EPA cap 2

# Lenguaje de expresión de problemas

Un lenguaje de expresión de problemas debe reunir las siguientes características:

• Debe estar formado por un número de instrucciones finito.

• Debe ser completo, es decir que todas las acciones de interés deben poder expresarse con dicho conjunto de instrucciones.

• Cada instrucción debe tener un significado (efecto) preciso.

• Cada instrucción debe escribirse de modo único.

# Sintaxis y semántica en un lenguaje

## La forma en que se debe escribir cada instrucción de un lenguaje y las reglas generales de expresión de un problema completo en un lenguaje constituyen su sintaxis.

## El significado de cada instrucción del lenguaje y el significado global de determinados símbolos del lenguaje constituyen su semántica.

## para cada instrucción, su sintaxis, es cómo debe escribirse, y su semántica es cómo se interpreta esa orden en el lenguaje

# Robot R-info

Es un robot móvil llamado R-info, controlado por un conjunto reducido de primitivas que permiten modelizar recorridos y acciones en una ciudad compuesta por calles y avenidas.

en este curso utilizaremos un único robot R-info que se desplazará en una única área de una ciudad compuesta por 100 avenidas y 100 calles.

El robot R-info que se utiliza posee las siguientes capacidades básicas:

1. Se mueve.

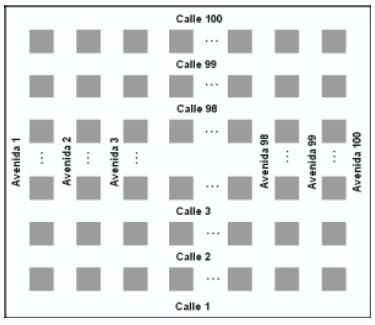
2. Se orienta hacia la derecha, es decir, gira 90 grados en el sentido de las agujas del reloj.

3. Dispone de sensores visuales que le permiten reconocer dos formas de objetos preestablecidas: flores y papeles. Los mismos se hallan ubicados en las esquinas de la ciudad.

4. Lleva consigo una bolsa donde puede transportar flores y papeles. Está capacitado para recoger y/o depositar cualquiera de los dos tipos de objetos en una esquina, pero de a uno a la vez. La bolsa posee capacidad ilimitada.

5. Puede realizar cálculos simples.

6. Puede informar los resultados obtenidos.



Debe considerarse a las arterias como rectas y a la esquina como el punto de intersección entre dichas rectas

La esquina se representará por dos coordenadas: la primera indicará el número de avenida y la segunda el número de calle. Por ejemplo, la esquina (2,4) es la intersección de la avenida 2 y la calle 4

El andar del robot queda asociado con un paso que equivale a una cuadra de recorrido;

Se aceptan convenciones (el robot solo inicia sus recorridos en la posición (1,1) de la ciudad);

|  |  |
| --- | --- |
| Sintaxis | Semántica |
| Iniciar | Instrucción primitiva que coloca al robot en (1,1), orientado hacia el norte |
| derecha | Primitiva que cambia la orientación del robot 90 grados a la derecha |
| mover | Primitiva que conduce al robot de una esquina a la siguiente espetando la dirección en la que se encuentra orientado |
| tomarFlor | Primitiva para recoger flor donde se encuentra el robot |
| TomarPapel | Primitiva para recoger papel donde se encuentra el robot |
| depositarFlor | Primitiva para depositar flor de la bolsa en la esquina que se encuentra el robot |
| depositarPapel | Primitiva para depositar papel de la bolsa en la esquina que se encuentra el robot |
| PosAv | Indica en que avenida está el robot |
| PosCa | Indica en que calle está el robot |
| HayFlorEnLaEsquina | Indica si hay una flor en la esquina que se encuentra el robot con V (si hay) y F (si no hay) |
| HayPapelEnLaEsquina | Indica si hay un papel en la esquina que se encuentra el robot con V (si hay) y F (si no hay) |
| HayFlorEnLaBolsa | Indica si hay al menos una flor en la bolsa (con V) o no hay (con F) |
| HayPapelEnLaBolsa | Indica si hay al menos un papel en la bolsa (con V) o no hay (con F) |
| Pos | Posiciona al robot donde se lo indique (Av,Ca), sin cambiar su orientacion |
| Informar | Instrucción que permite visualizar en pantalla el contenido almacenado en alguna variable. |
| {entre llaves} | Hacer un comentario |

## Un programa escrito en el lenguaje del robot comienza con la palabra clave programa, la cual debe estar seguida por un identificador que determina el nombre del programa.

## El cuerpo del programa principal es una secuencia de sentencias, delimitada por las palabras claves comenzar y fin.

**programa** nombre\_del\_programa

**areas**

se declara una única área que comprende toda la ciudad

**robots**

se declara un único tipo de robot “robot1”, junto al código correspondiente al programa que se quiere realizar

**variables**

se declara una variable que representa al robot, será llamada R-info

**comenzar**

se asigna el área donde se desplazará R-info (en este curso toda la ciudad) se indica el inicio para que cada robot se ejectue (en este curso solo se indica el comienzo de ejecución de R-info).

**fin**

Al escribir un programa se deben respetar las siguientes reglas de indentación:

• La palabra clave **programa** debe comenzar en la primer columna.

• Las palabras claves **comenzar** y **fin** de un programa deben comenzar en la misma columna que la palabra clave **programa**.

• Las sentencias del cuerpo del programa debe comenzar dos columnas más a la derecha que las palabras claves que lo delimitan: **comenzar** y **fin**.

• Las sentencias que pertenecen al cuerpo de una estructura de control deben comenzar dos columnas más a la derecha que la palabra clave que identifica a la estructura de control.

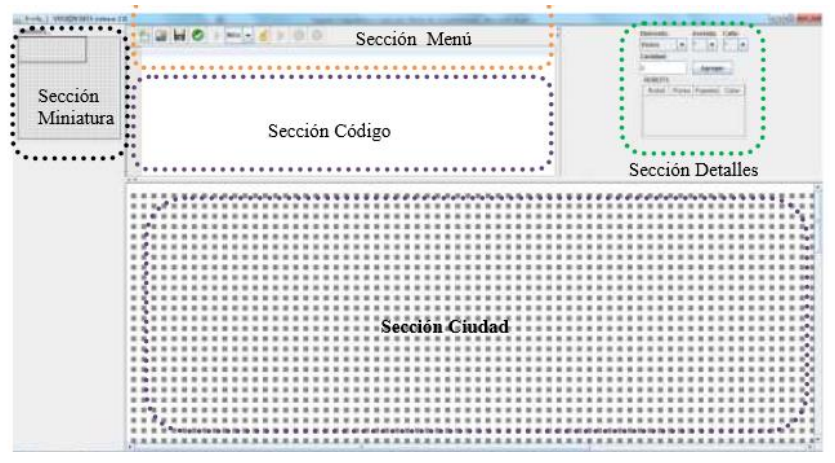
• La indentación es la única forma de indicar si una sentencia pertenece o no a la estructura de control en cuestión.

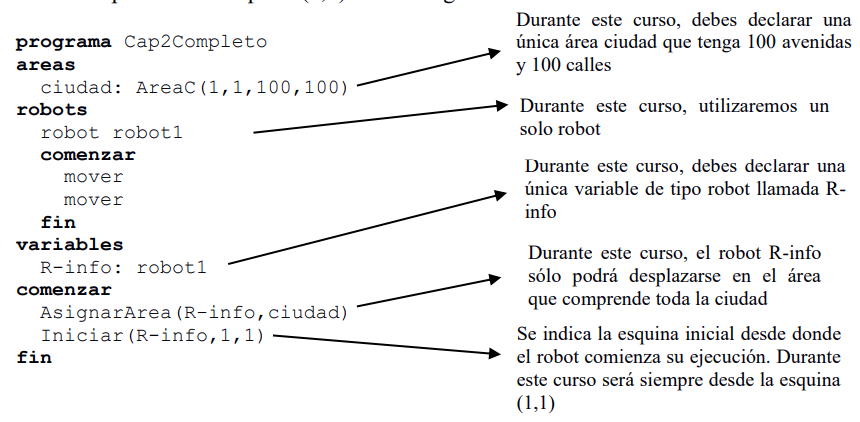
Para codificar un algoritmo en el lenguaje del robot R-info es necesario realizar las siguientes tres etapas:

1. Escribir el programa: se escriben los pasos a seguir utilizando la sintaxis descripta.

2. Compilar el programa: para lograr que la computadora ejecute el programa escrito en la etapa anterior es necesario traducirlo a un lenguaje que la computadora comprenda. Esta etapa de denomina compilación y permite detectar los errores de sintaxis.

3. Ejecutar el programa: una vez que el programa ha sido compilado, puede ejecutarse. El ambiente de programación del robot R-info permite visualizar durante la ejecución, el recorrido que realiza el robot R-info dentro de la ciudad.

****

****

# Ejercicios

1. • Programe al robot para que recorra la calle 6 desde la avenida 11 a la avenida 13.

programa Cap2Ejercicio1

areas

ciudad: AreaC(1,1,100,100)

robots

robot robot1

comenzar

Pos(11,6)

derecha

mover

mover

fin

variables

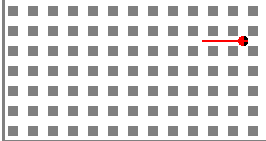
R-info: robot1

comenzar

AsignarArea(R-info,ciudad)

Iniciar(R-info,1,1)

fin



• Programe al robot para que recorra la avenida 17 desde la calle 31 hasta la calle 25.

programa Cap2Ejercicio2

areas

ciudad: AreaC(1,1,100,100)

robots

robot robot1

comenzar

Pos(17,31)

derecha

derecha

mover

mover

mover

mover

mover

mover

fin

variables

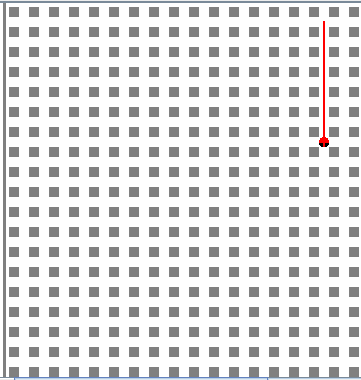
R-info: robot1

comenzar

AsignarArea(R-info,ciudad)

Iniciar(R-info,1,1)

fin



• Programe al robot para que, si puede, deposite un papel en (1,2) y una flor en (1,3).

programa Cap2Ejercicio3

areas

ciudad: AreaC(1,1,100,100)

robots

robot robot1

comenzar

mover

si HayPapelEnLaBolsa

depositarPapel

mover

si HayFlorEnLaBolsa

depositarFlor

fin

variables

R-info: robot1

comenzar

AsignarArea(R-info,ciudad)

Iniciar(R-info,1,1)

Fin

• Programe al robot para que intente recoger una flor de la esquina determinada por la calle 50 y la avenida 7. Solo si lo logra debe ir a la calle51 y avenida 8 e intentar recoger allí otra flor. Al finalizar debe informar en que esquina quedó parado.

programa Cap2Ejercicio4

areas

ciudad: AreaC(1,1,100,100)

robots

robot robot1

comenzar

Pos(7,50)

si HayFlorEnLaEsquina

tomarFlor

si HayFlorEnLaBolsa

mover

derecha

mover

si HayFlorEnLaEsquina

tomarFlor

Informar(PosAv, PosCa)

fin

variables

R-info: robot1

comenzar

AsignarArea(R-info,ciudad)

Iniciar(R-info,1,1)

Fin

programa Cap2Ejemplo6

areas

ciudad: AreaC(1,1,100,100)

robots

robot robot1

comenzar

derecha

repetir 2

si HayFlorEnLaBolsa

depositarFlor

sino

Informar(PosAv)

mover

fin

variables

R-info: robot1

comenzar

AsignarArea(R-info,ciudad)

Iniciar(R-info,1,1)

Fin

• ¿Por qué al finalizar el recorrido, el robot no queda posicionado en el mismo lugar que en el ejemplo 2.3? ¿Perjudica en algo este hecho a lo que debe ser informado?

Porque al repetir dos veces la acción mover, se desplaza dos avenidas y no una como en el ejemplo 2.3. No perjudica a lo que tiene que ser informado por el robot

¿Las acciones realizadas por el robot en los programas Cap2Ejemplo4 y Cap2Ejemplo7 son iguales?

No, no so iguales ya que la acción de tomar el papel se encuentra dos veces en cada repeticion

• ¿Es posible incluir la instrucción (\*) en la repetición? Si es así indique la manera de hacerlo y qué diferencias encuentra con el programa Cap2Ejemplo7.

Si, yo lo haría de la siguiente manera:

Programa Cap2Ejemplo7

areas

Ciudad: AreaC (1,1,100,100)

robots

robot robot1

comenzar

pos (15,12)

si HayPapelEnLaEsquina

tomarPapel

repetir 2

mover

si HayPapelEnLaEsquina

tomarPapel

fin

variables

R-info: robot 1

comenzar

AsignarArea (R-info, ciudad)

Iniciar (R-info ,1,1)

fin

La diferencia con el original es que esta dos veces el juntar una flor dentro de la repetición

• ¿El robot realiza la misma cantidad de pasos en ambos programas?

No, en la segunda opción el robot realiza un paso de mas cuando tiene que ir a (4,4)

• ¿Cuál solución prefiere Ud.? Justifique su respuesta pensando en que ahora debe hacer que el robot camine desde (3,5) hasta (3,7) y desde allí hasta (6,7).

En este caso en particular yo prefiero la primera acción, porque para adaptarla a la otra solo es necesario cambiar el posicionamiento inicial del robot

• ¿Qué función cumple la instrucción iniciar?

comenzar

• ¿Por qué todos los programas del robot deben comenzar con esta

instrucción?

Para que se comience a ejecutar el codigo

• ¿Qué diferencia hay entre las instrucciones PosAv y PosCa y la instrucción Pos?

Que las dos primeras son para informar ubicación, y la segunda para posicionar al robot

• Suponga que el robot se encuentra en (1,1) y se desea que salte a (3,4). ¿Es

posible realizar las siguientes asignaciones para lograrlo?

PosAv := 3

PosCa := 4

Justifique su respuesta. Indique la manera correcta de resolver este problema.

No, debería ser:

Pos (3,4)

• ¿Es posible para el robot depositar una flor sin verificar primero si tiene al

menos una flor en su bolsa?

si

• ¿El robot está capacitado para decir si en una misma esquina hay varios papeles?

No, salvo que ejecutes el código de HayPapelEnLaEsquina o de flores varias veces

## Una proposición es una expresión de la cual tiene sentido decir si es verdadera o falsa, o sea es posible asignarle un valor de verdad (verdadero o falso, pero no ambos)

Un ejemplo es la iteración y la selección

# Una proposición es considerada atómica si no puede ser descompuesta en otras proposiciones.

## Cuando en una expresión se unen varias proposiciones atómicas se forma una proposición molecular o compuesta. Dicha unión se realiza mediante conectivos lógicos o términos de enlace.

Los términos de enlace a utilizar son los siguientes: “y”, “o”, “no”. Los dos primeros se utilizan para conectar proposiciones atómicas; en tanto que el conectivo “no”, solamente se coloca frente a una proposición atómica

# Una proposición es atómica si no tiene conectivos lógicos, en caso contrario es molecular.

## Forma de expresarlos

|  |  |
| --- | --- |
| Conectivo | Simbolización en el ambiente de programación de R-info |
| y | & |
| o | | |
| no | ~ |

## Se utilizarán letras minúsculas para simbolizar las proposiciones atómicas.

Ejemplos de simbolización de proposiciones atómicas

1. Ayer fue un día ventoso.

Si se considera p = “ayer fue un día ventoso”, esta proposición puede ser simbolizada

como: p.

El proceso para simbolizar una proposición molecular consiste en tres pasos:

1. Determinar cuáles son las proposiciones atómicas que la componen.

2. Simbolizar las proposiciones como se explicó anteriormente.

3. Unir las proposiciones con los conectivos ya vistos. Por tal motivo, debe

definirse un símbolo para cada uno de los conectivos.

### ejemplo

• Juan es estudiante y es jugador de fútbol. p = “Juan es estudiante” q = “Juan es jugador de fútbol” Simbolizando p & q

## la palabra pero actúa como el conectivo lógico “y”.

# La tabla de verdad de una proposición molecular es, como su nombre lo indica, una tabla donde se muestran todas las combinaciones posibles de los valores de verdad de las proposiciones atómicas que la conforman.

## La conjunción de dos proposiciones es cierta únicamente en el caso en que ambas proposiciones lo sean.

Dadas dos proposiciones cualesquiera p y q, si ambas son verdaderas, entonces p & q, que representa la conjunción de p y q, es verdadera. Cualquier otra combinación da como resultado una proposición molecular falsa

Ejemplo 2.13: el robot debe depositar, de ser posible, una flor en la esquina (45,70) solamente si en la esquina no hay flores.

comenzar

Pos(45,70)

si ((HayFlorEnLaBolsa) & ~(HayFlorEnLaEsquina))

depositarFlor

fin

# o

# La disyunción entre dos proposiciones es cierta cuando al menos una de dichas proposiciones lo es

La disyunción utiliza el término de enlace “o” en su sentido incluyente. Esto significa que basta con que una de las dos proposiciones sea verdadera para que la disyunción sea verdadera.

## Dada una proposición p, su negación ~ p, permitirá obtener el valor de verdad opuesto.

## El operador “~” es el que tiene mayor prioridad, es decir, es el que primero es evalúa; seguido por “&”y “|”.

En determinados casos se tiene la necesidad de alterar este orden natural de resolución. Al igual que en Matemática, el uso de paréntesis permite resolver este problema.