1 Automates finis

Définition 1.1. Un automate fini A est universel ssi

$$A \models \quad \Phi_u \triangleq \left(\nu F \,.\, \lambda X \,.\, X \wedge \bigwedge_{a \in \Sigma} F \quad \langle a \rangle X\right) \quad final$$

où *final* représente le prédicat qui est vrai si un état est final et faux autrement.

En effet, cette formule correspond à l'ensemble de toutes les "traces" ou mots $t \in \Sigma^*$ qui précèdent au moins un état vérifiant *final*. Autrement dit, pour impliquer Φ_u, A doit avoir un état final après chaque trace $t \in \Sigma^*$, A accepte donc tous les mots de Σ^* et est donc un automate universel.

Définition 1.2. Un automate A représente le langage vide ssi

$$A \models \quad \Phi_\varnothing \triangleq \left(\nu F \,.\, \lambda X \,.\, X \wedge \bigwedge_{a \in \Sigma} F \quad [a] X\right) \quad \neg final$$

Si $A \models \Phi_\varnothing$, pour n'importe quelle trace $t \in \Sigma^*$, tous les états qui se trouvent après celle-ci ne sont pas finaux, donc aucun mot n'est reconnu par A, pas même le mot vide, donc A représente le langage vide.

Définition 1.3. Soit deux automates finis A et B, $L(A) \subseteq L(B)$ ssi

$$A,B \models \quad \Phi_\subseteq \triangleq \left(\nu F \,.\, \lambda X, Y \,.\, (X \Rightarrow Y) \wedge \bigwedge_{a \in \Sigma} F \quad \langle a \rangle_1 X \quad \langle a \rangle_2 Y\right) \quad final \quad final$$

Soit deux automates A et B, si $(A, B) \models \Phi_\subseteq$ alors pour chaque trace $t \in \Sigma^*$ qui correspond à un mot reconnu par A, la même trace correspond à un mot reconnu par B. Donc tous les mots reconnus par A sont reconnus par B et $A \subseteq B$.

2 Automates de Büchi

Définition 2.1. Un automate de Büchi A est vide ssi

il n'existe pas de mot infini qui mène à un état acceptant infiniment souvent. C'est équivalent à dire qu'il n'existe pas de pair de mots *(w1,w2)* tels que *w1* conduit à un état acceptant depuis l'état initial et depuis cet état acceptant *w2* conduit à un état acceptant infiniment souvent.

Un automate de Büchi A représente le langage vide ssi

$$A \models \quad \Psi_{\emptyset} \triangleq \neg\left(\left(\nu E\,.\,\lambda X\,.\,X \wedge \bigvee_{a\in\Sigma} E \quad \langle a\rangle X\right)\left(\left[\mu F\,.\,\lambda Y\,.\,Y \vee \bigvee_{b\in\Sigma} F \quad \langle b\rangle Y\right] \quad acc\right)\right)$$

où acc représente le prédicat qui est vrai si un état est acceptant et faux autrement.

On procède en construisant la formule qui est vraie ssi A est non-vide puis la négation permet d'obtenir une formule vraie ssi A est vide.

La partie droite de l'équation représente le plus petit point fixe de F qui renvoie les états acceptants les plus proches pour chaque état. Ainsi on applique l'expression de droite à l'expression de gauche et on obtient un prédicat vraie ssi X est vraie çàd on a trouvé un état acceptant (le plus proche, donc atteignable avec un mot fini) et en avançant d'un nouveau mot w on peut de nouveau vérifier cette propriété çàd trouver un nouvel état acceptant. En prenant le plus grand point fixe, on s'assure donc que il existe un mot infini passant par un nombre d'état acceptant infini, ce qui signifie bien que cette formule représente les langages non-vides et grace à la négation on a bien une formule vraie ssi A représente le langage vide.

Définition 2.2. Un automate de Büchi A est universel ssi

Il représente l'ensemble des ω-langages noté Σ^{ω}.

La façon choisie de caractériser un tel automate est de considérer que pour n'importe quel mot $w \in \Sigma^*$, il existe une *lecture* λ tel que il existe un futur état acceptant atteignable après λ. Cela implique donc que tous les mots dans Σ^{ω} sont reconnus, donc que $\Sigma^{\omega} \subseteq L(A)$, et il est évident que $L(A) \subseteq \Sigma^{\omega}$, car A est un automate de Büchi.

$L(A) = \Sigma^{\omega}$ ssi

$$A \models \quad \Psi_{\Sigma^{\omega}} \triangleq \left(\left(\nu E\,.\,\lambda X\,.\,X \wedge \bigwedge_{a\in\Sigma} E \quad \langle a\rangle X\right)\left(\left[\mu F\,.\,\lambda Y\,.\,Y \vee \bigvee_{b\in\Sigma} F \quad \langle b\rangle Y\right] \quad acc\right)\right)$$

On peut remarquer qu'il s'agit de la même formule utilisée pour caractériser les automates de Büchi non-vides à la différence près d'une disjonction remplacée par une conjonction. En effet, on ne s'enquiert pas qu'il existe un ω-mot reconnu, mais que tous les ω-mots soient reconnus. La partie droite reste inchangée et fait toujours le même travail, elle renvoie le prochain état acceptant atteignable.

Si on développe la partie gauche, on obtient que pour chaque mots w $\in \Sigma^{\omega}$ il existe un prochain état acceptant.

Il est à noter qu'il n'y a pas l'utilité d'utiliser l'opérateur box à la place du diamant ici, parce que on ne s'intéresse pas à ce que toutes les lectures d'un ω-mot soient reconnus, mais qu'il en existe au moins une, peu importe donc le chemin emprunté, tant que le mot est reconnu.

Définition 2.3. Soit deux automates de Büchi A et B, $L(A) \subseteq L(B)$ ssi
Soit $H_1 := \left(\left[\mu F \,.\, \lambda X \,.\, X \vee \bigvee_{b\in\Sigma} F \quad \langle b\rangle_1 X\right] \quad acc1\right)$
et $\quad H_2 := \left(\left[\mu F \,.\, \lambda Y \,.\, Y \vee \bigvee_{b\in\Sigma} F \quad \langle b\rangle_2 Y\right] \quad acc2\right)$

$$A, B \models \quad \Psi_{\subseteq} \triangleq \left(\left(\nu E \,.\, \lambda\, X, Y \,.\, (X \Rightarrow Y) \wedge \bigwedge_{a\in\Sigma} E \quad \langle a\rangle_1 X \quad \langle a\rangle_2 Y\right) \quad H_1 \quad H_2\right)$$

Le but est de caractériser le fait que chaque ω-mot reconnu par A est reconnu par B. H_1 et H_2 représente respectivement les prédicats vrais ssi il existe un chemin vers un état acceptant de A, de B. Le point fixe de E nous assure alors que si il existe un prochain état acceptant sur A après lecture d'un mot w, il existe également un état acceptant depuis B après une certaine lecture de w, et ce pour n'importe quel mot w. Donc tout ω-mot reconnu par A, est également reconnu par B.

3 Automates de Muller

La différence avec les automates de Büchi est la condition d'acceptation des ω-mots. $\mathfrak{F}$ n'est plus un ensemble d'états finaux ou acceptants comme pour les automates de Büchi ou les automates finis, mais un ensemble d'ensemble d'états acceptants. Un mot w est reconnu par A si il existe une lecture de ce mot ρ sur A, tel que l'ensemble des états dans ρ se répétant infiniment souvent noté $inf(\rho) \in \mathfrak{F}$.

Définition 3.1. Un automate de Muller A est vide ssi
il n'existe pas de mot infini w tel qu'il existe une lecture ρ de w dont $inf(\rho) \in \mathfrak{F}$.
Un automate de Muller A représente le langage vide ssi

Soit $C := \mu G \,.\, \lambda X \,.\, X \;\vee \bigvee_{b \in \Sigma} G \quad \langle b \rangle X$

$$A \models \quad \chi_{\varnothing} \triangleq \neg \left(\bigvee_{F_i \in \mathfrak{F}} \left[\left(\nu E \,.\, \lambda X, Y \,.\, X \wedge \bigvee_{a \in \Sigma} E \quad \langle a \rangle X \wedge (\neg Y \vee \bigwedge_{a \in \Sigma} \neg E \quad \langle a \rangle Y) \right) \bigwedge_{f \in F_i} C\, f \quad \bigvee_{f \notin F_i} C\, f \right] \right)$$

où f représente le prédicat vrai si l'état est f et faux autrement.

On procède par la même technique que pour les automates de Büchi vides : on caractérise ce que signifie avoir un langage non-vide avec une formule, puis on prend simplement la négation pour obtenir la caractérisation des langages vides.

C correspond au prédicat "il existe un chemin vers l'état passé en paramètre". E vérifie que pour à chaque étape il existe au moins un chemin menant vers tous les états $f \in F_i$" et également que à chaque étape soit on ne peut atteindre aucun des états qui ne sont pas dans F_i, soit ce sera le cas plus tard (aucun de ces états n'apparaîtra donc de manière infinie, à un moment plus aucun de ces états ne peut apparaître). En prenant la disjonction sur tous les ensembles de $\mathfrak{F}$, on obtient qu'il existe au moins un de ces ensembles tel que il existe un mot infini passant par chacun de ses états et par aucun autre. C'est donc la caractérisation d'un langage non-vide et la négation caractérise donc bien les langages vides.