathsf{E}\ x^{n\geq 0}\phi))$. Tenemos, por la definición de U, dos alternativas:

$$K^\beta_\Gamma, x^i \models \mathsf{INTEND}(x^{n\geq 0}\phi)$$

ó

$$K^\beta_\Gamma, x^i \models (\mathsf{BEL}(\phi) \vee \neg\mathsf{BEL}(\mathsf{E}\ x^{n\geq 0}\phi))$$

Si se da lo primero, entonces $\forall j, K^\beta_\Gamma, x^j \models \mathsf{INTEND}(x^{n\geq 0}\phi)$, que es la definición de *weak until*. Si lo segundo es el caso, entonces, para algún x^k, tenemos que $K^\beta_\Gamma, x^k \models (\mathsf{BEL}(\phi) \vee \neg\mathsf{BEL}(\mathsf{E}\ x^{n\geq 0}\phi))$, y para todo $0 \leq j < k, K^\beta_\Gamma, x^j \models \mathsf{INTEND}(x^{n\geq 0}\phi)$, que es de nuevo la defínción de *weak until*.
Ahora, para probar que $K^\beta_\Gamma, x^i \models (\mathsf{BEL}(\phi) \vee \neg\mathsf{BEL}(\mathsf{E}\ x^{n\geq 0}\phi))$. Hemos asumido que $K^\beta_\Gamma \models \mathsf{INTEND}(\phi)$. Dada la suposición, tenemos que mostrar que en la siguiente configuración x^{i+1} las propiedades deseadas se dan. Sabemos que $\Gamma \vdash x^i \rightarrow x^{i+1}$

ó $x^i = x^{i+1}$. Si lo segundo es el caso, la prueba termina aquí. Si lo primero es el caso, tenemos entonces $\mathsf{BEL}(\phi)$ ó $\neg(\mathsf{BEL}(\exists x = c_1, \ldots \in K | K_\Gamma^\beta \models x^{n \geq 0} \phi))$. Dado que asumimos $\mathsf{INTEND}(\mathsf{A}\Diamond\phi)$, existe un plan $p \in C_I \vee C_E$ con cabeza $+!\phi$ en c_1. Si existe una configuración $c_k \geq 1$ donde $\neg\mathsf{BEL}\Diamond\phi$ (se asume la definición de *weak until*), entonces $K, x^{0,\ldots,k} \models \mathsf{INTEND}(\mathsf{A}\Diamond\phi)$. Y siguiendo la demostración de *no-infinite-deferral*, en los casos de fallo el agente eventualmente satisfacerá $\bigcirc(\mathsf{INTEND}\phi)$ dado Rel_2; esto implica que para un evento de la forma $\langle te, i\,[+!\phi : c \leftarrow h\rangle]$ no hay planes relevantes y la intención i asociada será descartada, es decir, no existe un camino donde eventualmente se cumpla ϕ; razón por la cual es racional abandonar $\mathsf{INTEND}(\phi)$. Finalmente, el caso de ejecución exitosa cubre la segunda condición del *weak until* tal como en la demostración de *no-infinite deferral*. □

Esta es una forma limitada de compromiso flexible porque $\neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)$ no es representada explícitamente por el agente. De hecho, el agente no puede continuar intentando ϕ porque no hay planes para hacerlo y la intención completa falla. También es limitada debido a la incompletud intención-creencia que puede ser evitada al abandonar la premisa de *close world assumption* [6]. El aprendizaje intencional provee un enfoque alternativo para alcanzar una estrategia flexibe completa, que pueda representar las razones de fallo para usarlas en un modo menos reactivo, e.d., como un abandono de intenciones previsorio cercano a los postulados de Bratman.

6.6. Resumen

Las propiedades mostradas aquí no son propiedades arbitrarias. Al contrario, son propiedades que han sido consideradas como fundamentales en la teoría de agencia racional. Al formalizar y verificar tales propiedades en un lenguaje $CTL_{AgentSpeak(L)}$ hemos probado que tales propiedades se cumplen para cualquier agente programado en $AgentSpeak(L)$. Esto es relevante porque acerca a $AgentSpeak(L)$ a sus fundamentos filosóficos mientras ofrece un lenguaje para expresar y verificar sus propiedades.

También hemos extendido la metodología propuesta por Bordini *et al.* [6] para razonar acerca de los agentes $AgentSpeak(L)$. Luego probamos que cualquier agente $AgentSpeak(L)$ no sigue una estrategica de compromiso de tipo *blind*, sino que sigue una forma limitada de *single-minded commitment*. Las principales limitaciones de estos agentes son la incompletitud intención-creencia y la falta de una representación explícita de las razones de abandono. Guerra *et al* [25] ha argumentado que el aprendizaje intencional provee una solución para el problema de la representación de polítca de abandono. Trataremos esto en el siguiente capítulo.

(SelEv$_1$)	$\frac{\mathcal{S}_{\mathcal{E}}(C_E)=\langle te,i\rangle}{\langle ag,C,M,T,SelEv\rangle \longrightarrow \langle ag,C',M,T',RelPl\rangle}$	t.q.	$C'_E = C_E \setminus \{\langle te,i\rangle\}$ $T'_\epsilon = \langle te,i\rangle$
(Rel$_1$)	$\frac{T_\epsilon=\langle te,i\rangle, RelPlans(ag_{ps},te)\neq\{\}}{\langle ag,C,M,T,RelPl\rangle \longrightarrow \langle ag,C,M,T',AppPl\rangle}$	t.q.	$T'_R = RelPlans(ag_{ps},te)$
(Rel$_2$)	$\frac{RelPlans(ps,te)=\{\}}{\langle ag,C,M,T,RelPl\rangle \longrightarrow \langle ag,C,M,T,SelEv\rangle}$		
(Appl$_1$)	$\frac{ApplPlans(ag_{bs},T_R)\neq\{\}}{\langle ag,C,M,T,ApplPl\rangle \longrightarrow \langle ag,C,M,T',SelAppl\rangle}$	t.q.	$T'_{Ap} = AppPlans(ag_{bs},T_R)$
(SelAppl)	$\frac{\mathcal{S}_{\mathcal{O}}(T_{Ap})=(p,\theta)}{\langle ag,C,M,T,SelAppl\rangle \longrightarrow \langle ag,C,M,T',AddIM\rangle}$	t.q.	$T'_\rho = (p,\theta)$
(ExtEv)	$\frac{T_\epsilon=\langle te,\top\rangle, T_\rho=(p,\theta)}{\langle ag,C,M,T,AddIM\rangle \longrightarrow \langle ag,C',M,T,SelInt\rangle}$	t.q.	$C'_I = C_I \cup \{[p\theta]\}$
(SelInt$_1$)	$\frac{C_I\neq\{\},\mathcal{S}_{\mathcal{I}}(C_I)=i}{\langle ag,C,M,T,SelInt\rangle \longrightarrow \langle ag,C,M,T',ExecInt\rangle}$	t.q.	$T'_\iota = i$
(SelInt$_2$)	$\frac{C_I=\{\ \}}{\langle ag,C,M,T,SelInt\rangle \longrightarrow \langle ag,C,M,T,ProcMsg\rangle}$		
(AchvGl)	$\frac{T_\iota=i[head\leftarrow !at;h]}{\langle ag,C,M,T,ExecInt\rangle \longrightarrow \langle ag,C',M,T,ProcMsg\rangle}$	t.q.	$C'_E = C_E \cup \{\langle +!at,T_\iota\rangle\}$ $C'_I = C_I \setminus \{T_\iota\}$
(ClrInt$_1$)	$\frac{T_\iota=[head\leftarrow\top]}{\langle ag,C,M,T,ClrInt\rangle \longrightarrow \langle ag,C',M,T,ProcMsg\rangle}$	t.q.	$C'_I = C_I \setminus \{T_\iota\}$
(ClrInt$_2$)	$\frac{T_\iota=i[head\leftarrow\top]}{\langle ag,C,M,T,ClrInt\rangle \longrightarrow \langle ag,C',M,T,ClrInt\rangle}$	t.q.	$C'_I = (C_I \setminus \{T_\iota\}) \cup$ $\{k[(head' \leftarrow h)\theta]\}$ si $i = k[head' \leftarrow g;h]$ y $g\theta = TrEv(head)$
(ClrInt$_3$)	$\frac{T_\iota\neq[head\leftarrow\top]\wedge T_\iota\neq i[head\leftarrow\top]}{\langle ag,C,M,T,ClrInt\rangle \longrightarrow \langle ag,C,M,T,ProcMsg\rangle}$		

Cuadro 6.2: Reglas de la semántica operacional de $AgentSpeak(L)$.

Capítulo 7

Compromiso, reconsideración y aprendizaje

7.1. Introducción

Podemos reconciliar el concepto de razonamiento práctico con el concepto de reconsideración basada en políticas en la estrategia de compromiso flexible, la cual, a su vez, está modelada dentro del alcance de $AgentSpeak(L)$. Como decíamos en el capítulo 3, Bratman distingue tres tipos de reconsideración: la no-reflexiva (basada en hábitos o habilidades); la deliberativa (basada en razones creencia-deseo); y la basada en políticas: este tercer tipo de reconsideración es menos demandante que la deliberativa y más precisa que la no-reflexiva. Dado que reconsiderar una intención implica entretener la posibilidad de que dicha intención se asuma, la reconsideración y el compromiso están ligados. Y este hecho, como vimos, está modelado en las estrategias de compromiso.

Nuestro interés es relacionar el compromiso flexible limitado de $AgentSpeak(L)$ y la reconsideración basada en políticas, según Bratman, mediante el aprendizaje intencional, de tal modo que se logre un compromiso flexible completo.

7.2. Un marco para experimentar con reconsideración y compromiso

Para lograr tal cosa requerimos de un marco experimental para relacionar la reconsideración y el compromiso. Para ello supóngase que tenemos un agente situa-

do en un mundo de bloques que percibe el estado a mostrado en la figura 7.1. Y desea alcanzar el estado b. De este modo, el agente forma una intención de la forma $\langle +!on(b,c), \top \rangle$ usando un plan relevante y aplicable. Asúmase que dicho plan tiene la forma $+!on(X,Y) :\leftarrow .take(X); .put(X,Y)$. Esto significa que el agente es ingenuo (*bold* o *naive*) para apilar objetos: cree que X puede apilarse sobre Y en cualquier circunstacia.

Ahora, supóngase que después de formar dicho plan el agente percibe el estado c. Entonces la acción interna $.put(X = a, Y = b)$ fallará, por tanto la intención eventualmente fallará y la meta $on(b,c)$ no será lograda. Un agente con aprendizaje intencional puede hallar que el contexto correcto del plan fallido es $clear(Y) \wedge handfree(Ag)$ y modificar su plan de acuerdo a tal contexto. Y así, la próxima vez que procesa un evento $\langle +!on(b,c), \top \rangle$ en un estado similar a c, será el caso que $\neg(bs \models clear(c))$, y por tanto el plan ya no será aplicable. Este agente con aprendizaje intencional tiene un enfoque mejor que el agente ingenuo, pero no resuelve del todo el problema.

Ahora bien, el considerar también el fallo del plan para añadir creencias co-

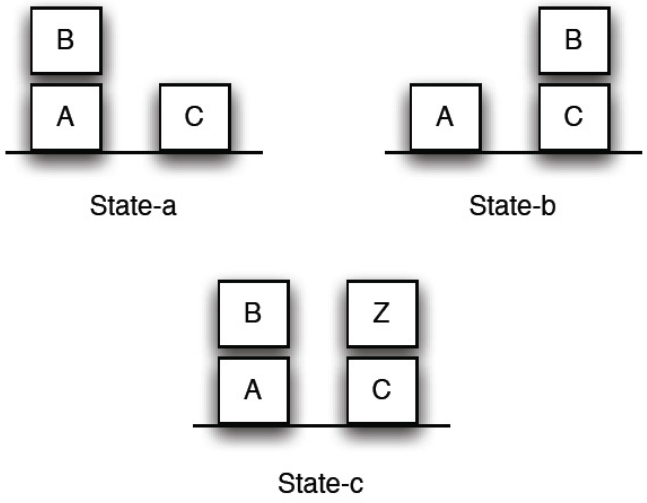

Figura 7.1: Mundo de bloques.

mo $abandon(p) \leftarrow not(clear(Y))$ permite un abandono previsorio de las intenciones a través de un mecanismo de limpiado. De este modo, si tal creencia aparece en el estado mental del agente mientras el plan p está siendo intentado, el agente tiene que abandonar dicha intención (y posiblemente volver a colocar los eventos $!clear(c)$ y $!on(b,c)$). Si esto no sucede, el agente puede continuar como un agente $AgentSpeak(L)$ normal. Esta estrategia corresponde a lo que hemos venido llamando estrategia flexible.

Se sabe que en ambientes dinámicos un agente cauteloso o flexible tiene mayor

eficiencia que un agente audaz o ingenuo e, inversamente, en ambientes estáticos un agente audaz tiene una eficiencia mayor que un agente flexible [32]. Sin embargo, el grado de cautela o audacia es algo difícil de definir *a priori*, por lo que la relevancia de aprender intencionalmente se muestra al proveer un mecanismo automático para definir tales grados, de tal suerte que el aprendizaje intencional provee adaptabilidad.

7.3. Aprendizaje

Como hemos mostrado, los agentes $AgentSpeak(L)$ no son agentes ciegos, es decir, no siguen una estrategia de compromiso de tipo *blind*. Antes bien, siguen una estrategia de tipo flexible; sin embargo, tal estrategia flexible es limitada. El aprendizaje intencional provee un enfoque alternativo para alcanzar un compromiso de tipo flexible completo, esto es, una forma de compromiso *single-minded* sin la limitación que los agentes $AgentSpeak(L)$ tienen (e.d., la falta de representación de razones de abandono)).

Los agentes con aprendizaje intencional usan una programación lógica inductiva para aprender sus razones prácticas para adoptar un plan como una intención. El contexto de los planes es el núcleo de este aprendizaje. Los ejemplos de entrenamiento están formados para cierto plan dado, registrando lo que el agente cree cuando el plan es adoptado como una intención, y etiqueta los ejemplos con la salida de la ejecución de la intención: éxito o fallo. Entonces, cuando cierta intención falla, una inducción basada en árboles de decisión es ejecutada para aprender un nuevo contexto en función de las ramas del árbol de decisión que llevan al éxito. La idea central es que es posible aprender, de modo similar, las razones de abandono de una intención: la política de la reconsideración en función de las ramas del árbol de decisión que llevan al fracaso o fallo. Así, en lugar de modificar el contexto del plan que está en reconsideración, los agentes flexibles añaden reglas (lo cual es posible en $Jason$) a las creencias del agente. Creencias de la forma:

$$abandon(I) : -at_1; \ldots; at_n$$

donde I es el átomo a ser reconsiderado, y la secuencia $at_1; \ldots; at_n$ es una rama del árbol de decisión que lleva a un fallo. Todo agente flexible tiene un plan en su librería para abandonar intenciones:

```
+abandon(I) : not I
<- -intending(I); .drop_intention(I).
```

El contexto del plan es *not* I para evitar el abandono de planes cuando otros agentes satisfacen I, cuando el ambiente lo satisface o cuando el propio agente lo satisface.

La creencia $inteding(I)$ es verdadera mientras el agente tiene I en el conjunto de intenciones. Cuando un agente verifica la secuencia $at_1; \ldots; at_n$, un evento de la forma $+abandon(I)$ es generado y el plan se convierte en un plan relevante y posiblemente aplicable.

7.4. Experimentación

Un experimento ha sido diseñado para probar este enfoque: el enfoque de compromiso flexible como un caso de reconsideración basada en políticas. El experimento ha sido implementado en *Jason*. Hay cuatro agentes en el mundo de los bloques:

```
MAS policy-based-rec {
    infrastructure: Centralised
       environment: Blocks
    agents:
       bold;
       learner agentClass Learner;
       single-minded agentClass SingleMinded;
       experimenter; }
```

El agente *experimenter* propone a los otros agentes (*bold*, *learner* y *single-minded*) la tarea de poner un bloque b sobre un bloque c y, con cierta probabilidad, introduce ruido en el experimento al colocar un bloque z sobre c antes de su pedido. Después recolecta la información sobre la eficiencia de los otros agentes para cierto número de iteraciones. Los otros agentes son, inicialmente audaces (*bold*), en el sentido de que los contextos de sus planes son vacíos (e.d., $\top$). Esto es, todos ellos comparten el plan:

```
+!on(X,Y) <- put(X,Y).
```

El agente audaz (*bold*) no puede aprender intencionalmente: es un agente por *default* de $AgentSpeak(L)$. El agente con aprendizaje intencional (*learner*) es capaz de aprender el contexto de sus planes. El agente flexible (*single-minded*) puede aprender tanto el contexto de los planes como razones de abandono. El agente con aprendizaje y el agente flexible pertenecen a una diferente clase de agentes debido a que usan acciones primitivas para lograr sus acciones de aprendizaje, y generan eventos especializados para usar su conocimiento generado por aprendizaje.

Tenemos entonces dos situaciones: la intención adoptada para colocar b sobre c

falla o tiene éxito. El agente con aprendizaje intencional modifica el contexto de su plan del siguiente modo:

```
+!on(X,Y) : clear(Y) <- put(X,Y).
```

Mientras que el agente flexible genera la regla:

```
abandon(on(X,Y)) :-
intending(on(X,Y)) & not clear(Y).
```

El ruido puede aparecer antes, después o durante la adopción de la intención. Esto depende de la organización de *Java* con respecto a los procesos de *Jason*. Aún cuando es posible controlar el momento donde el ruido aparece, es importante notar que los agentes en condiciones normales carecen de dicho control. Algunas veces el ruido aparece después de la ejecución de la intención, de tal suerte que el experimentador falla (esa es la razón por la cual el número de iteraciones no siempre es igual a la suma de fallos y éxitos en el cuadro 7.1).

Los resultados relevantes son los siguientes:

- El agente audaz (*bold*) siempre falla cuando hay ruido, ya que no puede aprender nada acerca de la adopción o abandono de las intenciones.

- El agente con aprendizaje intencional (*learner*), por otro lado, reduce el número de fallos debidos al ruido, puesto que aprende cierto contexto: en este caso, poner X sobre Y requiere que Y esté libre. Y cuando esto no es el caso, el plan deja de ser aplicable.

- El agente flexible (*single-minded*) reduce drásticamente el número de fallos al prevenir la adopción inconveniente de ciertos planes al abandonar intenciones cuando es necesario: cuando el agente adopta cierta intención pero el agente experimentador coloca un bloque z sobre c. En efecto, el agente flexible sólo falla cuando está listo para ejecutar su acción *put* y el ruido aparece.

Para los casos exitosos, hemos considerado la conducta previsora del agente como un éxito. Así, si el agente se rehusa a adoptar cierto plan, se considera como un caso exitoso. Usualmente se espera que el agente tenga diferentes planes para cierto evento. El rechazo de un plan resultaría en la generación de un plan diferente a ser adoptado para solventar el problema: un caso de verdadera reconsideración. Abandonar un plan también se considera como un caso de éxito: significa que el agente usó su reconsideración basada en políticas para prevenir un fallo real. En la práctica, esto resulta

en la eliminación del evento asociado a la intención perteneciente a los eventos del agente.

Claramente, tanto el agente audaz como el agente con aprendizaje intencional no pueden abandonar intenciones. Por tanto, como se esperaba, el agente flexible es más exitoso que el agente con aprendizaje intencional cuando hay tazas altas de ruido; mientras que estos dos agentes son más exitosos que un agente audaz.

Cuadro 7.1: Resultados experimentales

Agente	**Iters**	**Ruido**	**Fallo**				**Éxito**			
			antes	**desp**	**acción**	**total**	**no plan**	**abandon**	**éxito**	**total**
Bold	1,000	90 %	735	160	0	895	0	0	101	101
Learner	1,000	90 %	0	69	0	69	724	0	188	912
Single-minded	1,000	90 %	0	1	5	6	242	381	356	982
Bold	1,000	50 %	417	69	0	486	0	0	502	502
Learner	1,000	50 %	0	54	0	54	355	0	576	931
Single-minded	1,000	50 %	0	1	3	4	116	158	715	989
Bold	1,000	30 %	249	52	0	301	0	0	624	624
Learner	1,000	30 %	0	26	0	26	131	0	837	968
Single-minded	1,000	30 %	0	0	4	4	72	76	838	986
Bold	1,000	10 %	90	23	0	113	0	0	797	797
Learner	1,000	10 %	0	19	0	19	26	0	951	977
Single-minded	1,000	10 %	0	0	2	2	24	21	950	995
Bold	100	90 %	67	23	0	90	0	0	10	10
Learner	100	90 %	0	5	0	5	47	0	48	95
Single-minded	100	90 %	0	2	0	2	12	25	60	97
Bold	100	50 %	34	11	0	45	0	0	47	47
Learner	100	50 %	0	10	0	10	41	0	49	90
Single-minded	100	50 %	0	2	0	2	20	18	59	97
Bold	100	30 %	19	10	0	29	0	0	64	64
Learner	100	30 %	0	4	0	4	21	0	75	96
Single-minded	100	30 %	0	1	0	1	9	8	82	99
Bold	100	10 %	4	2	0	6	0	0	81	81
Learner	100	10 %	0	0	0	0	0	0	100	100
Single-minded	100	10 %	0	0	0	0	0	0	98	98

El cuadro 7.1 muestra el resultado de dos corridas de diferente tamaño: una con 1000 iteraciones y otra con 100. La dinámica del ambiente depende de la probablidad de ocurrencia de ruido (90 %, 50 % y30 %). Los valores bajos inducen ambientes estáticos. Luego tenemos dos situaciones: la intención adoptada para colocar b sobre c o bien falla o bien tiene éxito.

7.5. Resumen

Hemos discutido las bondades del aprendizaje intencional en relación con el compromiso y la reconsideración a través de la introducción del aprendizaje intencional

como una forma de aproximar una estrategia completa de compromiso flexible. Asimismo, hemos experimentado con el uso del aprendizaje intencional como una aproximación a la formación de políticas de abandono intencional en $AgentSpeak(L)$, y con buenos resultados. De este modo, el aprendizaje intencional provee un enfoque alternativo para alcanzar un compromiso flexible completo.

Capítulo 8

Conclusiones

8.1. Resumen final

Hemos relacionado el compromiso flexible con el concepto de reconsideración basada en políticas propuesta por Bratman. El aspecto clave de este proceso radica en la semántica de $CTL_{AgentSpeak(L)}$, la cual está basada en la semántica operacional del lenguage de programación $AgentSpeak(L)$. Usando este lenguaje hemos probado que los agentes programados en $AgentSpeak(L)$ exhiben una forma limitada de compromiso flexible (porque no representan explícitamente las razones de abandono de una intención). Entonces se propuso el uso de aprendizaje intencional para alcanzar un compromiso flexible completo. Una forma de aprendizaje intencional basada en el protocolo Smile ha sido formalizado como un conjunto de reglas de semántica operacional para $AgentSpeak(L)$ [26].

Se sabe que en ambientes dinámicos un agente cauteloso o flexible tiene mayor eficiencia que un agente audaz o ingenuo e, inversamente, en ambientes estáticos un agente audaz tiene una eficiencia mayor que un agente flexible [32]. Sin embargo, el grado de cautela o audacia es algo difícil de definir *a priori*, por lo que la relevancia de aprender intencionalmente se muestra al proveer de un mecanismo automático para definir tales grados, de tal suerte que el aprendizaje intencional provee adaptabilidad.

Finalmente, recordemos que una reconsideración, según Bratman, requiere dos aspectos descriptivos:

- **Espera.** Una reconsideración implica poner en espera a la intención.
  - . Ocurren procesos de cambios de razones (*reason-changing*).
  - . Ocurren procesos de preservación de razones (*reason-preserving*).

- **Costo.** Una reconsideración implica una revisión de los planes mismos, y esto implica ciertos costos de acuerdo a la jerarquía de los planes y a su alcance en razonamientos futuros.

La estrategia de compromiso flexible completa, aproximada gracias al aprendizaje intencional, apela a una regla general sobre cuándo reconsiderar y cuándo no reconsiderar, pues es mediante éste que obtenemos dos tipos de ramas en el árbol de decisión inductivo:

- Casos de éxito: procesos de preservación de razones.

- Casos de fallo: procesos de cambios de razones.

Y con ello, el costo de la reconsideración implica una revisión de los planes mismos, lo que conlleva a una modificación de la jerarquía de los planes y a su alcance en razonamientos futuros.

Normativamente, la cuestión de la reconsideración tiene que ver con cuándo es racional para un agente reconsiderar una intención. Sabemos que una vez que se forma un plan, este es estable, pero no irrevocable. Entonces tenemos dos casos en los cuales es racional reconsiderar:

. **Problemas para los planes.** Un problema para un plan consiste en un problema de tipo interno: *i)* el mundo esperado puede ser diferente al mundo actual; *ii)* los deseos pueden cambiar y *iii)* las propias intenciones pueden cambiar.

. **Reconsideración ocasional.** Si hay oportunidad y recursos suficientes para llevar a cabo una reconsideración, es razonable reconsiderar la intención.

La estrategia de compromiso flexible completa permite la reconsideración, como vimos con el experimento, en el primer sentido. De este modo, la reconsideración es una reconsideración basada en políticas.

8.2. Objetivos

A lo largo del trabajo hemos observado nuestros objetivos:

- Mostramos los conceptos básicos que fundamentaron y soportaron este trabajo:

  . El concepto de agencia: capítulo 2.

. El modelo formal BDI: capítulo 4.

. El lenguage $AgentSpeak(L)$ junto con su intérprete $Jason$: capítulo 5.

- Revisamos los fundamentos filosóficos de los conceptos de compromiso y política de abandono en las teorías computacionales de razonamiento práctico para $AgentSpeak(L)$: capítulo 3.

- Propusimos la lógica $CTL_{AgentSpeak(L)}$ como un lenguage formal para la especificación y verificación de agentes programados en $AgentSpeak(L)$:

  . Formalizamos los conceptos de compromiso de intenciones en una teoría basada en $AgentSpeak(L)$ a través de $CTL_{AgentSpeak(L)}$: capítulo 6.

  . Verificamos formalmente las propiedades sobre compromiso que cumplen los agentes $AgentSpeak(L)$ a través de $CTL_{AgentSpeak(L)}$: capítulo 6.

- Discutimos las bondades del aprendizaje intencional en relación con el compromiso y la reconsideración:

  . Introducimos el aprendizaje intencional como una forma de aproximar una estrategia de compromiso flexible: capítulo 7.

  . Experimentamos con el uso del aprendizaje intencional como una aproximación a la formación de políticas de abandono intencional en $AgentSpeak(L)$: capítulo 7.

Así pues, nuestro trabajo se ha perfilado como una triangulación entre fundamentos filosóficos, métodos formales y lenguajes de progr

amación para agencia racional.

8.3. Trabajo futuro

Dentro del trabajo futuro tenemos la tarea definir una teoría computacional completa de reconsideración en $AgentSpeak(L)$ partiendo de los postulados de Bratman. También, aunque el trabajo de esta investigación ha sido en torno a sistemas monoagente, hay una dimensión social para ser explorada.

Por parte de la filosofía y los métodos formales tenemos la tarea de mostrar las propiedades metalógicas de $CTL_{AgentSpeak(L)}$. Así como investigar las relaciones entre el formalismo $CTL_{AgentSpeak(L)}$ y las lógicas no-clásicas relevantes.

Las lógicas relevantes tradicionales dependen del principio de que las proposiciones deben compartir variables (*variable sharing principle*). Este principio es condición

necesaria de una sistema lógico para ser un sistema relevante; sin embargo, no es una condición suficiente. Más aún, este principio no provee un criterio que elimine todas las paradojas de la implicación y las falacias de relevancia: existen paradojas y falacias que satisfacen el principio en cuestion. Ahora bien, las lógicas relevantes proveen una noción de derivación relevante que garantiza una definición de prueba relevante; sin embargo, como es usual, no nos dice qué cuenta como una verdadera y relevante implicación a menos que una semántica sea adoptada. La semántica de $AgentSpeak(L)$ provee de una definición de plan relevante y plan aplicable que se puede traducir a una semántica relevante.

Bibliografía

[1] N. Alechina, R. H. Bordini, J. F. Hübner, M. Jago, and B. Logan.*Belief Revision for AgentSpeak Agents. AAMAS06* May 812 2006, Hakodate, Hokkaido, Japan.

[2] Aristóteles. *Acerca del alma*, III, 5, 430a, Gredos, Madrid, 1978.

[3] J. L. Austin. *How to Do Things With Words*. Oxford University Press. Oxford, England, 1962.

[4] J. Bates. *The role of emotion in believable agents. Communications of the ACM*, 37(7):122-125, 1994.

[5] R. H. Bordini, A. L. C. Bazzan, R. O. Jannone, D. M. Basso, R. M. Vicari, and V. R. Lesser. *AgentSpeak(XL): Efficient intention selection in BDI agents via decision-theoretic task scheduling*. In C. Castelfranchi and W. L. Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2002)*, 15-19 July, Bologna, Italy, pages 1294-1302, New York, NY, 2002. ACM Press.

[6] R. H. Bordini and Á. F. Moreira. *Proving BDI properties of agent-oriented programming languages. Annals of Mathematics and Artificial Intelligence*, 42:197-226, 2004.

[7] R. H. Bordini, J. F. Hübner, and M. Wooldridge. *Programming Multi-Agent Systems in AgentSpeak using Jason*. Wiley, England, 2007.

[8] M. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA., 1987.

[9] M. Bratman, M. E. Pollak, and Israel D. J. *Plans and resource-bounded practical reasoning. Computer Intelligence*, 4:349 355, 1988.

[10] F. Brentano *Psychology from an Empirical Standpoint* transl. by A. C. Rancurello, D. B. Terrell, and L. McAlister, London: Routledge, 1973. (2nd ed., intr. by Peter Simons, 1995).

[11] R. A. Brooks. *Cambrian Intelligence: the Early History of the New AI*. MIT Press, Cambridge, MA., USA, 1999.

[12] P. Cohen and H. Levesque. *Intention is choice with commitment. Artificial Intelligence*, 42(3):213-261, 1990.

[13] E. M. Jr. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, Boston, MA., USA, 1999.

[14] M. Dastani, M. Birna van Riemsdijk, J. J. Ch. Meyer. *A grounded specification for angent programs. AAMAS07*, May 14 - 18, Honolulu, Hawaii, USA, 2007.

[15] D. Dennett. *The Intentional Stance*. MIT Pres, Cambridge, Massachusetts, 1987.

[16] A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 996–1072. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands, Amsterdam, Holland, 1990.

[17] O. Etzioni. *Intelligence without robots. AI Magazine*, 14(4), 1993.

[18] J. Ferber. *Les Systemes Multi-Agents: vers une intelligence collective*. InterEditions, Paris, France, 1995.

[19] J. Ferrater Mora. *Diccionario de filosofía*, Ariel, Espaa, 2001.

[20] J. R. Galliers. *A Theoretical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi-Agent Conflict*. PhD thesis, Open University, UK, 1998.

[21] M. R. Genesereth, N. J. Nilsson. *Logical Foundations for Artificial Intelligence*. Morgan Kauffman Publishers, Inc., Palo Alto, CA., USA, 1987.

[22] M. R. Genesereth, S. P. Ketchpel. *Software agents. Communications of the ACM*, 37 (7):48-53, 1994.

[23] R. Goodwin. *Formalizing properties of agents*. Technical Report CMU-CS-93-159, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1993.

[24] A. Guerra-Hernández, A. El-Fallah-Seghrouchni, and H. Soldano. *Learning in BDI Multi-agent Systems*. In J. Dix and J. Leite, editors, *CLIMA IV, Revised and Selected Papers*, LNCS 3259:218–233. Springer, 2004.

[25] A. Guerra-Hernández and G. Ortíz-Hernández. *Toward BDI sapient agents: Learning intentionally*. In R. V. Mayorga and L. I. Perlovsky, editors, *Toward Artificial Sapience: Principles and Methods for Wise Systems*, pages 77–91. Springer, London, 2008.

[26] A. Guerra-Hernández, G. Ortíz-Hernández, and W. A. Luna-Ramírez. *Jason smiles: Incremental BDI MAS learning*. MICAI 2007. Special Session, IEEE CSP (In press).

[27] A. Guerra-Hernández, J. M. Castro-Manzano, A. El-Fallah-Seghrouchni, *Toward an AgentSpeak(L) theory of commitment and intentional learning*. In: Gelbuch, A., Morales, E.F. (eds.), *MICAI 2008*. LNCS, vol. 5317, pp. 848-858, Springer-Verlag, Berlin Heidelberg (2008)

[28] A. Haddadi. *Communication and Cooperation in Agent Systems : a Pragmatic Theory*. Number 1056 in Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin-Heidelberg, Germany, 1995.

[29] J. F. Hübner. *Um Modelo de Reorganizacão de Sistemas Multiagentes*. PhD thesis, Universidad de de São Paulo, Escola Politecnica, 2003.

[30] G. E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic*. Routledge, London, England, 1998 edition, 1996.

[31] M. d'Inverno et al. *A formal specification of dMARS*. In M. P Singh, A. S. Rao, and M. Wooldridge, editors, *Intelligent Agents IV: Proceedings of the Fourth International Workshop on Agent Theories, Architectures and Languages*, LNAI 1365:155–176, Springer Verlag, 1997.

[32] D. Kinny, M. Georgef. *Commitment and effectiveness of situated agents.Proceedings of the twelfth international joint conference on artificial intelligence* (IJCAI-91), Sydney, Australia, 1991.

[33] J. McCarthy. *Ascribing mental qualities to machines*. Technical report, Computer Science Department, Stanford University, Stanford, CA., USA, 1979.

[34] Á. F. Moreira, R. Bordini. *An operational semantics for a bdi agent-oriented programming language*. In *Proceedings of the Workshop on Logics for Agent-Based Systems* (LABS-2002) held with KR2000, April 22-25, Toulouse, France, pages 45-59, 2002.

[35] Á. F. Moreira, R. Vieira, and R. H. Bordini. *Extending the operational semantics of a BDI agent-oriented programming language for introducing speech-act based communication*. In J. Leite, A. Omicini, L. Sterling, and P. Torroni, editors, *Declarative Agent Languages and Technologies, Proceedings of the First International Workshop (DALT-03)*, held with AAMAS-03, 15 July, 2003, Melbourne, Australia (Revised Selected and Invited Papers), LNAI 2990:135-154,Springer-Verlag, 2004.

[36] Moro Simpson, T. *Semántica y filosofía: Problemas y discusiones*, Eudeba, Buenos Aires, Argentina, 1964.

[37] N. Nilsson. *Inteligencia artificial. Una nueva síntesis.* McGraw Hill, México, 2006.

[38] G. Plotkin. *A structural approach to operational semantics*. Laboratory for Foundations of Computer Science, School of Informatics, University of Edinburgh, King's Buildings, Edinburgh EH9 3JZ, Scotland, 2004.

[39] A. S. Rao and M. P. Georgeff. *Modelling Rational Agents within a BDI-Architecture*. In M. N. Huhns and M. P. Singh, editors, *Readings in Agents*, pages 317–328 Morgan Kaufmann Publishers, 1998.

[40] A. S. Rao and M. P. Georgeff. *Asymmetry thesis and side-effect problems in lineartime and branching-time intention logics.* Technical Note 13, Australian Artificial Intelligence Institute, Carlton, Victoria, April 1991. published in proceedings of IJCAI-91.

[41] A. S. Rao. *AgentSpeak(L): BDI agents speak out in a logical computable language*. In W. V. de Velde and J. W. Perram, editors, *MAAMAW*, LNCS 1038:42–55. Springer, 1996.

[42] A. S. Rao and M. P. Georgeff. *Decision procedures for BDI logics. Journal of Logic and Computation*, 8(3):293–342, 1998.

[43] J. S. Rosenschein, M. R. Genesereth. *Deals among rational agents*. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (IJCAI-85), pages 91-99, Los Angeles, CA, 1985.

[44] S. J. Russell, P. Norvig. *Artificial Intelligence, a modern approach*. Prentice Hall, New Jersey, USA, 1995.

[45] J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press: Cambridge, England, 1969.

[46] M. Singh. *A critical examination of the Cohen-Levesque Theory of Intentions*. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, 1992.

[47] M. Singh. *Multiagent Systems: A theoretical framework for intentions, know how, and communication*. Number 799 in *Lecture Notes in Computer Sciences*. Springer Verlag, Berlin-Heidelberg, Germany, 1995.

[48] M. Singh, A. Rao, and M. Georgeff. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter *Formal Methods in DAI: Logic-Based Representation and Reasoning*, pages 331–376. MIT Press, Cambridge, MA., USA, 1999.

[49] Y. Shoham. *Agent-oriented programming*. Technical Report STAN - CS - 1335 - 90, Computer Science Department, Stanford University, Stanford, CA., USA, 1990.

[50] J. E. White. Telescript technology: The foundation for the electronic marketplace. White paper, General Magic, Inc., 2465 Latham Street, Mountain View, CA 94040, 1994.

[51] M. Wooldridge. *Reasoning about Rational Agents*. MIT Press, Cambridge, MA., USA, 2000.

[52] M. Wooldridge. *Introduction to Multiagent Systems*, John Wiley and Sons,Ltd., 2001.

[53] M. Wooldridge, W. van der Hoek, W. Jamroga. *Towards a theory of intention revision*. Springer-Verlag, 2007.

Parte III

Apéndice

Los siguientes artículos fueron aceptados para presentación oral y publicación, respectivamente, en la séptima edición del MICAI (MICAI 08) y en la cuarta del LANMR (LANMR 08). El *Mexican International Conference on Artificial Intelligence* (MICAI por sus siglas en inglés) es una conferencia internacional de alto nivel y arbitrada que cubre todas las áreas de la IA, y es tradicionalmente llevada a cabo en México. Este año se realizó del 27 al 31 de octubre en la ciudad de México, en el ITESM Campus Estado de México y tuvo una taza de aceptación de 27 %. El MICAI es organizado por la SMIA (Sociedad Mexicana de Inteligencia Artificial).

El *Latin American Workshop on Logic/Languages, Algorithms, Non-Monotonic Reasoning* (LANMR) es un taller arbitrado para concentrar investigadores y estudiantes interesados en áreas de las ciencias computacionales. Este año se realizó del 22 al 24 de octubre en la ciudad de Puebla, México, en la Facultad de Ciencias de la Computación de la BUAP.

Las referencias bibliográficas son respectivamente:

- A. Guerra-Hernández, J. M. Castro-Manzano, A. El-Fallah-Seghrouchni, *Toward an AgentSpeak(L) theory of commitment and intentional learning*. In: Gelbuch, A., Morales, E.F. (eds.), *MICAI 2008*. LNCS, vol. 5317, pp. 848-858, Springer-Verlag, Berlin Heidelberg (2008)

- A. Guerra-Hernández, J. M. Castro-Manzano, A. El-Fallah-Seghrouchni, *$CTL_{AgentSpeak(L)}$: a Specification Language for Agent Programs*. In: Osorio, M., Olmos, I. (eds.), *Proceedings of the fourth Latin American Workshop on Non-Monotonic Reasoning 2008 (LANMR'08)*, vol. 408, CEUR Workshop Proceedings, (2008).

Toward an AgentSpeak(L) theory of commitment and intentional learning

Alejandro Guerra-Hernández[1], José Martín Castro-Manzano[1], and Amal El-Fallah-Seghrouchni[2]

[1] Departamento de Inteligencia Artificial
Universidad Veracruzana
Facultad de Física e Inteligencia Artificial
Sebastián Camacho No. 5, Xalapa, Ver., México, 91000
aguerra@uv.mx, e_f_s_s@hotmail.com
[2] Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie
Avenue du Président Kennedy, Paris, France, 75016
Amal.Elfalah@lip6.fr

Abstract. This work is about the commitment strategies used by rational agents programmed in *AgentSpeak*(L) and the relationship between single-minded commitment and intentional learning. Although agent oriented languages were proposed to reduce the gap between theory and practice of Multi-Agent Systems, it has been difficult to prove BDI properties of the agents programmed in such languages. For this reason, we introduce some ideas to reason temporally about the intentional state of rational agents in order to prove what kind of commitment strategy is used by *AgentSpeak*(L) agents, based on the operational semantics of the programming language. This enables us to prove that any agent programmed in this language follows by default a limited form of single-minded commitment. Then we analyze how intentional learning can enhance this commitment strategy allowing preemptive abandon of intentions.

1 Introduction

The philosophical foundation for intentional agency is provided by the theory of practical reasoning proposed by Bratman [3]. This theory is innovative because it does not reduce intentions to some combination of beliefs and desires, but indeed it assumes that they are composed by hierarchical, partial plans. Such assumption explains better temporal aspects of practical reasoning as future intentions, persistence, reconsideration and commitment. Three kinds of reconsideration (and nonreconsideration) are identified by Bratman: nondeliberative, based on habits; deliberative, based on belief-desire reasons; and policy based. Since reconsidering the intention that α always open the question of whether α, reconsideration is closely related to commitment. We are interested on how would policy based commitment strategies be approached by intentional learning.

Different multi-modal BDI (Belief-Desire-Intention) logics [11, 13, 14] formalise practical reasoning. They are used to reason about rational agents because

of their expressiveness, but not to program them. Agent oriented languages, as $AgentSpeak(L)$ [10], were proposed to reduce the gap between the theory and practice of Multi-Agent Systems (MAS). They have a well defined operational semantics, but verifying intentional properties of the agents programmed in them is not evident, since they dropped intentional and time modalities for the sake of efficiency.

The main question approached in this work is what kind of commitment is used by the rational agents implemented in $AgentSpeak(L)$? In order to answer that, we develop some ideas to reason temporally about intentional operators based on the operational semantics of this language. Then we prove that these agents follow a limited form of single minded commitment [9]. Finally, we discuss the use of intentional learning [6–8] to approach full single minded commitment in a similar way, Bratman argues, we form policies of reconsideration.

The paper is organized as follows: section 2 introduces briefly the subject of the commitment strategies in the BDI logics. Section 3 presents the agent oriented language $AgentSpeak(L)$ and its operational semantics. Section 4 explains the methodology used to prove BDI properties in such language. Section 5 presents the results on the temporal reasoning approach we use to inquire about the commitment strategy followed by $AgentSpeak(L)$ agents and the strategy found. The role of intentional learning in commitment is introduced briefly in section 6. Section 7 closes with discussion and future work.

2 Commitment

Different computational theories of practical reasoning have been proposed to capture the main ideas proposed by Bratman. Cohen and Levesque [4] defined intentions as choice with commitment, based on the concept of persistent goals. A critical examination of this theory [12] suggested that it fails to capture important aspects of commitment, as no-infinite deferral. Alternatively, commitment has been approached as a process of maintenance and revision of intentions, relating current and future intentions. Different types of commitment strategies define different types of agents. Three of them have been extensively studied in the context of BDI_{CTL} [11], where CTL [5] is the well known branched temporal logic:

- **Blind commitment.** An agent intending that inevitably (A, for all time branches) eventually ($\Diamond$, in a branch) is the case that ϕ, inevitably maintains his intentions until (U) he actually believes ϕ (his intention is achieved):

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ \mathsf{BEL}(\phi)) \tag{1}$$

- **Single-minded commitment.** An agent maintains his intentions as long as he believes they are not achieved or optionally (E) eventually are achievable:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi) \vee \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)) \tag{2}$$

– **Open-minded commitment.** An agent maintains his intentions as long as they are not achieved or they are still desired:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi) \vee \neg\mathsf{DES}(\mathsf{A}\Diamond\phi))) \quad (3)$$

BDI_{CTL} is expressive enough to capture these notions of commitment. It is possible to verify if an agent system satisfies them, proving they are valid formulae in such system. Convergence to what is intended, under certain conditions, can also be proved [9]. The problem is that the multimodal logics of rational agency, as BDI_{CTL}, were conceived to reason about agents and not to program them. Agent oriented languages, as *AgentSpeak*(L), were proposed to fill the gap between the theory and practice of MAS.

3 AgentSpeak(L)

The grammar of *AgentSpeak*(L) [10], as defined for its interpreter Jason [2], is shown in table 1. As usual an agent ag is formed by a set of plans ps and beliefs bs. Each belief $b_i \in bs$ is a ground first-order term. Each plan $p \in ps$ has the form $trigger\ event : context \leftarrow body$. A trigger event can be any update (addition or deletion) of beliefs (at) or goals (g).The context of a plan is an atom, a negation of an atom or a conjunction of them. A non empty plan body is a sequence of actions (a), goals, or belief updates. $\top$ denotes empty elements, e.g., plan bodies, contexts, intentions. Atoms (at) can be labelled with sources. Two kinds of goals are defined, achieve goals (!) and test goals (?).

ag	$::= bs \quad ps$		
bs	$::= b_1 \ldots b_n$	$(n \geq 0)$	
ps	$::= p_1 \ldots p_n$	$(n \geq 1)$	
p	$::= te : ct \leftarrow h$		
te	$::= +at \mid -at \mid +g \mid -g$		
ct	$::= ct_1 \mid \top$		
ct_1	$::= at \mid \neg at \mid ct_1 \wedge ct_1$		
h	$::= h_1; \top \mid \top$		
h_1	$::= a \mid g \mid u \mid h_1; h_1$		

at	$::= P(t_1, \ldots, t_n)$	$(n \geq 0)$
	$\mid P(t_1, \ldots, t_n)[s_1, \ldots, s_m]$	$(n \geq 0, m \geq 0)$
s	$::= \texttt{percept} \mid \texttt{self} \mid id$	
a	$::= A(t_1, \ldots, t_n)$	$(n \geq 0)$
g	$::= !at \mid ?at$	
u	$::= +b \mid -b$	

Table 1. Grammar of *AgentSpeak*(L) [2]

The operational semantics [2] of the language, is given by a set of rules that define a transition system between configurations $\langle ag, C, M, T, s\rangle$, where:

– ag is an agent program formed by a set of beliefs bs and plans ps.

– An agent circumstance C is a tuple $\langle I, E, A\rangle$, where: I is a set of intentions $\{i, i', \dots\}$, each $i \in I$ is a stack of partially instantiated plans $p \in ps$; E is a set of events $\{(te, i), (te', i'), \dots\}$, each te is a trigger event and each i is an intention (internal events) or the empty intention $\top$ (external events); and A is a set of actions to be performed in the environment.
– M is a tuple $\langle In, Out, SI\rangle$ working as a mailbox, where: In is the mailbox of the agent; Out is a list of messages to be delivered by the agent; SI is a register of suspended intentions (intentions that wait for an answer message).
– T is a tuple $\langle R, Ap, \iota, \epsilon, \rho\rangle$ that registers temporary information as follows: R is the set of relevant plans for a given event; Ap is the set of applicable plans (the subset of applicable plans which contexts are believed true); ι, ϵ, and ρ register the current intention, event and applicable plan along one cycle of execution.
– The label s indicates the current step in the reasoning cycle of the agent.

Figure 1 shows the interpreter for $AgentSpeak(L)$ as a transition system. The operational semantics rules [2] define the transitions. Because of space limitations, table 2 shows only the rules that are relevant for the next section.

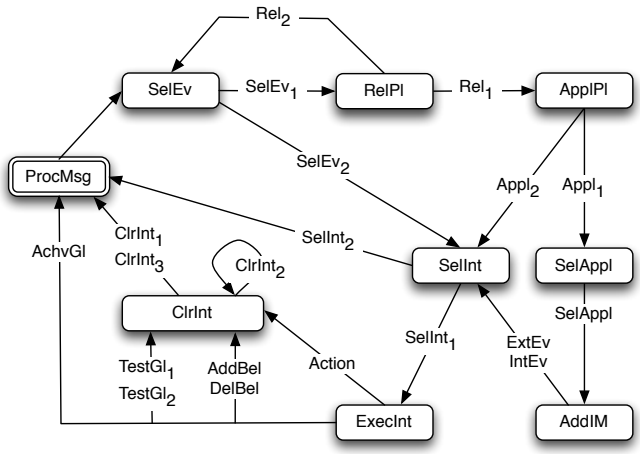

Fig. 1. The interpreter for $AgentSpeak(L)$ as a transition system.

Although the operational semantics defines clearly the practical reasoning performed by an agent, it is difficult to prove intentional properties using it. This is due to the abandonment of intentional and temporal modalities. The approach followed in this paper is to define these temporal and intentional operators in terms of the operational semantics, enabling the demonstration of BDI properties.

(SelEv$_1$)	$\frac{\mathcal{S}_{\mathcal{E}}(C_E)=\langle te,i\rangle}{\langle ag,C,M,T,SelEv\rangle \longrightarrow \langle ag,C',M,T',RelPl\rangle}$	s.t. $C'_E = C_E \setminus \{\langle te,i\rangle\}$ $T'_\epsilon = \langle te,i\rangle$
(Rel$_1$)	$\frac{T_\epsilon=\langle te,i\rangle, RelPlans(ag_{ps},te)\neq\{\}}{\langle ag,C,M,T,RelPl\rangle \longrightarrow \langle ag,C,M,T',AppPl\rangle}$	s.t. $T'_R = RelPlans(ag_{ps},te)$
(Rel$_2$)	$\frac{RelPlans(ps,te)=\{\}}{\langle ag,C,M,T,RelPl\rangle \longrightarrow \langle ag,C,M,T,SelEv\rangle}$	
(Appl$_1$)	$\frac{ApplPlans(ag_{bs},T_R)\neq\{\}}{\langle ag,C,M,T,ApplPl\rangle \longrightarrow \langle ag,C,M,T',SelAppl\rangle}$	s.t. $T'_{Ap} = AppPlans(ag_{bs},T_R)$
(SelAppl)	$\frac{\mathcal{S}_{\mathcal{O}}(T_{Ap})=(p,\theta)}{\langle ag,C,M,T,SelAppl\rangle \longrightarrow \langle ag,C,M,T',AddIM\rangle}$	s.t. $T'_\rho = (p,\theta)$
(ExtEv)	$\frac{T_\epsilon=\langle te,\top\rangle, T_\rho=(p,\theta)}{\langle ag,C,M,T,AddIM\rangle \longrightarrow \langle ag,C',M,T,SelInt\rangle}$	s.t. $C'_I = C_I \cup \{[p\theta]\}$
(SelInt$_1$)	$\frac{C_I\neq\{\}, \mathcal{S}_{\mathcal{I}}(C_I)=i}{\langle ag,C,M,T,SelInt\rangle \longrightarrow \langle ag,C,M,T',ExecInt\rangle}$	s.t. $T'_\iota = i$
(SelInt$_2$)	$\frac{C_I=\{\ \}}{\langle ag,C,M,T,SelInt\rangle \longrightarrow \langle ag,C,M,T,ProcMsg\rangle}$	
(AchvGl)	$\frac{T_\iota=i[head\leftarrow !at;h]}{\langle ag,C,M,T,ExecInt\rangle \longrightarrow \langle ag,C',M,T,ProcMsg\rangle}$	s.t. $C'_E = C_E \cup \{\langle +!at, T_\iota\rangle\}$ $C'_I = C_I \setminus \{T_\iota\}$
(ClrInt$_1$)	$\frac{T_\iota=[head\leftarrow\top]}{\langle ag,C,M,T,ClrInt\rangle \longrightarrow \langle ag,C',M,T,ProcMsg\rangle}$	s.t. $C'_I = C_I \setminus \{T_\iota\}$
(ClrInt$_2$)	$\frac{T_\iota=i[head\leftarrow\top]}{\langle ag,C,M,T,ClrInt\rangle \longrightarrow \langle ag,C',M,T,ClrInt\rangle}$	s.t. $C'_I = (C_I \setminus \{T_\iota\}) \cup$ $\{k[(head' \leftarrow h)\theta]\}$ if $i = k[head' \leftarrow g;h]$ and $g\theta = TrEv(head)$
(ClrInt$_3$)	$\frac{T_\iota\neq[head\leftarrow\top]\wedge T_\iota\neq i[head\leftarrow\top]}{\langle ag,C,M,T,ClrInt\rangle \longrightarrow \langle ag,C,M,T,ProcMsg\rangle}$	

Table 2. Some rules of the operational semantics of $AgentSpeak(L)$, relevant for the definition of intentional and temporal operators.

4 Methodology

Following Bordini [1] we define the intentional modalities of BDI_{CTL} in terms of $AgentSpeak(L)$ operational semantics. First the auxiliary function achievement goals is defined:

$$agoals(\top) = \{\},$$
$$agoals(i[p]) = \begin{cases} \{at\} \cup agoals(i) & \text{if } p = +!at : ct \leftarrow h, \\ agoals(i) & \text{otherwise} \end{cases}$$

which returns the set of atomic formulae (at) subject to an addition of achievement goal ($+!$) in the trigger events of the plans composing a given intention ($i[p]$ denotes an intention i which top plan is p).

Then the operators for BEL, DES, and INTEND are defined in terms of an agent ag and its circumstance C, given in a configuration:

$$\mathsf{BEL}_{\langle ag,C\rangle}(\phi) \equiv bs \models \phi. \tag{4}$$

$$\mathsf{INTEND}_{\langle ag,C\rangle}(\phi) \equiv \phi \in \bigcup_{i \in C_I} agoals(i) \vee \phi \in \bigcup_{\langle te,i\rangle \in C_E} agoals(i). \tag{5}$$

$$\mathsf{DES}_{\langle ag,C\rangle}(\phi) \equiv \langle +!\phi, i\rangle \in C_E \vee \mathsf{INTEND}_{\langle ag,C\rangle}(\phi). \tag{6}$$

These definitions are used to prove the assymetry thesis [3] (See table 3). The thesis expresses that intention-belief inconsistency is closer to irrationality than intention-belief incompletness (AT1-AT3), and the same for intention-desire (AT4-AT6) and desire-belief (AT7-AT8). Bordini and Moreira [1] prove that, under close world assumption, all $AgentSpeak(L)$ agents do not satisfy the asymmetry thesis AT1, AT5, and AT7, but they satisfy the rest of them. This means that $AgentSpeak(L)$ is not equivalent to any of the BDI modal systems studied previously by Rao and Georgeff [11].

Label	**Theorem**
AT1	$\models \mathsf{INTEND}_{\langle ag,C\rangle}(\phi) \implies \mathsf{BEL}_{\langle ag,C\rangle}(\phi)$
AT2	$\not\models \mathsf{INTEND}_{\langle ag,C\rangle}(\phi) \implies \mathsf{BEL}_{\langle ag,C\rangle}(\phi)$
AT3	$\not\models \mathsf{BEL}_{\langle ag,C\rangle}(\phi) \implies \mathsf{INTEND}_{\langle ag,C\rangle}(\phi)$
AT4	$\models \mathsf{INTEND}_{\langle ag,C\rangle}(\phi) \implies \mathsf{DES}_{\langle ag,C\rangle}(\phi)$
AT5	$\not\models \mathsf{INTEND}_{\langle ag,C\rangle}(\phi) \implies \mathsf{DES}_{\langle ag,C\rangle}(\phi)$
AT6	$\not\models \mathsf{DES}_{\langle ag,C\rangle}(\phi) \implies \mathsf{INTEND}_{\langle ag,C\rangle}(\phi)$
AT7	$\models \mathsf{DES}_{\langle ag,C\rangle}(\phi) \implies \mathsf{BEL}_{\langle ag,C\rangle}(\phi)$
AT8	$\not\models \mathsf{DES}_{\langle ag,C\rangle}(\phi) \implies \mathsf{BEL}_{\langle ag,C\rangle}(\phi)$
AT9	$\not\models \mathsf{BEL}_{\langle ag,C\rangle}(\phi) \implies \mathsf{DES}_{\langle ag,C\rangle}(\phi)$

Table 3. Asymmetry thesis expressed in $AgentSpeak(L)$

5 Results

The main contributions of this paper are the preliminary definition of temporal operators to reason about $AgentSpeak(L)$ programs; and a demonstration that $AgentSpeak(L)$ agents are not blind commited, but perform a limited single-minded commitment.

5.1 Time

Since $AgentSpeak(L)$ abandons time modalities for the sake of efficiency, it is necessary to redefine them. As it is well known, temporal modalities are defined

after a Kripke Structure $\langle S, R, L\rangle$ where S is a set of states, L the labeling for each state in S and R is a total relation on $S \times S$. Roughly, the states in $AgentSpeak(L)$ correspond to agent configurations $\langle ag, C, M, T, s\rangle$; R is defined by the operational semantics of the system, being certainly total ($\forall k\ \exists t\ (k,t) \in R$ s.t. $k, t \in S$), as shown in figure 1. L is the label of primitive formulae valid at a given state. Validity for intentional operators is defined as in the previous section. Paths, as usual, are sequences of states (configurations) $c_0, \ldots, c_n$.

Then, the definition of next is:

$$\models_{c_0} \bigcirc\alpha \ \equiv\ T_\iota = i[head \leftarrow \alpha; h] \tag{7}$$

for $\alpha \in \{a, g, u\}$ (action, goal, or belief update). c_0 is the current configuration, where formulae are evaluated. Observe that $T_\iota = \_$ always, except when an intention has been successfully selected to be executed at time t in $s = \texttt{SelInt}$, so that at time $t+1$ the system will be in $s = \texttt{ExecInt}$ since the selection was successful (otherwise the system goes to `ProcMsg`), then α will occur in the next state of the system. It is evident that we can now define the semantics for expressions like $\bigcirc\mathsf{BEL}(\phi)$, at least for intentional updates, e.g., changes that result from the execution of intentions.

As part of our current work, we are exploring formal definitions for until:

$$\models_{c_0} \phi\ \mathsf{U}\ \psi \ \equiv\ \exists k > 0\ \ \models_{c_k} \psi \wedge \forall 0 < j \leq k \models_{c_j} \phi \tag{8}$$

and eventually:

$$\models_{c_0} \Diamond\phi \ \equiv\ \exists k > 0 \models_{c_k} \phi \tag{9}$$

With these definitions we already can prove some properties about the commitment strategies of $AgentSpeak(L)$ agents.

5.2 Commitment strategies in *AgentSpeak(L)*

The main question at this stage of our research on commitment and intentional learning was what kind of commitment strategy is used by $AgentSpeak(L)$ agents? Knowing that is important, because intentional learning seems to be irrelevant for a blindly committed agent, while it can be really useful and explanatory if agents are single or open minded. So our first step was to prove that $AgentSpeak(L)$ agents do not satisfy the blind commitment axiom under no-infinite deferral.

Proposition 1. *$AgentSpeak(L)$ agents satisfy the no-infinite deferral axiom* $\mathsf{INTEND}(\phi) \Rightarrow \mathsf{A}\Diamond(\neg\mathsf{INTEND}(\phi))$.

Proof. Given the definition for intend (eq. 5), the no-infinite deferral axiom expresses that if a plan p with context $+!\phi$ is adopted to form an intention, this plan is eventually retired from C_I (active intentions) and C_E (suspended intentions). Given the finite nature of the plans and providing that intentions and events are always possible to be selected in $SelInt$ and $SelEv$ steps, there are different paths

satisfying $\mathsf{A}\Diamond\neg\mathsf{INTEND}_{\langle ag,C\rangle}(\phi)$, all of them via $\bigcirc\neg\mathsf{INTEND}_{\langle ag,C\rangle}(\phi) \equiv \bigcirc -!\phi$ (eq. 7) or $s = ClrInt$: i) Succesful execution of a plan: when the plan body becomes empty, the plan is removed from $i[p] \in C_I$ by $ClrInt_2$; ii) Succesful execution of intention: when an intention becomes empty, the full intention is removed from C_I by $ClrInt_1$; iii) Keep going execution of the intention: when the body of the plan in the top of an intention is not empty, the cycle continues and the intention will be eventually selected again by $SelInt$ arriving, if everything goes right, to one of the previous situations. If something goes wrong, a failure mechanism is activated by an event of the form $\langle -!\phi, i[p]\rangle$. Although Bordini et al. [2] only formalizes failures in finding relevant plans (Rel_2) which discards suspended intentions in C_E, other forms of failure detection have been considered in the context of intentional learning [6, 8]. By successful or failed execution of the plans every adopted intention is inevitable eventually dropped. □

Proposition 2. *AgentSpeak(L) agents do not satisfy the blind commitment axiom* $\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ \mathsf{BEL}(\phi))$.

Proof. Given that the no-infinite-deferral axiom is safisfied by *AgentSpeak(L)* agents, the blind axiom can be reduced to $\mathsf{INTEND}(\mathsf{A}\Diamond(\phi)) \implies (\mathsf{A}\Diamond(\mathsf{BEL}(\phi))$, given the blind commitment axiom (eq. 1) and assuming weak until [9]. In order to prove the proposition we will define an agent that does not satisfy the reduced blind commitment axiom. Consider an agent s.t. $ag = \langle bs, ps\rangle$ where $bs = \{\}$ and $ps = \{+b(t_1) : \top \leftarrow p(t_2).\ \ +!p(t_2) : \top \leftarrow +b(t_3).\}$ at its initial configuration. Suppose that from perception of the environment a belief $b(t_1)$ is added to the $ag_{bs} = \{b(t_1)\}$. An event is generated by this belief update, so that $C_E = \{\langle +b(t_1), \top\rangle\}$. Then following the state transitions defined by the semantic rules $SelEv_1$, Rel_1, $ApplPl_1$, we obtain a configuration where $C_I = \{[+b(t_1) : \top \leftarrow !p(t_2).]\}$ and $C_E = \{\}$, Then proceeding with the reasoning steps $SelAppl$, $ExtEv$, $SelInt_1$, $AchvGl$ we obtain a configuration where $C_E\langle +!p(t_2), +b(t_1) : \top \leftarrow \top\rangle$, $C_I = \{\}$. At this moment, the agent $\mathsf{DES}_{\langle ag,C\rangle}(p(t_2))$ (eq. 6). If we apply then $SelEv_1$, Rel_1, $AppPl_1$, $SelAppl$ then we obtain a configuration where $C_I = \{[+!p(t_2) : \top \leftarrow\ \ +b(t_3).]\}$ and $C_E = \{\}$, where the agent $\mathsf{INTEND}_{\langle ag,C\rangle}(p(t_2))$ (eq. 5). Then proceeding with $IntEv$, $SelInt_1$, $AddBel$ then $C_E = \langle +b(t_3)[self], \top\rangle$, $ag_{bs} = \{b(t_1)\}$ and $C_I = \{[+b(t_1) : \top \leftarrow \top \ddagger +!p(t_2) : \top \leftarrow \top]$ and $bs = \{b(t_1), b(t_3)\}$. The intention about $p(t_2)$ is maintained. Observe that the plan bodies in the intention are empty, so the $ClrInt$ rules will discard the whole intention, so that $\neg\mathsf{INTEND}_{\langle ag,C\rangle}(p(t_2))$ and $\neg\mathsf{BEL}_{\langle ag,C\rangle}(p(t_2))$. $\mathsf{INTEND}(\mathsf{A}\Diamond(\phi)) \implies (\mathsf{A}\Diamond(\mathsf{BEL}(\phi))$ is not satisfied for this agent. □

In fact, our agent does not safisfy the extended blind commitment axiom (eq. 1), since the agent did not keep its intention about $p(t_2)$ until she believed it. This reasoning is similar to the demonstration of intention-belief incompleteness (AT2) for *AgentSpeak(L)* [1].

Proposition 3. *AgentSpeak(L) agents satisfy a limited single-minded commitment* $\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi))$.

Proof. Following the no-infinite-deferral demonstration, in the failure cases the agent will eventually satisfy $\bigcirc\neg\mathsf{INTEND}_{\langle ag,C\rangle}(\phi)$ because Rel_2 which means that for an event $\langle te, i[+!\phi : c \leftarrow h.]\rangle$ there were not relevant plans and the associated intention will be discarded, e.g., there is not a path of configurations where eventually ϕ, so that it is rational to drop $\mathsf{INTEND}_{\langle ag,C\rangle}(\phi)$. □

This is a limited form of single-mind commitment because $\neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)$ is not represented explicitly by the agent. In fact, she only can not continue intending ϕ because there are no plans to do it and the full intention fails. It is also limited because of intention-belief incompleteness that can be avoided dropping the close world assumption [1]; or using the intentional and temporal definitions for studying the necessary conditions in the operational semantics and definition of the agents to warrant the expected properties of intentions, e.g., equivalence to a KD modal system.

Intentional learning provides a third alternative approach to achieve a full single-minded strategy, enabling the explicit representation of the reasons to abandon and intention in a less reactive basis, i.e., preemptive abandon of intentions.

6 Intentional learning and commitment

An agent can learn about his practical reasons. Learning examples are composed by the beliefs that supported the adoption of a plan as an intention. Because of the no-infinite deferral axiom, all examples become eventually labelled as success or failure cases. Then, first-order induction of logical decision trees is used to learn hypothesis about the reasons for successful adoption of intentions, in order to update the context of plans that have failed. If the examples of an agent do not offer enough evidence to learn, the agent can ask other agents sharing the plan for more examples. We have called this intentional learning [6–8].

For example, suppose we have an agent situated in the blocks world who perceives the state shown at figure 2, State-a; and desires to achieve the state shown at State-b. So, he forms an intention after the event $\langle +!on(b,c), \top\rangle$ using a relevant applicable plan. Suppose such plan has the form $[p1]$ $+!on(X,Y)$: $\top \leftarrow .take(X);\ .put(X,Y)$. It means our agent is bold (or naive) about stacking objects, he believes X can be stacked on Y in any circumstance using plan $p1$. Now, suppose that after forming an intention with this plan, the agent perceives the State-c. The internal action $.put(X/a, Y/b)$ will fail, the intention will eventually fail, and the goal $on(b,c)$ will not be achieved.

An intentional learning agent can find out that the right context for the failed plan is $clear(Y) \wedge handfree(Ag)$ and modify his plan definition accordingly. Then the next time he processes the event $\langle +!on(b,c), \top\rangle$ at a state similar to State-c, it will be the case that $bs \not\models clear(c)$, and the plan will not be applicable anymore. This is better than the original non-learning approach, but does not avoid the original problem. Considering also the failure branches of the induced logic tree, in order to add beliefs as $abandon(p1) \leftarrow not(clear(Y))$, enables preemptive abandon of intentions, via a cleaning mechanism to deal events of

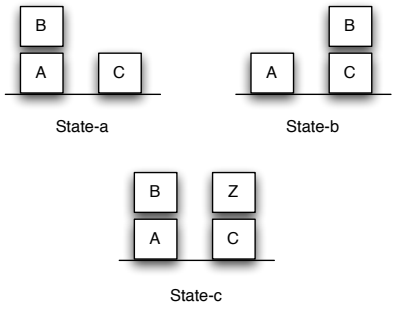

Fig. 2. Blocks world as perceived (State-a), desired (State-b), and perceived after forming an intention to put B on C, but before execute it (State-c).

the form $\langle +abandon(X), i\rangle$, as in Rel_2. We are currently experimenting with different forms of cleaning, implementing this scenario in Jason [2].

7 Discussion and future work

We have extended the methodology proposed by Bordini and Moreira [1] to reason about $AgentSpeak(L)$ agents. The extension consists in defining temporal operators based on the operational semantics of this agent oriented programming language. Then we proved that any $AgentSpeak(L)$ agent is not blindly committed, but follows a limited form of single-mind commitment. The main limitations for these agents are intention-belief incompleteness and the lack of an explicit representation for abandoning reasons. We have argued that intentional learning provides a solution for the second problem. Interestingly, modifying the context of the plans in ag_{ps} involves changes in the accessibility while forming intentions. We plan to study if intentional learning can approach intention-belief completeness in this way.

The degree of boldness and cautiousness for a given agent is something hard to define. It is well known that in dynamic environments a very cautious agent performs better than a bold one; and inversely, in static environments boldness pays better. The relevance of learning intentionally is that the right degree of cautioness is learned by the agents, instead of being established once and forever by the programmers.

Immediate future work includes to propose a $CTL_{AgentSpeak(L)}$ logic to reason about these agents. The preliminary results reported here are very encouraging in this sense. The main difficulty here is that CTL is propositional, while the content of $AgentSpeak(L)$ intentional operators is first-order, complicating the definition of L or a valuating function in the Kripke structure supporting temporal semantics. An extended $AgentSpeak(L)$ operational semantics that deals with intentional learning, for both incremental and batch inductive methods,

has been proposed [8]. So, it is possible to arrive to a full theory of commitment and intentional learning using the techniques presented here.

This computational theory should consider the concept of policy based reconsideration and commitment in practical reasoning. This is relevant because it brings *AgentSpeak*(L) closer to its philosophical foundation. But also, because policy based (non)reconsideration seems to be the more interesting of the three cases considered by Bratman. It is not so hard wired as non-deliberative cases, nor is so costly as deliberative ones.

Acknowledgments. The first and third authors have applied for Conacyt CB-2007-1 (project 78910) funding for this research. The second author is supported by Conacyt scholarship 214783.

References

1. Bordini, R.H., Moreira, Á.F.: Proving BDI properties of agent-oriented programming languages. Annals of Mathematics and Artificial Intelligence 42, 197–226 (2004)
2. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming Multi-Agent Systems in AgentSpeak using Jason. Wiley, England (2007)
3. Bratman, M.: Intention, Plans, and Practical Reason. Harvard University Press, Cambridge (1987)
4. Cohen, P., Levesque, H.: Intention is choice with commitment. Artificial Intelligence 42(3), 213–261 (1990)
5. Emerson, A.: Temporal and modal logic. In: van Leeuwen, J., (ed.) Handbook of Theoretical Computer Science, pp. 996–1072. Elsevier, Amsterdam (1990)
6. Guerra-Hernández, A., El-Fallah-Seghrouchni, A., Soldano, H.: Learning in BDI Multi-agent Systems. In: Dix, J., Leite, J. (eds.) CLIMA IV. LNCS, vol. 3259, pp. 218–233. Springer, Heidelberg (2004)
7. Guerra-Hernández, A., Ortíz-Hernández, G.: Toward BDI sapient agents: Learning intentionally. In: Mayorga, R.V., Perlovsky, L.I. (eds.) Toward Artificial Sapience: Principles and Methods for Wise Systems, pp. 77–91. Springer, London (2008)
8. Guerra-Hernández, A., Ortíz-Hernández, G., Luna-Ramírez, W.A.: Jason smiles: Incremental BDI MAS learning. In: MICAI 2007 Special Session, IEEE, (In press).
9. Rao, A.S., Georgeff, M.P.: Modelling Rational Agents within a BDI-Architecture. In: Huhns, M.N., Singh, M.P., (eds.) Readings in Agents, pp. 317–328. Morgan Kaufmann (1998)
10. Rao, A.S.: AgentSpeak(L): BDI agents speak out in a logical computable language. In: de Velde, W.V., Perram, J.W. (eds.) MAAMAW. LNCS, vol. 1038, pp. 42–55. Springer, Heidelberg (1996)
11. Rao, A.S., Georgeff, M.P.: Decision procedures for BDI logics. Journal of Logic and Computation 8(3), pp. 293–342 (1998)
12. Singh, M.P.: A critical examination of the Cohen-Levesque Theory of Intentions. In: Proceedings of the European Conference on Artificial Intelligence (1992).
13. Singh, M.P., Rao, A.S., Georgeff, M.P.: Formal Methods in DAI: Logic-Based Representation and Reasoning. In: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, pp. 331–376. MIT Press, Cambridge (1999)
14. Wooldridge, M.: Reasoning about Rational Agents. MIT Press, Cambridge (2000)

$CTL_{AgentSpeak(L)}$: a Specification Language for Agent Programs

Alejandro Guerra-Hernández[1], José Martín Castro-Manzano[1], and Amal El-Fallah-Seghrouchni[2]

[1] Departamento de Inteligencia Artificial
Universidad Veracruzana
Facultad de Física e Inteligencia Artificial
Sebastián Camacho No. 5, Xalapa, Ver., México, 91000
aguerra@uv.mx, e_f_s_s@hotmail.com
[2] Laboratoire d'Informatique de Paris 6
Université Pierre et Marie Curie
Avenue du Président Kennedy, Paris, France, 75016
Amal.Elfallah@lip6.fr

Abstract. This work introduces $CTL_{AgentSpeak(L)}$, a logic to specify and verify expected properties of rational agents implemented in the $AgentSpeak(L)$ agent oriented programming language. Our approach is similar to the classic BDI_{CTL} modal logic, used to reason about agents modelled in terms of belief (BEL), desires (DES), intentions (INTEND). A new interpretation for the temporal operators in CTL: next ($\bigcirc$), eventually ($\Diamond$), until(U), inevitable(A), etc., is proposed in terms of the transition system induced by the operational semantics of $AgentSpeak(L)$. The main contribution of the approach is a better understanding of the relation between the programming language and its logical specification, enabling us to proof expected or desired properties for any agent programmed in $AgentSpeak(L)$, e.g., commitment strategies.

1 Introduction

The theory of practical reasoning proposed by Bratman [3], expounds the philosophical foundation for the computational approaches to rational agency, known as BDI (Belief-Desire-Intention) systems. This theory is innovative because it does not reduce intentions to some combination of beliefs and desires, but indeed it assumes that intentions are composed by hierarchical, partial plans. Such assumption explains better temporal aspects of practical reasoning as future intentions, persistence, reconsideration and commitment. Different multi-modal BDI logics [14, 16, 17] have been proposed to formally characterize the rational behavior of such agents, in terms of the properties of the intentional attitudes and the relations among them, e.g., it is rational to intend desires that are believed possible; and temporal changes of these mental attitudes, e.g., commitment strategies define when it is rational to drop intentions. Due to their expressiveness, these logics are used to reason about the agents properties; but, because of their computational cost, they are not used to program them.

Agent oriented programming languages, such as $AgentSpeak(L)$ [13], have been proposed to reduce the gap between theory (logical specification) and practice (implementation) of rational agents. Even when this programming language has a well defined operational semantics, the verification of rational properties of the programmed agents is not evident, since intentional and time modalities are abandoned for the sake of efficiency. In order to reason about such properties, we propose $CTL_{AgentSpeak(L)}$ as a logic for specification and verification of $AgentSpeak(L)$ agents. The approach is similar to the classic BDI_{CTL} [14] logic, defined as a $B^{KD45}D^{KD}I^{KD}$ modal system, with temporal operators: next ($\bigcirc$), eventually ($\Diamond$), until(U), inevitable(A), etc., defined after the computational tree logic (CTL) [6].

Our main contribution is the definition of the semantics of the CTL temporal operators in terms of a Kripke structure, produced by a transition system defining the operational semantics of $AgentSpeak(L)$. The semantics of the intentional operators is adopted from the work of Bordini et al. [1]. As a result, the semantics of $CTL_{AgentSpeak(L)}$ is grounded in the operational semantics of programming language. In this way, we can prove if any agent programmed in $AgentSpeak(L)$ satisfies certain properties expressed in the logical specification. It is important to notice that our problem is different from that of model checking in the following sense: in model checking the problem consists in verifying if certain property holds in certain state in certain agent, while our work deals with verifying that certain general properties hold for any agent. The approach is exemplified verifying the commitment strategies for $AgentSpeak(L)$.

The paper is organized as follows: Section 2 exemplifies the specification of rational properties in BDI_{CTL} [14] with the definition of commitment strategies. Section 3 introduces the syntax and the semantics of the $AgentSpeak(L)$ agent oriented programming language. Section 4 presents the main contribution of the paper: a logic framework to reason about $AgentSpeak(L)$ agents, with semantics based on the operational semantics of the programming language. Section 5 shows how the commitment strategies introduced in section 2 can be verified for $AgentSpeak(L)$. Section 6 offers concluding remarks and discusses future work.

2 Commitment

As mentioned, different computational theories have been proposed to capture the theory of Bratman [3] on intentions, plans and practical reasoning. The foundational work of Cohen and Levesque [4], for example, defined intention as a combination of belief and desire based on the concept of persistent goal. A critical analysis of this theory [15] showed that the theory of Cohen and Levesque failed to capture important aspects of commitment. Alternatively, commitment has been approached as a process of maintenance and revision of intentions, relating current and future intentions.

Different types of commitment strategies define different types of agents. Three of them have been extensively studied in the context of BDI_{CTL} [14], where CTL denotes computational tree logic [6], the well known temporal logic:

– **Blind commitment.** An agent intending that inevitably (A) eventually ($\Diamond$) is the case that ϕ, inevitably maintains his intentions until (U) he actually believes ϕ:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ \mathsf{BEL}(\phi)) \tag{1}$$

– **Single-minded commitment.** An agent maintains his intentions as long as he believes they are achieved or optionally (E) eventually achievable:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi) \vee \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)) \tag{2}$$

– **Open-minded commitment.** An agent maintains his intentions as long as they are achieved or they are still desired:

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ (\mathsf{BEL}(\phi) \vee \neg\mathsf{DES}(\mathsf{A}\Diamond\phi))) \tag{3}$$

For example, a blind agent intending eventually to go to Paris, will maintain his intention, for any possible course of action (inevitable), until he believes he is going to Paris. A single-minded agent can drop such intention if he believes it is not possible any more to go to Paris. An open-minded agent can drop the intention if he does not desire anymore going to Paris. An interesting question is what kind of commitment strategy is followed by *AgentSpeak*(L) agents?

Furthermore, how can we relate commitment with policy-based reconsideration in *AgentSpeak*(L)? Bratman argues that there are three cases of reconsideration ([3], pp. 60–75) in practical reasoning. Non-reflective reconsideration has short effects, while deliberative one is very expensive. Policy-based reconsideration is a compromise between impact and cost. Obviously, if an agent is blindly committed, we can not talk about any form of reconsideration. But if the an agent is single-minded, then we could approach policy-based reconsideration through intentional learning [7–10]. In this way we would reconcile a relevant aspect of the computational theories of BDI agency (commitment) with its philosophical foundation (reconsideration *à la* Bratman).

3 *AgentSpeak*(L)

In this section, the syntax and semantics of *AgentSpeak*(L) [13], as defined for its interpreter Jason [2], are introduced. The operational semantics of the language is given in terms of a transition system that, being a Kripke structure, is used to define the semantic of the temporal operators in $CTL_{AgentSpeak(L)}$ in the next section; the BDI operators will also be defined in terms of the structures supporting the operational semantics of *AgentSpeak*(L)

3.1 Syntax

The syntax of $AgentSpeak(L)$ is shown in table 1. As usual, an agent ag is formed by a set of plans ps and beliefs bs (grounded literals). Each plan has the form $triggerEvent : context \leftarrow body$. The context of a plan is a literal or a conjunction of them. A non empty plan body is a finite sequence of actions, goals (achieve or test an atomic formula), or beliefs updates (addition or deletion). $\top$ denotes empty elements, e.g., plan bodies, contexts, intentions. The trigger events are updates (addition or deletion) of beliefs or goals.

$$
\begin{array}{lll|ll}
ag & ::= bs \quad ps & & at & ::= P(t_1, \ldots, t_n)\ (n \geq 0) \\
bs & ::= b_1 \ldots b_n & (n \geq 0) & a & ::= A(t_1, \ldots, t_n)\ (n \geq 0) \\
ps & ::= p_1 \ldots p_n & (n \geq 1) & g & ::= !at \mid ?at \\
p & ::= te : ct \leftarrow h & & u & ::= +b \mid -b \\
te & ::= +at \mid -at \mid +g \mid -g & & & \\
ct & ::= ct_1 \mid \top & & & \\
ct_1 & ::= at \mid \neg at \mid ct_1 \wedge ct_1 & & & \\
h & ::= h_1; \top \mid \top & & & \\
h_1 & ::= a \mid g \mid u \mid h_1; h_1 & & &
\end{array}
$$

Table 1. Sintax of $AgentSpeak(L)$. Adapted from Bordini et al. [2]

3.2 Operational semantics

The operational semantics of $AgentSpeak(L)$ is defined as a transition system between configurations $\langle ag, C, M, T, s \rangle$, where:

- ag is an agent program formed by beliefs bs and plans ps.
- An agent circumstance C is a tuple $\langle I, E, A \rangle$ where I is the set of intentions $\{i, i', \ldots, n\}$ s.t. $i \in I$ is a stack of partially instantiated plans $p \in ps$; E is a set of events $\{\langle te, i \rangle, \langle te', i' \rangle, \ldots, n\}$, s.t. te is a $triggerEvent$ and each i is an intention (internal event) or an empty intention $\top$ (external event); and A is a set of actions to be performed by the agent in the environment.
- M is a tuple $\langle In, Out, SI \rangle$ that works as a $mailbox$, where In is the mailbox of the agent, Out is a list of messages to be delivered by the agent and SI is a register of suspended intentions (intentions that wait for an answer message). It is not relevant for the purposes of this paper.
- T is a tuple $\langle R, Ap, \iota, \epsilon, \rho \rangle$ that registers temporal information: R is the set of relevant plans given certain $triggerEvent$; Ap is the set of applicable plans (the subset of R s.t. $bs \models ctx$); ι, ϵ y ρ register, respectively, the intention, the event and the current plan during an agent execution.
- The configuration label $s \in \{SelEv, RelPl, AppPl, SelAppl, SelInt, AddIM, ExecInt, ClrInt, ProcMsg\}$ indicates the current step in the reasoning cycle of the agent.

Transitions are defined in terms of semantic rules of the form:

$$(\textbf{rule id}) \quad \frac{cond}{C \rightarrow C'}$$

where $C = \langle ag, C, M, T, s\rangle$ is an $AgentSpeak(L)$ configuration that can be transformed to a new one C', if the conditions *cond* hold. Table 2 shows the semantic rules that are relevant for the purposes of this paper (communication processing rules are omitted for simplicity).

(SelEv$_1$)	$\frac{\mathcal{S}_{\mathcal{E}}(C_E)=\langle te,i\rangle}{\langle ag,C,M,T,SelEv\rangle \longrightarrow \langle ag,C',M,T',RelPl\rangle}$	s.t. $C'_E = C_E \setminus \{\langle te, i\rangle\}$ $T'_\epsilon = \langle te, i\rangle$
(Rel$_1$)	$\frac{T_\epsilon=\langle te,i\rangle, RelPlans(ag_{ps},te)\neq\{\}}{\langle ag,C,M,T,RelPl\rangle \longrightarrow \langle ag,C,M,T',AppPl\rangle}$	s.t. $T'_R = RelPlans(ag_{ps}, te)$
(Rel$_2$)	$\frac{RelPlans(ps,te)=\{\}}{\langle ag,C,M,T,RelPl\rangle \longrightarrow \langle ag,C,M,T,SelEv\rangle}$	
(Appl$_1$)	$\frac{ApplPlans(ag_{bs},T_R)\neq\{\}}{\langle ag,C,M,T,ApplPl\rangle \longrightarrow \langle ag,C,M,T',SelAppl\rangle}$	s.t. $T'_{Ap} = AppPlans(ag_{bs}, T_R)$
(SelAppl)	$\frac{\mathcal{S}_{\mathcal{O}}(T_{Ap})=(p,\theta)}{\langle ag,C,M,T,SelAppl\rangle \longrightarrow \langle ag,C,M,T',AddIM\rangle}$	s.t. $T'_\rho = (p, \theta)$
(ExtEv)	$\frac{T_\epsilon=\langle te,\top\rangle, T_\rho=(p,\theta)}{\langle ag,C,M,T,AddIM\rangle \longrightarrow \langle ag,C',M,T,SelInt\rangle}$	s.t. $C'_I = C_I \cup \{[p\theta]\}$
(SelInt$_1$)	$\frac{C_I\neq\{\}, \mathcal{S}_{\mathcal{I}}(C_I)=i}{\langle ag,C,M,T,SelInt\rangle \longrightarrow \langle ag,C,M,T',ExecInt\rangle}$	s.t. $T'_\iota = i$
(SelInt$_2$)	$\frac{C_I=\{\ \}}{\langle ag,C,M,T,SelInt\rangle \longrightarrow \langle ag,C,M,T,ProcMsg\rangle}$	
(AchvGl)	$\frac{T_\iota=i[head\leftarrow !at;h]}{\langle ag,C,M,T,ExecInt\rangle \longrightarrow \langle ag,C',M,T,ProcMsg\rangle}$	s.t. $C'_E = C_E \cup \{\langle +!at, T_\iota\rangle\}$ $C'_I = C_I \setminus \{T_\iota\}$
(ClrInt$_1$)	$\frac{T_\iota=[head\leftarrow\top]}{\langle ag,C,M,T,ClrInt\rangle \longrightarrow \langle ag,C',M,T,ProcMsg\rangle}$	s.t. $C'_I = C_I \setminus \{T_\iota\}$
(ClrInt$_2$)	$\frac{T_\iota=i[head\leftarrow\top]}{\langle ag,C,M,T,ClrInt\rangle \longrightarrow \langle ag,C',M,T,ClrInt\rangle}$	s.t. $C'_I = (C_I \setminus \{T_\iota\}) \cup$ $\{k[(head' \leftarrow h)\theta]\}$ if $i = k[head' \leftarrow g; h]$ and $g\theta = TrEv(head)$
(ClrInt$_3$)	$\frac{T_\iota\neq[head\leftarrow\top] \wedge T_\iota\neq i[head\leftarrow\top]}{\langle ag,C,M,T,ClrInt\rangle \longrightarrow \langle ag,C,M,T,ProcMsg\rangle}$	

Table 2. Operational semantics rules of $AgentSpeak(L)$.

The reasoning cycle of an $AgentSpeak(L)$ agent starts processing messages and updating perception ($s = ProcMsg$). This adds events to C_E (C_α denotes the element α of circumstance C, the same for other element of configurations)

to be processed by the agent. One of these events is selected ($\mathcal{S}_{\mathcal{E}}(C_E) = \langle te, i\rangle$), and relevant and applicable plans are generated using the following definitions:

Definition 1 (Relevant plans) *Given a set of plans ag_{ps}, the subset of relevant plans for a selected event $\langle te, i\rangle \in C_E$, is defined as:*

$$RelPlans(ps, te) = \{(p, \theta) | p \in ps \wedge \theta = mgu(te, TrEv(p))\}$$

where $TrEv(te' : ctxt \leftarrow h) = te'$ gets the trigger event te of a plan, C_E denotes the events E in a given circumstance C, and mgu is the most general unifier.

Definition 2 (Applicable plans) *Given a set of relevant plans T_R and a set of beliefs ag_{bs}, the set of applicable plans is defined as:*

$$AppPlans(bs, R) = \{(p, \theta\theta') | (p, \theta) \in R \wedge \theta' \quad is\ s.t. \quad bs \models Ctxt(p)\theta\theta'\}$$

where θ' is the substitution computed when verifying if the context of relevant plan p ($Ctxt(p)$), affected by its relevant substitution θ, is a logical consequence of the beliefs of the agent bs.

Then the agent proceeds selecting a relevant plan to form an intention ($Appl_1$ and $SelAppl$) or, if no relevant plans were found, selecting an intention to be executed ($Appl_2$ and $SelInt_1$). The execution of an intention changes the environment and the mental attitudes of the agent (including the abandoning of accomplished intentions in $ClrInt$). $ProcMsg$ generates the new events, and so on. Figure 1 shows the transition system induced by these semantic rules. States are labeled with possible values for s. Transitions correspond to the semantic rules identifiers. The initial state is $s = ProcMsg$.

Although the operational semantics of $AgentSpeak(L)$ clearly defines the practical reasoning performed by an agent, it is difficult to prove BDI properties, such as the commitment strategies, for any given agent. This is due to the abandon of intentional and temporal modalities in $AgentSpeak(L)$, the main reason to propose $CTL_{AgentSpeak(L)}$.

4 $CTL_{AgentSpeak(L)}$

$CTL_{AgentSpeak(L)}$ may be seen as an instance of BDI_{CTL}. Similar approaches have been explored for other instances of agent oriented programming languages, e.g, a simplified version of 3APL [5]. The idea is to define the semantics of temporal and intentional operators in terms of the operational semantics of the programming language. This grounds the semantics to reason about particular kinds of agents, in this case $AgentSpeak(L)$ agents. Once this is done, we can use the logic to reason about general properties of such agents.

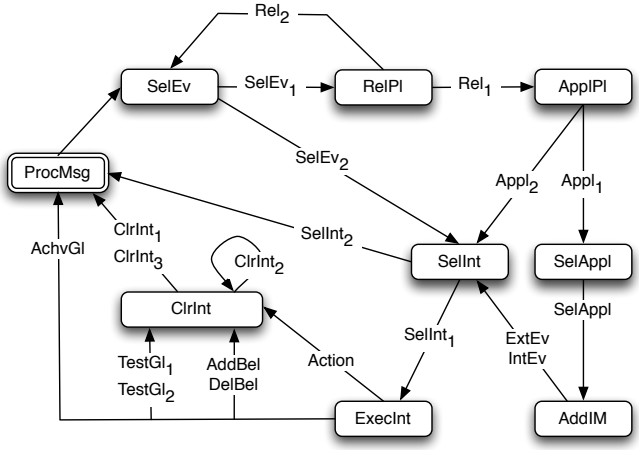

Fig. 1. The interpreter for $AgentSpeak(L)$ as a transition system.

4.1 Syntax

Formulae for intentional and temporal operators are required:

Definition 3 (BDI syntax) *If ϕ is an atomic formula in $AgentSpeak(L)$, then ϕ, $\mathsf{BEL}(\phi)$, $\mathsf{DES}(\phi)$, $\mathsf{INTEND}(\phi)$ are $CTL_{AgentSpeak(L)}$ well formed formulae (BDI formulae).*

Definition 4 (Temporal syntax) *Temporal formulae are divided, as usual, in state and path formulae. State formulae are defined as:*

s1 *Each well formed BDI formula is a state formula.*
s2 *If ϕ and ψ are state formulae, $\phi \wedge \psi$ and $\neg\phi$ are state formulae.*
s3 *If ϕ is a path formula, then $\mathsf{E}\phi$ and $\mathsf{A}\phi$ are state formulae.*

Path formulae are defined as:

p1 *Each state formula is also a path formula.*
p2 *If ϕ and ψ are path formulae, then $\neg\phi$, $\phi \wedge \psi$, $\bigcirc\phi$, $\Diamond\phi$ and $\phi\ \mathsf{U}\ \psi$ are path formulae.*

For example, $\mathsf{INTEND}(\mathsf{A}\Diamond go(paris))\ \mathsf{U}\ \neg\mathsf{BEL}(go(paris, summer))$, expressing that the agent will intend inevitably (A) for every course of action, eventually ($\Diamond$) going to Paris until (U) he does not believe go to Paris in summer, is a well formed formula.

4.2 Semantics

The semantics of the intentional operators BEL, DES and INTEND is adopted from Bordini et al. [1]. First an auxiliary function for getting the atoms (at) subject of an achieve goal (+!) in a given intention, is defined: required:

$$\begin{aligned} agoals(\top) &= \{\}, \\ agoals(i[p]) &= \begin{cases} \{at\} \cup agoals(i) & \text{if } p = +!at : ct \leftarrow h, \\ agoals(i) & \text{otherwise} \end{cases} \end{aligned}$$

Definition 5 (BDI semantics) *The semantics of the* BEL, DES, *and* INTEND *operators for a given agent ag and its circumstance C is:*

$$\mathsf{BEL}_{\langle ag,C\rangle}(\phi) \equiv ag_{bs} \models \phi$$

$$\mathsf{INTEND}_{\langle ag,C\rangle}(\phi) \equiv \phi \in \bigcup_{i \in C_I} agoals(i) \vee \bigcup_{\langle te,i\rangle \in C_E} agoals(i)$$

$$\mathsf{DES}_{\langle ag,C\rangle}(\phi) \equiv \langle +!\phi, i\rangle \in C_E \vee \mathsf{INTEND}(\phi)$$

If the agent ag and his circumstance C are explicit, we simply write $\mathsf{BEL}(\phi)$, $\mathsf{DES}(\phi)$, and $\mathsf{INTEND}(\phi)$. So an agent ag is said to believe the atomic formula ϕ, if ϕ is a logical consequence of the beliefs bs of ag. An agent is said to intend the atomic formula ϕ, if ϕ is the subject of an achieve goal in the active intentions of the agent (C_I) or in his suspended intentions associated to events to be processed (C_E). An agent is said to desire the atomic formula ϕ, if there is an event in C_E which trigger is an achieve goal about ϕ or if ϕ is intended.

The semantics of the temporal operators: next($\bigcirc$), eventually ($\Diamond$), and until (U), as well as the path quantifier inevitable (A), required a Kripke structure $\langle S, R, L\rangle$ where S is a set of states, R is a serial relation on $S \times S$ and L is a labelling or a valuation function for formulae in the states. The transition system of $AgentSpeak(L)$ induce a Kripke structure:

Definition 6 ($AgentSpeak(L)$ Kripke structure) *$K = \langle S, R, V\rangle$ is a Kripke structure specified by the transition system (Γ) defining the $AgentSpeak(L)$ operational semantics rules:*

- *S is a set of agent configurations $\langle ag, C, M, T, s\rangle$.*
- *$R \subseteq S^2$ is a serial relation s.t. for all $(c_i, c_j) \in R$, $(c_i, c_j) \in \Gamma$ or $c_i = c_j$.*
- *V is a valuation function over the intentional and temporal operators, defined after their semantics (see definitions 5 and 7), e.g., $V_{\mathsf{BEL}}(c, \phi) \equiv \mathsf{BEL}(phi)$ at the configuration $c = \langle ag, C, M, T, s\rangle$, etc.*

As usual, $x = c_0, \ldots, c_n$ denotes a path in the Kripke structure, i.e., a sequence of configurations s.t. for all $c_i \in S$, $(c_i, c_{i+1}) \in R$. The expression x^i denotes the suffix of path x starting at configuration c_i.

Definition 7 (Temporal semantics) *The semantic of the state formulae is defined for a given current configuration $c_i \in K$:*

$$
\begin{aligned}
&K, c_i \models \mathsf{BEL}(\phi) \Leftrightarrow \phi \in V_{\mathsf{BEL}}(c_i, \phi)\\
&K, c_i \models \mathsf{INTEND}(\phi) \Leftrightarrow \phi \in V_{\mathsf{INTEND}}(c_i, \phi)\\
&K, c_i \models \mathsf{DES}(\phi) \Leftrightarrow \phi \in V_{\mathsf{DES}}(c_i, \phi)\\
&K, c_i \models \mathsf{E}\phi \Leftrightarrow \exists x^i \exists c_{j \geq i} \in x^i \ \textit{s.t.}\ K, c_j \models \phi\\
&K, c_i \models \mathsf{A}\phi \Leftrightarrow \forall x^i \exists c_{j \geq i} \in x^i \ \textit{s.t.}\ K, c_j \models \phi
\end{aligned}
$$

The semantic of the path formulae is defined as follows:

$$
\begin{aligned}
&K, c_i \models \phi \Leftrightarrow K, x^i \models \phi, \textit{where } \phi \textit{ is a state formula}\\
&K, c_i \models \bigcirc\phi \Leftrightarrow K, x^{i+1} \models \phi\\
&K, c_i \models \Diamond\phi \Leftrightarrow \exists c_{j \geq i} \in x^i \ \textit{s.t.}\ K, c_j \models \phi\\
&K, c_i \models \phi \ \mathsf{U}\ \psi \Leftrightarrow (\exists c_{k \geq i} \ \textit{s.t.}\ K, x^k \models \psi \wedge \forall c_{i \leq j < k} \quad K, x^j \models \phi)\\
&\qquad\qquad \vee \quad (\exists c_{j \geq i} \quad K, x^j \models \phi).
\end{aligned}
$$

Observe that the semantics of until corresponds to weak until (ψ can never occur). Once satisfaction over state and path formulae has been defined, we can define satisfaction and validity over *AgentSpeak*(L) runs.

Definition 8 (Run) *Given an initial AgentSpeak(L) configuration c_0, the run K_Γ^0 denotes the sequence of configurations $c_0, c_1, \ldots$ such that $\forall i \geq 1, c_i = \Gamma(c_{i-1})$.*

Definition 9 (Satisfaction over runs) *Given an AgentSpeak(L) run K_Γ^0 the property $\phi \in CTL_{AgentSpeak(L)}$ is satisfied if $\forall i \geq 0, K_\Gamma^0, c_i \models \phi$.*

Definition 10 (Validity) *A property $\phi \in CTL_{AgentSpeak(L)}$ is valid if for any initial configuration $K_\Gamma^0, c_0 \models \phi$, e.g., if it is satisfied for any run of any agent.*

5 Results about commitment

Now we proceed to show some properties that expressed in the logical specification hold for any *AgentSpeak*(L) agent. We choose to verify blind and single-minded commitment strategies. In principle, open-minded commitment can be verified in a similar way to the single-minded one, but since it is related to emotional aspects of agency more than to policy-based reconsideration (see section 2), it has not been included here. First the no-infinite deferral axiom is verified. It expresses that there is no agent that maintains forever his intentions.

Proposition 1 *An AgentSpeak(L) agent satisfies the axiom of no-infinite deferral:* $\mathsf{INTEND}(\phi) \Rightarrow \mathsf{A}\Diamond(\neg\mathsf{INTEND}(\phi))$.

Proof. Assume $K, c_0 \models \mathsf{INTEND}(\phi)$, then given the definition for INTEND (definition 5), there is a plan $p \in C_I \cup C_E$ with head $+!\phi$ at c_0. The non-infinite deferral axiom expresses that for all runs K^0_Γ eventually this plan will be removed from C_I (active intentions) and C_E (suspended intentions). While p is being executed successfully, three runs are possible given the transition rules $ClrInt_X \in \Gamma$ (table 2): i) $ClrInt_3$ applies when the body of p is not empty, nothing is cleaned; ii) $ClrInt_2$ applies when the plan p, with an empty body, is at the top of an intention i, then p is dropped from i; $ClearInt_1$ applies when the intention i has only a plan p with and empty body, the whole intention i is dropped. Given the finite nature of the plans (section 3.1), if everything goes right, conditions (ii) or (iii) are eventually reached. If something goes wrong with p, a failure mechanism is activated by an event of the form $\langle -!\phi, i[p]\rangle$ resulting in p being dropped. Although [2] only formalizes failures in finding relevant plans (Rel_2), which discards suspended intentions in C_E, other forms of failure detection have been considered in the context of intentional learning [7, 8]. By successful or failed execution of the plans every adopted intention is eventually dropped. □

Proposition 2 *An AgentSpeak(L) agent does not satisfy the axiom of blind commitment.*

Proof. Given that *AgentSpeak(L)* agents satisfy the no-infinite deferral property (proposition 1) and the weak until semantics, the blind commitment axiom is reduced to (a similar reduction is used in Rao et al. [14]):

$$\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \Rightarrow \mathsf{A}\Diamond\mathsf{BEL}(\phi)$$

Consider an initial configuration c_0 s.t. $ag = \langle bs, ps\rangle$ where $bs = \{\}$ and $ps = \{+b(t_1) : \top \leftarrow p(t_2).\ \ +!p(t_2) : \top \leftarrow +b(t_3).\}$. Suppose that from perception of the environment a is added $ag_{bs} = \{b(t_1)\}$. An event is generated by this belief update, so that $C_E = \{\langle +b(t_1), \top\rangle\}$. Then following the semantic rules defining Γ, $SelEv_1$, Rel_1, $ApplPl_1$, are applied obtaining a configuration where $C_I = \{[+b(t_1) : \top \leftarrow !p(t_2).]\}$ and $C_E = \{\}$. Then proceeding with the rules $SelAppl$, $ExtEv$, $SelInt_1$, $AchvGl$ a configuration where $C_E\langle +!p(t_2), +b(t_1) : \top \leftarrow \top\rangle$, $C_I = \{\}$ is obtained. At this configuration c', $K, c' \models \mathsf{DES}(p(t_2))$ (Definition. 5). If we apply then $SelEv_1$, Rel_1, $AppPl_1$, $SelAppl$, a configuration where $C_I = \{[+!p(t_2) : \top \leftarrow\ +b(t_3).]\}$ and $C_E = \{\}$, is obtained. In this configuration c'' $K, c'' \models \mathsf{INTEND}(p(t_2))$ (Definition. 5). Then proceeding with $IntEv$, $SelInt_1$, $AddBel$ gets $C_E = \langle +b(t_3), \top\rangle$, $ag_{bs} = \{b(t_1)\}$ and $C_I = \{[+b(t_1) : \top \leftarrow \top \ddagger +!p(t_2) : \top \leftarrow \top]$ and $bs = \{b(t_1), b(t_3)\}$. The intention about $p(t_2)$ is maintained. Observe that the plan bodies in the intention are empty, so the $ClrInt$ rules will discard the whole intention, so that at the next configuration c''', $K, c''' \models \neg\mathsf{INTEND}(p(t_2))$ and $K, c''' \models \neg\mathsf{BEL}(p(t_2))$. By counter-example $\mathsf{INTEND}(\mathsf{A}\Diamond(\phi)) \implies (\mathsf{A}\Diamond(\mathsf{BEL}(\phi))$ is not valid. □

In fact, our agents do neither satisfy the extended blind commitment axiom (eq. 1), since the agent did not keep its intention about $p(t_2)$ until she believed it.

This reasoning is similar to the demonstration of intention-belief incompleteness (AT2) for *AgentSpeak*(L) [1].

Proposition 3 *AgentSpeak*(L) *agents satisfy a limited single-minded commitment:* $\mathsf{INTEND}(\mathsf{A}\Diamond\phi) \implies \mathsf{A}(\mathsf{INTEND}(\mathsf{A}\Diamond\phi)\ \mathsf{U}\ \neg\mathsf{BEL}(\mathsf{E}\Diamond\phi))$.

Proof. This case is similar to the no-infinite deferral demonstration. Assume the agent $\mathsf{INTEND}(\mathsf{A}\Diamond\phi)$ at c_0, then there is a plan $p \in C_I \cup C_E$ with head $+!\phi$ at c_0. If there is a configuration $c_k \geq 0$ where $\neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)$ (the weak until definition has been adopted), then $K, x^{0,\ldots,k} \models \mathsf{INTEND}(\mathsf{A}\Diamond\phi)$. Following the no-infinite-deferral demonstration, in the failure cases the agent will eventually satisfy $\bigcirc\neg\mathsf{INTEND}(\phi)$ because Rel_2 which means that for an event $\langle te, i[+!\phi : c \leftarrow h.]\rangle$ there were not relevant plans and the associated intention will be discarded, e.g., there is not a path of configurations where eventually ϕ, so that it is rational to drop $\mathsf{INTEND}_{\langle ag,C\rangle}(\phi)$. The case of no-infinite deferral by successful execution of intentions covers the second condition of the weak until $\neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)$ does not occur. □

This is a limited form of single-minded commitement because $\neg\mathsf{BEL}(\mathsf{E}\Diamond\phi)$ is not represented explicitly by the agent. In fact, the agent can not continue intending ϕ because there are no plans to do it and the full intention fails. It is also limited because of intention-belief incompleteness that can be avoided dropping the close world assumption [1]; or using the intentional and temporal definitions for studying the necessary conditions in the operational semantics and definition of the agents to warranty the expected properties of intentions, e.g., in the case of intentions, what does it mean in terms of *AgentSpeak*(L) to be equivalent to a KD modal system?

Given that, *AgentSpeak*(L) agents are not blind committed, intentional learning [7–10] provides a third alternative approach to achieve a full single-minded commitment. The idea is that the agents can learn, in the same way they learn the successful adoption of plans as intentions, the reasons behind a plan adoption that lead to an intention failure. In this way a policy-based reconsideration can be approached.

6 Conclusion

We have extended the methodology proposed by Bordini et al. [1] to reason about *AgentSpeak*(L) agents. Then we proved that any *AgentSpeak*(L) agent is not blindly committed, but follows a limited form of single-minded commitment. The main limitations for these agents are intention-belief incompleteness and the lack of a explicit representation for abandoning reasons. Guerra et al. [8, 9] have suggested that intentional learning provides a solution for the latter, enabling a policy-based reconsideration.

Interestingly, the degree of boldness and cautiousness for a given agent is something hard to define. It is well known [11] that in dynamic environments a very cautious agent performs better than a bold one; and inversely, in static environments boldness pays better. The relevance of learning intentionally is

that the right degree of cautioness is learned by the agents, instead of being established once and forever by the programmers. An extended $AgentSpeak(L)$ operational semantics that deals with intentional learning, for both incremental and batch inductive methods, has been proposed [10]. So, it is possible to arrive to a full theory of commitment and intentional learning, using the techniques presented here. We are currently experimenting these ideas.

As the example of commitment, reconsideration and learning illustrates, the properties verified in this paper are not arbitrary ones. Proving these properties using $CTL_{AgentSpeak(L)}$, prove that they hold for any $AgentSpeak(L)$ agent. This also illustrates the relevance of the specification language proposed in this paper, to bring $AgentSpeak(L)$ closer to its philosophical foundation and to extend our computational theories of practical reasoning.

Acknowledgments. The first and third authors are supported by Conacyt CB-2007-1 (project 78910) funding for this research. The second author is supported by Conacyt scholarship 214783.

References

1. Bordini, R.H., Moreira, Á.F.: Proving BDI properties of agent-oriented programming languages. Annals of Mathematics and Artificial Intelligence 42, 197–226 (2004)
2. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming Multi-Agent Systems in AgentSpeak using Jason. Wiley, England (2007)
3. Bratman, M.: Intention, Plans, and Practical Reason. Harvard University Press, Cambridge (1987)
4. Cohen, P., Levesque, H.: Intention is choice with commitment. Artificial Intelligence 42(3), 213–261 (1990)
5. Dastani, M., van Riemsdijk, M.B., Meyer, J.C.: A grounded specification language for agent programs. In: AAMAS '07. ACM, New York, NY, pp. 1-8 (2007)
6. Emerson, A.: Temporal and modal logic. In: van Leeuwen, J., (ed.) Handbook of Theoretical Computer Science, pp. 996–1072. Elsevier, Amsterdam (1990)
7. Guerra-Hernández, A., El-Fallah-Seghrouchni, A., Soldano, H.: Learning in BDI Multi-agent Systems. In: Dix, J., Leite, J. (eds.) CLIMA IV. LNCS, vol. 3259, pp. 218–233. Springer, Heidelberg (2004)
8. Guerra-Hernández, A., Ortíz-Hernández, G.: Toward BDI sapient agents: Learning intentionally. In: Mayorga, R.V., Perlovsky, L.I. (eds.) Toward Artificial Sapience: Principles and Methods for Wise Systems, pp. 77–91. Springer, London (2008)
9. Guerra-Hernández, A., Castro-Manzano, J.M., El-Fallah-Seghrouchni, A.: Toward an AgentSpeak(L) theory of commitment and intentional learning. In: Gelbuch, A., Morales, E.F. (eds.), MICAI 2008. LNCS, vol. 5317, pp. 848–858, Springer-Verlag, Berlin Heidelberg (2008)
10. Guerra-Hernández, A., Ortíz-Hernández, G., Luna-Ramírez, W.A.: Jason smiles: Incremental BDI MAS learning. In: MICAI 2007 Special Session, IEEE, Los Alamitos (In press)
11. Kinny, D., Georgeff, M.P.: Commitment and effectiveness of situated agents. In: Proceedings of the twelfth international joint conference on artificial intelligence (IJCAI-91), Sydney, Australia (1991)

12. Rao, A.S., Georgeff, M.P.: Modelling Rational Agents within a BDI-Architecture. In: Huhns, M.N., Singh, M.P., (eds.) Readings in Agents, pp. 317–328. Morgan Kaufmann (1998)
13. Rao, A.S.: AgentSpeak(L): BDI agents speak out in a logical computable language. In: de Velde, W.V., Perram, J.W. (eds.) MAAMAW. LNCS, vol. 1038, pp. 42–55. Springer, Heidelberg (1996)
14. Rao, A.S., Georgeff, M.P.: Decision procedures for BDI logics. Journal of Logic and Computation 8(3), pp. 293–342 (1998)
15. Singh, M.P.: A critical examination of the Cohen-Levesque Theory of Intentions. In: Proceedings of the European Conference on Artificial Intelligence (1992).
16. Singh, M.P., Rao, A.S., Georgeff, M.P.: Formal Methods in DAI: Logic-Based Representation and Reasoning. In: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, pp. 331–376. MIT Press, Cambridge (1999)
17. Wooldridge, M.: Reasoning about Rational Agents. MIT Press, Cambridge (2000)