Décider si un automate reconnait un langage régulier.

Arthur MILCHIOR

Liafa
Université Paris Diderot

22 mai 2013

Sommaire

Langage

- $r \in \mathbb{N}$ l'arité
- $p \geq 2$ la base

Definition (Langage)

- Alphabet $[0, p-1]^r$ de taille p^r,
- petitboutiste, $0, 1, 01, 11, 001, 101, 011, 111,$
- interprété par $\bar{\cdot} : [0, p-1]^{r*} \to \mathbb{N}^r$
  - $\overline{\epsilon} = (0, \cdots, 0)$
  - $\overline{sw} = s + p\overline{w}$ pour $s \in [0, p-1]$.

Automate

- $A = (Q, [0, p-1]^r, \delta, q_0, F)$
- $\delta(q, \overline{0}^r) \in F \Leftrightarrow q \in F$.
- $|A|$ l'ensemble des mots acceptés,
- $|\overline{A}| = \{\overline{w} \mid w \in |A|\}$,
- tous les automates sont minimaux.

Definition (Arithmétique de Buchi de base p)

- $\mathrm{FO}[+, V_P]$
- $V_P(p^c d) = d$ pour $p \not| d$.

Theorem

$|\overline{A}|$ appartient à l'arithmétic de Büchi de base p.

Automate et FO[+]

Theorem ([Ler06])

Savoir si $|\overline{A}| \in \mathrm{FO}[+]$ *est décidable en temps polynomial si le langage est* $[0, p-1]$.

Si l'automate est sur $[0, p-1]^r$, sa taille sera multiplié par p^r, la taille de l'alphabet

Ensemble régulier

Definition

$R \subseteq \mathbb{N}^r$ est régulier s'il est définissable dans $\mathrm{FO}[<, \mathrm{mod}]$

Theorem ([Str94])

Si R est régulier, son écriture unaire est reconnu par un automate fini.

Ensemble régulier

Theorem ([Mil13])

Les deux énoncés sont équivalents

1. $R \subseteq \mathbb{N}^r$ *définissable dans* $\mathrm{FO}[<, \mathrm{mod}\, k]$.
2. 1. *il y a* $l \in \mathbb{N}$ *tel que pour tout* $\overline{x} \in \mathbb{N}^r$ *avec pour tout* $x_i > l$, *alors soit* $y_i = x_i + k$, $\overline{y} \in F \Leftrightarrow \overline{x} \in F$,
   2. *toutes les section est diagonales sont définissables dans* $\mathrm{FO}[<, \mathrm{mod}\ k]$,

Theorem

$\phi \in \mathrm{FO}[<]$ *est équivalent à* $\psi \in \Pi_0[\mathbb{N}, +\mathbb{N}, <]$.

Principal résultat

Theorem

On peut décider si un automate minimal reconnait un langage de FO[$<$] *en* $O(p^r \frac{r!}{\log(\frac{3}{2})^{n+1}}|Q|)$

- Polynomial pour un alphabet fixé,
- Degré fixé pour tout alphabet.

Theorem

On peut décider si un automate minimal reconnait un langage régulier en $O(p^r \frac{r!}{\log(\frac{3}{2})^{n+1}}|Q|^{2^{r+1}r+1})$

- Degré variable, donc hors de la complexité paramétré.

Liste des ensembles traités

- $\Pi_0[]$,
- $\Pi_0[0]$,
- $\Pi_0[+1]$,
- $\Pi_0[+1, 0]$,
- $\Pi_0[<]$,
- $\Pi_0[<, 0]$,
- $\Pi_0[<, +1]$,
- $\Pi_0[<, +1, 0]$,
- $\mathrm{FO}[<]$,
- $\mathrm{FO}[(\mathrm{mod}\, k)_{k \wedge p=1}]$ sans égalité entre les variables
- $\mathrm{FO}[\mathrm{mod}, \mathbb{N}]$ sans égalité entre les variables
- $\mathrm{FO}[<, \mathrm{mod}]$

Pourquoi 0,+1

Soit $x \in \mathbb{N}^r$, $x/l = \frac{x}{p^l}$.

Lemma

Let $c \in \mathbb{N}$ and let $l = \lceil \log_p(c+1) \rceil$,

1. *$x = c$ imply that $x/l = 0$,*
2. *$x = y + c$ imply that $x/l = y/l$ ou $x/l = y/l + 1$,*
3. *$x > y + c$ implies that $x/l \geq y/l$ and*
4. *$x < y + c$ implies that $x/l \leq y/l + 1$.*

Sommaire

Méthode générale

Soit $\mathcal{L}$ un ensemble de langage

Proposition

- *I ensemble dénombrables de valeur des paramètre, calculable en temps $O(g(|Q|))$,*
- *pour tout $i \in I$, α_i un alphabet de taille a_i,*
- *$\mathcal{L}_i \subseteq \mathcal{L} \cap \alpha_i^*$, $\mathcal{L} = \bigcup_{i \in I} \mathcal{L}_i$,*
- *J_i un ensemble fini de cardinal c_i,*
- *$(P_{i,j})_{j \in J_i}$ une partition de α_i^*,*
- *$L \in \mathcal{L}_i$ implique que $L = \bigcup_{j \in F_L} (P_{i,j})$ pour $F_L \subseteq J_i$,*
- *$A_{i,j} = ([1, c_i], \alpha_i, \delta_i, 0, j)$ accept $P_{i,j}$, calculable en temps $O(f(i))$,*

On peut décider si $|\overline{A}| \in \mathcal{L}$ en $O((g(|Q|) + f(i) + a_i c_i + |Q|^2)$.

Application $\Pi_0[]$

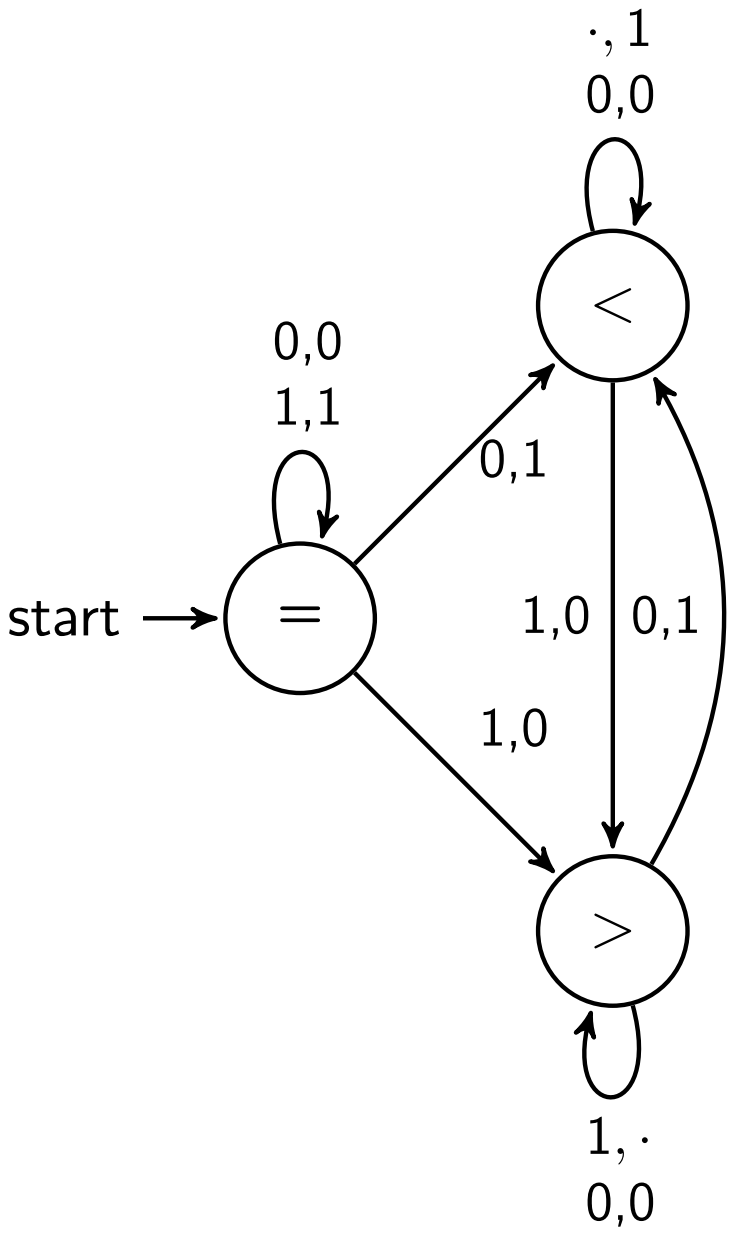
- $I = \mathbb{N}^2$, les paramètres sont r et p,
- J_i ensemble de partitions $c_i = \frac{1}{r+1}\binom{2n}{n}$, nombre de Bell,
- $P_{i,j}$ les nombres dont l'ordre respectent la partition,
- si $L \in \Pi_0[]$ et que $x, y \in \mathbb{N}^r$ respectent la même partition, $x \in L \Leftrightarrow y \in L$,
- A_i calculable en $O(p^r \frac{1}{r+1}\binom{2n}{n})$.

Décidable en temps $O(p^r \frac{1}{r+1}\binom{2n}{n} + |Q|^2)$.

Application $\Pi_0[<]$

- $I = \mathbb{N}^2$, les paramètres sont r et p,
- J_i ensemble de partitions totalement ordonnées, $c_i \approx \frac{r!}{2\log(2)^{n+1}}$, nombre de Fubini,
- $P_{i,j}$ les nombres dont l'ordre respectent la partition ordonnée,
- si $L \in \Pi_0[<]$ et que $x, y \in \mathbb{N}^r$ respectent la même partition ordonnée, $x \in L \Leftrightarrow y \in L$,
- A_i calculable en $O(p^r \frac{r!}{\log(2)^{n+1}})$.

Décidable en temps $O(p^r \frac{r!}{\log(2)^{n+1}})$.


·, 1
0,0
<
0,0
1,1
0,1
start
=
1,0
0,1
1,0
>
1, ·
0,0

Application $\Pi_0[<, 0]$

- $I = \mathbb{N}^2$, les paramètres sont r et p,
- J_i ensemble de partitions totalement ordonnées, plus un bit d'information pour 0, $c_i \approx 2\frac{r!}{2\log(2)^{n+1}}$,
- $P_{i,j}$ les nombres dont l'ordre respectent la partition ordonnée et le plus petit est 0 si le bit est 0,
- si $L \in \Pi_0[<, 0]$ et que $x, y \in \mathbb{N}^r$ respectent la même partition ordonnée et ont les mêmes 0, $x \in L \Leftrightarrow y \in L$,
- A_i calculable en $O(p^r \frac{r!}{\log(2)^{n+1}})$.

Décidable en temps $O(p^r \frac{r!}{\log(2)^{n+1}})$.

Application $\Pi_0[<, 0, +1]$

- $I = \mathbb{N}^2$, les paramètres sont r et p,
- J_i ensemble de partitions totalement ordonnées, plus un bit par classe signifiant que deux classes sont successeur, un bit pour 0 et un pour $(p-1)^*$, $c_i \approx \frac{r!}{3\log(\frac{3}{2})^{n+1}}$,
- $P_{i,j}$ les nombres dont l'ordre respectent la partition ordonnée, les successeur, et le plus petit est 0 si le bit est 0, le plus grand de la forme $(p-1)^*$ si le deuxième bit est 1,
- si $L \in \Pi_0[<, 0, +1]$ et que $x, y \in \mathbb{N}^r$ respectent la même partition ordonnée et ont les mêmes 0, $x \in L \Leftrightarrow y \in L$,
- A_i calculable en $O(p^r \frac{r!}{\log(\frac{3}{2})^{n+1}})$.

Décidable en temps $O(p^r \frac{r!}{\log(\frac{3}{2})^{n+1}})$.

FO[<]

- $I = \mathbb{N}^3$, les paramètres sont r et p et l la longueur du plus long chemin sans cycle,
- J_i ensemble de partitions totalement ordonnées, plus la distance entre deux classes, la valeur des classes si inférieure à p^l et pareil pour la classe maximale, $c_i = O(p^{lr}\frac{r!}{2\log(\frac{3}{2})^{n+1}})$,
- $P_{i,j}$ les nombres dont l'ordre respectent la partition
- si $L \in \Pi_0[<, [0, p^l - 1], +[0, p^l - 1]]$ et que $x, y \in \mathbb{N}^r$ respectent la même partition, $x \in L \Leftrightarrow y \in L$,
- A_i calculable en $O(p^{(l+1)r}\frac{r!}{\log(\frac{3}{2})^{n+1}})$.

Décidable en temps
$O(p^{(l+1)r}\frac{r!}{\log(\frac{3}{2})^{n+1}} + |Q|^2) = O(p^{|Q|r}\frac{r!}{\log(\frac{3}{2})^{n+1}} + |Q|^2)$.
Plus polynomiale quand l'alphabet est fixé !

FO[<]

Lemma

Un automate reconnait un langage de FO[<] *si tout sous-automate reconnait un langage de* FO[<] *et s'il a un seul composant fortement connexe qui est une feuille.*

Theorem

On décide en temps $O(p^r \frac{r!}{\log(\frac{3}{2})^{n+1}}|Q|)$ *si un automate minimal accepte un langage de* FO[<].

Sommaire

Pascal automaton

Definition (Pascal automaton)

- $\overline{k} = k_1, \ldots, k_r$ premier avec p,
- $P \subseteq \Pi_{i=1}^{r}[0, k_i - 1]$,
- soit ψ_i avec $p^{\psi_i} \equiv 1 \mod k_i$
- donc $p^c = p^{c \bmod \psi_i} \mod k_i$.
- $\psi = \mathrm{lcm}\{i \in [1, r] \mid \psi_i\}$,
- donc $k_i^c = k_i^{c \bmod \psi} \mod p$.

Let $\mathcal{P}_{\overline{k}}^{P}$ be

$$\mathcal{P}_{\overline{k}}^{P} = (\Pi_{i=1}^{r}\mathbb{Z}/k_i\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z}, S_{p,r}, \delta, (\overline{0}^r, P \times [0, \psi - 1])) \quad (1)$$

with $\delta((\overline{x}, t), (\overline{s})) = (\overline{x} + p^t\overline{s}, t + 1)$.

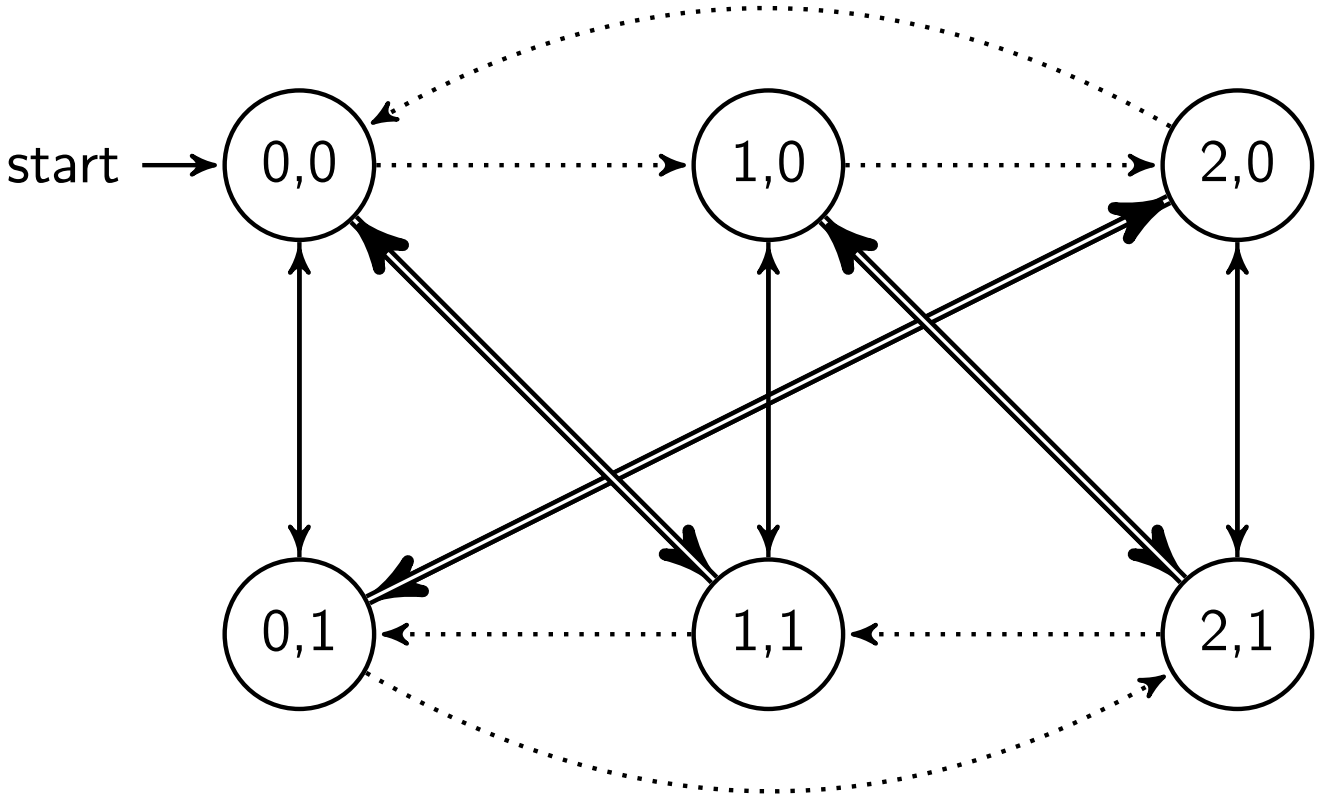


Figure: $\mathcal{P}_3, \Pi_0[\text{mod } 3], r = 1, p = 2, \psi = 2$

Strong arrows represent 0, double arrows represent 1, dotted arrows represent g_1

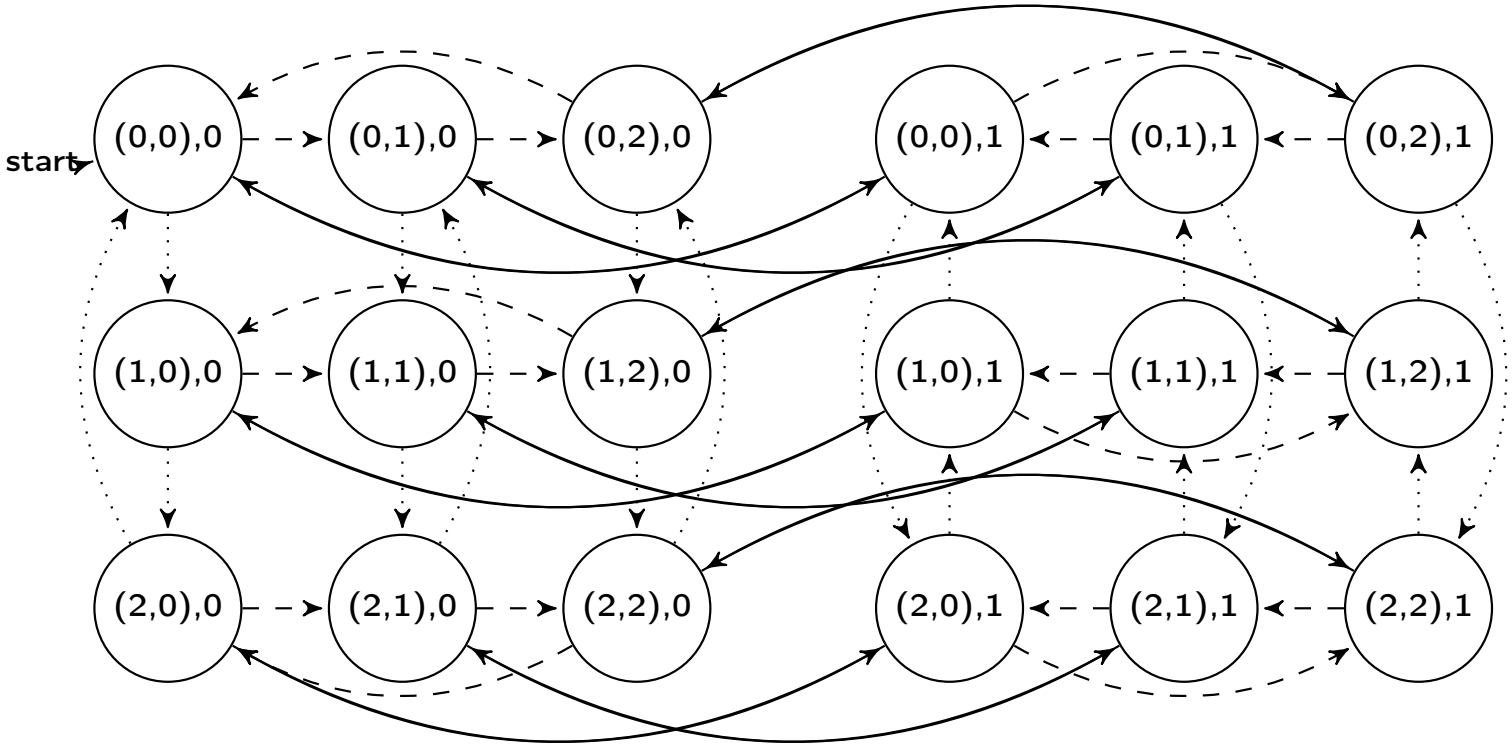


Figure: $\mathcal{P}_{(3,3)}, \Pi_0[\mathrm{mod}(3,3)], r = 2, p = 2, \psi = 2$

Dotted arrows represent g_1 and dashed arrows represent g_2

Theorem

Un automate reconnait un langage de FO[mod k] *si c'est le quotient d'un automate de Pascal.*

Corollary (du slide suivant)

On décide en temps $O(|Q|(\log(|Q|)p + r^2 + p^r r))$ *si un automate reconnait un ensemble périodique de périodicité première avec* p.

Quotient de Pascal

Proposition

Soit $A = (Q, S_{p,r}, \delta, q_0, F)$ un automate et M son monoïde de transition. C'est le quotient d'un automate de pascal ssi

1. *Le graphe de A est fortement connexe,*
2. *M est une groupe,*
3. *pour $i \in [1, r]$, soit $g_i = 1^i(\overline{0}^r)^{-1}$. Pour tout $1 \leq i < j \leq r$, g_i et g_j commutent,*
4. *pour tout $\overline{a} \in [0, p-1]^r$, $i \in [1, r]$ avec $a_i > 0$, soit $\overline{b} = \overline{a} - 1^i$, $a = g_i b$,*
5. *$g_i \overline{0}^r = \overline{0}^r g_i^p$,*
6. *les g_i-cycle ont une taille première avec p,*
7. *soit ψ comme dans la définition, $(\overline{0}^r)^\psi = \epsilon$ et*

Definition

Un ensemble R est ultimement périodique(UP) de retard (lag) $l \in \mathbb{N}$ et périodicité $k \in \mathbb{N}$ si pour tout $\overline{x} \in \mathbb{N}^r$ tel que $x_i \geq l$ est dans R si $\overline{x} + k^i \in R$.

Il est périodique (P) si le retard est 0.

Lemma

Un ensemble est ultimement périodique de périodicité k et retard l s'il est définissable dans $\mathrm{FO}[[0, l-1], \mathrm{mod}\, k]$.

Algorithme

Theorem

$|\overline{A}|$ est UP(P) s'il respecte la caractérisation de la slide suivante. On peut tester cette caractérisation en temps $O(|Q|(r^2 + p^r r))$.

Le cas $r = 1$ vient de [MS13].

UP-automaton I

Soit $A = (Q, S_{p,r}, \delta, q_0, F)$ un automate et M son monoide de transition, A est un UP-automata (resp. P-automata) si

1. tout état est accessible,
2. $q \in F \Leftrightarrow \delta(q, \overline{0}^r) \in F$,
3. les successeurs d'un état avec un cycle ont un cycle,
4. soit $\mathcal{C}$ l'ensemble des composants fortement connexe avec un cycle. $\overline{0}^r$ est une permutation sur chaque composante,
5. $\overline{0}^{r-1}$ est l'inverse de $\overline{0}^r$ sur $\mathcal{C}$, $g_i = 1^i(\overline{0}^r)^{-1}$. Pour tout $\overline{a} \in [0, p-1]^r$, $i \in [1, r]$ avec $a_i > 0$, let $\overline{b} = \overline{a} - 1^i$, $a = g_i b$,
6. g_i et g_j commutent,
7. $g_i \overline{0}^r = \overline{0}^r g_i^p$,

UP-automaton II

8. $w \in [0, p-1]^{r*}$ s'écrit comme $\overline{0}^{r\,t} \Pi_{i=1}^{r} g_i^{x_i}$ avec $x_i \in [1, k]$ et $t \in [0, \psi - 1]$,
9. si $C \in \mathcal{C}$ et $i \in [1, r]$, $\delta(C, g_i g_i) = \delta(C, g_i)$,
10. pour $i \in [1, r]$, soit $q, q' \in C$, et q' accessible depuis q, alors $\delta(q', g_i)$ est accessible depuis $\delta(q', g_i)$. δ est étendu sur $\mathcal{C} \times S_{p,r}^{*}$ vers $\mathcal{C}$,
11. si C est une feuille, $P(C) = (C, [0, p-1]^r, \delta, q_0', F)$ où q_0' est un état de C, alors $P(C)$ est le quotient d'un automate de Pascal,
12. (resp. toutes les composantes sont des feuilles).

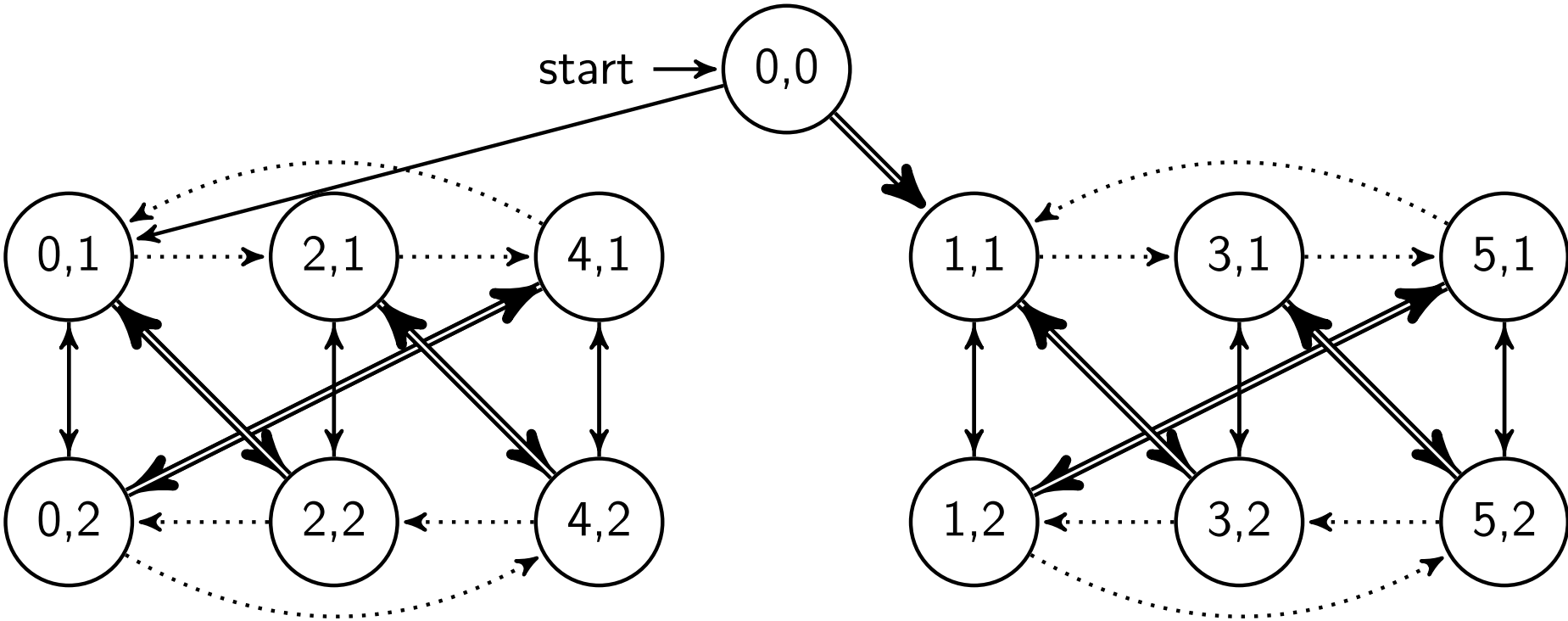


Figure: $\Pi_0[\mathrm{mod}\, 6], r = 1, p = 2$

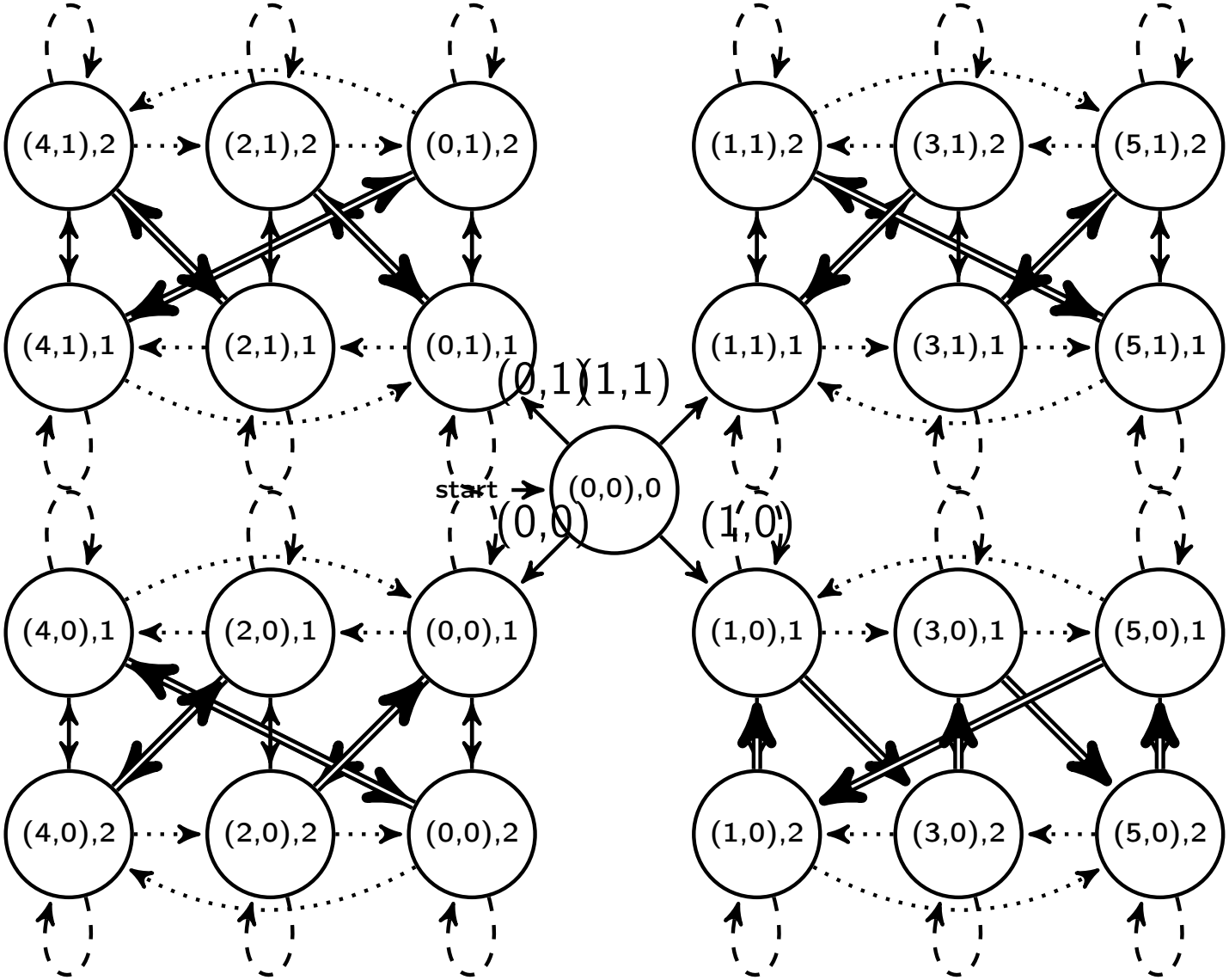


Figure: $\Pi_0[\mathrm{mod}(6,2)]$

Automate avec un cycle au départ, $\Pi_0[<, +1, 0, \mod k]$

k premier avec p.

Lemma

La périodicité est au plus $|Q|^{2^r}$, ψ aussi.

Calculable en $O(|Q|2^r)$.

Application $\Pi_0[<, 0, +1, \text{mod } k]$

- $I = \mathbb{N}^2$, les paramètres sont r et p et k
- J_i ensemble de partitions totalement ordonnées, une valeur mod k et une mode ψ par classe, plus un bit par classe signifiant que deux classes sont successeur, un bit pour 0 et un pour $(p-1)^*$, $c_i = O(\frac{r!}{\log(\frac{3}{2})^{n+1}}(k\psi)^r)$,
- A_i calculable en $O(p^r \frac{r!}{\log(\frac{3}{2})^{n+1}}(k\psi)^r)$.

Décidable en temps $O(p^r \frac{r!}{\log(\frac{3}{2})^{n+1}}(k\psi)^r) =$

$O(p^r \frac{r!}{\log(\frac{3}{2})^{n+1}}(|Q|^{2^r})^{2r}) = O(p^r \frac{r!}{\log(\frac{3}{2})^{n+1}}|Q|^{2^{r+1}r})$

- Polynomial pour un alphabet fixé
- Quitte la complexité paramétrée

FO[$<$, mod]

Lemma

Si un automate n'a pas de cycle sur son état initial, il reconnait un langage régulier ssi tous ses successeurs reconnaissent un langage régulier.

Theorem

On peut décider si un automate reconnait un langage régulier en $O(p^r r \frac{r!}{\log(\frac{3}{2})^{n+1}} |Q|^{2^{r+1} r+1})$

Sommaire

Commentaire de la méthode

Avantage :

- très flexible
- Peut fonctionner sur des langages non numérique

Inconvéniant

- génére un automate potentiellement gros, pour tester si un autre automate est son quotient,
- nécessite de calculer des paramètres,
- temps de calcul dépendant des paramètres

Pistes

- Associer à chaque état une unique partition.
- La calculer itérativement sur l'automate
- fonctionne déjà pour $\Pi_0[]$, $\Pi_0[0]$
- pour $\Pi_0[<]$ a nécessité des définitions complexes
- Question ouverte, comment étendre à $\Pi_0[+1]$?

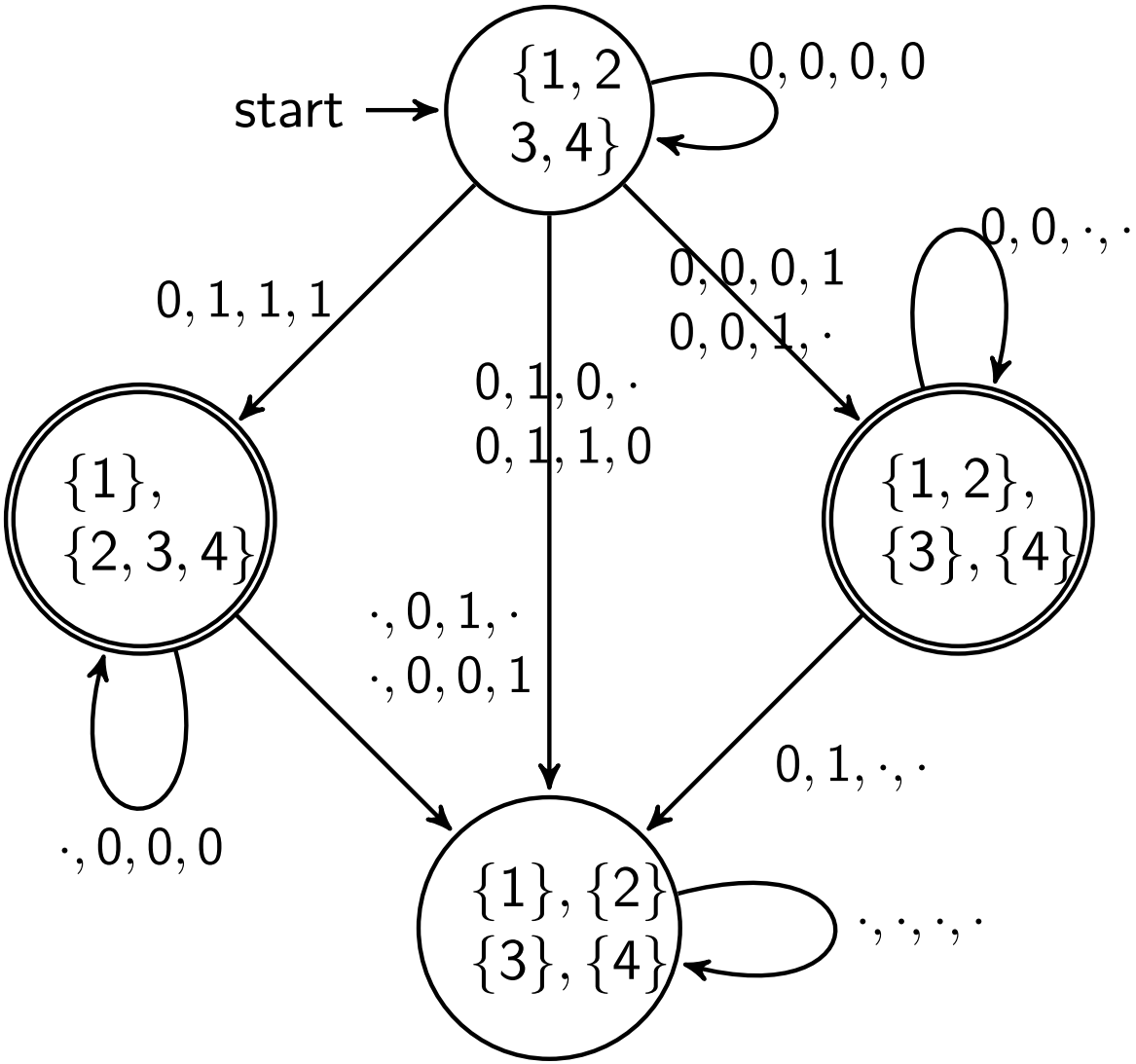


Figure: $x_1 \neq x_2 \leftrightarrow (x_2 = x_3 = x_4)$

Merci

Merci

Bibliography I


Jérôme Leroux.
Least significant digit first presburger automata.
CoRR, abs/cs/0612037, 2006.


Arthur Milchior.
Undecidability of satisfiability of expansions of fo[<] with a semilinear non regular predicate over words.
CIE, 2013.


Victor Marsault and Jacques Sakarovitch.
Ultimate periodicity of b-recognisable sets : a quasilinear procedure.
CoRR, abs/1301.2691, 2013.

Bibliography II


H. Straubing.
Finite Automata, Formal Logic, and Circuit Complexity.
1994.