Le décalage à droite est la multiplication par 2 en binaire (30 min)

En notation *little-endian* c'est-à-dire bit de point fort à droite, un tableau $U[0..N]$ de bits représente le nombre entier $\sum_{k=0}^{k=N} U[k] \times 2^k$. Pour effectuer la multiplication par 2 en binaire, il suffit donc de décaler les bits d'un cran vers la droite et d'ajouter un 0 à gauche. L'automate ci-dessous exploite ce principe : il prend en paramètre un nombre binaire U représenté par le tableau $U[0..N]$ et remplit le tableau $D[0..N]$ qui représente le double de U.

Exemple : Si on donne le tableau U suivant au programme

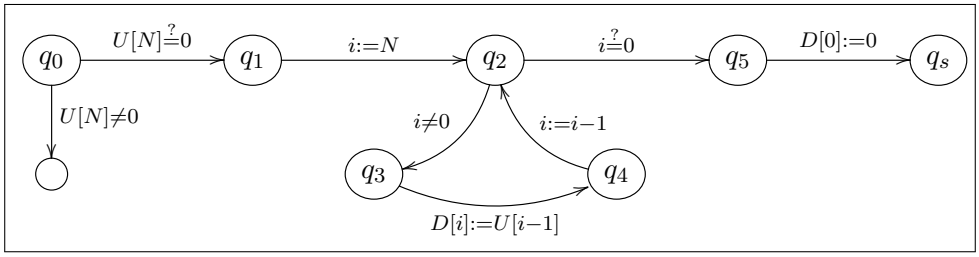
k	0	1	2	3	4	...	N
$U[k]$	1	0	1	0	0	...	0

$\simeq 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 = 1 + 4 = 5$

il construit le tableau

ℓ	0	1	2	3	4	...	N
$D[\ell]$	0	1	0	1	0	...	0

$\simeq 0 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 = 2 + 8 = 10$



On prétend que l'automate ci-dessus calcule le double de U par décalage à droite. Le but de l'exercice est de montrez qu'en q_s la propriété ψ_s suivante est vérifiée :

$$\psi_s \equiv \left\{ \left(\sum_{\ell=0}^{\ell=N} D[\ell] \times 2^\ell \right) \stackrel{?}{=} 2 \times \left(\sum_{k=0}^{k=N} U[k] \times 2^k \right) \right\}$$

Indication : Pour effectuer correctement les affectations en arrière vous aurez besoin de réécrire les $\sum$ sous une forme qui fait apparaître explicitement le terme que vous devez substituer.

A

```
if (U[N] == 0){
  i := N ;
  while(i ≠ 0){
    D[i] := U[i-1] ;
    i := i-1 ;
  }
  D[0]:=0;
}
```

B

```
  U[N] := 0 ;
  i := N ;
  while(i ≠ 0){
    D[i] := U[i-1] ;
    i := i-1 ;
  }
  D[0]:=0;
```

C

```
if (U[N] == 0){
  i := N ;
  while(i ≠ 0){
    D[i] := U[i-1] ;
    i := i-1 ;
  }
} else {
  D[0]:=0 ;
}
```

D

```
if (U[N] == 0){
  i := N ;
}
while(i ≠ 0){
  D[i] := U[i-1] ;
  i := i-1 ;
  D[0]:=0;
}
```

FIGURE 1 – Quel est le programme correspondant à l'automate ?

Multiplication rapide (30 min)

Le but de l'exercice est de démontrer la correction partielle de l'algorithme de multiplication rapide (MR) suivant :

MR

```
float x,r ;
x:=A ; y:=B ; r:=0 ; i:=0 ;
while(y>0){
  if (y mod 2 == 0){
    x := x+x ; y := y ÷ 2 ; i:=i+1 ;
  } else {
    r := r+x ; y := y-1
  }
}
```

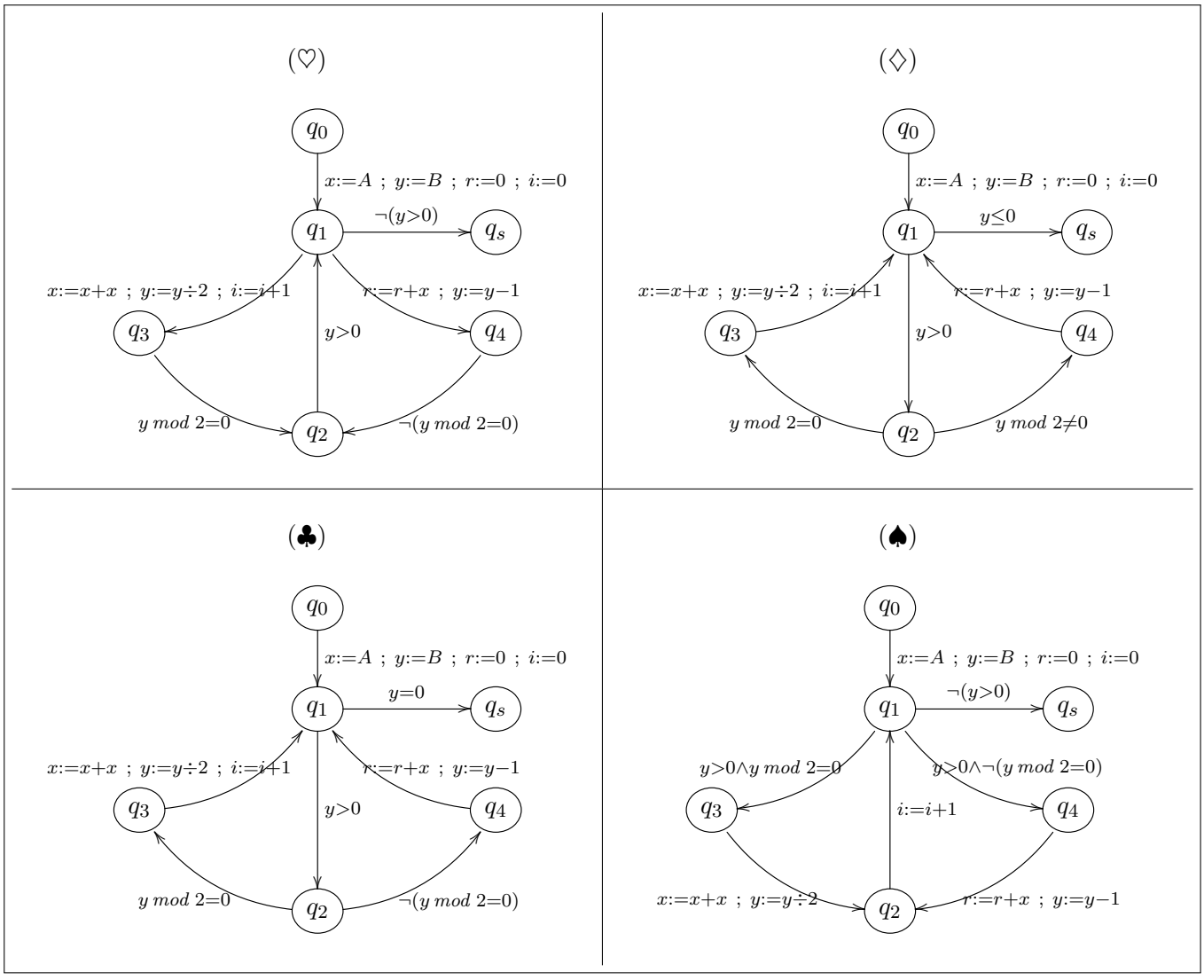


FIGURE 2 – Lequel des automates correspond à l'ALGORITHME MR ?