



RoboLang



Objetivo del reto

Diseñar y formalizar una **gramática libre de contexto (GLC)** en **BNF** para un mini-lenguaje de control de robots llamado **RoboLang**

El lenguaje debe permitir describir **programas completos** que combinan:

- **Movimientos básicos**
- **Bloques de repetición**
- **Condicionales basados en sensores**
- **Bucles condicionales**

La gramática debe ser **clara, no ambigua**.



Partes del lenguaje a cubrir

1. **Comandos simples** (cada uno termina con ;):
 - UP;, DOWN;, LEFT;, RIGHT;
 - PENUP;, PENDOWN;
 2. **Bloques de repetición**
 - Forma: REPEAT n { <instrucciones> }
 - n es un número entero positivo.
 3. **Condicionales**
 - Forma: IF (<condición>) { <instrucciones> } [ELSE { <instrucciones> }]
 - El ELSE es opcional.
 4. **Bucles condicionales**
 - Forma: WHILE (<condición>) { <instrucciones> }
 5. **Sensores disponibles**
 - Numéricos: DIST, TEMP, LIGHT, BATTERY
 - Booleano: OBSTACLE
 6. **Operadores**
 - Comparación: =, !=, <, <=, >, >=
 - Lógicos: NOT, AND, OR
 - Precedencia: NOT > AND > OR
-



Requerimientos del ejercicio

1. **Definir formalmente la gramática**
 - Expresarla en **BNF**.

- Representar correctamente la **precedencia** y **asociatividad** de los operadores lógicos.
 - 2. **Probar el lenguaje**
 - Diseñar al menos **5 programas válidos** y **5 inválidos**.
 - Justificar por qué cada uno debe aceptarse o rechazarse según la gramática propuesta.
 - 3. **Implementación**
 - Implementar lo en ANTLR
 - Generar árboles sintácticos de ejemplos válidos.
-

✓ Ejemplos de uso del lenguaje (para guiar la definición)

Programas válidos

- Movimientos simples:

```
None
PENDOWN; UP; RIGHT; LEFT; PENUP;
```

- Repetición:

```
None
REPEAT 3 { UP; RIGHT; }
```

- Condicional:

```
None
IF (OBSTACLE = true) { LEFT; } ELSE { UP; }
```

- Bucle:

```
None
WHILE (BATTERY > 20 AND NOT (OBSTACLE = true)) { UP; RIGHT; }
```

Programas inválidos

- Falta de ;:

None

UP

- REPEAT mal formado:

None

REPEAT 5 UP; RIGHT;

- Paréntesis y llaves incorrectos:

None

IF DIST < 5 { UP; }

- Recorrer un laberinto

None

```
WHILE (LIGHT < 90) {  
  /* Intentar mover a la derecha si no hay obstáculo */  
  IF (OBSTACLE = false) { RIGHT; }  
  ELSE {  
    /* Intentar mover hacia abajo si no hay obstáculo */  
    IF (OBSTACLE = false) { DOWN; }  
    ELSE {  
      /* Intentar mover a la izquierda si no hay obstáculo */  
      IF (OBSTACLE = false) { LEFT; }  
      ELSE {  
        /* Intentar mover hacia arriba si no hay obstáculo */  
        IF (OBSTACLE = false) { UP; }  
        ELSE {  
          /* CALLEJÓN SIN SALIDA:  
            Retroceder: girar 180° y avanzar */  
          LEFT; LEFT; UP;  
        }  
      }  
    }  
  }  
}
```



Entregable esperado

Un documento en **Markdown o PDF** que contenga:

1. La **gramática completa en BNF**.
2. Los **programas válidos e inválidos** con justificación.
3. Implementar en **ANTLR4**.

👉 **El reto está en que ustedes definan la gramática.**

Aquí tienen las reglas del juego y los ejemplos, pero deben **formalizar el lenguaje de manera rigurosa**.