Lógica Computacional 2017-2
Práctica 5:Monerías de Prolog

Lourdes del Carmen González Huesca
Roberto Monroy Argumedo
Fernando A. Galicia Mendoza

Facultad de Ciencias, UNAM

Fecha de entrega: Viernes, 28 de abril del 2017

Esta práctica puede ser entregada en equipo: máximo dos personas.

1. Propósito general

1.1. Autómatas finitos

En un archivo llamado `Automatas.pl` realiza los siguientes ejercicios:

1. Considera el siguiente autómata finito determinista $(Q, \Sigma, q_0, \delta, F)$ donde:

   - $Q = \{q_0, q_1, q_2, q_3\}$
   - $\Sigma = \{a, b, c, d, e\}$
   - q_0 es el estado inicial.
   - δ es la función de transición definida como:

$$
\begin{align}
\delta(q_0, a) &= q_1 \tag{1}\\
\delta(q_1, b) &= q_2 \tag{2}\\
\delta(q_2, c) &= q_1 \tag{3}\\
\delta(q_2, d) &= q_3 \tag{4}\\
\delta(q_3, e) &= q_0 \tag{5}\\
& \tag{6}
\end{align}
$$

- $F = \{q_3\}$

Define una relación afd(ℓ) la cual se cumpla syss ℓ es una lista de símbolos aceptados por el autómata descrito.
Por ejemplo:

```
?- afd([a,b,d]).
true.
?- afd([a,b,d,e]).
false.
```

Sugerencia: Define una relación tExt(q, ℓ) que se cumpla syss a partir del estado q es posible llegar a q_3 a través de la cadena ℓ.

2. Considera el siguiente autómata finito no determinista $(Q, \Sigma, q_0, \delta, F)$ donde:

   - $Q = \{q_0, q_1, q_2, q_3\}$
   - $\Sigma = \{0, 1\}$
   - q_0 es el estado inicial.
   - δ es la función de transición definida como:

$$\delta(q_0, 0) = \{q_0\} \tag{1}$$
$$\delta(q_0, 1) = \{q_0, q_1\} \tag{2}$$
$$\delta(q_1, 0) = \{q_2\} \tag{3}$$
$$\delta(q_2, 1) = \{q_3\} \tag{4}$$
$$\tag{5}$$

   - $F = \{q_3\}$

Define una relación afn(ℓ) la cual se cumpla syss ℓ es una lista de símbolos aceptados por el autómata descrito.
Por ejemplo:

```
?- afn([1,1,0,1]).
true.
?- afn([1,1,1,0]).
false.
```

Sugerencia: Define una relación afnGen(ℓ, D, I, F) que represente a cualquier autómata finito no determinista, es decir, ℓ es una cadena de entrada al autómata, D es una lista describiendo la función de transición utilizando una tupla de 3 elementos, I es el estado inicial y F es una lista de estados finales.

1.2. Operadores de corte

En un archivo llamado Corte.pl realiza los siguientes ejercicios utilizando el operador de corte:

1. La relación exp(X,Y,Z) la cual se cumpla syss $Z = X^Y$.

2. La relación fact(X,Y) la cual se cumpla syss $Y = X!$.

3. La relación `ackerman(X,Y,Z)` la cual se cumpla syss Z es el resultado de aplicarle la función de Ackerman a X y Y.
   La definición de la función de Ackerman es la siguiente:

$$\begin{aligned}
ackerman(0, n) &= n - 1 \\
ackerman(m, 0) &= ackerman(m - 1, 1) \\
ackerman(m, n) &= ackerman(m - 1, ackerman(m, n - 1))
\end{aligned}$$

   Chisme: La función de Ackerman es de suma importancia en el estudio de funciones recursivas, ya que, es la función que crece más rápido que cualquier función recursiva primitiva.

1.3. Listas diferenciables

En el laboratorio se definió la relación `flatten`(ℓ_1, ℓ_2) tal que `flatten`(ℓ_1, ℓ_2) es válida syss ℓ_2 es la lista con todos los elementos que ocurren en la lista de listas ℓ_1, esto localizado en el archivo `Diff.pl`.

En tal archivo agrega la versión de `flatten` que utilice listas diferenciables.
Sugerencia: Recuerda que la concatenación de listas se define de la siguiente forma:

```
app(A-B, B-C, A-C).
```

2. Introducción a inteligencia artificial

2.1. Inferencia

Crea un archivo llamado `cartoons.pl` el cual resuelva el siguiente problema:

Hay cinco casas consecutivas, cada una tiene un color diferente y habitada por personajes de distintas caricaturas. Cada una tiene una mascota distinta y manejan un tipo distinto de arma y tienen una bebida favorita.
Tus hipótesis son:

1. Bugs Bunny vive en la casa roja.
2. Steven tiene un león.
3. En la casa verde se bebe whisky.
4. Star bebe chocolate.
5. La casa verde está inmediatamente al lado de la morada.
6. El dueño de la espada tiene caracoles.
7. El personaje que ataca con un pistola vive en la casa amarilla.
8. En la casa de en medio se bebe leche.
9. Mickey Mouse vive en la primera casa (de izquierda a derecha).

10. El que usa un sable vive en la casa siguiente del que tiene un zorro.
11. La pistola la usa el personaje que vive en la casa siguiente a donde está el caballo.
12. El que usa rayos como arma bebe coca cola.
13. Batman usa sus puños como arma.
14. Mickey Mouse vive en la casa siguiente a la casa azul.

Las preguntas son:

- ¿Quién es el dueño del perro?
- ¿Quién bebe agua?

Sugerencia: Cada casa la puedes ver como una tupla de 5 elementos, es decir:

```
(_Color,_Personaje,_Arma,_Bebida,_Mascota)
```

Por lo que el problema completo es una lista de casas que cumplen con las descripciones dadas.

2.2. Redes semánticas

En el laboratorio se explicó que es una red semántica, recordemos las definiciones necesarias:

Definición 1 *Una red semántica es una representación del conocimiento que consiste en la gráfica dirigida $G =< V, A >$ donde V es un conjunto de nodos y A un conjunto de flechas tales que:*

- $V = \{t | t \text{ es término } \}$
- $A = \{a_{i,j} | a \text{ es una relación entre los términos } t_i, t_j\}$

Definición 2 (Herencia) *En una red semántica la forma de deducción es por medio de herencia, es decir, dados dos términos $t, s \in V$, t se deduce de s bajo la relación r syss existe un camino de t a s donde r se cumpla en el último paso del camino.*

Crea un archivo llamado `mired.pl` y realiza los siguientes ejercicios:

1. Crea una red semántica que cumpla:
    - Tener al menos 10 nodos y al menos 13 flechas.
    - Tener al menos 5 nodos cuyo grado de salida de cada uno sea 2.
    - Tener al menos 4 nodos cuyo grado de salida de cada uno sea 3.

    Dada la red semántica previamente definida brinda la representación impresa en tu archivo `README`. Puedes mandar la imagen scanneada (que sea clara la imagen) o bien utilizar `graphviz` para dibujarla.

2. A partir de la red semántica del ejercicio anterior define una relación llamada `satisface`(R, X, Y) que sea válida syss Y es heredada de X por medio de R.

Reglas:

- Para esta práctica no es necesario importar biblioteca alguna.

En el ejercicio de inferencia (2.1) la solución por análisis exhaustivo requiere de $5!^5$ casos a analizar, que es un total de $24,883,200,000$ casos... bendita sea la computadora.