Méthodes symboliques pour les jeux à pile

séminaire VERIMAG

18 avril 2003

Thierry CACHAT

RWTH Aachen (Aix-la-Chapelle)

http://www-i7.informatik.rwth-aachen.de/~cachat/

Résumé

On s'intéresse aux jeux à deux joueurs sur les graphes de transitions des automates à pile, avec différentes conditions de gain : accessibilité, Büchi (récurrence), et une nouvelle condition Σ_3. Après avoir défini ces jeux et rappelé le lien étroit avec la vérification (model-checking et synthèse de contrôleur), on évoquera les problèmes algorithmiques particuliers qu'ils soulèvent. À la suite des travaux de Bouajjani, Esparza et Maler sur l'accessibilité, on utilisera une méthode symbolique basée sur la construction d'un automate fini qui reconnaît la région gagnante, et on déduira de celui-ci une stratégie gagnante. On peut ensuite étendre ces techniques aux conditions de Büchi et Σ_3.

Plan

1. jeux : définition, problèmes algorithmiques, vérification
2. graphes infinis : jeux à pile
3. accessibilité : méthode symbolique
4. Büchi : accélération
5. condition Σ_3 : borne

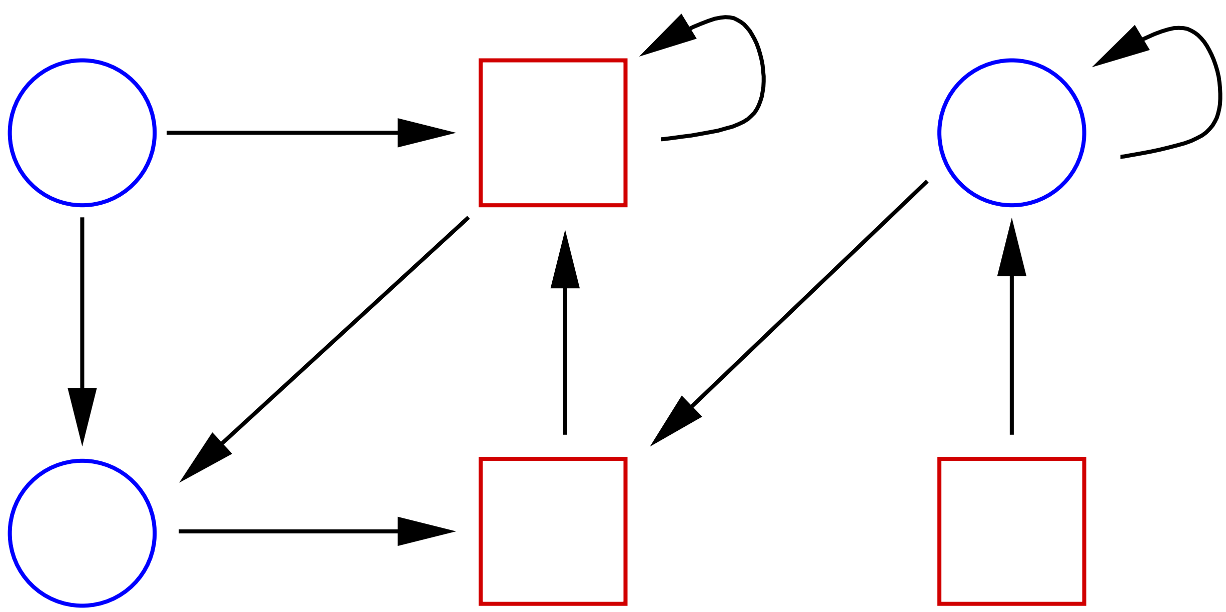
Références

E. Grädel, W. Thomas and T. Wilke eds.,
Automata, Logics, and Infinite Games,
A guide to current research, LNCS 2500, 2002.

projet "Games"
`http://www.games.rwth-aachen.de/index.html`

Jeu infini sur un graphe (fini)

Arène : graphe orienté (V, E), $E \subseteq V \times V$,
$V = V_0 \uplus V_1$
Depuis un sommet v :
si $v \in V_0$, i.e. ◯, Joueur 0 choisit v', vEv'
si $v \in V_1$, i.e. □, Joueur 1 choisit v', vEv'



partie $\pi = \pi_0\pi_1 \cdots \in V^\omega$, $\forall i\ \pi_i E \pi_{i+1}$
partie gagnée pour le Joueur 0 $\Leftrightarrow \pi \in L \subseteq V^\omega$

Différentes conditions de gain

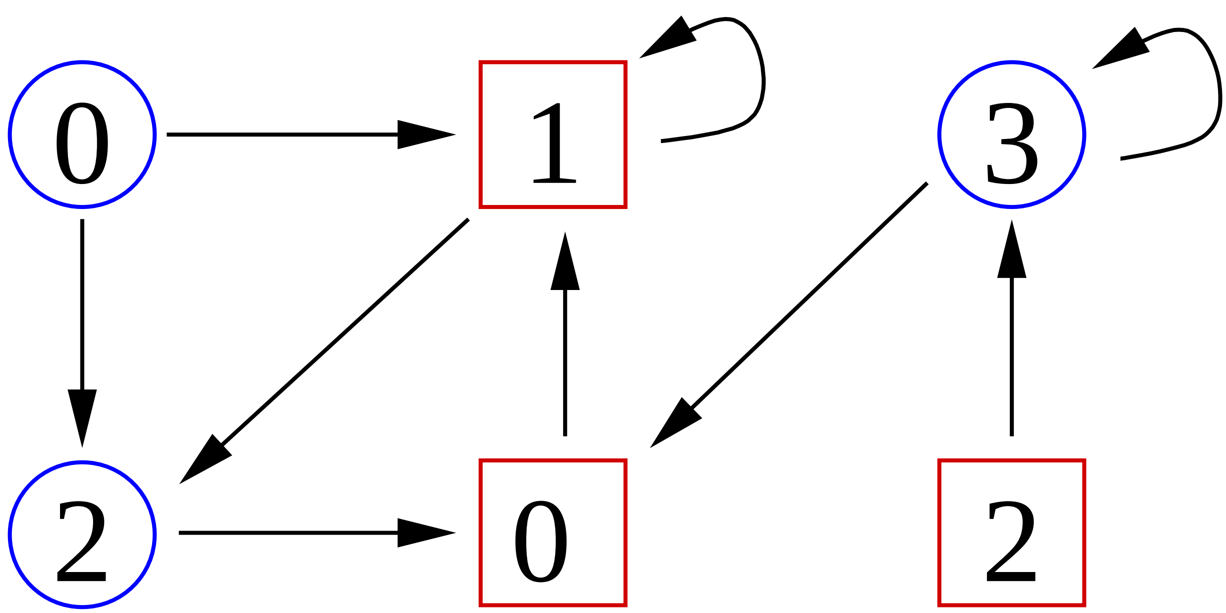
Accessibilité : étant donné $R \subseteq V$,

$$L = \{\pi \mid \exists i,\ \pi_i \in R\}$$

Récurrence (Büchi) : $L = \{\pi \mid \exists^\infty i,\ \pi_i \in R\}$

Parité : étant donné $c : V \longrightarrow \{0, \cdots, n\}$

$$L = \{\pi \mid \max(Inf(c(\pi))) \text{ est pair } \}$$



Problèmes algorithmiques

Depuis quelles positions le Joueur 0 peut-il gagner “à coup sûr” ?

Comment ?

Stratégie pour le Joueur 0

fonction (partielle) $f_0 : \begin{array}{ccc} V^*V_0 & \longrightarrow & V \\ \pi_0 \cdots \pi_n & \longmapsto & \pi_{n+1} \end{array}$

stratégie gagnante depuis une position : toutes les parties jouées conformément à la stratégie sont gagnantes pour le Joueur 0.
Cas particulier stratégie de position

$$f_0 : V_0 \longrightarrow V$$

Jeu déterminé.
Algorithmes connus pour les graphes finis.

Applications

Systèmes réactifs :

$$\begin{aligned} \text{Contrôleur} &= \text{Joueur } 0 \\ \text{Environnement} &= \text{Joueur } 1 \end{aligned}$$

Vérification, synthèse de contrôleurs

$$\left.\begin{array}{l} \text{Système fini } S \\ \text{formule } \phi \text{ du } \mu\text{-calcul} \end{array}\right\} \longmapsto \text{jeu} \qquad \text{(pol.)}$$

$$S \models \phi \qquad \Leftrightarrow \left\{ \begin{array}{l} \text{Joueur 0 a une} \\ \text{stratégie gagnante} \end{array}\right.$$

[EJ 91], [EJS 93]

déterminer le gagnant *d'un jeu* de parité est NP∩co-NP [Jur 00].

Jeu à pile

Automate à pile (PDS) $\mathscr{P} = (P, \Gamma, \Delta)$

$P = P_0 \uplus P_1$ ensemble fini, “partagé”, d’états de contrôle,
Γ alphabet de pile, fini,
$\Delta \subseteq P \times \Gamma \times P \times \Gamma^*$ ensemble fini de règles de transition.

Graphe de transitions (V, E)

$P\Gamma^* = V$ ensemble de sommets (configurations)
$E = \{(p\gamma v, q\nu v) \mid (p, \gamma, q, \nu) \in \Delta,\ v \in \Gamma^*\}$
Joueur 0 : $V_0 = P_0\Gamma^*$, Joueur 1 : $V_1 = P_1\Gamma^*$

condition de gain :

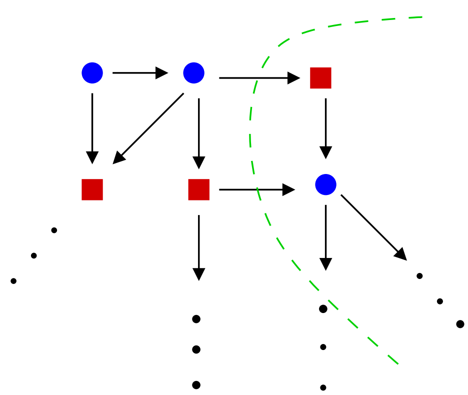
accessibilité, Büchi, Parité (définies par l’état de contrôle), Σ_3.

Jeu d'accessibilité

atteindre un ensemble R de “bonnes” configurations

Attracteur :
Région gagnante = $\mathit{Attr}(R) = \bigcup_{i\in\mathbb{N}} \mathit{Attr}^i$
$\mathit{Attr}^0 = R$,

$$\mathit{Attr}^{i+1} = \mathit{Attr}^i \cup \left\{v \in V_0 \mid \exists\ u,\ v \hookrightarrow u,\ u \in \mathit{Attr}^i\right\} \cup \left\{v \in V_1 \mid \forall\ u,\ v \hookrightarrow u \Rightarrow u \in \mathit{Attr}^i\right\}.$$



Approche symbolique : ensemble R rationnel reconnu par un automate fini $\mathscr{A}$. (alphabet $P \uplus \Gamma$)

Algorithme : ajouter successivement à $\mathscr{A}$ des transitions pour construire $\mathscr{A}_{Att}$, automate alternant qui reconnaît $\mathit{Attr}(R)$, rationnel.
[BH 1970] [BEM 97] [EHRS 00] [icalp 02]

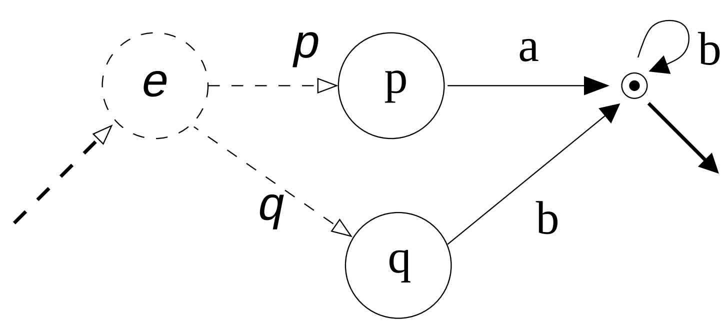
ensemble rationnel de configurations

alphabet $P \uplus \Gamma$, automate fini...
en particulier : $\mathscr{P}$-automate

$$\mathscr{A} = (Q, \Gamma, \longrightarrow, P, F), \text{ tel que}$$

Q ensemble fini d'états,
$\longrightarrow \ \subseteq \ Q \times \Gamma \times Q$ ensemble de transitions,
$P \subseteq Q$ ensemble des états initiaux : les états de contrôle de $\mathscr{P}$,
$F \subseteq Q$ ensemble des états finaux.

Pas de transitions vers P.
$R = \{pw \in P\Gamma^* \mid p \xrightarrow{w}_* f \in F\} \ \in \mathsf{Rat}(P\Gamma^*)$

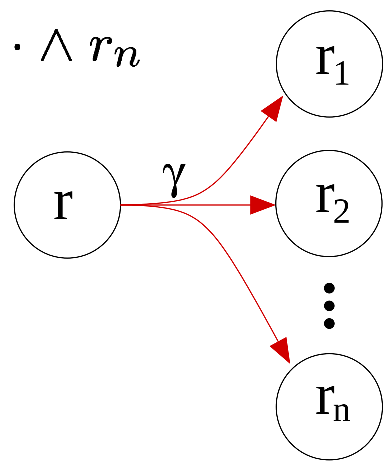


$pab^* \cup qb^+$

Automate alternant

$\mathscr{A}$ est vu comme un automate alternant (EA)

transitions-ET :
$r \xrightarrow{\gamma} \{r_1, \cdots, r_n\}$, avec $r, r_1, \cdots, r_n \in Q$,
i.e. conjonction : $r \xrightarrow{\gamma} r_1 \wedge \cdots \wedge r_n$



$$\longrightarrow_* \ \subseteq \ Q \times \Gamma^* \times 2^Q,$$

$\forall r \in Q,\ r \xrightarrow{\epsilon}_* \{r\}$,

$r \xrightarrow{\gamma} r' \quad \Rightarrow \quad r \xrightarrow{\gamma}_* \{r'\}$

$$\left.\begin{array}{l} r \xrightarrow{\gamma} \{r_1, \cdots, r_n\} \\ \forall i,\ r_i \xrightarrow{w}_* S_i \end{array}\right\} \Rightarrow \quad r \xrightarrow{\gamma w}_* \bigcup_i S_i.$$

$\mathscr{A}$ accepte le mot pw ssi il existe un chemin $p \xrightarrow{w}_* S$ avec $S \subseteq F$.

Ici $r \xrightarrow{\gamma} (r_1 \wedge r_2) \vee (r_3 \wedge r_4)$ représenté par deux transitions $r \xrightarrow{\gamma} \{r_1, r_2\}$ et $r \xrightarrow{\gamma} \{r_3, r_4\}$.

Algorithme

procédure de saturation

Répéter

- (Joueur 0) si $p \in P_0$, $p\gamma \hookrightarrow qw$ et
$q \xrightarrow{w}_* S$ dans l'automate actuel,
alors ajouter la nouvelle transition $p \xrightarrow{\gamma} S$.

- (Joueur 1) si $p \in P_1$,

$$\begin{cases} p\gamma & \hookrightarrow & q_1 w_1 \\ \vdots & & \vdots \\ p\gamma & \hookrightarrow & q_n w_n \end{cases}$$ sont *toutes* les transitions depuis $p\gamma$ et

$$\begin{cases} q_1 & \xrightarrow{w_1}_* & S_1 \\ \vdots & & \vdots \\ q_n & \xrightarrow{w_n}_* & S_n \end{cases}$$ dans l'automate actuel,

alors ajouter la nouvelle transition $p \xrightarrow{\gamma} \bigcup_i S_i$.

jusqu'à aucune nouvelle transition ne peut être ajoutée.

$\rightsquigarrow \mathscr{A}_{Att}$, $L(\mathscr{A}_{Att}) = Attr(R)$

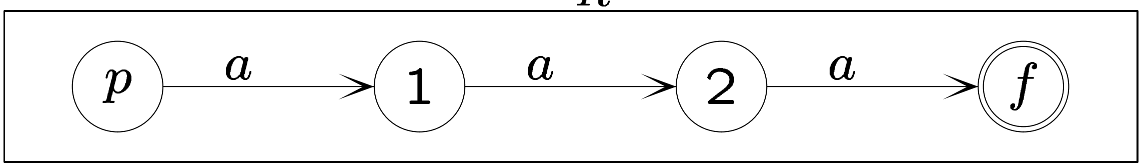
au plus $|\Gamma|.|P|.2^{|Q|}$ transitions, temps $|\Delta|\ 2^{\mathcal{O}(|Q|^2)}$

[BH 1970] $\left(2^{2^{|Q|}}\right)^{2^{|Q|}}$.

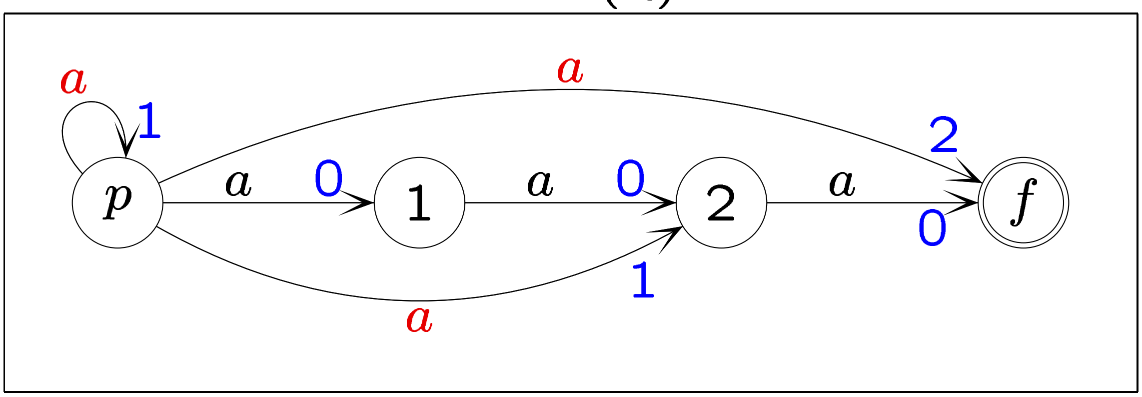
Exemple de saturation

$R = \{paaa\}$, $P = P_0 = \{p\}$, $P_1 = \emptyset$, $\Gamma = \{a\}$
$\Delta = \{\ (p, a, p, \varepsilon),\ (p, a, p, aa)\ \}$

$\mathscr{A}_R$



$\mathscr{A}_{Attr(R)}$



configuration paa acceptée par deux chemins différents, avec les coûts 3 et 1 :
Coût minimal = coût de paa = 1

La stratégie choisit la "configuration suivante" qui a le plus petit coût.

Stratégie (pour le Joueur 0)

À une configuration $p\gamma u \in L(\mathscr{A}_{Att})$ associer le coup suivant. Basée sur $\mathscr{A}_{Attr(R)}$.

idée : transition $p \xrightarrow{\gamma} S$ de $\mathscr{A}_{Attr(R)}$ associée a un unique $p\gamma \hookrightarrow qw$.
$\rightsquigarrow$ jouer vers $qwu \in Attr(R)$, MAIS, cela ne définit pas une stratégie gagnante : exemple

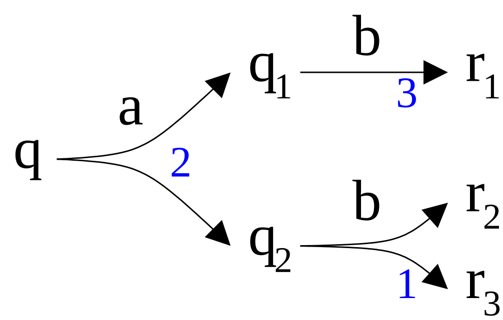
2 solutions
- définir des coûts, calculer la "distance" à R
$\rightsquigarrow$ stratégie positionnelle, optimale, mais le calcul du "coup suivant" est linéaire

- enregistrer dans la pile d'un transducteur à pile la description des chemins acceptants
$\rightsquigarrow$ stratégie à pile, "coup suivant" en temps constant.

Stratégie de position avec des coûts

But : depuis $pv \in Attr_0(R)$,
déterminer le $\min\{i \mid pv \in Attr_0^i\} = rang(pv)$.

coût d'une transition



coût d'un chemin : somme maximale le long d'une branche

$rang(pv)$ est le coût minimal d'un chemin acceptant.

Jouer vers $Attr_0^{i-1}$.
Stratégie optimale.

Stratégie à pile

Automate à pile avec entrée/sortie, qui "lit" les coup du Joueur 1 et produit les coup du Joueur 0.

$\mathscr{S} = (P, A, \Pi)$, $A = \Gamma \times \Sigma$, Σ un ensemble fini

$$\begin{array}{rcl} \Pi & \subseteq & (P_1 \times A \times \Delta_1) \times (P \times A^*) \\ & & \cup (P_0 \times A) \times (P \times A^* \times \Delta_0) \end{array}$$

configuration du jeu : $pw = p\gamma_0 \cdots \gamma_n$
et de la stratégie : $p(\gamma_0, \sigma_0) \cdots (\gamma_n, \sigma_n)$

exécution en temps constant

Ici $\sigma_0 \cdots \sigma_n$ code le chemin acceptant de $\mathscr{A}_{Attr(R)}$.

Jeux de Büchi

But du Joueur 0 : visiter R ∞-souvent.

Définition ensembliste de la région gagnante : avec $Attr_0^+(X)$: atteindre X en au moins un coup.

$Buchi^0(R) = V = P\Gamma^*$,
$Buchi^{i+1}(R) = Attr^+(Buchi^i(R) \cap R)$

$Buchi(R) = \bigcap_{i \in \mathbb{N}} Buchi^i(R)$.

Première idée : Construire un automate $\mathscr{B}^i$ qui reconnaît $Buchi_0^i(R)$, $i = 0, 1, 2, 3, \cdots$

Mais, pas de terminaison : suite strictement décroissante de langages.
Exm : Graphe $\cdots \hookrightarrow paaa \hookrightarrow paa \hookrightarrow pa \hookrightarrow p$
$Buchi^i(pa^*) = pa^i a^*$

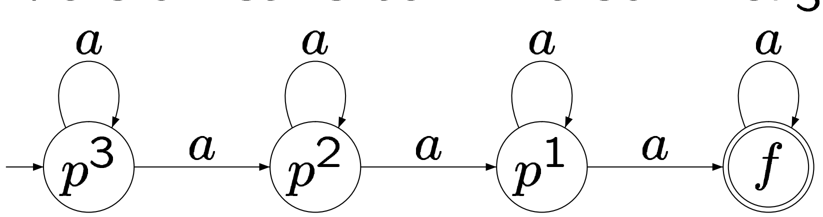
Ici le nombre d'état augmente. On a besoin d'une "projection" pour éliminer les états inutiles. $\rightsquigarrow$ Accélération

Deux même types de stratégies.

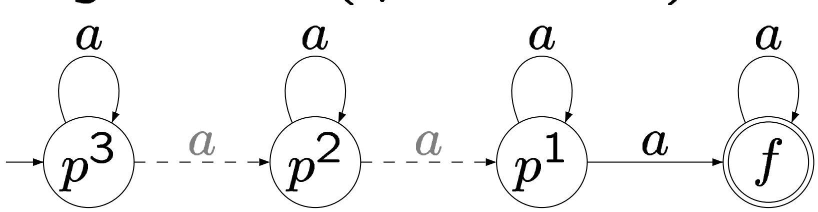
Assurer la convergence

On obtient $\mathscr{B}_{i+1}$ de $\mathscr{B}_i$ en ajoutant une nouvelle "génération" d'états et de transitions.

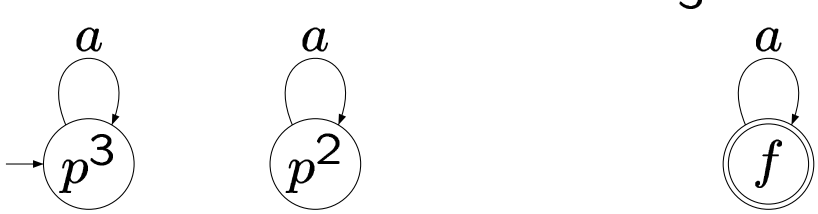
Version sans terminaison : $\mathscr{B}_3 =$



Algorithme (qui termine) :



Les "projections" éliminent les transitions discontinues. On obtient $\mathscr{B}'_3 =$



Propriété :

$$\bigcap_i L(\mathscr{B}_i) = \bigcap_i L(\mathscr{B}'_i)$$

Il existe i_0 t.q. $L(\mathscr{B}'_{i_0}) = L(\mathscr{B}'_i)$ pour $i \geqslant i_0$

Nouvelle condition (Σ_3)

Joueur 0 gagne la partie $(q_i, \mu_i)_{i \geq 0} \iff$
$\exists (q, \mu) \in Q \times \Gamma^* \; \forall j > 0 \; \exists i > j : \; (q_i, \mu_i) = (q, \mu)$

Ou, de façon équivalente
$\exists n > 0 \; \forall j > 0 \; \exists i > j \; |\mu_i| = n$

cf Explosion de pile

Formule $\Sigma_3 \Rightarrow$ ensemble Σ_3.

Thm : Étant donné un jeu à pile avec cette condition Σ_3, la région gagnante du Joueur 0 est rationnelle et peut être calculée. [csl 02]

Borne

Région gagnante $W_0 = \bigcup_{M>0} Buchi_0(P\Gamma^{\leqslant M})$ (degré borné)

Solution : "borne" sur M

$N = 1 + |\Gamma||Q| \max\{|\nu| - 1 \mid (p, \alpha, q, \nu) \in \Delta\}$

Prop : $\forall M \geq N$,
$Buchi_0(P\Gamma^{\leqslant M}) \bullet \Gamma^* = Buchi_0(P\Gamma^{\leqslant N}) \bullet \Gamma^*$

Conséquence :
$W_0 = Attr_0(Buchi_0(P\Gamma^{\leqslant N}) \bullet \Gamma^*)$.

Stratégie correspondante, définie pour toute la région gagnante.

Résultats antérieurs

jeux de parité avec un sommet initial fixé

[Wal 96] : *existence* de stratégie à pile [infinity 02]

[KV 00] : model checking du μ-calcul pour les graphes d'automates à piles et les graphes préfixe-reconnaissables [alig 02]

[MS 85] Logique monadique du second ordre : définition de la région gagnante. Mais ne peux pas définir la stratégie ?

Résultats nouveaux

description uniforme de l'ensemble gagnant et de la stratégie

jeux d'accessibilité, de Büchi et Σ_3

Extensions

Graphes préfixe-reconnaissables :
Alphabet Γ, langages rationnels U_i, V_i, W_i,
sommets $V = \Gamma^*$, $E \subseteq \Gamma^* \times \Gamma^*$, $E =$

$$\{uw \hookrightarrow vw \mid u \in U_i, v \in V_i, w \in W_i, 1 \leqslant i \leqslant N\}$$

Accessibilité, Büchi

Automates à pile de pile, jeux de parité : jeu-réduction [icalp 03]

Autres classes de graphes... indécidabilité.

Références

BEM 97 Ahmed Bouajjani, Javier Esparza, Oded Maler, *Reachability analysis of pushdown automata : Application to model-checking*, CONCUR '97, LNCS 1243, pp 135-150, 1997.

BH 70 J. Richard Büchi and William H. Hosken, *Canonical systems which produce periodic sets* . Mathematical Systems Theory, 4-1, pp 81-90, 1970, or in *The collected works of J.Richard Büchi*, Saunders Mac Lane, Dirk Siefkes, 1990.

icalp 02 Thierry Cachat, *Symbolic strategy synthesis for games on pushdown graphs*, ICALP'02, LNCS 2380, pp. 704-715, 2002.

infinity 02 Thierry Cachat, *Uniform solution of parity games on prefix-recognizable graphs*, INFINITY 2002, ENTCS 68(6), 2002.

alig 02 Thierry Cachat, *Two-way tree automata solving pushdown games*, in Automata, Logics, and Infinite Games, E. Grädel, W. Thomas and T. Wilke eds., ch. 17, pp. 303-317, LNCS 2500, 2002.

csl 02 Thierry Cachat, Jacques Duparc and Wolfgang Thomas, *Solving Pushdown Games with a $Sigma_3$ Winning Condition*, CSL'02, LNCS 2471, pp. 322-336, 2002.

icalp 03 Thierry Cachat, *Higher Order Pushdown Automata, the Caucal Hierarchy of Graphs and Parity Games*, accepted at ICALP'03.

EHRS 00 Javier Esparza, David Hansel, Peter Rossmanith and Stefan Schwoon, *Efficient algorithm for model checking pushdown systems*. Technische Universität München, 2000.

EJ 91 E. Allen Emerson and Charanjit S. Jutla *Tree automata, mu-calculus and determinacy*, FoCS '91, IEEE Computer Society Press, pp. 368–377, 1991.

EJS 93 E. Allen Emerson, Charanjit S. Jutla, and A. Prasad Sistla, *On model-checking for fragments of μ-calculus*, CAV '93, LNCS 697, pp. 385–396, 1993.

Jur 00 Marcin Jurdziński, *Small progress measures for solving parity games*, STACS 2000, LNCS 1770, pp. 290–301, 2000.

KV 00 Onar Kupferman und Moshe Y.Vardi, *An automata-theoretic approach to reasoning about infinite-sate systems*, CAV'00, LNCS 1855, 2000.

MS 85 David E. Muller and Paul E. Schupp, *The theory of ends, pushdown automata, and second-order logic*, TCS 37, 1985

Var 98 Moshe Y. Vardi, *Reasoning about the past with two-way automata*, ICALP'98, vol. 1443 of LNCS, 1998

Wal 96 Igor Walukiewicz, *Pushdown processes : games and model checking*, CAV'96, LNCS 1102, pp. 62–74, 1996. Full version in Information and Computation 164, pp. 234-263, 2001.