Automates à états finis : Déterminisation et Minimisation

Définition : Un automate à états finis ***A = (X, Q, q_0, F, ∂)*** est dit ***déterministe*** si et seulement si :

1. $\forall \boldsymbol{q} \in \boldsymbol{Q}, \forall \boldsymbol{a} \in \boldsymbol{X}, |\boldsymbol{\partial}(\boldsymbol{q}, \boldsymbol{a})| \leq \mathbf{1}$;
2. ***A*** ne comporte pas des ε-transitions.

Algorithme de déterminisation d'un automate à états finis sans ε-transitions :

Soit un automate à états finis non déterministe ***A = (X, Q, q_0, F, ∂)***. Son automate équivalent et déterministe ***A'= (X, {E_0, E_1,….}, E_0, F', ∂')*** est défini comme suit :

1. ***E_0 = {q_0}*** (c'est l'état initial du nouvel automate);
2. Pour chaque état ***E_i*** et chaque symbole ***a***, construire un nouvel état ***E_{i+1}*** qui contient les états obtenus par toutes les transitions ***a*** sur les états de ***E_i***, d'où : ***E_{i+1}*** = $\bigcup_{q \in E_i} \boldsymbol{\partial}(\boldsymbol{q}, \boldsymbol{a})$. Une nouvelle transition ***a*** est définie dans le nouvel automate entre les deux états par : ***∂'(E_i, a)*** = ***E_{i+1}***.
3. Chaque état ***E_i*** qui contient au moins un état final du premier automate devient état final du nouvel automate, d'où : $\boldsymbol{E_i} \in \boldsymbol{F'}$ ***ssi***: $\exists\, \boldsymbol{q} \in \boldsymbol{E_i}$ ***tel que*** $\boldsymbol{q} \in \boldsymbol{F}$.
4. Renommer les états en tant qu'états simples.

Algorithme de minimisation d'un automate à états finis :

Rappel : Deux états **q** et **p** sont ***β–équivalents*** si et seulement si ils permettent d'atteindre les états finaux avec ***le même ensemble de mots***.

Objectif de l'algorithme: Le but de la minimisation est de ***fusionner*** les états ***β–équivalents***.

Soit un automate à états finis ***A = (X, Q, q_0, F, ∂)***. La procédure de minimisation est la suivante :

1. Eliminer les états inaccessibles et/ou puits ;
2. Déterminiser l'automate s'il ne l'est pas ;
3. Construire la partition initiale de classes $\boldsymbol{\beta}_0$ définie par **{*A, B*}** tel que : ***A*** est la classe contenant les états finaux, et ***B*** est celle contant les états non finaux ;
4. **Séparation des états :** Pour chaque partition de classes $\boldsymbol{\beta}_i$ faire :
   a) Pour toute paire d'états ***q*** et ***p*** de la même classe de $\boldsymbol{\beta}_i$, s'il existe un symbole ***a*** dans ***X*** tel que les états des transitions ***∂(q, a)*** et ***∂(p, a)*** ne sont pas dans la même classe de $\boldsymbol{\beta}_i$, alors les états ***q*** et ***p*** doivent être séparés dans la partition $\boldsymbol{\beta}_{i+1}$. On note cela par $\boldsymbol{q}\ \cancel{\boldsymbol{\beta}_{i+1}}\ \boldsymbol{p}$.
   b) Construire la nouvelle partition $\boldsymbol{\beta}_{i+1}$ tel qu'il n'existe aucune classe qui contient deux états ***q*** et ***p*** tel que $\boldsymbol{q}\ \cancel{\boldsymbol{\beta}_{i+1}}\ \boldsymbol{p}$.
5. Répéter l'étape (4) jusqu'à ce que $\boldsymbol{\beta}_i = \boldsymbol{\beta}_{i+1}$.

A la fin de l'algorithme on obtient une partition finale de classe $\boldsymbol{\beta}_n$ telle que chaque classe de cette partition contient que des états qui sont ***β–équivalents*** entre eux.

Chaque classe de cette partition devient un état de l'automate minimal.

Un état ***A*** est initial s'il contient l'état initial ***q_0***. Ainsi un état ***A*** devient état final s'il contient au moins un état $\boldsymbol{q} \in \boldsymbol{F}$.

Finalement, on définit une transition entre deux classes ***A*** et ***B*** de $\boldsymbol{\beta}_n$ par ***∂(A, a)=B*** si et seulement si : $\exists$ ***q*** ∈ ***A*** **et** ***p*** ∈ ***B*** **tel que** ***∂(q, a) = p.***

H. Mahfoud