

Funciones de Selección Cualitativas y Cuantitativas para la Revisión de Planes en Agentes Inteligentes.

Gerardo Parra
Departamento de Ciencias de la Computación
UNIVERSIDAD NACIONAL DEL COMAHUE

Guillermo R. Simari
Departamento de Ciencias e Ingeniería de la Computación
UNIVERSIDAD NACIONAL DEL SUR

e-mail: gparra@uncoma.edu.ar, grs@cs.uns.edu.ar

Palabras Clave: INTELIGENCIA ARTIFICIAL, PLANNING, DINÁMICA DE CREENCIAS.

Resumen

Los agentes inteligentes autónomos, por su proactividad, se ven obligados a considerar la satisfacción de sus metas a través de un conjunto estructurado de acciones que conforman un plan. El modelo BDI (*Belief, Desires and Intentions*) para representar el conjunto cognitivo de un agente es una posibilidad interesante que permite estudiar el problema que introduce el dinamismo natural del entorno en el que un plan particular se desenvuelve.

El dinamismo del entorno provoca que algunos de los planes deban ser modificados para poder alcanzar las metas finales. Es esta una actividad de replaneamiento o revisión del plan.

En esta propuesta de trabajo postulamos la conveniencia de adoptar el punto de vista del área de Dinámica de Creencias al considerar la actividad de replaneamiento de un agente inteligente. En trabajos previos se ha introducido un modelo para representar expansiones, contracciones y revisiones de grafos de planning. La operación de revisión ocurre cuando es necesario remover algunas piezas del plan global y reemplazarlas por subplanes alternativos que permitan llevar a cabo el plan de manera exitosa. En este trabajo, presentamos funciones de selección basadas en criterios cualitativos y cuantitativos que permiten determinar las piezas que deben ser removidas y guiar, de este modo, toda la operación.

1. Introducción

Los agentes inteligentes autónomos, debido a su proactividad, se ven obligados a considerar la satisfacción de sus metas a través de un conjunto estructurado de acciones que conforman un plan. El modelo BDI (*Belief, Desires and Intentions*)[GPP⁺99] para representar el conjunto cognitivo de un agente es una posibilidad interesante que permite estudiar el problema que introduce el dinamismo natural del entorno en el que un plan particular se desenvuelve. El entorno corriente, el mundo actual del agente, es representado con un modelo de creencias (*beliefs*) adecuado. Las metas del agente representan sus deseos (*desires*) y describen en forma parcial estados del entorno preferidos. Finalmente, los planes para alcanzar alguno de aquellos estados constituyen, en cierta forma, las intenciones (*intentions*).

El dinamismo del entorno provoca que algunos de los planes deban ser modificados para poder alcanzar las metas finales. Esta actividad de replaneamiento puede considerarse, una revisión del mismo. Ciertas partes pueden ser conservadas, pero otras deben ser removidas y reemplazadas por subplanes convenientes que ofrezcan la posibilidad de éxito para el plan global.

Este trabajo postula la conveniencia de adoptar el punto de vista del área de Dinámica de Creencias[AGM85, Gär88, Han96] al considerar la actividad de replaneamiento de un agente inteligente. En trabajos previos[PS01, PS02a, PS02b] se ha introducido un modelo para representar expansiones, contracciones y revisiones de grafos de planning. La operación de revisión ocurre cuando es necesario remover algunas piezas del plan global y reemplazarlas por subplanes alternativos que permitan llevar a cabo el plan de manera exitosa. En este trabajo, introducimos mecanismos de selección que permiten determinar las piezas que deben ser removidas y guiar, de este modo, toda la operación.

El artículo está organizado de la siguiente manera. En la sección 2 se presentan los principales conceptos relacionados al área de planning en Inteligencia Artificial. Se analiza un dispositivo de planning particular y se trata, en este contexto, la construcción de grafos de planning. En la siguiente sección presentamos los aspectos más relevantes relacionados con la dinámica de creencias. La sección 4 contiene las principales contribuciones de este trabajo. Se analiza el operador de revisión de grafos de planning y se presentan funciones de selección basadas en criterios cualitativos y cuantitativos. Finalmente, la sección 5 incluye las conclusiones del trabajo.

2. Planeamiento. Graphplan

El objetivo central del área de planning en el contexto de Inteligencia Artificial es construir algoritmos que hagan posible a un agente elaborar un curso de acción para lograr sus metas. El resultado producido por un dispositivo de planning es una secuencia de acciones las cuales, cuando son ejecutadas en un mundo que satisface la descripción del estado inicial, lograrán la obtención de la meta. Existe una amplia variedad de lenguajes para representar el mundo, las metas del agente y las acciones posibles. En este trabajo adoptamos la representación STRIPS[FN71] como lenguaje de representación.

La representación STRIPS describe el estado inicial del mundo mediante un conjunto completo de literales básicos (*ground*) y las metas son definidas como una conjunción proposicional. La

teoría de dominio, es decir, la descripción formal de las acciones disponibles para el agente, completa la descripción del problema de planning.

Cada acción se describe mediante dos fórmulas: la precondition y el efecto o poscondición. Ambas están constituídas por una conjunción de literales y definen una función de transición de un mundo a otro. Una acción puede ser ejecutada en cualquier mundo w que satisfaga la fórmula de la precondition. El resultado de ejecutar una acción en un mundo w es especificado tomando la descripción de w , adicionando cada literal de la poscondición de la acción ejecutada y eliminando literales contradictorios.

Lo expuesto hasta aquí caracteriza un problema de planning clásico. Es importante destacar que existen muchas suposiciones que simplifican el problema: tiempo atómico, no existen eventos exógenos, los efectos de las acciones son determinísticos, omnisciencia por parte del agente, etc.

Uno de los dispositivos de planning más interesantes es Graphplan[BK95, BK97, Wel99]. El funcionamiento de Graphplan alterna entre dos fases: la construcción del grafo de planning y la extracción de la solución. La primera fase construye, a través de sucesivas etapas, un grafo de planning hacia adelante en el tiempo hasta que se logra una condición necesaria (pero que puede ser insuficiente) para la existencia de un plan. Luego, la fase de extracción de solución realiza un recorrido hacia atrás sobre el grafo, buscando un plan que resuelva el problema. Si no es hallada una solución, el ciclo se repite llevando a cabo una nueva etapa en la construcción del grafo de planning.

A continuación describimos, de manera resumida, cada una de las fases del funcionamiento de Graphplan. Una presentación detallada puede encontrarse en [Wel99, PS02b].

2.1. Grafos de Planning

El grafo de planning contiene dos tipos de nodos: nodos de proposición y nodos de acción, organizados en niveles. Los niveles con numeración par contienen nodos de proposición (es decir, literales *ground*) y, en particular, el nivel cero consiste precisamente de las proposiciones que son verdaderas en el estado inicial del problema de planning. Los nodos presentes en niveles con numeración impar corresponden a instancias de acción. Existe uno de tales nodos por cada instancia de acción cuyas condiciones estén presentes (y sean mutuamente consistentes) en el nivel previo.

Los nodos correspondientes a instancias de acción están conectados mediante arcos a los nodos de proposición (en el nivel anterior) que constituyen las condiciones de la acción. Existen arcos adicionales (arcos de poscondición) que conectan los nodos de acción con los nodos de proposiciones (en el nivel siguiente) que se hacen verdaderas por efecto de la acción.

Es importante destacar que el grafo de planning representa acciones que pueden desarrollarse *en paralelo* en cada nivel de acciones. Sin embargo, el hecho que dos acciones estén presentes en el mismo nivel de un grafo de planning no significa que sea posible ejecutar ambas a la vez.

Central a la eficiencia de Graphplan es la inferencia de una relación binaria de exclusión mutua, denominada *mutex*, entre nodos presentes en el mismo nivel. La relación se define recursivamente

como sigue:

- Dos instancias de acción en el nivel i son *mutex* si ocurre alguno de los casos siguientes:
 - el efecto de una acción es la negación del efecto de la otra (*Efectos Inconsistentes*)
 - el efecto de una acción elimina la precondition de la otra (*Interferencia*)
 - las acciones tienen precondiciones que son mutuamente exclusivas a nivel $i - 1$ (*Necesidades Conflictivas*).
- Dos proposiciones en el nivel i son *mutex* si una es la negación de la otra o si todas las formas de arribar a estas proposiciones (es decir, las acciones en el nivel $i - 1$) son *mutex* tomadas de a dos (*Soporte Inconsistente*).

A continuación presentamos, a modo de ejemplo, la especificación STRIPS del *dinner-date problem*[Wel99].

```
Condiciones Iniciales: (and (basura) (manosLimpias) (silencio))
Meta:                  (and (cena) (regalo) (not (basura)))
Acciones:

Cocinar                :precondition (manosLimpias)
                       :effect (cena)
Envolver               :precondition (silencio)
                       :effect (regalo)
Llevar_a_mano          :precondition
                       :effect and ( (not (basura)) (not (manosLimpias)))
Llevar_en_carretilla  :precondition
                       :effect and ( (not (basura)) (not (silencio)))
```

Figura 1: *Dinner-date problem*.

El *dinner-date problem* consiste en encontrar un plan para preparar una cita sorpresa para nuestra amada que se encuentra durmiendo. La meta del problema es sacar las bolsas de basura, preparar la cena y envolver el regalo. Existen cuatro acciones posibles: `cocinar`, `envolver`, `llevar_a_mano` y `llevar_en_carretilla`. `Cocinar` requiere `manosLimpias` y produce `cena`. `Envolver` tiene como precondition `silencio` porque es una sorpresa y produce `regalo`. La acción `llevar_a_mano` elimina `basura` pero, el contacto manual con las bolsas, niega `manosLimpias`. La acción final, `llevar_en_carretilla`, también elimina `basura` pero, a causa del desplazamiento ruidoso, niega `silencio`. Inicialmente, tenemos `manosLimpias`, la casa tiene `basura` y está en `silencio`. Las demás proposiciones se consideran falsas.

El grafo de planning se construye de la siguiente manera. Todas las condiciones iniciales son ubicadas en el primer nivel de proposiciones (nivel cero) del grafo. Construir un nivel de acción genérico consiste en lo siguiente. Para cada operador y para cada forma de instanciar las precondiciones de ese operador a proposiciones del nivel previo, se inserta un nodo de acción si

no existen dos precondiciones que sean mutuamente exclusivas. Además, se insertan todas las acciones de mantenimiento (acciones nulas) y los arcos de las precondiciones. Luego se chequea la relación de exclusión mutua entre los nodos de acción y se crea una lista, que mantiene esas relaciones, para cada acción. Para crear un nivel de proposiciones genérico, simplemente se tienen en cuenta todos los efectos de las acciones presentes en el nivel previo (incluyendo las acciones de mantenimiento) y se los ubica en el siguiente nivel como proposiciones, conectándolos vía los apropiados arcos de poscondición. Finalmente, dos proposiciones son marcadas como mutuamente exclusivas, si todas las formas de generar la primera son mutuamente exclusivas con respecto a todas las formas de generar la segunda de las proposiciones marcadas.

Como condición necesaria para la existencia de un plan se debe verificar que la totalidad de los literales de la meta estén presentes en el máximo nivel proposicional alcanzado y que ninguno de ellos sea *mutex* con cada uno de los otros. En este caso, la segunda fase de Graphplan es ejecutada: la extracción de la solución.

2.2. Extracción de la Solución

La fase de extracción de la solución se aboca a la tarea de hallar un plan considerando cada una de las n submetas (términos de la conjunción) que conforman la meta. Para cada uno de tales literales presentes en un nivel i , Graphplan elige una acción a en el nivel $i - 1$ tal que produzca esa submeta. Este es un punto de *backtracking*: si más de una acción produce una submeta, entonces Graphplan debe considerar a todas ellas con el fin de asegurar completitud. Si a es consistente (es decir, no *mutex*) con todas las acciones que han sido escogidas hasta ahora en este nivel, entonces se procede a la siguiente submeta. De lo contrario, se realiza *backtracking* a la elección previa.

Una vez que ha encontrado un conjunto consistente de acciones en el nivel $i - 1$, Graphplan trata recursivamente de encontrar un plan para el conjunto formado a partir de la unión de todas las precondiciones de las aquellas acciones en el nivel $i - 2$. El caso base de la recursión es el nivel cero: si las proposiciones están presentes en ese nivel entonces Graphplan ha encontrado una solución. De lo contrario, si el *backtracking* falla para todas las combinaciones de las posibles acciones que soportan cada submeta (para cada nivel), entonces Graphplan extiende el grafo de *planning* con un nuevo nivel de acción y un nuevo nivel de proposición y luego intenta nuevamente la extracción de una solución.

Es importante destacar que Graphplan genera un plan inherentemente paralelo o parcialmente ordenado. Las acciones seleccionadas para el último nivel de acción del ejemplo planteado, *llevar a mano* y *envolver* pueden ser ejecutadas en cualquier orden y producirán el mismo efecto. Así, si uno desea una secuencia de acciones totalmente ordenada como plan definitivo, uno podría elegir arbitrariamente: *cocinar*, *llevar a mano* y *envolver*.

Una vez obtenido el plan definitivo, surge inmediatamente el siguiente interrogante. ¿Qué sucedería si, una vez ejecutado el plan, el agente descubre que alguna de las acciones necesarias para lograr la meta no produjo los resultados esperados? Ante esta situación sería necesaria una tarea de replaneamiento. El mayor problema, desde un punto de vista computacional, consiste en que el grafo de *planning* debe volver a construirse desde cero para intentar encontrar nuevamente la solución. En las secciones subsiguientes, consideraremos la conveniencia de adoptar el punto

de vista del área de Dinámica de Creencias con el fin de intentar simplificar este problema.

3. Dinámica de Creencias

Dinámica de creencias es el proceso por el cual un agente cambia sus creencias, realizando una transición desde un estado epistémico a otro. Cuando tal agente aprende nueva información puede concluir que esta información contradice sus creencias previas. En este caso, el agente debe revisar sus creencias y decidir cuáles tienen que ser abandonadas con el fin de incorporar la nueva información.

Uno de las más fundamentales aproximaciones a la formalización de la dinámica de creencias es el modelo AGM[AGM85]. En este enfoque, los estados epistémicos son representados por conjuntos de creencias que son conjuntos de sentencias cerrados bajo consecuencia lógica.

Notation: Se adopta un lenguaje proposicional \mathcal{L} con un conjunto completo de conectivos booleanos: negación, conjunción, disyunción e implicación. Las fórmulas en \mathcal{L} serán denotadas por letras griegas minúsculas y los conjuntos de sentencias en \mathcal{L} serán denotadas mediante letras mayúsculas. Se emplea un operador de consecuencia Cn . Cn toma un conjunto de sentencias en \mathcal{L} y produce un nuevo conjunto de sentencias. Se asume que el operador Cn satisface las propiedades de *inclusión* ($A \subseteq Cn(A)$), *iteración* ($Cn(A) = Cn(Cn(A))$) y *monotonidad* (si $A \subseteq B$ entonces $Cn(A) \subseteq Cn(B)$).

Sea $\mathbf{K} = Cn(\mathbf{K})$ un conjunto de creencias y α una sentencia en un lenguaje proposicional \mathcal{L} . Los tres principales tipos de operaciones de cambio de creencias son los siguientes[Gär88]:

- **Expansión:** Una nueva sentencia es incorporada a un estado epistémico. Si '+' es un operador de expansión entonces $\mathbf{K} + \alpha$ denota el conjunto de creencias \mathbf{K} expandido por α .
- **Contracción:** Alguna sentencia presente en el estado epistémico es retraída sin incorporar nueva información. Si '-' es un operador de contracción entonces $\mathbf{K} - \alpha$ denota el conjunto de creencias \mathbf{K} contraído por α .
- **Revisión:** Una nueva sentencia es incorporada de manera consistente al estado epistémico. Con el fin de hacer posible esta operación, algunas sentencias deben ser retraídas del estado epistémico original. Si '*' es un operador de revisión entonces $\mathbf{K} * \alpha$ denota el conjunto de creencias \mathbf{K} revisado por α .

La operación de expansión puede ser definida explícitamente tomando la clausura lógica del conjunto de creencias \mathbf{K} unido a la sentencia α : $\mathbf{K} + \alpha = Cn(\mathbf{K} \cup \alpha)$.

No desarrollaremos los postulados de racionalidad propuestos para las operaciones de expansión, contracción y revisión puesto que no son necesarios para el resto del artículo. El lector interesado, puede consultar las referencias citadas en esta sección. Además, una presentación exhaustiva de los diferentes modelos de cambios de creencias puede encontrarse en [Fal99, Par98].

Las operaciones de contracción y revisión pueden ser definidas usando nociones lógicas y algún

mecanismo de selección. Los mecanismos de selección pueden responder a criterios cualitativos o cuantitativos. En los primeros se respeta un cierto orden de preferencia entre las sentencias. Aquí podemos mencionar, entre otros, los operadores basados en *importancia epistémica*[Gär88, Rot92].

En los criterios cuantitativos se utilizan mecanismos de selección basados en nociones numéricas. Entre los enfoques más representativos podemos mencionar a los operadores basados en *sistemas de esferas*[Gro88].

4. Revisión de Planes

Asumamos que un agente descubre que, una de las acciones requeridas en el plan definitivo no pudo ser ejecutada adecuadamente, es decir, no ha dado los resultados esperados. Ante esta situación, una porción del plan debe ser removida y reemplazada por un subplan conveniente que ofrezca la posibilidad de éxito para el plan global.

Analicemos un ejemplo concreto tomando como base el *dinner-date problem* (Figura 1). Supongamos que, una vez construido y ejecutado el plan definitivo, el agente descubre que se ha quemado la cena. Ante tal situación, una posibilidad interesante sería apelar a una nueva acción que produzca el efecto esperado. Asumamos que se dispone, para tal fin, de una acción `llamar_rotiseria` con efecto `cena` y sin precondiciones. El reemplazo de la porción del plan que no produjo los resultados esperados por este subplan alternativo permite al agente seguir el curso de acción establecido. Sin embargo, esto implica, en el contexto de Graphplan, la reconstrucción del grafo de planning desde el nivel cero.

Es importante tener en cuenta que, la construcción del grafo de planning para un problema determinado no es una tarea trivial. Por lo tanto, es imperativo conservar la mayor parte del grafo ante una modificación del problema original.

Con esta motivación, proponemos la definición de una operación de revisión para grafos de planning. En primer lugar, necesitamos tratar a cada acción y a todas sus precondiciones y poscondiciones como una unidad.

Definición 4.1. Un esquema de acción es una terna (Pre, a, Pos) donde, a es una acción, Pre es un conjunto finito de proposiciones que constituyen las precondiciones de a y Pos es un conjunto finito de proposiciones que se verifican como resultado de aplicar la acción a (poscondiciones de a).

Notación: Sea A un esquema de acción. Emplearemos A_{Pre} y A_{Pos} para denotar, respectivamente, las precondiciones y poscondiciones de A .

A continuación, necesitamos especificar cuándo un esquema de acción pertenece a un nivel determinado de un grafo de planning.

Definición 4.2. Sea Π un grafo de planning. Sea n un nodo de proposición o de acción perteneciente al grafo. Mediante la función $Lev_{\Pi}(n)$ se indica el nivel correspondiente al nodo n en el grafo Π . Además, $Lev_{\Pi}(n)$ es indefinido si y sólo si el nodo n no pertenece a Π .

Definición 4.3. Sea Π un grafo de planning y sea $A = (Pre, a, Pos)$ un esquema de acción. Decimos que A pertenece a nivel i a Π si y sólo si $Lev_{\Pi}(a) = i$, cada uno de los elementos de los conjuntos Pre y Pos existen como nodos en el grafo y si existen los arcos que modelan las relaciones correspondientes.

La tarea básica del operador de revisión es obtener, dado un grafo de planning Π y un esquema de acción A , un nuevo grafo de planning Π' . Este nuevo grafo tiene dos características principales: el esquema A pertenece a nivel i al grafo Π' y, posiblemente, esquemas de acción adicionales han sido removidos de Π' .

La operación de revisión de grafos de planning debería entenderse como un proceso mediante el cual se reemplaza un esquema de acción B por un esquema A que contiene los efectos o poscondiciones de B . Sin embargo, es posible que el esquema a reemplazar y el nuevo esquema difieran en sus precondiciones.

Varios esquemas de acción pueden ser candidatos a ser reemplazados. Una función de selección determina, dentro de este conjunto, el esquema de acción escogido. A continuación, definiremos formalmente las nociones de conjuntos de esquemas de acción *pos-equivalentes*, función de selección y revisión de grafos de planning.

Definición 4.4. Sea Π un grafo de planning y sea A un esquema de acción. El conjunto de esquemas de acción *pos-equivalentes* a nivel i con respecto a A , denotado $\Pi\Psi^i A$, se define de la siguiente manera.

$$\Pi\Psi^i A = \{E : E \text{ pertenece a nivel } i \text{ a } \Pi \text{ y } E_{Pos} \subseteq A_{Pos}\}.$$

El conjunto de esquemas pos-equivalentes a nivel i con respecto a A , está formado por los esquemas de acción de nivel i cuyas poscondiciones están incluidas en las poscondiciones del esquema de acción A .

Definición 4.5. Sea Π un grafo de planning. Una *función de selección para Π* es una función γ tal que, para cualquier esquema de acción A , se verifica que:

$$\gamma(\Pi\Psi^i A) = \begin{cases} \{E' : E' \in \Pi\Psi^i A\} & \text{si } \Pi\Psi^i A \neq \emptyset \\ \emptyset & \text{si } \Pi\Psi^i A = \emptyset \end{cases}$$

La función de selección γ escoge un elemento de $\Pi\Psi^i A$ si el conjunto no es vacío. En caso contrario, la función de selección devuelve el conjunto vacío.

Definición 4.6. Sea Π un grafo de planning, A un esquema de acción y γ una función de selección para Π . La operación de *revisión* de Π con respecto a A , denotada como $\Pi \otimes^i A$, se define como:

$$\Pi \otimes^i A = (\Pi \ominus^i \gamma(\Pi\Psi^i A)) \oplus^i A$$

De acuerdo a esta definición, para realizar la revisión a nivel i de un grafo de planning Π con respecto a un esquema de acción A debemos, en primer lugar, contraer a nivel i por el esquema de acción $\gamma(\Pi\Psi^i A)$ y luego, expandir a nivel i por el esquema de acción A .

4.1. Funciones de Selección Cualitativas

En esta aproximación, la idea fundamental es que la función de selección γ escoja los *peores* elementos de $\Pi\Psi^i A$. Esto puede concretarse asumiendo que existe un *ordenamiento* de los esquemas de acción pos-equivalentes de $\Pi\Psi^i A$ que puede ser usado para determinar los elementos mínimos.

Asumimos que, para todo esquema de acción A , existe una relación reflexiva y transitiva ' \leq ' que conforma un orden total sobre los elementos de $\Pi\Psi^i A$ ¹. De esta manera, cuando $\Pi\Psi^i A$ es no vacío, esta relación puede emplearse para definir una función de selección que seleccione los elementos menores del ordenamiento.

Definición 4.7. Sea γ una función de selección para Π . Decimos que γ es una *función de selección cualitativa* si se verifica que:

$$\gamma(\Pi\Psi^i A) = \{E' \in \Pi\Psi^i A : E' \leq E'' \text{ para todo } E'' \in \Pi\Psi^i A\}$$

De esta manera, la función γ selecciona los menores esquemas de acción, de acuerdo al ordenamiento impuesto por la relación ' \leq '.

4.2. Funciones de Selección Cuantitativas

Las funciones de selección cualitativas utilizan criterios basados en la calidad, peso o importancia de los esquemas de acción pos-equivalentes. Un enfoque alternativo es considerar criterios *numéricos* para determinar qué esquemas remover.

En una primera aproximación en este sentido, proponemos una clase de funciones de selección que escogen los esquemas de acción con el mínimo número de pos-condiciones. De esta manera, se trata de minimizar el impacto causado en el grafo de planning al remover este esquema de acción.

Definición 4.8. Sea γ una función de selección para Π . Decimos que γ es una *función de selección pos-condicional minimal* si, para cualquier otra función de selección γ' para Π , se verifica que:

$$|\gamma(\Pi\Psi^i A)_{Pos}| \leq |\gamma'(\Pi\Psi^i A)_{Pos}|$$

De acuerdo a la Definición 4.8, la función de selección γ escoge los esquemas de acción con el mínimo número de pos-condiciones.

Una segunda aproximación, consiste en analizar una clase de funciones que seleccionan los esquemas de acción con el mínimo número de pre-condiciones. La motivación, nuevamente, es tratar de minimizar el impacto causado en el grafo de planning al remover un esquema de acción.

¹En un enfoque menos restrictivo, se podría considerar un orden *parcial* además de un mecanismo de decisión para el caso de elementos incomparables.

Definición 4.9. Sea γ una función de selección para Π . Decimos que γ es una *función de selección pre-condicional minimal* si, para cualquier otra función de selección γ' para Π , se verifica que:

$$|\gamma(\Pi\Psi^i A)_{Pre}| \leq |\gamma'(\Pi\Psi^i A)_{Pre}|$$

Conforme a la Definición 4.9, la función de selección γ escoge los esquemas de acción con el mínimo número de pre-condiciones.

Claramente, las funciones de selección pre-condicionales minimales y pos-condicionales minimales no son equivalentes. El ejemplo a continuación es ilustrativo en este sentido.

Ejemplo 4.1. Sean los esquemas de acción $B = (\{p_1, p_2\}, a_1, \{e_1\})$, $C = (\{\}, a_2, \{e_1, e_2\})$ y $D = (\{p_1\}, a_3, \{e_3\})$ pertenecientes a nivel i al grafo de planning Π . Sea el esquema de acción $A = (\{\}, a, \{e_1, e_2\})$. Luego, el conjunto $\Pi\Psi^i A$ es igual a $\{B, C\}$. Entonces, si γ es una función de selección pos-condicional minimal, se verifica que

$$\gamma(\{B, C\}) = B$$

dado que $|B_{Pos}| \leq |C_{Pos}|$. Si, por el contrario, γ es una función de selección pre-condicional minimal, entonces se verifica que

$$\gamma(\{B, C\}) = C$$

dado que $|C_{Pre}| \leq |B_{Pre}|$.

5. Conclusiones

La operación de revisión de grafos de planning modela situaciones en las cuales es necesario remover algunas piezas del plan global y reemplazarlas por subplanes alternativos que permitan al agente seguir un determinado curso de acción. En este trabajo, hemos propuesto funciones de selección cualitativas y cuantitativas que permiten determinar las piezas del plan que deben ser removidas.

Si bien existen muchos argumentos para sostener que, en los procesos de decisión de un agente, se deben preservar los elementos de mayor importancia, en muchos casos, puede ser necesario y útil contar con métodos cuantitativos. Por ejemplo, cuando se debe decidir qué elementos preservar entre varios elementos que pueden ser incomparables bajo algún criterio cualitativo.

Otro punto a favor de los mecanismos de selección cuantitativos radica en que no es necesario almacenar ninguna información *externa* a los esquemas de acción. Los mecanismos de selección cualitativos requieren que se almacene, además de los esquemas de acción, información de preferencia entre los mismos. Si pensamos en teorías de dominio con miles de acciones, el espacio necesario para almacenar la información de preferencia puede ser considerablemente alto.

Referencias

- [AGM85] Carlos Alchourrón, Peter Gärdenfors, and David Makinson. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *Journal of Symbolic Logic*, (50):510–530, 1985.
- [BF95] A. Blum and M. Furst. Fast planning through planning graph analysis. In *Proceedings of the XIV International Joint Conference of AI*, pages 1636–1642, 1995.
- [BF97] A. Blum and M. Furst. Fast planning through planning graph analysis. *J. Artificial Intelligence*, (90):281–300, 1997.
- [Fal99] Marcelo Falappa. *Teoría de Cambio de Creencias y sus Aplicaciones sobre Estados de Conocimiento*. PhD thesis, Departamento de Ciencias de la Computación, Universidad Nacional del Sur, Bahía Blanca, 1999.
- [FN71] Richard Fikes and Nils Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *J. Artificial Intelligence*, (2), 1971.
- [Gär88] Peter Gärdenfors. *Knowledge in Flux: Modelling the Dynamics of Epistemic States*. MIT Press, Cambridge, England, 1988.
- [GPP⁺99] M. Georgeff, B. Pell, M. Pollack, M. Tambe, and M. Wooldridge. The Belief-Desire-Intention Model of Agency. In M.P.Singh J.P.Müller and A.S. Rao, editors, *Intelligent Agents V*, volume 1555 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, Germany, 1999.
- [Gro88] Adam Grove. Two Modellings for Theory Change. *The Journal of Philosophical Logic*, (17):157–170, 1988.
- [Han96] Sven Owe Hansson. *A Textbook of Belief Dynamics*. Kluwer Academic Press, 1996.
- [Par98] Gerardo Parra. Semi Revisión Plausible en Bases de Ccreencias. Master’s thesis, Departamento de Ciencias de la Computación, Universidad Nacional del Sur, Bahía Blanca, 1998.
- [PS01] G. Parra and G. Simari. Replaneamiento en Agentes Inteligentes. Contracción de Grafos de Planning. In *VII Congreso Argentino de Ciencias de la Computación*, volume 2, pages 1081–1093, Universidad Nacional de la Patagonia Austral - El Calafate - Santa Cruz, 2001.
- [PS02a] G. Parra and G. Simari. Reelaboración de Planes en Agentes Inteligentes. Expansión de Grafos de Planning. *Jornadas Chilenas de Computación 2002*, 2002.
- [PS02b] G. Parra and G. Simari. Replaneamiento en Agentes Inteligentes. Revisión de Grafos de Planning. In *VIII Congreso Argentino de Ciencias de la Computación*, Buenos Aires, 2002.
- [Rot92] Hans Rott. Preferential Belief Change Using Generalized Epistemic Entrenchment. *Journal of Logic, Language and Information*, 1992.
- [Wel99] Daniel S. Weld. Recent Advances in AI Planning. *AI Magazine*, 1999.