Nouveaux types d'automates

DURAND Ulysse

En caml, voilà une manière d'implémenter les automates :

```
type ('q, 'sig) automate = ('q -> bool)*('q -> bool)*(q'*sig'*('q list));;
(*('q,'ssig) automate = (i,f,delta) correspond a un automate ('q,'sig,i,f,delta)*)
```

1 Automate déterministe qui écrit

Definition 1.1 (Automate déterministe qui écrit)**.** Ce type d'automate permet de transformer un mot en un autre.

$$\mathcal{A} = (Q, \Sigma_1, \Sigma_2, i, F, \delta, \eta)$$
$$i \in Q, F \in \mathcal{P}(Q)$$

$$\begin{aligned}
\delta &: Q \times \Sigma_1 \to Q \\
\eta &: Q \times \Sigma_1 \to (\Sigma_2)^\star \\
\\
\delta^\star &: Q \times \Sigma_1^\star \to Q \\
(q, \epsilon) &\mapsto q \\
(q, l.m) &\mapsto \delta^\star(\delta(q, l), m), \text{ avec } l \in \Sigma_1 \\
\\
\eta^\star &: Q \times \Sigma_1^\star \to (\Sigma_2)^\star \\
(q, \epsilon) &\mapsto \epsilon' \\
(q, l.m) &\mapsto \eta(q, l).\eta^\star(\delta(q, l), m), \text{ avec } l \in \Sigma_1
\end{aligned}$$

Implementation en Caml :

```
type ('q, 'sig1, 'sig2) automatequiecrit = ('q*('sig2 list), 'sig1 ) automate;;
(*('q,'sig1,'sig2) automatequiecrit = (i,f,g) correspond a un automate ('q,'sig1,'sig2,i,f,delta,e
supposons g sous la forme g((q,m),l) = (q',p::m), alors delta(q,l) = q' et eta(q,l) = p*)
```

2 Automate qui écrit avec une mémoire

Definition 2.1 (Automate qui écrit avec mémoire)**.** Ce type d'automate n'a pas de d'intérêt théorique vu que la mémoire est finie, mais a un intérêt pratique.

$$\mathcal{A} = (Q, \Sigma_1, \Sigma_2, S_m, i, F, \delta, \eta)$$
$$Q, \Sigma_1, \Sigma_2, i, F \text{ restent inchangés}$$

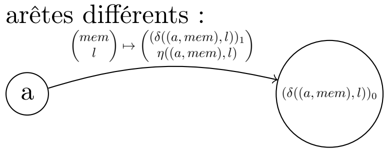
$$\begin{aligned}
\delta &: Q \times S_m \times \Sigma_1 \to Q \times S_m \\
\eta &: Q \times S_m \times \Sigma_1 \to (\Sigma_2)^\star
\end{aligned}$$

$$\begin{aligned}
\delta^\star : Q \times S_m \times \Sigma_1^\star &\to Q \\
(q, M, \epsilon) &\mapsto q \\
(q, M, l.m) &\mapsto \delta^\star(\delta(q, M, l), m), \text{ avec } l \in \Sigma \\
\eta^\star : Q \times S_m \times \Sigma_1^\star &\to (\Sigma_2)^\star \\
(q, M, \epsilon) &\mapsto \epsilon' \\
(q, M, l.m) &\mapsto \eta(q, M, l).\eta^\star(\delta(q, M, l), m), \text{ avec } l \in \Sigma_1
\end{aligned}$$

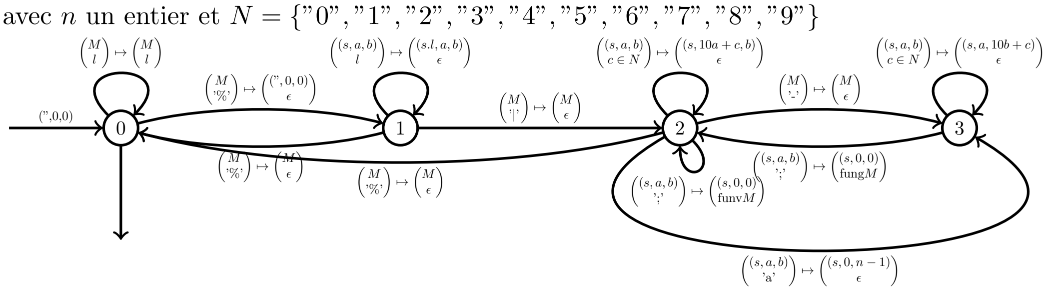
Il s'agit en fait d'un automate qui écrit avec $Q' = Q \times S_m$
Implementation en Ocaml :

```
type ('q, 'sm, 'sig1, 'sig2) automatequiecritavecmemoire = ('q*'sm,'sig1,'sig2) automatequiecrit
(*('q, 'sm, 'sig1, sig2) automatequiecritavecmemoire : (i,f,g) correspond a un automate qui ecrit ave
supposons g sous la forme g((q,m,mem),l) = ((q',mem'),p@m), alors delta((q,mem),l) = (q',mem') et
```

Pour les représenter graphiquement, nous ferons comme les automates, mais avec des annotation sur les arêtes différents :



Un exemple d'automate qui écrit avec une mémoire :
avec n un entier et $N = \{"0", "1", "2", "3", "4", "5", "6", "7", "8", "9"\}$



Si A est cet automate qui écrit,
$n = 6$,
$\text{funv}(s, a, b) = "x\_a"$,
$\text{fung}(s, 0, 3) = \text{funv}(s, 0, 3).s.\text{funv}(s, 1, 3).s.\text{funv}(s, 2, 3).s.\text{funv}(s, 3, 3)$,

$\eta^\star(0, (”, 0, 0),$ `\frac{{%|0;%+%|15;%}^{(%*|1-4;%)}}{%+|a;%}` $) =$
`\frac{{x_0+x_15}^{x_1*x_2*x_3}}{x_0+x_1+x_2+x_3+x_4+x_5}`