5 Satisfaisabilité et vérification par des techniques d'automates : le cas du temps linéaire

La notion clef pour obtenir des algorithmes efficaces est celle d'automates alternants : ceux-ci permettent de décrire de façon compacte les modèles d'une formule de logique temporelle. C'est ce que nous montrons tout d'abord. Puis nous étudions l'algorithmique des automates alternants, montrant qu'on obtient bien des algorithmes optimaux pour le model-checking (PSPACE-complet) et la satisfaisabilité (PSPACE-complet). (cf. [Var96]).

5.1 Automates de Büchi alternants

Un automate de Büchi alternant est un tuple $(Q, \Sigma, Q_f, q_0, \delta)$ où Q ensemble d'états (finis), Σ alphabet, $Q_f \subseteq Q$, $q_0 \in Q$, transitions : $\delta : Q \times \Sigma \to \mathcal{B}^+(Q)$. $\mathcal{B}^+(Q)$ est l'ensemble des combinaisons Booléennes positives d'éléments de Q : ce sont des formules propositionnelles sans négation sur l'alphabet Q de variables propositionnelles,

Un *calcul* de A sur $w \in \Sigma^\omega$ est un arbre (fini ou infini) étiqueté par Q et tel que : $r(\epsilon) = q_0$ et si p est une position de longueur i telle que $r(p) = q$ et $w(i) = a$ et $\delta(q, a) = \phi$, l'ensemble $\{p \cdot 1, ..., p \cdot n\}$ des successeurs de p est tel que $r(p \cdot 1), ..., r(p \cdot n) \models \phi$.

La notion de satisfaction est ici définie par : $q_1, \dots, q_n \models q$ ssi $q \in \{q_1, \dots, q_n\}$ puis s´étend au calcul Booléen comme d'habitude. (En particulier, pour tout ensemble d'états E, $E \models$ **true** et $E \not\models$ **false**).

Un calcul de l'automate alternant est *réussi* si tout chemin infini contient une infinité d'états finaux. (NB : il peut ne pas exister de calcul au cas où **false** est présent comme résultat d'une transition ; il peut exister des calculs non réussis. Les calculs peuvent être des arbres finis, au quel cas ils sont nécessairement réussis).

Exemple 15 Soit l'automate alternant défini par $Q = \{q_0, q_1\}$, $\Sigma = \{a, b, c, d\}$ et la fonction de transition donnée par la table :

δ	a	b	c	d
q_0	**true**	**true**	$q_0 \wedge q_1$	q_0
q_1	**false**	**true**	q_1	**true**

Un exemple de calcul sur le mot $cccdcdad\cdots$ est représenté dans la figure 10.

On peut passer des automates alternants aux automates non-déterministes au prix d'une exponentielle :

Lemme 16 *Soit $\mathcal{A}$ un automate de Büchi alternant, d'ensemble d'états Q et acceptant le langage L. Il existe un automate de Büchi non déterministe $\mathcal{A}'$ dont l'ensemble des états est $2^Q \times 2^Q$ et qui accepte L.*

Preuve: On construit $\mathcal{A}'$ de la manière suivante : les états sont constitués des paires d'ensembles d'états de $\mathcal{A}$. L'état initial est $(\{q_0\}, \emptyset)$ et les états finaux sont les paires $(\emptyset, S)$. Si F est l'ensemble des états finaux de $\mathcal{A}$, les transitions $((S, S'), a, (S_1, S_1')) \in \delta'$ sont données par :

Si $S \neq \emptyset$ alors S_1 et S_1' sont tels que

$$\bigwedge_{q_1 \in S_1} q_1 \models \bigwedge_{q \in S \setminus F} \delta(q, a)$$

$$\bigwedge_{q_1 \in S_1'} q_1 \models \bigwedge_{q \in S'} \delta(q, a) \wedge \bigwedge_{q \in S \cap F} \delta(q, a)$$

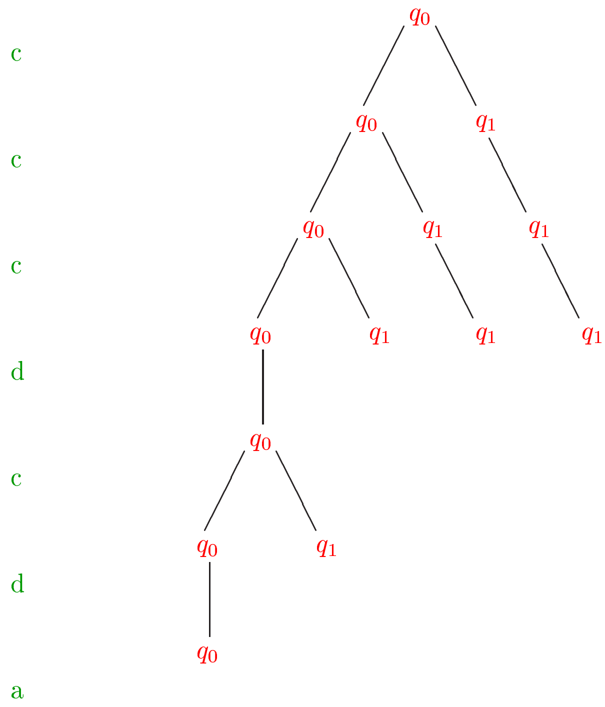

FIG. 10 – Un calcul de l'automate alternant sur le mot $ccc\ldots$

Si $S = \emptyset$ alors $S'_1 = \emptyset$ et S_1 est tel que

$$\bigwedge_{q_1 \in S_1} q_1 \models \bigwedge_{q \in S'} \delta(q, a)$$

Supposons d'abord que ρ est un calcul réussi de $\mathcal{A}$ sur w et construisons un calcul réussi r de $\mathcal{A}'$ sur w. (S_i, S'_i) est défini par récurrence : $S_0 = \{q_0\}$ et $S'_0 = \emptyset$. $S_i \cup S'_i$ restera l'ensemble des étiquettes des positions de longueur i dans ρ. On partage les positions de longueur i en deux ensembles disjoints : P_i et P'_i qui sont définis par récurrence par $P_0 = \{\epsilon\}$, $P'_0 = \emptyset$.

Si $S_i \neq \emptyset$

$$P_{i+1} \stackrel{\text{def}}{=} \{p \cdot j \mid p \in P_i, \rho(p) \notin F, p \cdot j \in \mathrm{Pos}(\rho)\}$$

$$P'_{i+1} \stackrel{\text{def}}{=} \{p \cdot j \mid p \in P'_i, p \cdot j \in \mathrm{Pos}(\rho)\} \cup \{p \cdot j \mid p \in P_i, \rho(p) \in F, p \cdot j \in \mathrm{Pos}(\rho)\}$$

Si $S_i = \emptyset$ P_{i+1} est l'ensemble des positions de longueur $i+1$ de ρ et $P'_{i+1} = \emptyset$.

Dans tous les cas, $S_{i+1} \stackrel{\text{def}}{=} \{\rho(p) \mid p \in P_{i+1}\}, S'_{i+1} \stackrel{\text{def}}{=} \{\rho(p) \mid p \in P'_{i+1}\}$. On vérifie d'abord que $((S_i, S'_i), a, (S_{i+1}, S'_{i+1})) \in \delta'$ pour tout i, ce qui est une conséquence des définitions de δ' et de ρ. Il reste à montrer que le calcul r défini par $r(i) = (S_i, S'_i)$ est un calcul réussi. Par l'absurde : si r ne contient qu'un nombre fini de lettres telles que $S_i = \emptyset$, alors, pour $i > i_0$, $S_i \neq \emptyset$. Donc, pour tout $i > i_0$, $P_i \neq \emptyset$. Soit ρ' le sous-arbre de ρ ne contenant que les positions de P_i, pour $i > i_0$ (et les mêmes positions que ρ pour $i \leq i_0$). ρ' est bien un arbre puisque, si $p \cdot j \in P_{i+1}$ et $i > i + 0$, alors $p \in P_i$. D'autre part ρ' est un arbre infini puisque P_i est non vide pour tout i. Donc ρ' contient un chemin infini (lemme de König). Sur ce chemin, il n'y a aucun état final, ce qui est absurde.

Réciproquement, supposons que r est un calcul réussi de $\mathcal{A}'$ sur w. On construit par récurrence un calcul ρ de l'automate alternant, gardant comme invariant le fait que $S_i \cup S'_i$ est l'ensemble