

## Control LP (Compiladors): The Roomba Language!

Cal fer un compilador per interpretar el llenguatge per controlar un robot escombra. El llenguatge té instruccions per definir una posició inicial, moure en 2 dimensions, interaccionar amb sensors de proximitat i llum, executar condicionalment instruccions, executar tasques predeterminades i executar llistes d'operacions predefinides. El següent exemple mostra el llenguatge:

```
position 400 10           // initial position of the robot
startcleaning             // starting the cleaning instructions
move right 12             // move 12 units to the right
if sensorprox == ON then  // if an object is found
    flush 180             // flushes 180 units of cleaning liquid
exec t1                   // execute task t1 defined below
ops [move left 2, exec t1, flush 90] // ops executes a list of instructions
if sensorlight > 16 AND sensorprox == OFF then
    ops [flush 90, exec t1, ops [flush 10, move left 10]] // ops can be within an if
endcleaning

TASK t1                   // task definition
    exec t2               // other tasks can be invoked!
    move left 100
    flush 180
ENDTASK

TASK t2
    ops [flush 120, if sensorprox == OFF then move up 100]
ENDTASK
```

S'assumeix que inicialment esta la posicio inicial del robot, les instruccions del programa i finalment la llista de definició de tasques. Heu de tenir en compte que:

- Les instruccions `if` només tenen una instrucció associada.
- Els únics operadors relacionals són el `>` i `==`. La precedència respecte els operadors lògics és: `>, == > (OR, AND)`.

**[Part 1: 50% nota]** Assumint que tots els tokens ja estan definits, defineix la part sintàctica (gramàtica). Fès la gramàtica per a que PCCTS pugui reconèixer-la i decorar-la per generar l'AST mostrat a l'anvers de la pàgina. Assumeix la següent regla inicial:

```
roomba: position STARTC! linstr ENDC! tasks <<#0=createASTlist(_sibling);>>;
```

**[Part 2: 50% nota]** Interpretació: fès el mètode

```
void novaPosicio(AST *a)
```

on **a** apunta al node arrel de l'arbre. La funció interpretarà el programa en temps real per determinar la nova posició del robot, és a dir, s'imprimirà per la sortida estàndard la posició del robot després d'executar el programa partint de la posició inicial definida al principi. En el codi que has de fer, assumeix que ja hi ha definida la funció:

```
AST *findTask(string id)
```

que donat el nom d'una tasca, retorna el node de l'AST on està definida. Tambè teniu les funcions:

```
int SensorLight()
```

```
bool SensorProx()
```

que donen els valors dels sensors que té el robot.

```
list
  \__position
  |      \__400
  |      \__10
  \__list
  |      \__move
  |      |      \__right
  |      |      \__12
  |      \__if
  |      |      \__==
  |      |      |      \__sensorprox
  |      |      |      \__ON
  |      |      |      \__flush
  |      |      |      \__180
  |      \__exec
  |      |      \__id(t1)
  |      \__ops
  |      |      \__move
  |      |      |      \__left
  |      |      |      \__2
  |      |      \__exec
  |      |      |      \__id(t1)
  |      |      \__flush
  |      |      |      \__90
  |      \__if
  |      |      \__AND
  |      |      |      \__>
  |      |      |      |      \__sensorlight
  |      |      |      |      \__16
  |      |      |      \__==
  |      |      |      |      \__sensorprox
  |      |      |      |      \__OFF
  |      |      \__ops
  |      |      |      \__flush
  |      |      |      |      \__90
  |      |      |      \__exec
  |      |      |      |      \__id(t1)
  |      |      |      \__ops
  |      |      |      |      \__flush
  |      |      |      |      |      \__10
  |      |      |      |      \__move
  |      |      |      |      |      \__left
  |      |      |      |      |      \__10
  \__list
  |      \__TASK
  |      |      \__id(t1)
  |      |      \__list
  |      |      \__exec
```

```

|           |      \__id(t2)
|           |      \__move
|           |      \__left
|           |      \__100
|           |      \__flush
|           |      \__180
\__TASK
  \__id(t2)
  \__list
    \__ops
      \__flush
      |      \__120
      \__if
        \__==
        |      \__sensorprox
        |      \__OFF
        \__move
          \__up
          \__100

```