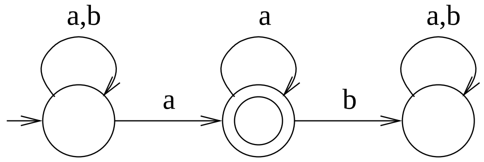

Figure 12: Un automate de Büchi acceptant $(a+b)^* a^\omega$

6 Jeux à parité et le théorème de complémentation

L'objectif de ce paragraphe est de démontrer la clôture par complémentaire de la classe des langages reconnus par un automate de Büchi. Nous avons choisi la preuve donnée dans [Tho99] parce qu'elle utilise les automates alternants et permet d'introduire des notions utiles pour la suite, typiquement la vérification en μ-calcul. Il s'agit des conditions d'acceptance à parité, des automates faibles, des jeux à deux joueurs et du théorème de détermination des jeux.

6.1 Quelques remarques sur les automates de Büchi

Tout d'abord, il n'est pas toujours possible de déterminiser un automate de Büchi:

Exercice 11 Montrer qu'il n'existe aucun automate de Büchi déterministe qui accepte le langage défini par l'automate non-déterministe de la figure 12.

En fait, ce problème de non-déterminisme va pouvoir être résolu grâce aux automates alternants: les automates non-déterministes correspondent à des automates alternants ne comportant que des disjonctions dans les transitions (et pas de conjonction). L'introduction de formules Booléennes positives arbitraires permet de symétriser les transitions et on peut donc espérer que la complémentation est plus simple.

Mais il existe un deuxième problème orthogonal, lié à la condition d'acceptance de Büchi elle-même:

Exercice 12 Montrer que, pour chacun des automates donnés dans les figures 13 et 14, bien que déterministes, il ne suffit pas d'échanger les états finaux et non finux pour obtenir un automate qui reconnaisse le complémentaire.

Ces exemples montrent que le problème de complémentation ne peut être traîté comme dans le cas des automates de mots finis.

Il existe cependant un cas où l'on sait bien complémenter les langages de mots infinis. Il a été vu dans la section 5: si $\mathcal{A}_\phi$ est un automate (alternant) acceptant les modèles d'une formule LTL ϕ, $\mathcal{A}_{\neg\phi}$ accepte le complémentaire. La raison pour laquelle ce passage au complémentaire est simple est dû à la *faiblesse* de l'automate. On dira qu'un automate ed Büchi alternant est *faible* s'il existe un ordre sur les états tel que, pour tout état q, tout symbole a de l'alphabet et

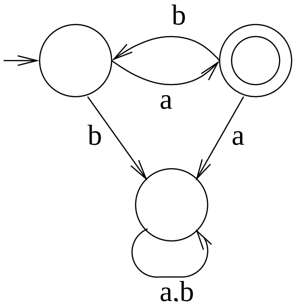

Figure 13: Un automate de Büchi déterministe acceptant $(ab)^\omega$

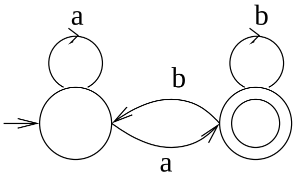

Figure 14: Un automate de Büchi déterministe acceptant $(a^*b)^\omega$

tout état q' apparaissant dans $\delta(q, a)$, $q' \leq q$. Les automates correspondant aux formules LTL sont faibles en ce sens: il suffit de choisir l'ordre sous-formule.

Exercice 13 Montrer que, pour un automate de Büchi déterministe et faible, il suffit d'échanger les états finaux et non-finaux pour obtenir un automate qui accepte le complémentaire.

Les automates de Büchi alternants faibles ne sont cependant pas assez expressifs pour que n'importe quel langage Büchi-reconnaissable soit reconnu par un tel automate. Il nous faut donc changer la condition pour accepter (c'est pourquoi on introduit une condition de parité) puis montrer que tout langage accepté par un automate de Büchi est accepté par un automate alternant à parité faible, puis montrer la complémentation des automates à parité faible et enfin retourner aux automates de Büchi. C'est ce que nous faisons dans les paragraphes qui suivent.

6.2 Automates alternants faibles à condition de parité

On suppose qu'à chaque état q est associé un entier $r(q)$ appelé *rang*. Un automate avec condition de parité est un automate fini comme précédemment, mais la condition pour accepter est remplacée par:

> Il existe un calcul ρ de $\mathcal{A}$ sur w tel que le minimum des rangs des états de ρ apparaissant infiniment souvent est pair.

Un automate alternant accepte un mot avec la condition de parité ssi il existe un calcul ρ de l'automate tel que tout chemin infini satisfait la condition de parité.

Un automate (alternant ou non) avec condition de parité est *faible*, si pour toute transition $\delta(q,a) = \phi$ et tout état $q' \in \phi$, $r(q') \leq r(q)$.

Exercice 14 Les indices dans la notation des états indiquent leur rang. L'état initial est q_3^1.

$$
\begin{array}{lcl}
q_3^1 & \xrightarrow{a} & (q_2^1 \vee q_3^1) \wedge q_1^1 \\
q_3^1 & \xrightarrow{b} & (q_2^2 \vee q_3^1) \wedge q_0^1 \\
q_2^1 & \xrightarrow{b} & \textbf{true} \\
q_2^2 & \xrightarrow{a} & \textbf{true} \\
q_1^1 & \xrightarrow{b} & q_1^1 \\
q_1^1 & \xrightarrow{a} & q_1^1 \vee q_0^1 \\
q_0^1 & \xrightarrow{a} & q_0^1
\end{array}
$$

les transitions non indiquées vont vers **false**.

L'automate est-il faible ?

Donner un calcul de l'automate sur le mot aba^ω.

Donner un calcul sur le mot $aaba^\omega$.

Décrire l'ensemble des mots acceptés.

Exercice 15 Calculer un automate alternant faible avec condition de parité pour le langage $(ab)^\omega$.

Exercice 16 Montrer que, pour toute formule LTL, on peut calculer en temps linéaire un automate alternant faible avec condition de parité qui accepte les modèles de ϕ.

Donner un automate alternant faible avec condition de parité qui n'accepte les modèles d'aucune formule de LTL (la preuve n'est pas demandée).

6.3 Passage des automates de Büchi aux automates alternants faibles avec condition de parité

Lemme 22 *Etant donné un automate de Büchi quelconque $\mathcal{A}$, on peut construire en temps polynômial un automate alternant faible avec condition de parité $\mathcal{A}'$ tel que $L(\mathcal{A}) = L(\mathcal{A}')$.*

Preuve: Soit $\mathcal{A} = (Q, Q_f, q_0, \Delta)$ un automate de Büchi. On construit $\mathcal{A}'$ de la manière suivante:

- l'ensemble des états est $Q \times \{0, \ldots, 2 \times n\}$ où n est le nombre d'éléments de Q (si bien que $\mathcal{A}'$ contient $O(n^2)$ états).
- Le rang est donné par la deuxième composante des états.

- L'état initial est $(q_0, 2n)$.
- La fonction de transition est définie par:
  - pour tout $q \in Q$, $(q,0) \xrightarrow{a} \bigvee_{(q,a,q')\in\Delta}(q',0)$. Autrement dit, sur la composante 0, $\mathcal{A}'$ se comporte comme $\mathcal{A}$.
  - Si $i > 0$ est pair, $(q,i) \xrightarrow{a} \bigvee_{(q,a,q')\in\Delta}(q,i) \wedge (q,i-1)$. La conjonction force alors à rencontrer de nouveau un état pair, ce qui ne sera possible que si l'on rencontre un état final.
  - Si i est impair et $q \notin Q_f$, alors $(q,i) \xrightarrow{a} \bigvee_{(q,a,q')\in\Delta}(q',i)$. Autrement dit, on reste sur le rang impair tant qu'on a pas rencontré un état final.
  - Si i est impair et $q \in Q_f$, alors $(q,i) \xrightarrow{a} \bigvee_{(q,a,q')\in\Delta}(q',i-1)$.

Supposons tout d'abord que ρ est un calcul réussi de $\mathcal{A}$ sur w. On construit ρ', calcul de $\mathcal{A}'$ sur w, par récurrence sur la longueur de la position p de ρ', tel que $\rho'(p) = (\rho(|p|), i)$ pour un certain i :

- La racine est étiquetée par $(q_0, 2n)$.
- Si p est une position de ρ' étiquetée par (q,i) et $\rho(|p|) = q$, $w(|p|) = a$ et $\rho(|p|+1) = q'$, on ajoute un ou deux fils à p dans ρ', en respectant la relation de transition ci-dessus définie et de telle manière que chacun des fils soit étiqueté par un état de la forme (q', j).

Tous les chemins de ρ' sont alors infinis et contiennent une infinité d'états de la forme (q_f, i) où $q_f \in Q_f$. Comme l'automate est faible, sur chaque chemin le rang se stabilise sur une certaine valeur. Cette valeur ne peut pas être impaire puisque, lorsque le rang est impair, en passant par un état final, le rang décroit. Il en résulte que les rangs se stabilisent sur chaque branche sur des valeurs paires et donc le calcul est réussi.

Réciproquement, supposons que ρ' est un calcul acceptant de $\mathcal{A}'$ sur w. On définit une *tranche* de ρ' comme étant l'ensemble de tous les états étiquetant des nœuds de même profondeur. Remarquons que, si une tranche contient $(q, 2i)$, alors la tranche suivante contient un état $(q', 2i-1)$. Si le calcul est acceptant le nœud étiqueté $(q', 2i-1)$ doit avoir un descendant étiqueté par $(q'', 2i-2)$. Donc, par récurrence et puisque l'état initial est $(q_0, 2n)$, à partir d'une certaine profondeur, toutes les tranches d'un calcul acceptant contiennent, pour chaque $i \in \{0, \ldots, n\}$ un état $(q_i, 2i)$. de plus comme, il n'y a qu'un nombre fini de tranches possibles, il existe une tranche S répétée infiniment souvent et telle que S contient au moins un état (q,i) pour chaque rang pair i. De plus, on peut supposer sans perdre de généralité (comme vu ci-dessus) que, de chaque état $(q, 2i)$ de S, il existe un chemin dans ρ' vers un état $(q', 2i-2) \in S$. On choisit alors de numéroter les états de Q de manière à ce qu'il existe un chemin dans ρ de $(q_i, 2i)$ vers $(q_{i-1}, 2i-2)$ pour $i > 0$. Par définition de $\mathcal{A}'$, ce chemin passe par un état $(q_i^f, 2i-1)$ tel que $q_i^f \in Q_f$.

Maintenant, au moins deux des états $q_0, \ldots, q_n$ sont identiques. Disons $q_i = q_{i-k}$. On construit alors un calcul de $\mathcal{A}$ comme suit; On commence par un chemin quelconque π_0 qui conduit à une position $(q_i, 2i)$ à une profondeur dont la tranche est S. On construit ensuite π_1 conduisant de $(q_i, 2i)$ à $(q_{i-k}, 2i-2k)$ (passant par $k+1$ tranches S). Si l'on oublie les deuxièmes composantes du mot $\pi_0\pi_1^\omega$, on obtient un calcul réussi de $\mathcal{A}$ sur w: comme $q_i = q_{i-k}$, il s'agit bien d'un calcul de $\mathcal{A}$ et, comme π_1 contient nécessairement une paire (q_f, j) avec $q_f \in Q_f$, il s'agit d'un calcul réussi.

Exercice 17 Construire un automate alternant faible avec condition de parité qui accepte le langage $(a+b)^*a^\omega$

Exercice 18 Soit $\mathcal{A}$ un automate de Büchi déterministe et complètement spécifié à n états. Construire un automate alternant faible avec condition de parité qui accepte le même langage et qui comporte $2n$ états.

Exercice 19 Construire un automate alternant à parité faible pour chacun des langages définis dans les figures 13 et 14.

Exercice 20 Montrer que, pour tout entier k, il existe un automate de Büchi $\mathcal{A}_k$ à n états tel que tout automate alternant faible à condition de parité qui accepte le même langage que $\mathcal{A}_k$ comporte au moins $k \times n$ états.

6.4 Jeux à parité

Pour démontrer le résultat de dualité des automates alternants faibles à condition de parité, il est commode de voir le problème en termes de jeux, ce qui d'ailleurs permettra plus loin de classifier certains problèmes de vérification en termes de stratégies gagnantes, puis de se ramener à des problèmes algorithmiques sur des graphes finis.

Un *jeu à parité* est un tuple $G = (Q_1, Q_2, \delta_1, \delta_2, r, q_0)$ où Q_1, Q_2 sont des ensembles d'états (pas nécessairement finis) disjoints, δ_1 est une application de $Q_1 \times \mathbb{N}$ dans 2^{Q_2} et δ_2 est une application de $Q_2 \times \mathbb{N}$ dans 2^{Q_1} telles que $\delta_k(q, i)$ n'est jamais vide. r est une fonction de *rang* de $Q_1 \cup Q_2$ dans $\mathbb{N}$. Enfin, $q_0 \in Q_1$ est l'*état initial*.

Une *configuration* du jeu G est un élément de $Q_1 \times \mathbb{N} \cup Q_2 \times \mathbb{N}$. Une *partie* est un mot infini P sur $Q_1 \cup Q_2$ tel que $w(0) = q_0$ et, si $w(i) = q \in Q_j$, alors $w(i+1) \in \delta_j(q, i)$. Le joueur 1 *gagne la partie* P si le minimum des rangs des états apparaissant infiniment souvent dans P est pair. Sinon, c'est le joueur 2 qui gagne.

Une *stratégie* (sans mémoire) s pour le joueur 1 (resp. le joueur 2) est une application de $Q_1 \times \mathbb{N}$ (resp. $Q_2 \times \mathbb{N}$) dans Q_2 (resp. Q_1) telle que $s(q, i) \in \delta_1(q, i)$ (resp. $\in \delta_2(q, i)$) pour tous q, i.

Un joueur B a une *stratégie gagnante* s'il gagne la partie en appliquant sa stratégie, quels que soient les coups de son adversaire.

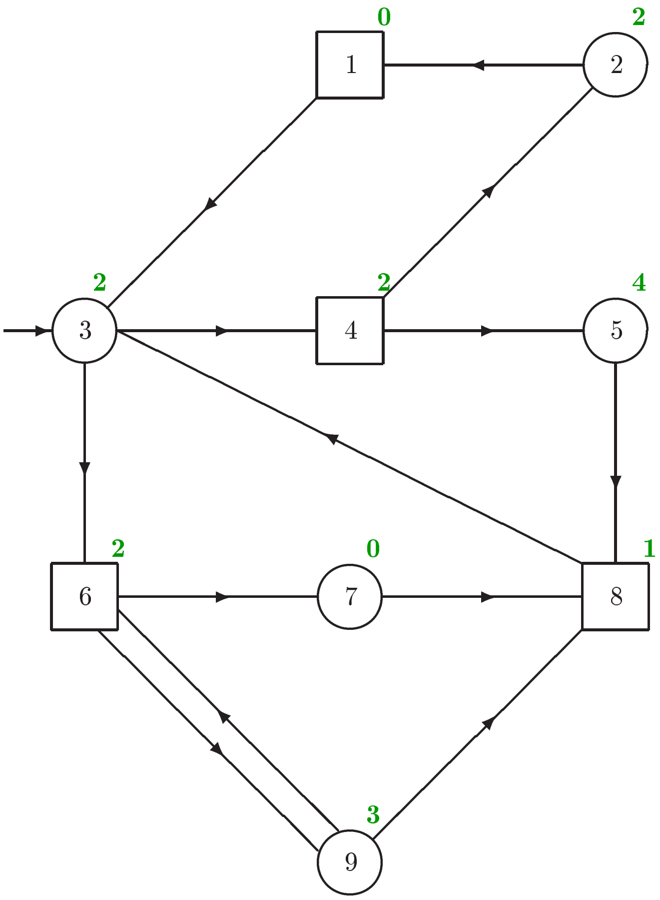

Figure 15: Un jeu à parité