Examen pré-doctoral oral

Autour de l'automate de Parikh

Une étude de l'automate de Parikh,
ses restrictions et ses généralisations

Michaël Cadilhac

14 octobre 2010

<http://michael.cadilhac.name/public/writings/10-predoc.pdf>

Introduction

D'un problème pratique. . .

À des fins pratiques, besoin de modèles de calculs :

- (Efficacement) décidables

Introduction

D'un problème pratique. . .

À des fins pratiques, besoin de modèles de calculs :

- (Efficacement) décidables : automates

Introduction

D'un problème pratique. . .

À des fins pratiques, besoin de modèles de calculs :

- (Efficacement) décidables : automates
- Expressifs

Introduction

D'un problème pratique. . .

À des fins pratiques, besoin de modèles de calculs :

- (Efficacement) décidables : automates
- Expressifs : exprimer au moins $a^n b^n c^n$

Introduction

D'un problème pratique. . .

À des fins pratiques, besoin de modèles de calculs :

- (Efficacement) décidables : automates
- Expressifs : exprimer au moins $a^n b^n c^n$
- Quelques autres problèmes décidables

Introduction

D'un problème pratique. . .

À des fins pratiques, besoin de modèles de calculs :

- (Efficacement) décidables : automates
- Expressifs : exprimer au moins $a^n b^n c^n$
- Quelques autres problèmes décidables

Solutions :

- Langages sensibles au contexte

Introduction

D'un problème pratique. . .

À des fins pratiques, besoin de modèles de calculs :

- (Efficacement) décidables : automates
- Expressifs : exprimer au moins $a^n b^n c^n$
- Quelques autres problèmes décidables

Solutions :

- ~~Langages sensibles au contexte~~ Vide indécidable

Introduction

D'un problème pratique. . .

À des fins pratiques, besoin de modèles de calculs :

- (Efficacement) décidables : automates
- Expressifs : exprimer au moins $a^n b^n c^n$
- Quelques autres problèmes décidables

Solutions :

- ~~Langages sensibles au contexte~~ Vide indécidable
- Langages *légèrement* sensibles au contexte

Introduction

D'un problème pratique. . .

À des fins pratiques, besoin de modèles de calculs :

- (Efficacement) décidables : automates
- Expressifs : exprimer au moins $a^n b^n c^n$
- Quelques autres problèmes décidables

Solutions :

- ~~Langages sensibles au contexte~~ Vide indécidable
- ~~Langages *légèrement* sensibles au contexte~~
  Adaptés aux langues, peu aux systèmes

Introduction

D'un problème pratique...

À des fins pratiques, besoin de modèles de calculs :

- (Efficacement) décidables : automates
- Expressifs : exprimer au moins $a^n b^n c^n$
- Quelques autres problèmes décidables

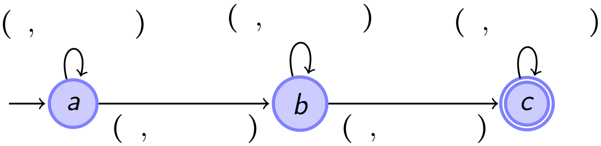
Solutions :

- ~~Langages sensibles au contexte~~ Vide indécidable
- ~~Langages *légèrement* sensibles au contexte~~
  Adaptés aux langues, peu aux systèmes
- Adjoindre des capacités à l'automate (pile, mémoire, arithmétique, ...)

Introduction

. . . à une solution . . .

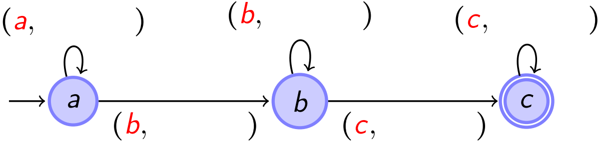
[Klaedtke and Rueß, 2003] proposent l'automate de Parikh :



Introduction

. . . à une solution . . .

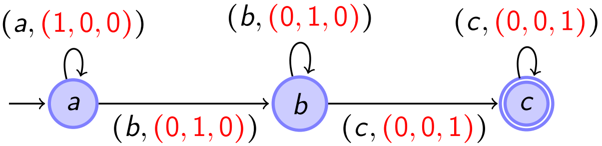
[Klaedtke and Rueß, 2003] proposent l'automate de Parikh :



Introduction

. . . à une solution . . .

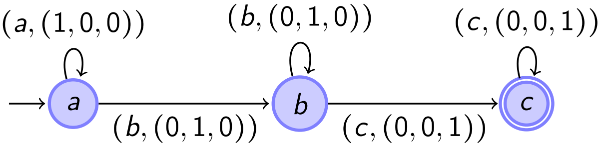
[Klaedtke and Rueß, 2003] proposent l'automate de Parikh :



Introduction

. . . à une solution . . .

[Klaedtke and Rueß, 2003] proposent l'automate de Parikh :



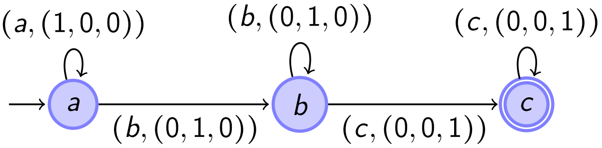
- Définissons la concaténation par :

$$(l, \overline{v}).(l', \overline{v'}) = (ll', \overline{v} + \overline{v'})$$

Introduction

. . . à une solution . . .

[Klaedtke and Rueß, 2003] proposent l'automate de Parikh :



- Définissons la concaténation par :

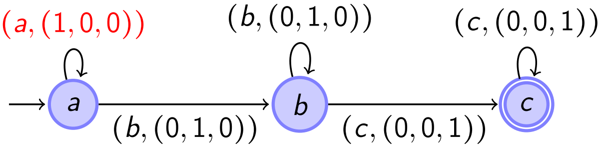
$$(l,\overline{v}).(l',\overline{v'}) = (ll',\overline{v}+\overline{v'})$$

Exemple :

Introduction

. . . à une solution . . .

[Klaedtke and Rueß, 2003] proposent l'automate de Parikh :



- Définissons la concaténation par :

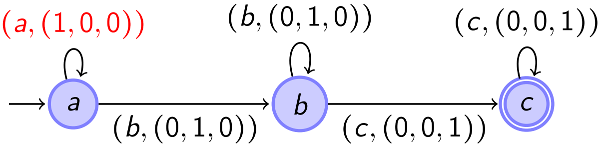
$$(l,\overline{v}).(l',\overline{v'}) = (ll',\overline{v}+\overline{v'})$$

Exemple : $(a,(1,0,0))$

Introduction

. . . à une solution . . .

[Klaedtke and Rueß, 2003] proposent l'automate de Parikh :



- Définissons la concaténation par :

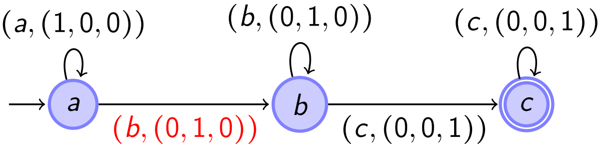
$$(l, \overline{v}).(l', \overline{v'}) = (ll', \overline{v} + \overline{v'})$$

Exemple : $(aa, (2, 0, 0))$

Introduction

. . . à une solution . . .

[Klaedtke and Rueß, 2003] proposent l'automate de Parikh :



- Définissons la concaténation par :

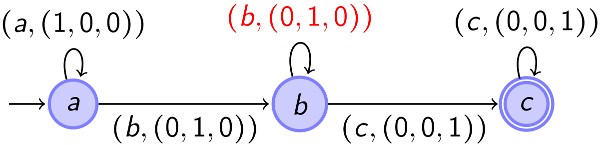
$$(l,\overline{v}).(l',\overline{v'}) = (ll',\overline{v}+\overline{v'})$$

Exemple : $(aab,(2,1,0))$

Introduction

. . . à une solution . . .

[Klaedtke and Rueß, 2003] proposent l'automate de Parikh :



- Définissons la concaténation par :

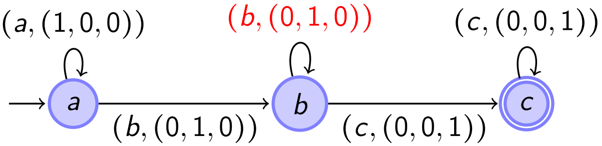
$$(l, \overline{v}).(l', \overline{v'}) = (ll', \overline{v} + \overline{v'})$$

Exemple : $(aabb, (2, 2, 0))$

Introduction

. . . à une solution . . .

[Klaedtke and Rueß, 2003] proposent l'automate de Parikh :



- Définissons la concaténation par :

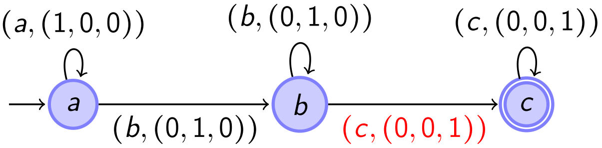
$$(l,\overline{v}).(l',\overline{v'}) = (ll',\overline{v}+\overline{v'})$$

Exemple : $(aabbb,(2,3,0))$

Introduction

. . . à une solution . . .

[Klaedtke and Rueß, 2003] proposent l'automate de Parikh :



- Définissons la concaténation par :

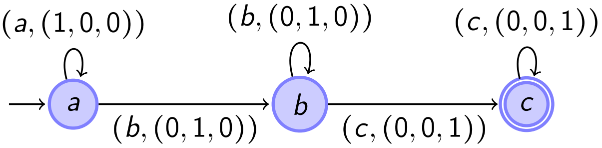
$$(l,\overline{v}).(l',\overline{v'}) = (ll',\overline{v}+\overline{v'})$$

Exemple : $(aabbbc,(2,3,1))$

Introduction

. . . à une solution . . .

[Klaedtke and Rueß, 2003] proposent l'automate de Parikh :



- Définissons la concaténation par :

$$(l,\overline{v}).(l',\overline{v'}) = (ll',\overline{v}+\overline{v'})$$

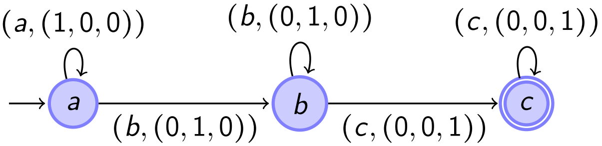
- Un mot est accepté s'il est de la forme :

$$(w,(|w|_a,|w|_b,|w|_c))$$

Introduction

. . . à une solution . . .

[Klaedtke and Rueß, 2003] proposent l'automate de Parikh :



- Définissons la concaténation par :

$$(l,\overline{v}).(l',\overline{v'}) = (ll',\overline{v}+\overline{v'})$$

- Un mot est accepté s'il est de la forme :

$$(w,(|w|_a,|w|_b,|w|_c))$$

- Contraindre la seconde partie : sous-ensemble de $\mathbb{N}^3$

Introduction

. . . à des problèmes théoriques

Nous proposons :

- Une étude de l'expressivité du modèle

Introduction

. . . à des problèmes théoriques

Nous proposons :

- Une étude de l'expressivité du modèle
- Des caractérisations (plus) naturelles

Introduction

. . . à des problèmes théoriques

Nous proposons :

- Une étude de l'expressivité du modèle
- Des caractérisations (plus) naturelles
- Des restrictions

Introduction

. . . à des problèmes théoriques

Nous proposons :

- Une étude de l'expressivité du modèle
- Des caractérisations (plus) naturelles
- Des restrictions
- Des généralisations

Introduction

. . . à des problèmes théoriques

Nous proposons :

- Une étude de l'expressivité du modèle
- Des caractérisations (plus) naturelles
- Des restrictions
- Des généralisations

On touche à :

- Théorie des langages

Introduction

...à des problèmes théoriques

Nous proposons :

- Une étude de l'expressivité du modèle
- Des caractérisations (plus) naturelles
- Des restrictions
- Des généralisations

On touche à :

- Théorie des langages
- Logique

Introduction

. . . à des problèmes théoriques

Nous proposons :

- Une étude de l'expressivité du modèle
- Des caractérisations (plus) naturelles
- Des restrictions
- Des généralisations

On touche à :

- Théorie des langages
- Logique
- Complexité

Introduction

. . . à des problèmes théoriques

Nous proposons :

- Une étude de l'expressivité du modèle
- Des caractérisations (plus) naturelles
- Des restrictions
- Des généralisations

On touche à :

- Théorie des langages
- Logique
- Complexité
- Algèbre

Plan

Plan

Ensembles semi-linéaires

Définition

Un ensemble $E \subseteq \mathbb{N}^d$ est *linéaire* s'il existe $\overline{c}$ et $\overline{p_1}, \ldots, \overline{p_n} \in \mathbb{N}^d$ tels que :

$$E = \{\overline{c} + \sum_{i=1}^{n} k_i \overline{p_i} \mid k_1, \ldots, k_n \in \mathbb{N}\}$$

Ensembles semi-linéaires

Définition

Un ensemble $E \subseteq \mathbb{N}^d$ est *linéaire* s'il existe $\overline{c}$ et $\overline{p_1}, \ldots, \overline{p_n} \in \mathbb{N}^d$ tels que :

$$E = \{\overline{c} + \sum_{i=1}^{n} k_i \overline{p_i} \mid k_1, \ldots, k_n \in \mathbb{N}\}$$

Ensemble semi-linéaire (SL) : union finie de linéaires

Ensembles semi-linéaires

Définition

Un ensemble $E \subseteq \mathbb{N}^d$ est *linéaire* s'il existe $\overline{c}$ et $\overline{p_1}, \ldots, \overline{p_n} \in \mathbb{N}^d$ tels que :

$$E = \{\overline{c} + \sum_{i=1}^{n} k_i \overline{p_i} \mid k_1, \ldots, k_n \in \mathbb{N}\}$$

Ensemble semi-linéaire (SL) : union finie de linéaires

Pour $w \in \Sigma^*$, on pose $\mathrm{Pkh}(w) = (|w|_a, |w|_b, \ldots) \in \mathbb{N}^{|\Sigma|}$

Ensembles semi-linéaires

Définition

Un ensemble $E \subseteq \mathbb{N}^d$ est *linéaire* s'il existe $\overline{c}$ et $\overline{p_1}, \ldots, \overline{p_n} \in \mathbb{N}^d$ tels que :

$$E = \{\overline{c} + \sum_{i=1}^{n} k_i \overline{p_i} \mid k_1, \ldots, k_n \in \mathbb{N}\}$$

Ensemble semi-linéaire (SL) : union finie de linéaires

Pour $w \in \Sigma^*$, on pose $\mathrm{Pkh}(w) = (|w|_a, |w|_b, \ldots) \in \mathbb{N}^{|\Sigma|}$

Théorème ([Parikh, 1966])

L *est HC* $\Rightarrow \mathrm{Pkh}(L) = \{\mathrm{Pkh}(w) \mid w \in L\}$ *effectivement SL*

Ensembles semi-linéaires

Définition

Un ensemble $E \subseteq \mathbb{N}^d$ est *linéaire* s'il existe $\overline{c}$ et $\overline{p_1}, \ldots, \overline{p_n} \in \mathbb{N}^d$ tels que :

$$E = \{\overline{c} + \sum_{i=1}^{n} k_i \overline{p_i} \mid k_1, \ldots, k_n \in \mathbb{N}\}$$

Ensemble semi-linéaire (SL) : union finie de linéaires

Pour $w \in \Sigma^*$, on pose $\mathrm{Pkh}(w) = (|w|_a, |w|_b, \ldots) \in \mathbb{N}^{|\Sigma|}$

Théorème ([Parikh, 1966])

L est HC $\Rightarrow \mathrm{Pkh}(L) = \{\mathrm{Pkh}(w) \mid w \in L\}$ *effectivement SL*

- Pour tout E SL, il existe L rationnel, $\mathrm{Pkh}(L) = E$

Ensembles semi-linéaires

Définition

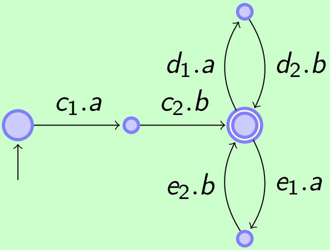
Un ensemble $E \subseteq \mathbb{N}^d$ est *linéaire* s'il existe $\overline{c}$ et $\overline{p_1}, \ldots,$

Ensem

Pour *w*

Exemple

$E = \{(c_1, c_2) + k_1(d_1, d_2) + k_2(e_1, e_2) \mid k_i \in \mathbb{N}\}$:



Théorè

L est ... *SL*

- Po

Ensembles semi-linéaires

Définition

Un ensemble $E \subseteq \mathbb{N}^d$ est *linéaire* s'il existe $\overline{c}$ et $\overline{p_1}, \ldots, \overline{p_n} \in \mathbb{N}^d$ tels que :

$$E = \{\overline{c} + \sum_{i=1}^{n} k_i \overline{p_i} \mid k_1, \ldots, k_n \in \mathbb{N}\}$$

Ensemble semi-linéaire (SL) : union finie de linéaires

Pour $w \in \Sigma^*$, on pose $\mathrm{Pkh}(w) = (|w|_a, |w|_b, \ldots) \in \mathbb{N}^{|\Sigma|}$

Théorème ([Parikh, 1966])

L est HC $\Rightarrow \mathrm{Pkh}(L) = \{\mathrm{Pkh}(w) \mid w \in L\}$ *effectivement SL*

- Pour tout E SL, il existe L rationnel, $\mathrm{Pkh}(L) = E$

Ensembles semi-linéaires

Définition

Un ensemble $E \subseteq \mathbb{N}^d$ est *linéaire* s'il existe $\overline{c}$ et $\overline{p_1}, \ldots, \overline{p_n} \in \mathbb{N}^d$ tels que :

$$E = \{\overline{c} + \sum_{i=1}^{n} k_i \overline{p_i} \mid k_1, \ldots, k_n \in \mathbb{N}\}$$

Ensemble semi-linéaire (SL) : union finie de linéaires

Pour $w \in \Sigma^*$, on pose $\mathrm{Pkh}(w) = (|w|_a, |w|_b, \ldots) \in \mathbb{N}^{|\Sigma|}$

Théorème ([Parikh, 1966])

L est HC $\Rightarrow \mathrm{Pkh}(L) = \{\mathrm{Pkh}(w) \mid w \in L\}$ *effectivement SL*

- Pour tout E SL, il existe L rationnel, $\mathrm{Pkh}(L) = E$
- $L \subseteq \{a, b\}^*$ indécidable $\rightarrow \mathrm{Pkh}(a\{a, b\}^* \cup bLa)$ SL...

Pourquoi les ensembles SL sont pratiques

- Passage effectif → décidabilité du vide, du fini

Pourquoi les ensembles SL sont pratiques

- Passage effectif → décidabilité du vide, du fini

Théorème ([Ginsburg and Spanier, 1966])

$E \subseteq \mathbb{N}^d$ est semi-linéaire ssi il existe φ de $\mathrm{FO}[+,<]$ à d variables libres avec :

$$(x_1, \ldots, x_d) \in E \leftrightarrow \mathbb{N} \vDash \varphi(x_1, \ldots, x_d)$$

Pourquoi les ensembles SL sont pratiques

- Passage effectif → décidabilité du vide, du fini

$\varphi \equiv x_1 \leq x_2 \wedge (\exists y)[x_1 = y + y]$

Théorème ([Ginsburg and Spanier, 1966])

$E \subseteq \mathbb{N}^d$ est semi-linéaire ssi il existe φ de $\mathrm{FO}[+,<]$ à d variables libres avec :

$$(x_1, \ldots, x_d) \in E \leftrightarrow \mathbb{N} \models \varphi(x_1, \ldots, x_d)$$

Pourquoi les ensembles SL sont pratiques

- Passage effectif → décidabilité du vide, du fini

$E = \{(0,0) + k_1(2,2) + k_2(0,1)\}$ $\qquad$ $\varphi \equiv x_1 \leq x_2 \wedge (\exists y)[x_1 = y + y]$

Théorème ([Ginsburg and Spanier, 1966])

$E \subseteq \mathbb{N}^d$ est semi-linéaire ssi il existe φ de FO$[+,<]$ *à d variables libres avec :*

$$(x_1, \ldots, x_d) \in E \leftrightarrow \mathbb{N} \models \varphi(x_1, \ldots, x_d)$$

Pourquoi les ensembles SL sont pratiques

- Passage effectif → décidabilité du vide, du fini

Théorème ([Ginsburg and Spanier, 1966])

$E \subseteq \mathbb{N}^d$ est semi-linéaire ssi il existe φ de $\mathrm{FO}[+,<]$ à d variables libres avec :

$$(x_1, \ldots, x_d) \in E \leftrightarrow \mathbb{N} \vDash \varphi(x_1, \ldots, x_d)$$

Pourquoi les ensembles SL sont pratiques

- Passage effectif → décidabilité du vide, du fini

Théorème ([Ginsburg and Spanier, 1966])

$E \subseteq \mathbb{N}^d$ est semi-linéaire ssi il existe φ de $\mathrm{FO}[+,<]$ à d variables libres avec :

$$(x_1, \ldots, x_d) \in E \leftrightarrow \mathbb{N} \vDash \varphi(x_1, \ldots, x_d)$$

- Clôture par intersection, union, complémentation

Plan

Principe de base

- Idée première : contraindre de l'information supplémentaire « calculée » par l'automate

Principe de base

- Idée première : contraindre de l'information supplémentaire « calculée » par l'automate

Exemple

$L = a^* b^*$, et on contraint $\mathrm{Pkh}(\cdot)$:

$$\{a^n b^n \mid n \in \mathbb{N}\} = \{w \in L \mid (\exists n)[\mathrm{Pkh}(w) = (n, n)]\}$$

Principe de base

- Idée première : contraindre de l'information supplémentaire « calculée » par l'automate

Exemple

$L = a^* b^*$, et on contraint $\mathrm{Pkh}(\cdot)$:

$$\{a^n b^n \mid n \in \mathbb{N}\} = \{w \in L \mid (\exists n)[\mathrm{Pkh}(w) = (n, n)]\}$$

- Insuffisant : $a^m b^m a^n b^n \sim a^m b^n a^n b^m$

Principe de base

- Idée première : contraindre de l'information supplémentaire « calculée » par l'automate

Exemple

$L = a^* b^*$, et on contraint $\mathrm{Pkh}(\cdot)$:

$$\{a^n b^n \mid n \in \mathbb{N}\} = \{w \in L \mid (\exists n)[\mathrm{Pkh}(w) = (n, n)]\}$$

- Insuffisant : $a^m b^m a^n b^n \sim a^m b^n a^n b^m$

  - Permettre au modèle de calculer plus d'information

Principe formel

- Σ un alphabet, $D \subset_{\text{fin}} \mathbb{N}^d$: $T = (\Sigma \times D)$ alphabet

Principe formel

- Σ un alphabet, $D \subset_{\text{fin}} \mathbb{N}^d$: $T = (\Sigma \times D)$ alphabet
- Pour $w = (a_1, \overline{d_1}) \dots (a_n, \overline{d_n}) \in T^*$ on pose :
  $\text{Mot}(w) = a_1 \dots a_n \qquad \text{Val}(w) = \overline{d_1} + \dots + \overline{d_n}$

Principe formel

- Σ un alphabet, $D \subset_{\text{fin}} \mathbb{N}^d$: $T = (\Sigma \times D)$ alphabet
- Pour $w = (a_1, \overline{d_1}) \ldots (a_n, \overline{d_n}) \in T^*$ on pose :
$$\text{Mot}(w) = a_1 \ldots a_n \qquad \text{Val}(w) = \overline{d_1} + \ldots + \overline{d_n}$$

- $L \subseteq T^*$ et $C \subseteq \mathbb{N}^d$ semi-linéaire :
$$L{\upharpoonright}_C = \{\text{Mot}(w) \mid w \in L \wedge \text{Val}(w) \in C\} \subseteq \Sigma^*$$

Principe formel

- Σ un alphabet, $D \subset_{\text{fin}} \mathbb{N}^d$: $T = (\Sigma \times D)$ alphabet
- Pour $w = (a_1, \overline{d_1}) \dots (a_n, \overline{d_n}) \in T^*$ on pose :
  $$\mathrm{Mot}(w) = a_1 \dots a_n \qquad \mathrm{Val}(w) = \overline{d_1} + \dots + \overline{d_n}$$

- $L \subseteq T^*$ et $C \subseteq \mathbb{N}^d$ semi-linéaire :
  $$L\!\upharpoonright_C = \{\mathrm{Mot}(w) \mid w \in L \wedge \mathrm{Val}(w) \in C\} \subseteq \Sigma^*$$

▸▸ $L \in \mathrm{REG}, L \in \mathrm{HC}, \dots$

Principe formel

- Σ un alphabet, $D \subset_{\text{fin}} \mathbb{N}^d$: $T = (\Sigma \times D)$ alphabet
- Pour $w = (a_1, \overline{d_1}) \ldots (a_n, \overline{d_n}) \in T^*$ on pose :
  $$\text{Mot}(w) = a_1 \ldots a_n \qquad \text{Val}(w) = \overline{d_1} + \ldots + \overline{d_n}$$
- $L \subseteq T^*$ et $C \subseteq \mathbb{N}^d$ semi-linéaire :
  $$L{\upharpoonright}_C = \{\text{Mot}(w) \mid w \in L \wedge \text{Val}(w) \in C\} \subseteq \Sigma^*$$

- $L \in \text{REG}, L \in \text{HC}, \ldots$

Exemple ($\mathbb{1}_{2/4} = (0, 1, 0, 0)$)

$L = (a, \mathbb{1}_{1/4})^*(b, \mathbb{1}_{2/4})^*(a, \mathbb{1}_{3/4})^*(b, \mathbb{1}_{4/4})^*$,
$C = \{k_1 \times (1, 1, 0, 0) + k_2 \times (0, 0, 1, 1) \mid k_i \in \mathbb{N}\}$:

$$L{\upharpoonright}_C =$$

Principe formel

- Σ un alphabet, $D \subset_{\text{fin}} \mathbb{N}^d$: $T = (\Sigma \times D)$ alphabet
- Pour $w = (a_1, \overline{d_1}) \ldots (a_n, \overline{d_n}) \in T^*$ on pose :
  $$\mathrm{Mot}(w) = a_1 \ldots a_n \qquad \mathrm{Val}(w) = \overline{d_1} + \ldots + \overline{d_n}$$
- $L \subseteq T^*$ et $C \subseteq \mathbb{N}^d$ semi-linéaire :
  $$L{\restriction_C} = \{\mathrm{Mot}(w) \mid w \in L \wedge \mathrm{Val}(w) \in C\} \subseteq \Sigma^*$$

  - $L \in \mathrm{REG}, L \in \mathrm{HC}, \ldots$

Exemple ($\mathbb{1}_{2/4} = (0,1,0,0)$)

$L = (a, \mathbb{1}_{1/4})^*(b, \mathbb{1}_{2/4})^*(a, \mathbb{1}_{3/4})^*(b, \mathbb{1}_{4/4})^*$,
$C = \{k_1 \times (1,1,0,0) + k_2 \times (0,0,1,1) \mid k_i \in \mathbb{N}\}$:

$$L{\restriction_C} = \{a^m b^m a^n b^n \mid m, n \in \mathbb{N}\}$$

Automate de Parikh

Automate de Parikh (AP) de dimension d :

- Automate A sur $\Sigma \times D$ où $D \subset_{\text{fin}} \mathbb{N}^d$
- Ensemble de contraintes C semi-linéaires $\subseteq \mathbb{N}^d$

Automate de Parikh

Automate de Parikh (AP) de dimension d :

- Automate A sur $\Sigma \times D$ où $D \subset_{\text{fin}} \mathbb{N}^d$
- Ensemble de contraintes C semi-linéaires $\subseteq \mathbb{N}^d$
- Son langage : $L(A)\restriction_C$

Automate de Parikh

Automate de Parikh (AP) de dimension d :

- Automate A sur $\Sigma \times D$ où $D \subset_{\text{fin}} \mathbb{N}^d$
- Ensemble de contraintes C semi-linéaires $\subseteq \mathbb{N}^d$
- Son langage : $L(A)\restriction_C$
- Déterministe si A sans les vecteurs est dét.

Automate de Parikh

Automate de Parikh (AP) de dimension d :

- Automate A sur $\Sigma \times D$ où $D \subset_{\text{fin}} \mathbb{N}^d$
- Ensemble de contraintes C semi-linéaires $\subseteq \mathbb{N}^d$
- Son langage : $L(A){\restriction}_C$
- Déterministe si A sans les vecteurs est dét.

De Parikh ?...

Automate de Parikh

Automate de Parikh (AP) de dimension d :

- Automate A sur $\Sigma \times D$ où $D \subset_{\text{fin}} \mathbb{N}^d$
- Ensemble de contraintes C semi-linéaires $\subseteq \mathbb{N}^d$
- Son langage : $L(A)\restriction_C$
- Déterministe si A sans les vecteurs est dét.

De Parikh ?...

Théorème

Tout langage L d'AP est tel que $\mathrm{Pkh}(L)$ *est SL*

Automate de Parikh

Automate de Parikh (AP) de dimension d :

- Automate A sur $\Sigma \times D$ où $D \subset_{\text{fin}} \mathbb{N}^d$
- Ensemble de contraintes C semi-linéaires $\subseteq \mathbb{N}^d$
- Son langage : $L(A)\restriction_C$
- Déterministe si A sans les vecteurs est dét.

De Parikh ?...

Théorème

Tout langage L d'AP est tel que $\text{Pkh}(L)$ *est SL*

- (Converse toujours aussi fausse : PAL)

Propriétés de l'automate de Parikh

- Modèle non déterministe plus puissant

Propriétés de l'automate de Parikh

- Modèle non déterministe plus puissant
- Bonnes propriétés de clôture :

	$\cup$	$\cap$	$\cdot$	$\bar{\cdot}$	$h(\cdot)$
Dét.	☻	☻	T ☹	☻	☹
Non dét.	☻	☻	☻	☹	☻

Propriétés de l'automate de Parikh

- Modèle non déterministe plus puissant
- Bonnes propriétés de clôture :

	$\cup$	$\cap$	$\cdot$	$\bar{\cdot}$	$h(\cdot)$
Dét.	☻	☻	T ☹	☻	☹
Non dét.	☻	☻	☻	☹	☻

- Décidabilité du vide, T du fini

Propriétés de l'automate de Parikh

- Modèle non déterministe plus puissant
- Bonnes propriétés de clôture :

	$\cup$	$\cap$	$\cdot$	$\bar{\cdot}$	$h(\cdot)$
Dét.	☻	☻	T ☹	☻	☹
Non dét.	☻	☻	☻	☹	☻

- Décidabilité du vide, T du fini
- Indécidabilité de l'univers (non-déterministe)

Une logique pour les AP

$\Sigma = \{a, b\}$. Pour $w = w_0 \ldots w_n \in \Sigma^*$, on définit :

$$S^\infty(w) = (\qquad \mathbb{N} \qquad , Q_a, Q_b), \quad Q_a = \{p \mid w_p = a\}$$

Une logique pour les AP

$\Sigma = \{a, b\}$. Pour $w = w_0 \ldots w_n \in \Sigma^*$, on définit :

$$S^\infty(w) = (\quad \mathbb{N} \quad , Q_a, Q_b), \quad Q_a = \{p \mid w_p = a\}$$

[Klaedtke and Rueß, 2003] proposent $\exists$MSO$^\infty$[card] :

Théorème

Pour tout AP de langage L, il existe une formule φ $\exists$MSO équipée de :

$$|X_1| + \ldots + |X_n| < |Y_1| + \ldots + |Y_n|$$

telle que $w \in L \Leftrightarrow S^\infty(w) \models \varphi$, et réciproquement

Une logique pour les AP

$\Sigma = \{a, b\}$. Pour $w = w_0 \dots w_n \in \Sigma^*$, on définit :

$$S^\infty(w) = (\quad \mathbb{N} \quad, Q_a, Q_b), \quad Q_a = \{p \mid w_p = a\}$$

[Klaedtke and Rueß, 2003] proposent $\exists\mathrm{MSO}^\infty[\mathrm{card}]$:

Théorème

Pour tout AP de langage L, il existe une formule φ ∃MSO équipée de :

$$|X_1| + \dots + |X_n| < |Y_1| + \dots + |Y_n|$$

telle que $w \in L \Leftrightarrow S^\infty(w) \models \varphi$, et réciproquement

T Une logique sur modèles finis connectée à des formalismes connus (quantificateur de comptage unaire)

Une logique pour les AP

$\Sigma = \{a, b\}$. Pour $w = w_0 \ldots w_n \in \Sigma^*$, on définit :

$$S\quad(w) = (\{0, \ldots, n\}, Q_a, Q_b), \qquad Q_a = \{p \mid w_p = a\}$$

[Klaedtke and Rueß, 2003] proposent $\exists\mathrm{MSO}^\infty[\mathrm{card}]$:

Théorème

Pour tout AP de langage L, il existe une formule φ ∃MSO équipée de :

$$|X_1| + \ldots + |X_n| < |Y_1| + \ldots + |Y_n|$$

telle que $w \in L \Leftrightarrow S^\infty(w) \vDash \varphi$, et réciproquement

T Une logique sur modèles finis connectée à des formalismes connus (quantificateur de comptage unaire)

Une logique pour les AP

$\Sigma = \{a, b\}$. Pour $w = w_0 \dots w_n \in \Sigma^*$, on définit :

$$S\ \ (w) = (\{0, \dots, n\}, Q_a, Q_b), \quad Q_a = \{p \mid w_p = a\}$$

[Klaedtke and Rueß, 2003] proposent $\exists\mathrm{MSO}^\infty[\mathrm{card}]$:

Théorème

Pour tout AP de langage L, il existe une formule φ ∃MSO équipée de :

$$|X| < |Y|$$

telle que $w \in L \Leftrightarrow S\ \ (w) \models \varphi$, et réciproquement

T Une logique sur modèles finis connectée à des formalismes connus (quantificateur de comptage unaire)

Caractérisations

T Plusieurs caractérisations ; en particulier :

Théorème

Pour tout langage L d'AP, il existe un automate A et un ensemble semi-linéaire C tel que :

$$L = \{\mathrm{Etiq}(\pi) \mid \pi \in \mathrm{ChemAcc}(A) \wedge \mathrm{Pkh}(\pi) \in C\}$$

et réciproquement

Caractérisations

T Plusieurs caractérisations ; en particulier :

Théorème

Pour tout langage L d'AP, il existe un automate A et un ensemble semi-linéaire C tel que :

$$L = \{\mathrm{Etiq}(\pi) \mid \pi \in \mathrm{ChemAcc}(A) \wedge \mathrm{Pkh}(\pi) \in C\}$$

et réciproquement

- En substance : l'automate ne fait que compter les transitions

Expressivité

T Quelques lemmes à la lemme de pompage :

Lemme

Soit L un langage d'AP. Il existe $0 < p < l$ tels que tout $w \in L$ avec $|w| \geq l$ s'écrit $w = uvxvz$ où :

Expressivité

T Quelques lemmes à la lemme de pompage :

Lemme

Soit L un langage d'AP. Il existe $0 < p < l$ tels que tout $w \in L$ avec $|w| \geq l$ s'écrit $w = uvxvz$ où :

- $0 < |v| \leq p$ *et* $|x| > p$,
- $|uvxv| \leq l$,
- $uv^2xz \in L$ *et* $uxv^2z \in L$

Expressivité

T Quelques lemmes à la lemme de pompage :

Lemme

Soit L un langage d'AP. Il existe $0 < p < l$ tels que tout $w \in L$ avec $|w| \geq l$ s'écrit $w = uvxvz$ où :

- $0 < |v| \leq p$ *et* $|x| > p$,
- $|uvxv| \leq l$,
- $uv^2xz \in L$ *et* $uxv^2z \in L$

- COPIE n'est pas représentable : $(a^p b)^{2l}$

Expressivité

T Quelques lemmes à la lemme de pompage :

Lemme

Soit L un langage d'AP. Il existe $0 < p < l$ tels que tout $w \in L$ avec $|w| \geq l$ s'écrit $w = uvxvz$ où :

- $0 < |v| \leq p$ *et* $|x| > p$,
- $|uvxv| \leq l$,
- $uv^2xz \in L$ *et* $uxv^2z \in L$

- COPIE n'est pas représentable : $(a^p b)^{2l}$
- **T** Langage séparant explicitement déterministes et non-déterministes

Des considérations en complexité

T On propose des bornes supérieurs :

Des considérations en complexité

T On propose des bornes supérieurs :

Théorème

- *Vérifier l'appartenance dans un ensemble SL :* $\mathsf{ACC}^0 \setminus \mathsf{AC}^0$

Des considérations en complexité

T On propose des bornes supérieurs :

Théorème

- *Vérifier l'appartenance dans un ensemble SL :* $\mathrm{ACC}^0 \backslash \mathrm{AC}^0$
- *Tout langage d'AP est dans NL* ($\mathrm{ACC}^0 \subseteq L$)

Des considérations en complexité

T On propose des bornes supérieurs :

Théorème

- *Vérifier l'appartenance dans un ensemble SL :* $\mathrm{ACC}^0 \backslash \mathrm{AC}^0$
- *Tout langage d'AP est dans NL* ($\mathrm{ACC}^0 \subseteq L$)
- *Tout langage d'AP déterministe est dans* NC^1 *(Notons : REG* $\subsetneq \mathrm{NC}^1$*)*

Plan

T Automate de Parikh sur lettres

- Idée : si *a* apparaît deux fois, son « poids » ne doit pas être influencé par sa position

T Automate de Parikh sur lettres

- Idée : si a apparaît deux fois, son « poids » ne doit pas être influencé par sa position
- En pratique : pour un AP (A, C), si $(a, \overline{d})$ et $(a, \overline{e})$ apparaissent dans $w \in L(A)$, alors $\overline{d} = \overline{e}$

T Automate de Parikh sur lettres

- Idée : si a apparaît deux fois, son « poids » ne doit pas être influencé par sa position
- En pratique : pour un AP (A, C), si $(a, \overline{d})$ et $(a, \overline{e})$ apparaissent dans $w \in L(A)$, alors $\overline{d} = \overline{e}$
- Caractérisations similaires : il existe L' rationnel et C' SL tels que :

$$L(A)\upharpoonright_C = \{w \in L' \mid \mathrm{Pkh}(w) \in C'\}$$

T Automate de Parikh sur lettres

- Idée : si a apparaît deux fois, son « poids » ne doit pas être influencé par sa position
- En pratique : pour un AP (A, C), si $(a, \overline{d})$ et $(a, \overline{e})$ apparaissent dans $w \in L(A)$, alors $\overline{d} = \overline{e}$
- Caractérisations similaires : il existe L' rationnel et C' SL tels que :

$$L(A)\restriction_C = \{w \in L' \mid \mathrm{Pkh}(w) \in C'\}$$

- En substance : l'automate ne fait que compter les lettres

T Automate de Parikh sur lettres

- Idée : si a apparaît deux fois, son « poids » ne doit pas être influencé par sa position
- En pratique : pour un AP (A, C), si $(a, \overline{d})$ et $(a, \overline{e})$ apparaissent dans $w \in L(A)$, alors $\overline{d} = \overline{e}$
- Caractérisations similaires : il existe L' rationnel et C' SL tels que :

$$L(A)\restriction_C = \{w \in L' \mid \mathrm{Pkh}(w) \in C'\}$$

- En substance : l'automate ne fait que compter les lettres
- Plus aucune différence entre déterministe et non-déterministe

T Expressivité

On montre :

- Des résultats d'expressivité :

Lemme

Si L est un langage d'AP sur lettres, alors pour tout R langage rationnel :

$$(L \cap R) \notin \mathrm{REG} \Rightarrow (\exists w \in R)[\mathrm{Pkh}(w) \notin \mathrm{Pkh}(L)]$$

T Expressivité

On montre :

- Des résultats d'expressivité :

Lemme

Si L est un langage d'AP sur lettres, alors pour tout R langage rationnel :

$$(L \cap R) \notin \mathrm{REG} \Rightarrow (\exists w \in R)[\mathrm{Pkh}(w) \notin \mathrm{Pkh}(L)]$$

Ex. : $L = \{a^n b a^n\}$,

T Expressivité

On montre :

- Des résultats d'expressivité :

Lemme

Si L est un langage d'AP sur lettres, alors pour tout R langage rationnel :

$$(L \cap R) \notin \mathrm{REG} \Rightarrow (\exists w \in R)[\mathrm{Pkh}(w) \notin \mathrm{Pkh}(L)]$$

Ex. : $L = \{a^n b a^n\}, R = (aa)^+ b (aa)^+$,

Expressivité

On montre :

- Des résultats d'expressivité :

Lemme

Si L est un langage d'AP sur lettres, alors pour tout R langage rationnel :

$$(L \cap R) \notin \mathrm{REG} \Rightarrow (\exists w \in R)[\mathrm{Pkh}(w) \notin \mathrm{Pkh}(L)]$$

Ex. : $L = \{a^n b a^n\}, R = (aa)^+ b (aa)^+$, et $a^{|w|_a/2} b a^{|w|_a/2}$

T Expressivité

On montre :

- Des résultats d'expressivité :

Lemme

Si L est un langage d'AP sur lettres, alors pour tout R langage rationnel :

$$(L \cap R) \notin \mathrm{REG} \Rightarrow (\exists w \in R)[\mathrm{Pkh}(w) \notin \mathrm{Pkh}(L)]$$

Ex. : $L = \{a^n b a^n\}, R = (aa)^+ b (aa)^+$, et $a^{|w|_a/2} b a^{|w|_a/2}$

- Des résultats de clôture :

	$\cup$	$\cap$	$\cdot$	$\bar{\cdot}$	$h(\cdot)$
AP lettres	☹	☻	☹	☹	☹

T Langages bornés

- Un langage L est borné si $L \subseteq w_1^* \dots w_n^*$

T Langages bornés

- Un langage L est borné si $L \subseteq w_1^* \dots w_n^*$

Théorème

Les langages bornés d'AP sont exactement tous les langages bornés L tels que $\mathrm{Pkh}(L)$ est SL

T Langages bornés

- Un langage L est borné si $L \subseteq w_1^* \dots w_n^*$

Théorème

Les langages bornés d'AP sont exactement tous les langages bornés L tels que $\mathrm{Pkh}(L)$ est SL

Or tout langage d'AP est sensible au contexte

T Langages bornés

- Un langage L est borné si $L \subseteq w_1^* \dots w_n^*$

Théorème

Les langages bornés d'AP sont exactement tous les langages bornés L tels que $\mathrm{Pkh}(L)$ est SL

Or tout langage d'AP est sensible au contexte

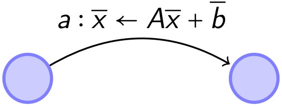
Théorème

Tout langage borné d'image de Parikh SL est SC

Plan

T Automate de Parikh affine

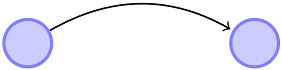
- L'idée : mettre à jour *des registres* au franchissement de chaque transition, avec une fonction affine



T Automate de Parikh affine

- L'idée : mettre à jour *des registres* au franchissement de chaque transition, avec une fonction affine

$$a : \overline{x} \leftarrow A\overline{x} + \overline{b}$$

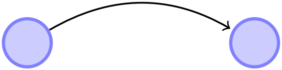


- Cas particulier : l'automate de Parikh

T Automate de Parikh affine

- L'idée : mettre à jour *des registres* au franchissement de chaque transition, avec une fonction affine

$$a : \overline{x} \leftarrow A\overline{x} + \overline{b}$$



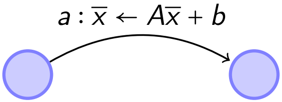
- Cas particulier : l'automate de Parikh

Un modèle décidable :

- Tout langage d'AP affine est dans NP ∩ SC

T Automate de Parikh affine

- L'idée : mettre à jour *des registres* au franchissement de chaque transition, avec une fonction affine



- Cas particulier : l'automate de Parikh

Un modèle décidable :

- Tout langage d'AP affine est dans NP $\cap$ SC

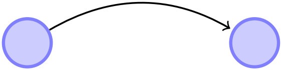
Aux problèmes indécidables :

- Problème du vide indécidable pour les AP affines det.

T Automate de Parikh affine

- L'idée : mettre à jour *des registres* au franchissement de chaque transition, avec une fonction affine

$$a : \overline{x} \leftarrow A\overline{x} + \overline{b}$$



- Cas particulier : l'automate de Parikh

Un modèle décidable :

- Tout langage d'AP affine est dans NP $\cap$ SC

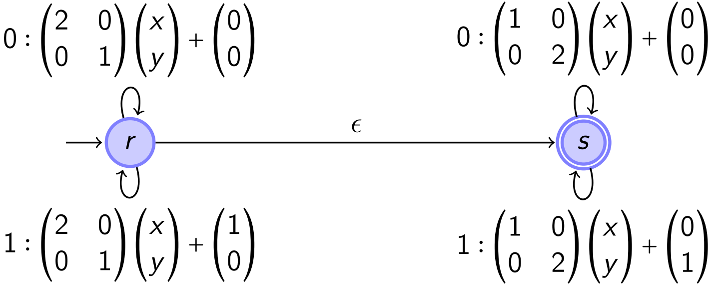
Aux problèmes indécidables :

- Problème du vide indécidable pour les AP affines det.

Des propriétés de clôture similaires aux AP :

	$\cup$	$\cap$	$\cdot$	$\bar{\cdot}$	$h(\cdot)$
AP af. dét.	☻	☻	O	☻	☹
AP af.	☻	☻	☻	O	☹

T Exemple, COPIE

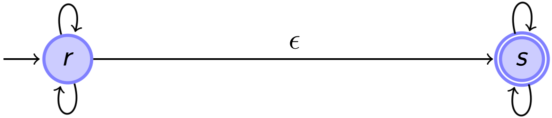


Mot : 0101 1001 Registres : (0, 0)

T Exemple, COPIE

$$0 : \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$0 : \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

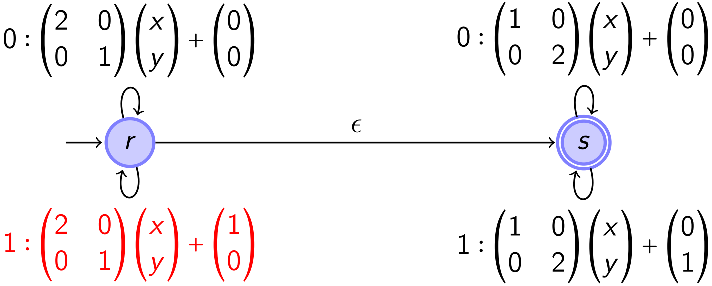


$$1 : \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$1 : \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Mot : 0101 1001 Registres : (0, 0)

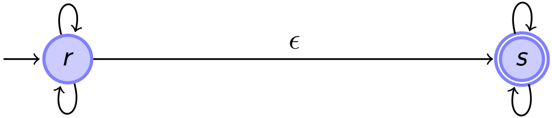
T Exemple, COPIE



Mot : 0101 1001 Registres : (1, 0)

T Exemple, COPIE

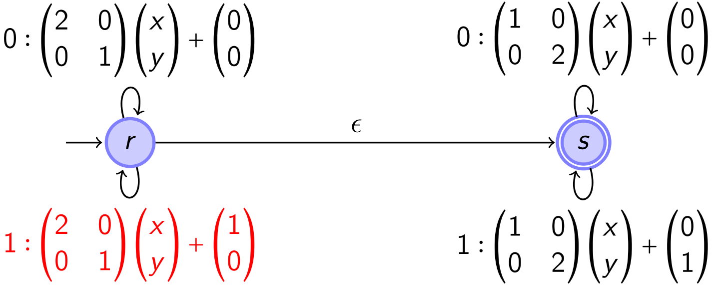
$$0 : \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad 0 : \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$



$$1 : \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad 1 : \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Mot : 0101 1001 Registres : (2, 0)

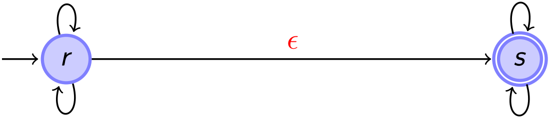
T Exemple, COPIE



Mot : 0101 1001 Registres : (5, 0)

T Exemple, COPIE

$$0 : \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad 0 : \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

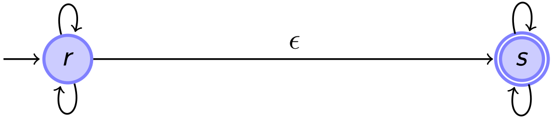


$$1 : \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad 1 : \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Mot : 0101 1001 Registres : (5, 0)

T Exemple, COPIE

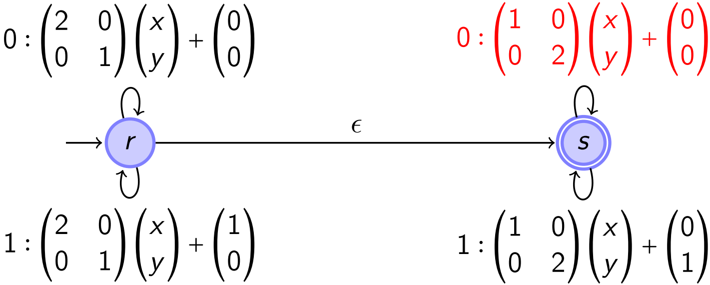
$$0 : \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad 0 : \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$



$$1 : \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad 1 : \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

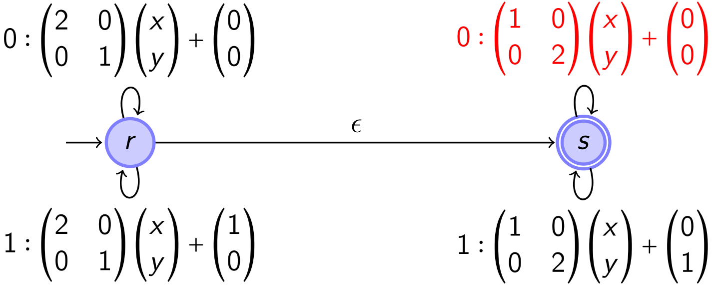
Mot : 0101 1001 Registres : (5, 1)

Exemple, COPIE



Mot : 0101 1001 Registres : (5, 2)

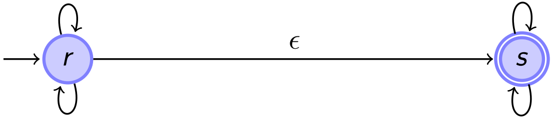
T Exemple, COPIE



Mot : 0101 1001 Registres : (5, 4)

T Exemple, COPIE

$$0 : \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad 0 : \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

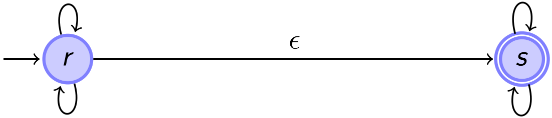


$$1 : \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad 1 : \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Mot : 0101 1001 Registres : (5, 9)

T Exemple, COPIE

$$0 : \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad 0 : \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

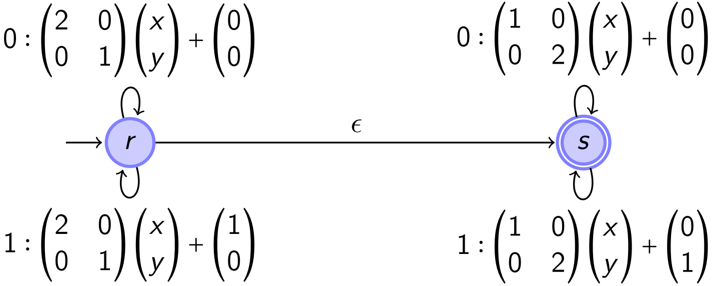


$$1 : \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad 1 : \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Mot : 0101 1001 Registres : (5, 9)

- i-ème composante : valeur binaire de la i-ème moitié

T Exemple, COPIE



Mot : 0101 1001 Registres : (5, 9)

- i-ème composante : valeur binaire de la i-ème moitié
- Ensemble de contraintes : $\{k_1(1,1) \mid k_1 \in \mathbb{N}\}$

T Les limites d'une généralisation ?

L'automate ne semble plus être utile dans le cas déterministe :

Théorème

Tout langage d'AP affine déterministe peut être représenté par un AP affine déterministe à 2 états

T Les limites d'une généralisation ?

L'automate ne semble plus être utile dans le cas déterministe :

Théorème

Tout langage d'AP affine déterministe peut être représenté par un AP affine déterministe à 2 états

De plus, trop limiter les fonctions fait perdre l'intérêt du modèle :

Théorème

Pour un AP affine, si les parties linéaires des fonctions affines se combinent de manière finie, alors il existe un AP de même langage

Plan

Caractérisation et étude des AP 1/2

- Caractérisation des langages d'AP bornés ?
  (Ex. dans les réguliers : w, w^*, clos par union et produit)

Caractérisation et étude des AP 1/2

- Caractérisation des langages d'AP bornés ?
  (Ex. dans les réguliers : w, w^*, clos par union et produit)
- Bornes inférieures en fonction du type d'automate sous-jacent ?
  (REG $\subset$ NC^1, plus petites classes de rationnels ?)

Caractérisation et étude des AP 1/2

- Caractérisation des langages d'AP bornés ?
  (Ex. dans les réguliers : w, w^*, clos par union et produit)
- Bornes inférieures en fonction du type d'automate sous-jacent ?
  (REG $\subset$ NC^1, plus petites classes de rationnels ?)
- Caractérisation des monoïdes syntactiques ?

Caractérisation et étude des AP 1/2

- Caractérisation des langages d'AP bornés ?
  (Ex. dans les réguliers : w, w^*, clos par union et produit)
- Bornes inférieures en fonction du type d'automate sous-jacent ?
  (REG $\subset$ NC^1, plus petites classes de rationnels ?)
- Caractérisation des monoïdes syntactiques ?

De la logique :

- Restriction de la logique aux déterministes ? aux AP sur lettres ?

Caractérisation et étude des AP 1/2

- Caractérisation des langages d'AP bornés ?
  (Ex. dans les réguliers : w, w^*, clos par union et produit)
- Bornes inférieures en fonction du type d'automate sous-jacent ?
  (REG $\subset$ NC^1, plus petites classes de rationnels ?)
- Caractérisation des monoïdes syntactiques ?

De la logique :

- Restriction de la logique aux déterministes ? aux AP sur lettres ?
- Jeux d'Ehrenfeucht-Fraïssé-Ajtai-Fagin ?

Caractérisation et étude des AP 1/2

- Caractérisation des langages d'AP bornés ?
  (Ex. dans les réguliers : w, w^*, clos par union et produit)
- Bornes inférieures en fonction du type d'automate sous-jacent ?
  (REG $\subset$ NC^1, plus petites classes de rationnels ?)
- Caractérisation des monoïdes syntactiques ?

De la logique :

- Restriction de la logique aux déterministes ? aux AP sur lettres ?
- Jeux d'Ehrenfeucht-Fraïssé-Ajtai-Fagin ?
- Lettre neutre/Conjecture de la plage de Crane

Caractérisation et étude des AP 2/2

Du décidable :

- Peut-on décider si un langage d'AP est rationnel ?
- Peut-on décider si un langage d'AP est hors-contexte ?
- Peut-on décider si un langage d'AP est borné ?

Caractérisation et étude des AP 2/2

Du décidable :

- Peut-on décider si un langage d'AP est rationnel ?
- Peut-on décider si un langage d'AP est hors-contexte ?
- Peut-on décider si un langage d'AP est borné ?

De généralisations/restrictions :

- Automates de Parikh avec alternance ?
  (Notons : REG=MSO=$\exists$MSO, pas le cas ici)

Caractérisation et étude des AP 2/2

Du décidable :

- Peut-on décider si un langage d'AP est rationnel ?
- Peut-on décider si un langage d'AP est hors-contexte ?
- Peut-on décider si un langage d'AP est borné ?

De généralisations/restrictions :

- Automates de Parikh avec alternance ?
  (Notons : REG=MSO=∃MSO, pas le cas ici)
- D'autres restrictions : épars ? Automate sans boucles imbriquées ?

Caractérisations et expressivité des AP affines

- Quels langages ne sont pas d'AP affines ? (3SAT ? PAL ?)

Caractérisations et expressivité des AP affines

- Quels langages ne sont pas d'AP affines ? (3SAT ? PAL ?)
- Et pas d'AP affines déterministes ? (COPIE ?)

Caractérisations et expressivité des AP affines

- Quels langages ne sont pas d'AP affines ? (3SAT ? PAL ?)
- Et pas d'AP affines déterministes ? (COPIE ?)
- Est-ce que les AP affines sont « de Parikh » ?

Caractérisations et expressivité des AP affines

- Quels langages ne sont pas d'AP affines ? (3SAT ? PAL ?)
- Et pas d'AP affines déterministes ? (COPIE ?)
- Est-ce que les AP affines sont « de Parikh » ?
- Quelle caractérisation logique ?

Caractérisations et expressivité des AP affines

- Quels langages ne sont pas d'AP affines ?
  (3SAT ? PAL ?)
- Et pas d'AP affines déterministes ?
  (COPIE ?)
- Est-ce que les AP affines sont « de Parikh » ?
- Quelle caractérisation logique ?
- Bornes inférieures ?

Des mots infinis

- En *model checking*, on considère des calculs infinis (« Dans le futur . . . »)

Des mots infinis

- En *model checking*, on considère des calculs infinis (« Dans le futur . . . »)

- Mots infinis

Des mots infinis

- En *model checking*, on considère des calculs infinis (« Dans le futur . . . »)
- Mots infinis
- Modèle usuel : automate de Büchi (= LTL)

Des mots infinis

- En *model checking*, on considère des calculs infinis (« Dans le futur ... »)

- Mots infinis

- Modèle usuel : automate de Büchi (= LTL)
- Ici, acceptation une infinité de fois

Des mots infinis

- En *model checking*, on considère des calculs infinis (« Dans le futur . . . »)

- Mots infinis

- Modèle usuel : automate de Büchi (= LTL)
- Ici, acceptation une infinité de fois

- Quelles caractérisations ? (Logiques temporelles ? Logiques classiques ?)

Plan

Conclusion

Un sujet d'avenir :

- Des modèles à étudier, généraliser, restreindre

Conclusion

Un sujet d'avenir :

- Des modèles à étudier, généraliser, restreindre
- Un sujet connectant 3 piliers de l'info. théorique

Conclusion

Un sujet d'avenir :

- Des modèles à étudier, généraliser, restreindre
- Un sujet connectant 3 piliers de l'info. théorique
- Une quantité de questions ouvertes dans les 3 domaines et en algèbre

Questions ?

¿ — ?

Ensembles semi-linéaires, arithmétique de Presburger

Automate de Parikh

Restrictions d'AP

Généralisations d'AP

Questions ouvertes

Conclusion

Bibliographie I

Ginsburg, S. and Spanier, E. H. (1966).
Semigroups, Presburger formulas and languages.
Pacific Journal of Mathematics, 16(2) :285–296.

Klaedtke, F. and Rueß, H. (2003).
Monadic second-order logics with cardinalities.
In *Proceedings of the 30th International Colloquium on Automata, Languages, and Programming (ICALP 2003)*, volume 2719 of *Lecture Notes in Computer Science*, pages 681–696. Springer-Verlag.

Parikh, R. J. (1966).
On context-free languages.
Journal of the ACM, 13(4) :570–581.