

Lenguaje de Programación

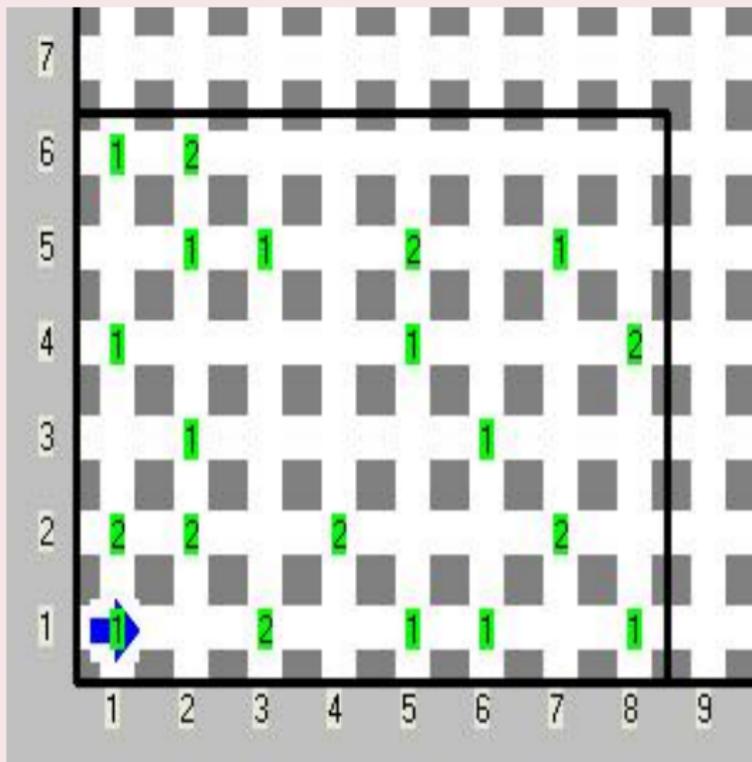
José Luis Alonzo Velázquez

Universidad de Guanajuato

Agosto 2010

Sembrando

Cuando el Dr. Isaac invento a Karel, decidió que Karel debía sembrar el campo de beepers de su patio. Sin embargo, el Dr. Karelov es algo descuidado con su jardín. Este campo tiene esquinas en las cuales ya hay beepers y no solo eso, hay esquinas en donde hay dos beepers. Lamentablemente, cuando una esquina tiene más beepers, la tierra se agota y los beepers mueren antes de poder ser cosechados. La tarea de Karel es dejar el campo con un beeper por cada esquina para tener una excelente cosecha. Karel inicia mirando al este sobre el campo de beepers.



Estrategia a seguir

Para resolver el problema, debemos recorrer el campo por completo. Para lo cual dividiremos el problema en problemas más pequeños. Si ya supiéramos sembrar una avenida el programa seria:

```
program()  
  {  
    siembra_avenida();  
    iterate(7)  
      {  
        move();  
        siembra_avenida();  
      }  
    turnoff();  
  }
```

2do paso

Ahora hay que programar `siembra_avenida()`, en cada avenida hay 5 esquinas además de la que Karel estará parado. Supongamos que ya tenemos la función `siembra_esquina()` que como su nombre lo indica le dice a Karel que hacer en cada esquina entonces la función `siembra_avenida()` vendría dada por:

```
define siembra_avenida()  
  {  
    turnleft();  
    siembra_esquina();  
    iterate(5)  
      {  
        move();  
        siembra_esquina();  
      }  
    regresa();  
  }
```

Donde por supuesto la función `regresa` nos devuelve a la primera calle

```
define regresa()  
  {  
    halfturn();  
    iterate(5)  
      {  
        move();  
      }  
    turnleft();  
  }
```

El último paso es resolver que hacer en cada esquina del campo, para eso analizaremos las diferentes situaciones:

- 1 No hay beeper, entonces poner beeper y moverse a la siguiente esquina.
- 2 Hay beepers: Pregunta cuantos beepers hay, entonces quitar beeper y ver si aun hay:
 - 1 Si hay un beeper, movernos a la siguiente esquina.
 - 2 Si no hay beeper, poner beeper y moverse a la siguiente esquina.

Con lo cual `siembra_esquina()` estaría dada por:

```
define siembra_esquina()
{
  if(notNextToABeeper)
    {
      putbeeper();
    }
  else
    {
      pickbeeper();
      if(notNextToABeeper)
        {
          putbeeper();
        }
    }
}
```

Simplificando Condicionales

Escribir código para mirar al norte sin usar y usando `iterate()`.

Invirtiendo Condicionales

```
if(frontIsClear)
    {
    move();
    }
else
    {
    halfturn();
    }
```

Invirtiendo Condicionales

```
if(frontIsBlocked)
    {
    halfturn();
    }
else
    {
    move();
    }
```

```
if(Condicional X)
    {
    has_nada();
    }
else
    {
    <instrucciones>;
    }
```

```
if(condicional inversa a X)
    {
    <instrucciones>;
    }
```

Factorización Superior

```
if(Condicional X)
    {
    <instrucion 1>;
    <instrucion 2>;
    }
else
    {
    <instrucion 1>;
    <instrucion 3>;
    }
```

Factorización Superior

```
    <instrucion 1>;  
if(Condicional X)  
    {  
    <instrucion 2>;  
    }  
else  
    {  
    <instrucion 3>;  
    }
```

Factorización Inferior

```
if(Condicional X)
    {
    <instrucion 1>;
    <instrucion 3>;
    }
else
    {
    <instrucion 2>;
    <instrucion 3>;
    }
```

Factorización Inferior

```
if(Condicional X)
    {
    <instrucion 1>;
    }
else
    {
    <instrucion 2>;
    }
<instrucion 3>;
```

Doble Condicional Anidada

```
if(Condicional X)
{
  <instruccion 1>;
  if(Condicional X)
  {
    <instruccion 2>;
  }
}
```

Se reduce a en muchos casos

```
if(Condicional X)
{
  <instruccion 1>;
  <instruccion 2>;
}
```

En que casos no ocurre esto???