



INTELIGENCIA ARTIFICIAL

Proyecto N° 2

Lógica Default: Semántica Operacional

Segundo Cuatrimestre de 2009

El objetivo del proyecto es automatizar el cómputo de extensiones de una teoría default empleando PROLOG. Una teoría default T es un par $\langle W, D \rangle$, donde W es un conjunto de fórmulas de la lógica de predicados (denominadas hechos o axiomas de T) y D es un conjunto de defaults. A modo de simplificación, restringiremos la sintaxis de los axiomas y defaults de acuerdo a la siguiente gramática:

$$\begin{aligned} \langle \text{axioma} \rangle &::= \langle \text{literal} \rangle \{ \text{and} \langle \text{literal} \rangle \}^* \mid \langle \text{implicacion} \rangle \\ \langle \text{implicacion} \rangle &::= \langle \text{antecedente} \rangle \rightarrow \langle \text{literal} \rangle \\ \langle \text{antecedente} \rangle &::= \langle \text{literal} \rangle \{ \text{and} \langle \text{literal} \rangle \}^* \\ \langle \text{literal} \rangle &::= \langle \text{letra_prop} \rangle \mid \neg \langle \text{letra_prop} \rangle \\ \langle \text{letra_prop} \rangle &::= a|b|c|d|\dots|x|y|z \\ \\ \langle \text{default} \rangle &::= \langle \text{prerrequisito} \rangle : \langle \text{justificacion} \rangle / \langle \text{consecuente} \rangle \\ \langle \text{prerrequisito} \rangle &::= \text{true} \mid \langle \text{literal} \rangle \{ \text{and} \langle \text{literal} \rangle \}^* \\ \langle \text{justificacion} \rangle &::= \text{true} \mid \langle \text{literal} \rangle \{ , \langle \text{literal} \rangle \}^* \\ \langle \text{consecuente} \rangle &::= \langle \text{literal} \rangle \{ \text{and} \langle \text{literal} \rangle \}^* \end{aligned}$$

Notar que de acuerdo a la gramática, $\neg\neg a$ **no** es una fórmula válida. Por lo tanto, siempre utilizaremos a en lugar de $\neg\neg a$.

A continuación se listan algunas convenciones de representación:

- deberán definirse en PROLOG operadores ' \neg ', ' and ' y ' \rightarrow ' empleando el predicado predefinido `op/3`. Concretamente, deberán incluirse las siguientes declaraciones en el archivo `.pl` con el código PROLOG implementado:

```
:-op(1050, xfx, user:(->)).
:-op(200, fx, user:(\neg)).
:-op(201, xfy, user:(and)).
```

- se utilizarán los predicados PROLOG `w/1` y `d/1` para representar los conjuntos W y D respectivamente. Por ejemplo, la teoría default T se representará mediante los siguientes hechos PROLOG:

<code>w(a->\neg c).</code>	<code>d(x:(a,\neg c)/a and \neg z).</code>
<code>w(b->c).</code>	<code>d(y:c/b).</code>
<code>w(\neg b->f).</code>	<code>d(a and x:true/e).</code>
<code>w(x).</code>	<code>d(true:b/e).</code>
<code>w(y).</code>	<code>d(b:\neg e/\neg x).</code>
<code>w(x and t ->z).</code>	<code>d(f:g/g).</code>

El cómputo de extensiones de una teoría default (en este caso según la semántica operacional) requiere computar la clausura lógica de conjuntos de fórmulas. En particular, las extensiones asociadas a una teoría default son conjuntos de fórmulas clausurados lógicamente. Dado que la clausura lógica de un dado conjunto de fórmulas es un conjunto infinito, computarlo por completo resulta imposible. En tal sentido, para el presente proyecto abordaremos simplemente el cálculo de los literales presentes en las extensiones, y no de las extensiones completas. Además, la cátedra proveerá de predicados para determinar si una dada fórmula pertenece a la clausura lógica de un dado conjunto, y para determinar el conjunto de literales presentes en una dada clausura lógica. La clave de estos predicados es que computan solo una porción (finita) de la clausura lógica del conjunto en cuestión.

ACLARACIÓN: Dado que, en el contexto de la semántica operacional, el conjunto W de axiomas de una teoría default siempre forma parte del conjunto de fórmulas a clausurar, los predicados provistos por la cátedra consideran implícitamente a los elementos de W .

Sea S un conjunto de consecuentes de defaults (literales y conjunciones de literales), entonces:

- `in_thWUS(+F, +S)` determina si una dada fórmula F (respetando la sintaxis especificada para los axiomas según la gramática) pertenece a $Th(W \cup S)$, donde W es el conjunto de axiomas de la teoría default considerada (especificados mediante el predicado `w/1`).
- `lits_in_thWUS(+S, -LitsInThWUS)` calcula el conjunto `LitsInThWUS` de literales presentes en $Th(W \cup S)$, donde W es el conjunto de axiomas de la teoría default considerada.

La implementación de estos predicados será provista por la cátedra en un archivo `th.pl`, el cual deberá ser consultado por el `.pl` con el código prolog implementado por la comisión. ACLARACIÓN: el archivo `th.pl` ya incluye las declaraciones de los operadores ‘ \neg ’, ‘and’ y ‘ \rightarrow ’ a las que se hizo referencia previamente en el enunciado.

Para el presente proyecto se pide implementar los siguientes predicados:

- `lits_en_extension(-LitsInE, -P)`: retorna en `LitsInE` el conjunto de literales de una extensión E de la teoría default, junto al proceso cerrado y exitoso asociado, P . Un proceso se representará mediante una lista `PROLOG` de defaults. Por practicidad, puede asumirse que los defaults aparecen en la lista en el orden inverso al de aplicación. La implementación de este predicado debe estar basada en la semántica operacional para la Lógica Default. Además, debe implementarse de manera tal que puedan obtenerse todas las extensiones de la teoría mediante el pedido reiterado de respuestas alternativas.

Por ejemplo, en base a la teoría T :

```
?- lits_in_extension(LitsInE,P).
LitsInE=[true, x, y, a, ¬z, ¬t, ¬c, ¬b, f, g, e]
P=[x:(a,¬c)/a and ¬z, a and x:true/e, f:g/g];
LitsInE=[true, x, y, b, c, ¬a, e]
P=[y:c/b, true:b/e];
...
```

- `cauta(+C)`: determina si un dado literal C es una conclusión cauta del sistema.
- `osada(+C)`: determina si in dado literal C es una conclusión osada del sistema.

Condiciones de Entrega

1. El proyecto deberá ser entregado el día miércoles 9 de diciembre de 2009 en el horario de clase.
2. Deberá presentarse un folio plástico (cerrado) conteniendo los siguientes elementos:
 - Un disquete o cd (con el nombre de los integrantes de la comisión) conteniendo el código PROLOG de los programas implementados en un archivo `pr-2.pl`. Se recomienda **incluir múltiples copias**.
 - Un informe impreso completo que documente decisiones de diseño, implementación, restricciones de uso, etc. El informe debe incluir ejemplos de corrida de los predicados pedidos y un listado del código fuente debidamente documentado. **Importante:** Se va a prestar especial atención a los informes.

Opcional

1. Diseñar una teoría default (conjuntos W y D) a ser empleado como (parte) del mecanismo de razonamiento y toma de decisiones de un agente para el juego de rol presentado en el proyecto 1.
2. Implementar un agente que efectivamente emplee la teoría default definida en a , junto a la implementación desarrollada para el presente proyecto, para razonar y tomar decisiones en el juego. Por supuesto, puede emplear el agente desarrollado para el proyecto 1 como base.
3. Documentar apropiadamente en el informe el desarrollo realizado.