

## Práctica 9

**Calendario y puntuación máxima:** 28/04/16: 10 puntos. Hasta el 05/05 inclusive: 5 puntos.

Se quieren generar *casos de prueba* para comprobar operaciones sobre strings en un cierto lenguaje. En particular se quieren representar:

- `substring(a,p,b)` = a es una subcadena de b comenzando en la posición p. Es decir  $b[p]=a[1] \dots b[p+n-1]=a[n]$  con n la longitud de a.
- `reverse(a,b)`: b es la cadena inversa de a.
- `count(a,n,c)`: la cadena a contiene n copias del carácter c.

Se debe escribir un predicado por cada una de las operaciones. Todos los predicados se encontrarán en un fichero, "lstrings.mzn":

```
% fichero lstrings.mzn
set of int:CHAR=1..4;

predicate substring(array [int] of var CHAR:a, var int:ini,
                  array [int] of var CHAR:b) =...
predicate reverse...
....
```

```
include "lstrings.mzn";

int:n=3;
int:m=6;
array [1..n] of var CHAR:a;
array [1..m] of var CHAR:b;
var int:p;
var int:q;

constraint p=2;
constraint a[1]=2 /\ a[2]=3;
constraint q=p+2;
constraint substring(a,p,b) /\ substring(a,q,b);

solve satisfy;

output(["a: ",show(a), " p: ",show(p)," b: ", show(b)]);
```

debe mostrar:

```
D:\minizinc>minizinc -a pstrings.mzn
a: [2, 3, 2] p: 2 b: [1, 2, 3, 2, 3, 2]
-----
a: [2, 3, 2] p: 2 b: [2, 2, 3, 2, 3, 2]
-----
a: [2, 3, 2] p: 2 b: [3, 2, 3, 2, 3, 2]
-----
a: [2, 3, 2] p: 2 b: [4, 2, 3, 2, 3, 2]
```

