

Inteligencia Artificial 2018-1

Proyecto 2: Búsqueda

IIMAS-PCIC

Luis Alejandro Lara Patiño
Roberto Monroy Argumedo
Alejandro Ehécatl Morales Huitrón

23 de noviembre de 2017

Índice

| | |
|---|----------|
| 1. Funcionamiento del proyecto | 1 |
| 1.1. Carga de información | 1 |
| 1.2. Ejecución | 1 |
| 1.2.1. Desde una terminal UNIX | 2 |
| 1.2.2. Desde el listener de Prolog | 2 |
| 2. Estructura de la base de conocimiento | 2 |
| 2.1. Conocimiento conceptual | 2 |
| 2.2. Conocimiento factual | 2 |
| 2.3. Conocimiento del mundo | 2 |
| 3. Módulos | 2 |
| 3.1. Simulador | 2 |
| 3.2. Módulo de diagnóstico | 3 |
| 3.3. Módulo de toma de decisión | 3 |
| 3.4. Módulo de planeación | 3 |
| 3.5. Utilitarios | 3 |

1. Funcionamiento del proyecto

1.1. Carga de información

Toda la información necesaria para el funcionamiento del programa debe cargarse en la base de conocimiento previamente a la ejecución. Las bases se guardan en el directorio **bases**, y se pasan como argumento al programa escribiendo el nombre de archivo, sin la extensión.

1.2. Ejecución

El punto de entrada al proyecto es el archivo `main.pl`. Este archivo está acondicionado para ejecutarse tanto desde el listener de Prolog como desde una terminal en sistemas operativos UNIX.

1.2.1. Desde una terminal UNIX

El archivo `main.pl` puede ejecutarse como cualquier otro *script*; solamente se debe navegar hasta el directorio del proyecto y desde allí ejecutar el comando:

```
./main.pl <nombre-base>
```

1.2.2. Desde el listener de Prolog

Asimismo, es posible ejecutar el programa desde el entrono de Prolog. Para ello, se debe consultar el archivo `main.pl`, y realizar la siguiente consulta:

```
?- main([<nombre-base>])
```

2. Estructura de la base de conocimiento

La estructura de base de conocimiento que utiliza este proyecto puede dividirse en tres secciones principales: conocimiento conceptual, conocimiento factual y conocimiento del mundo.

2.1. Conocimiento conceptual

En esta sección se almacena la información acerca de los objetos que participan en las actividades del robot. En el caso del asistente de supermercado, aquí se almacenan los productos de la tienda en forma de objetos, los cuales heredan las propiedades y relaciones de sus clases padre de acuerdo a la jerarquía implementada en el primer proyecto.

2.2. Conocimiento factual

En esta sección se almacenan las acciones realizables tanto por el empleado de la tienda como por el robot. También se mantienen aquí las decisiones que puede tomar el robot, así como la creencia que tiene el robot del mundo y las observaciones que éste adquiere durante la ejecución del programa.

Finalmente, se almacenan aquí los resultados de cada uno de los módulos del proyecto, dentro de los objetos `diagnostico`, `decision` y `agenda`.

2.3. Conocimiento del mundo

En esta sección se guarda la configuración que tiene el mundo del robot; las ubicaciones a las que éste puede acceder y qué objetos hay actualmente en ellas.

También existe un objeto `robot`, que mantiene los objetos que éste tiene en sus brazos y la posición en la que se encuentra.

3. Módulos

3.1. Simulador

La primera sección del proyecto que se ejecuta cuando éste inicia es el simulador. Éste se encarga de coordinar la ejecución de los tres módulos cuando sea necesario y de ejecutar el plan encontrado.

Al inicio del programa, el simulador ejecuta los tres módulos para generar un plan y cumplir las órdenes del cliente. En esta primera iteración, el robot confía plenamente en el reporte inicial del empleado de la tienda, y solamente busca entregar los productos que el cliente le pidió.

Una vez generado el plan, el simulador ejecuta las acciones secuencialmente. para determinar si una acción tuvo o no éxito, se extrae la probabilidad de éxito de la misma, almacenada en la base de conocimiento

y mediante un generador de números aleatorios, se aplica dicha probabilidad. Adicionalmente, la acción buscar puede fallar si se recibe una observación que no concuerda con la creencia que tiene el robot del mundo en ese momento.

Si alguna de las acciones falla, el simulador vuelve a ejecutar los tres módulos del proyecto, con el fin de generar un nuevo plan para superar los obstáculos que pueda haber encontrado. Las acciones anteriores se repiten hasta que el robot cumple con todas sus encomiendas. En este momento, el programa termina.

3.2. Módulo de diagnóstico

Para hacer un diagnóstico se necesita un conjunto de observaciones y una locación actual. También una lista de objetos y posiciones restantes. La forma en como se obtiene es realizando una búsqueda en la que en cada paso agregamos una acción “mover” o “colocar” hasta que la lista de objetos y posiciones restantes sea vacía en cuyo caso agregamos la locación inicial para así obtener (en una lista) el conjunto de acciones de final a principio que realizó el almacenista.

La forma de nuestra función sucesor toma en cuenta la mínima falla que pudo cometer el almacenista, es decir suponemos que los productos están en su lugar.

3.3. Módulo de toma de decisión

3.4. Módulo de planeación

3.5. Utilitarios