2-ways outer-nondeterministic finite automata

and

descriptional complexity

Bruno Guillon

29 Septembre 2011

Stage de M1, réalisé à l'*Università degli studi di Milano*,
encadré par *Giovanni Pighizzini*

Nous savons...

- ce qu'est un automate fini (unidirectionel)
  - déterministe ($1\textsc{dfa}$)
  - non-déterministe ($1\textsc{nfa}$)
- déterminiser un automate fini (*l'algorithme des parties*)

Complexité descriptive

On ne s'interesse plus à la puissance de calcul

Complexité descriptive

On ne s'interesse plus à la puissance de calcul

mais à la *taille* du modèle.

Complexité descriptive

On ne s'interesse plus à la puissance de calcul

mais à la *taille* du modèle.

Fonction *taille* :

Complexité descriptive

On ne s'interesse plus à la puissance de calcul

mais à la *taille* du modèle.

Fonction *taille* : nombre d'états.

Comparons les modèles

Deux types de résultats :

- Borne inférieure (langage témoin)
- Borne supérieure (algorithme)

Exemple

1DFA versus 1NFA

Exemple

1DFA versus 1NFA

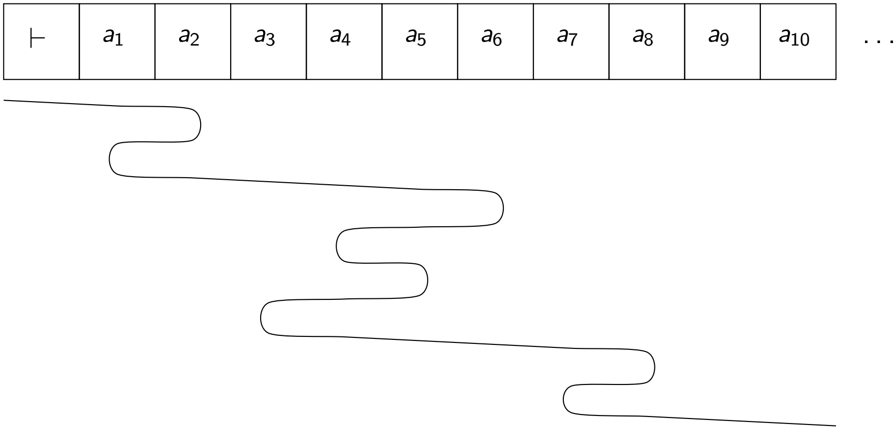
- langage $L_n = \{w_1 w_2 \ldots w_{|w|} \in \{0,1\}^*, w_{|w|-n+1}\}$
  - accepté par un automate fini non-déterministe à n états
  - un automate fini déterministe équivalent a au moins 2^n états

Exemple

1DFA versus 1NFA

- langage $L_n = \{w_1 w_2 \dots w_{|w|} \in \{0,1\}^*, w_{|w|-n+1}\}$
  - accepté par un automate fini non-déterministe à n états
  - un automate fini déterministe équivalent a au moins 2^n états
- Algorithme des parties :
  pour tout 1NFA à n états il existe un 1DFA équivalent avec 2^n états.

Automates finis bidirectionels



Automates finis bidirectionels

Formellement,
$\delta \in Q \times \Sigma \cup \{\vdash, \dashv\} \to 2^{Q \times \{-1,0,+1\}}$

Puissance de calcul

Théorème 1

Les automates finis bidirectionels reconnaissent exactement les langages rationnels.

Sakoda & Sipser : formalisation et questions

Classes de complexité descritpive $1D$, $1N$, $2D$, $2N$...

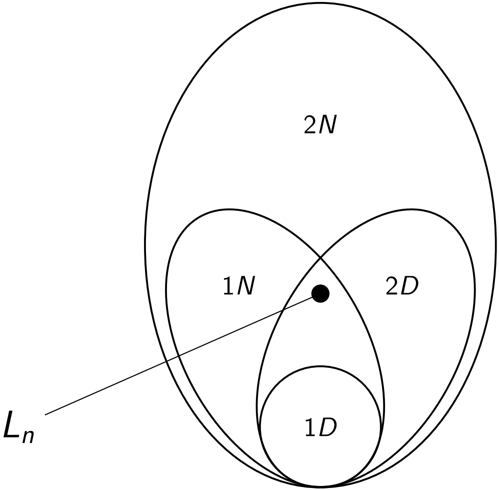
Sakoda & Sipser : formalisation et questions

Classes de complexité descritpive $1D$, $1N$, $2D$, $2N$...

Questions 2

- *Relation* $2D - 1N$
- *Relation* $2N - 2D$

Diagram



Approches

- *Sweeping automata* : Sipser (1980)
- *Oblivious automata* : Hromkovič (2003)
- automate avec nombre de changements de direction limité : Kapoutsis (2011)
- cas unaire : Geffert, Pighizzini, Mereghetti (2003 - 2011)
- ...

Automate "self-verifying" (SVFA)

Automate non-déterministe,

Automate “self-verifying” (SVFA)

Automate non-déterministe, qui a trois types de réponse :

- OUI
- NON
- JE NE SAIS PAS

Automate "self-verifying" (SVFA)

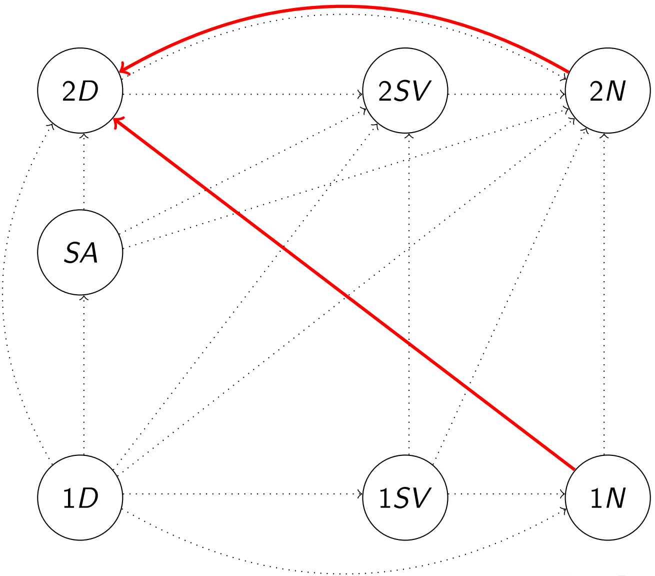
Automate non-déterministe, qui a trois types de réponse :

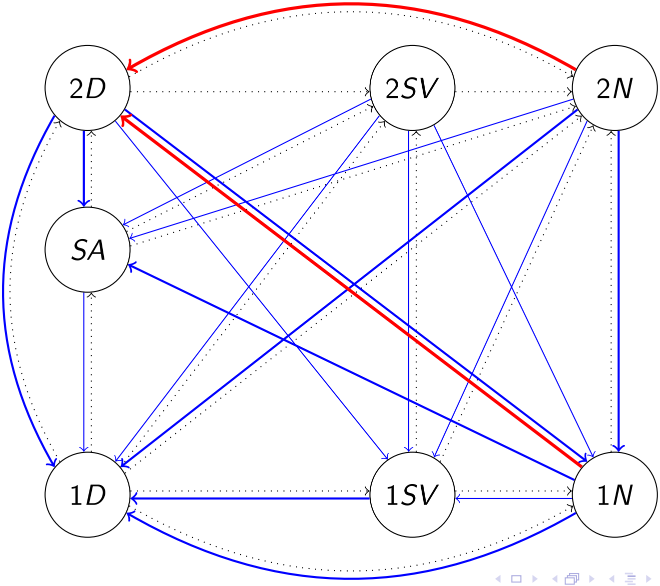
- OUI
- NON
- JE NE SAIS PAS

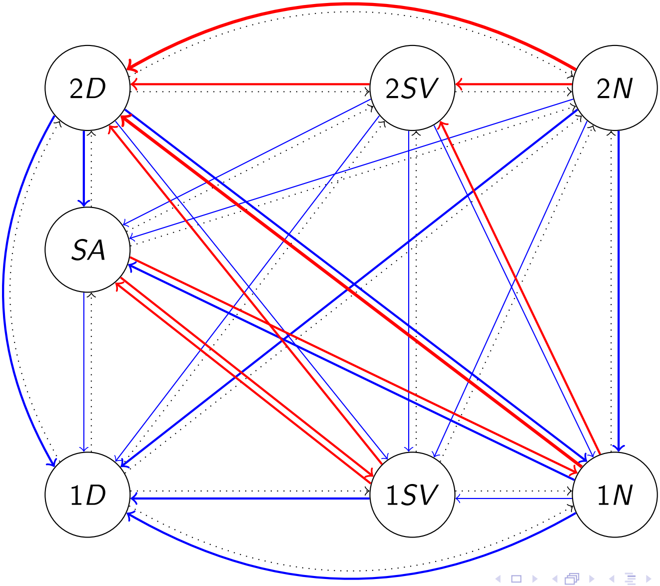
Pour tout mot w,

soit il existe une chemin acceptante

soit il existe une chemin rejetante


2D
2SV
2N
SA
1D
1SV
1N


2D
2SV
2N
SA
1D
1SV
1N


2D
2SV
2N
SA
1D
1SV
1N

Notre variante : 2ONFA

Autoriser les choix non-déterministes seulement aux marqueurs de fin.

Notre variante : 2ONFA

Autoriser les choix non-déterministes seulement aux marqueurs de fin.

Lemme 2 (*Forme normale*)

Pour tout 2ONFA $\mathcal{A}$ à n états il existe un automate $\mathcal{A}'$ équivalent, tel que :

- *$\mathcal{A}'$ a au plus $3n$ états*
- *$\mathcal{A}'$ a un unique état acceptant q_F, stoppant.*
- *q_F n'est atteignable qu'à partir du marqueur de fin gauche par une transition stationaire*
- *les choix non-déterministes ne sont pris qu'au marqueur de fin gauche*
- *les transitions stationnaires sont utilisées uniquement pour atteindre q_F*

Comportement

Définition 1

Un segment est une sequence de configurations successives qui démarre au marqueur gauche et finit au premier marquer gauche rencontré.

Comportement

Définition 2

Un segment est une sequence de configurations successives qui démarre au marqueur gauche et finit au premier marquer gauche rencontré.

Question 2

Peut-on décider déterministiquement l'existence d'un segment entre deux états donnés ?

OUI

OUI

Deux problèmes :

OUI

Deux problèmes :

- Un choix non-déterministe

OUI

Deux problèmes :

- Un choix non-déterministe
- Possibilité d'entrer dans une boucle

OUI

Deux problèmes :

- Un choix non-déterministe
- Possibilité d'entrer dans une boucle

voir tableau

Automates $reach_{s,t}$

Pour toutes pairs d'états s, t il existe un automate $reach_{s,t}$ tel que :

Automates $reach_{s,t}$

Pour toutes pairs d'états s, t il existe un automate $reach_{s,t}$ tel que :

- $reach_{s,t}$ a $4n - 3$ états

Automates $reach_{s,t}$

Pour toutes pairs d'états s, t il existe un automate $reach_{s,t}$ tel que :

- $reach_{s,t}$ a $4n - 3$ états
- $reach_{s,t}$ accepte w ssi il existe un segment de s à t (sur w)

Automates $reach_{s,t}$

Pour toutes pairs d'états s, t il existe un automate $reach_{s,t}$ tel que :

- $reach_{s,t}$ a $4n - 3$ états
- $reach_{s,t}$ accepte w ssi il existe un segment de s à t (sur w)
- $reach_{s,t}$ s'arrete toujours

Simulation de 2ONFA par un 2SVFA

Remarque 1

Tout mot du langage accepté, est accepté par une computation de moins de n segments
(sinon ça boucle).

Simulation de 2ONFA par un 2SVFA

Remarque 2

Tout mot du langage accepté, est accepté par une computation de moins de n segments
(sinon ça boucle).

- Idée :
  simuler segments par segments jusqu'à n segments (en utilisant *reach*)

Simulation de 2ONFA par un 2SVFA

Remarque 3

Tout mot du langage accepté, est accepté par une computation de moins de n segments
(sinon ça boucle).

- Idée :
  simuler segments par segments jusqu'à n segments (en utilisant *reach*)
- Problème :
  se souvenir des états accessibles après chaque segments prend trop de place

Simulation de 2ONFA par un 2SVFA

Remarque 4

Tout mot du langage accepté, est accepté par une computation de moins de n segments
(sinon ça boucle).

- Idée :
  simuler segments par segments jusqu'à n segments (en utilisant *reach*)
- Problème :
  se souvenir des états accessibles après chaque segments prend trop de place
- Solution :
  mémoriser seulement leur nombre, et les retrouver
  inductive counting

Algorithme

```
m' ← 1;
for k ← 1 to |Q| − 1 do
    m ← m'; m' ← 0;
    for each t ∈ Q do
        for i ← 1 to m do
            s ← simulation(k);
            if (i > 1 and s ≤ s_prev) then halt in q_?;
            s_prev ← s;
            if reach(s, t) then
                if t = q_accept then halt in q_yes;
                m' ← m' + 1;
                break;
            end
        end
    end
end
halt in q_no;
```

Théorèmes

Théorèmes

Théorème 6

$2\textsc{onfa}$ *à* n *états* $\rightarrow 2\textsc{svfa}$ *à* $O(n^8)$ *états*

Théorèmes

Théorème 10

$2\textsc{onfa}$ *à n états* $\rightarrow$ $2\textsc{svfa}$ *à* $O(n^8)$ *états*

Théorème 11

$2\textsc{onfa}$ *à n états* $\rightarrow$ $2\textsc{dfa}$ *à* $O(n^{\log_2 n+6})$ *états*

Théorèmes

Théorème 14

$2\textsc{onfa}$ *à* n *états* $\rightarrow$ $2\textsc{svfa}$ *à* $O(n^8)$ *états*

Théorème 15

$2\textsc{onfa}$ *à* n *états* $\rightarrow$ $2\textsc{dfa}$ *à* $O(n^{\log_2 n+6})$ *états*

Théorème 16

si $L = NL$ *alors :*

$2\textsc{onfa}$ *à* n *états* $\rightarrow$ $2\textsc{dfa}$ *à* $p(n)$

Théorèmes

Théorème 18

2ONFA *à n états* $\to$ 2SVFA *à* $O(n^8)$ *états*

Théorème 19

2ONFA *à n états* $\to$ 2DFA *à* $O(n^{\log_2 n+6})$ *états*

Théorème 20

si $L = NL$ *alors :*
2ONFA *à n états* $\to$ 2DFA *à* $p(n)$

Théorème 21

2ONFA *à n états* $\to$ 2UFA *à* $p(n)$ *états*

Conclusion

- découverte : complexité descriptive
- et automates bidirectionnels
- Lecture d'articles
- J'ai fait une présentation (en anglais)
- Rédaction d'article
- NCMA
- Ho imparato l'italiano

Conclusion

- découverte : complexité descriptive
- et automates bidirectionnels
- Lecture d'articles
- J'ai fait une présentation (en anglais)
- Rédaction d'article
- NCMA
- Ho imparato l'italiano

Merci, avez-vous des questions ?

Bibliographie I

J. Berman and A. Lingas.
On the complexity of regular languages in terms of finite automata.
Technical Report 304, Polish Academy of Sciences, 1977.

Piotr Berman.
A note on sweeping automata.
In *ICALP*, pages 91–97, 1980.

Pavol Duris, Juraj Hromkovic, José D. P. Rolim, and Georg Schnitger.
Las vegas versus determinism for one-way communication complexity, finite automata, and polynomial-time computations.
In Rüdiger Reischuk and Michel Morvan, editors, *STACS*, volume 1200 of *Lecture Notes in Computer Science*, pages 117–128. Springer, 1997.

Bibliographie II

V. Geffert, C. Mereghetti, and G. Pighizzini.
Converting two-way nondeterministic automata into simpler automata.
Theoret. Comput. Sci., 295:189–203, 2003.

Viliam Geffert, Carlo Mereghetti, and Giovanni Pighizzini.
Complementing two-way finite automata.
Inf. Comput., 205(8):1173–1187, 2007.

Viliam Geffert and Giovanni Pighizzini.
Two-way unary automata versus logarithmic space.
Inf. Comput., 209(7):1016–1025, 2011.

Jonathan Goldstine, Martin Kappes, Chandra M.R̃. Kintala, Hing Leung, Andreas Malcher, and Detlef Wotschke.
Descriptional complexity of machines with limited resources.
J. UCS, 8(2):193–234, 2002.

Bibliographie III

J. Hopcroft and J. Ullman.
Introduction to automata theory, languages, and computation.
Addison-Wesley, Reading, MA, 1979.

Juraj Hromkovic and Georg Schnitger.
Nondeterminism versus determinism for two-way finite automata: Generalizations of sipser's separation.
In Jos C.M̃. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *ICALP*, volume 2719 of *Lecture Notes in Computer Science*, pages 439–451. Springer, 2003.

Christos A. Kapoutsis.
Removing bidirectionality from nondeterministic finite automata.
In *MFCS*, pages 544–555, 2005.

Bibliographie IV

Christos A. Kapoutsis.
Small sweeping 2nfas are not closed under complement.
In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP (1)*, volume 4051 of *Lecture Notes in Computer Science*, pages 144–156. Springer, 2006.

Christos A. Kapoutsis.
Size complexity of two-way finite automata.
In Volker Diekert and Dirk Nowotka, editors, *Developments in Language Theory*, volume 5583 of *Lecture Notes in Computer Science*, pages 47–66. Springer, 2009.

Christos A. Kapoutsis.
Nondeterminism is essential in small 2fas with few reversals.
In *ICALP (2)*, pages 198–209, 2011.

Bibliographie V

Christos A. Kapoutsis.
Two-way automata versus logarithmic space.
In *CSR*, pages 359–372, 2011.

Martin Kappes, Andreas Malcher, and Detlef Wotschke.
Remembering chandra kintala.
In *DCFS*, pages 15–26, 2010.

R. Karp and R. Lipton.
Turing machines that take advice.
Enseign. Math., 28:191–209, 1982.

Hing Leung.
Separating exponentially ambiguous finite automata from polynomially ambiguous finite automata.
SIAM J. Comput., 27(4):1073–1082, 1998.

Bibliographie VI

Hing Leung.
Tight lower bounds on the size of sweeping automata.
J. Comput. Syst. Sci., 63(3):384–393, 2001.

Albert R. Meyer and Michael J. Fischer.
Economy of description by automata, grammars, and formal systems.
In *FOCS*, pages 188–191. IEEE, 1971.

Silvio Micali.
Two-way deterministic finite automata are exponentially more succinct than sweeping automata.
Inf. Process. Lett., 12(2):103–105, 1981.

M. Rabin and D. Scott.
Finite automata and their decision problems.
IBM J. Res. Develop., 3:114–125, 1959.

Bibliographie VII

Klaus Reinhardt and Eric Allender.
Making nondeterminism unambiguous.
SIAM J. Comput., 29(4):1118–1131, 2000.

William J. Sakoda and Michael Sipser.
Nondeterminism and the size of two way finite automata.
In *STOC*, pages 275–286. ACM, 1978.

Michael Sipser.
Halting space-bounded computations.
Theor. Comput. Sci., 10:335–338, 1980.

Michael Sipser.
Lower bounds on the size of sweeping automata.
J. Comput. Syst. Sci., 21(2):195–202, 1980.