

Control LP (Compiladors): The Lego Game

Cal fer un compilador per interpretar un llenguatge simple de posicionament de blocs rectangulars de Lego en una graella bidimensional $N \times M$. La posició (1, 1) de la graella està a la cantonada superior esquerra, mentre que la (N, M) està a la cantonada inferior dreta. Inicialment, la graella està buida, i assumirem que els blocs tenen la mateixa alçada unitària però diferents dimensions, que es poden definir amb un parell d'enters ordenats que especificaran l'amplada (eix de les abscises), i la longitud (eix de les ordenades). Els blocs tenen una codificació idèntica a la de la graella. Un bloc es pot empilar a la graella o a sobre d'altre bloc, sempre que totes les connexions del bloc a empilar quedin connectades. També es poden moure blocs sempre que es compleixin les condicions de no deixar connexions desconnectades. Amb l'ànim d'automatitzar el procés, el jugador pot definir bucles i funcions per a anar fent l'emplaçament dels blocs en la graella. Quan una instrucció sobre un bloc no compleix les regles, s'ignora. A continuació podeu veure un exemple complet d'aquest llenguatge.

```
Grid 10 20                                     // Comença el joc Graella NxM
B1 = PLACE (2,2) AT (1,1)                       // Crea el bloc B1 que es situa a la posició (1,1)
B2 = PLACE (2,4) AT (3,1)                       // Crea el bloc B2 que es situa just al costat de B1
MOVE B2 NORTH 1                                // Incorrecta: solapa parcialment B2 a sobre de B1
MOVE B1 SOUTH 2                                // Correcta: posar B1 a sobre de B2
MOVE B2 EAST 5                                  // Mou B2 cinc unitats cap a l'est

B2 = PLACE (2,3) AT B2                          // Incorrecta: no hi ha espai al bloc B2 pel nou bloc 2x3
// noteu que no s'especifica la posició exacta, només si hi ha espai
WHILE (FITS(B2,2,1,2)) [                       // Bucle que omple amb blocs (2,1) el segon tram (alçada 2) de B2
    B2 = PLACE (2,1) AT B2                     // Empila el bloc en un espai lliure de B2
]

TOWER10                                         // Crida a la funció TOWER10

DEF TOWER10
WHILE (HEIGHT(B2) > 1 AND HEIGHT(B2) < 10) [ // Bucle per fer una torre d'alçada 10 amb blocs 2x2
    B2 = PLACE (2,2) AT B2
]
ENDEF
```

Figura 1: The Lego Game.

Assumiu que com a condicions dels bucles només poden aparèixer els operadors relacionals ($<$, $>$), que poden ser combinats amb l'operador **AND**. Fixeu-vos que la operació **MOVE** actua com assignació, és a dir modifica la definició del block que es mou.

[Part 1: 50% nota] Defineix la part lèxica i sintàctica. Fes la gramàtica per a que PCCTS pugui reconèixer-la i decorar-la per generar l'AST mostrat a l'anvers de la pàgina.

[Part 2: 50% nota] Interpretació: fes els mètodes

```
AST *findDefinition(string id)
void executeListInstructions(AST *a)
```

que determinen el node de l'AST on hi ha definició d'una rutina, i executa la llista d'instruccions d'entrada, respectivament. Pots assumir que ja hi han implementades les següents funcions:

```
void setGrid(int width, int height)           // Crea la graella
bool evaluateExpression(AST *a)               // Avalua una expressio apuntada per a
void executeMove(string id, string direction, int steps) // Executa una instruccio de move
void executePlace(string id, AST *orig_bloc, AST *dest) // Executa una instruccio de place
```

```

list
  \__Grid
  | \__10
  | \__20
  \__list
  | \__PLACE
  | | \__B1
  | | | \__list
  | | | | \__2
  | | | | \__2
  | | | \__list
  | | | | \__1
  | | | | \__1
  | | \__PLACE
  | | | \__B2
  | | | | \__list
  | | | | | \__2
  | | | | | \__4
  | | | | \__list
  | | | | | \__3
  | | | | | \__1
  | | \__MOVE
  | | | \__B2
  | | | \__NORTH
  | | | \__1
  | | \__MOVE
  | | | \__B1
  | | | \__SOUTH
  | | | \__2
  | | \__MOVE
  | | | \__B2
  | | | \__EAST
  | | | \__5
  | | \__PLACE
  | | | \__B2
  | | | | \__list
  | | | | | \__2
  | | | | | \__3
  | | | \__B2
  | | \__WHILE
  | | | \__FITS
  | | | | \__B2
  | | | | | \__list
  | | | | | | \__2
  | | | | | | \__1
  | | | | | | \__2
  | | | | \__list
  | | | | | \__PLACE
  | | | | | | \__B2
  | | | | | | | \__list
  | | | | | | | | \__2
  | | | | | | | | \__1
  | | | | | \__B2
  | | \__TOWER10
  \__list
  | \__DEF
  | | \__TOWER10
  | | | \__list
  | | | | \__WHILE
  | | | | | \__AND
  | | | | | | \__>
  | | | | | | | \__HEIGHT
  | | | | | | | | \__B2
  | | | | | | | \__1
  | | | | | | \__<

```

```

|          \__HEIGHT
|          |          \__B2
|          \__10
\__list
      \__PLACE
            \__B2
            \__list
            |          \__2
            |          \__2
            \__B2

```