Descriptional Complexity and Parikh Equivalence

Giovanna J. Lavado[1], G. Pighizzini[1], S. Seki[2]

[1]Dipartimento di Informatica, Università degli Studi di Milano, Italy

[2]Department of Information and Computer Science, Aalto University, Finland

Progetto PRIN 2010 - 2011
Automi e Linguaggi Formali: Aspetti Matematici e Applicativi
November 28, 2013

Standard equivalence: 1NFAs vs 1DFAs

Subset construction [Rabin&Scott '59]

1NFA
n states
L

$\Longrightarrow$

1DFA
2^n states
L

Moreover, this state bound cannot be reduced in the worst case [Meyer&Fischer '71, Moore '71]

What happens if we do not care of the order of symbols in the strings?

This problem is related to the concept of *Parikh equivalence* [Parikh '66]

Standard equivalence: 1NFAs vs 1DFAs

Subset construction [Rabin&Scott '59]

1NFA		1DFA
n states	$\Longrightarrow$	2^n states
L		L

Moreover, this state bound cannot be reduced in the worst case [Meyer&Fischer '71, Moore '71]

What happens if we do not care of the order of symbols in the strings?

This problem is related to the concept of *Parikh equivalence* [Parikh '66]

Parikh equivalence: preliminaries

- $\Sigma = \{a_1, \ldots, a_m\}$ alphabet of m symbols
- $|w|_a$ be the number of occurrences of a in $w \in \Sigma^*$
- *Parikh's map* $\psi : \Sigma^* \to \mathbb{N}^m$

$$\psi(w) = (|w|_{a_1}, |w|_{a_2}, \ldots, |w|_{a_m})$$

  for each string $w \in \Sigma^*$
- *Parikh's image* of a language $L \subseteq \Sigma^*$:

$$\psi(L) = \{\psi(w) \mid w \in L\}$$

- $w_1 =_\pi w_2$ iff $\psi(w_1) = \psi(w_2)$
- $L_1 =_\pi L_2$ iff $\psi(L_1) = \psi(L_2)$

Parikh equivalence: Parikh's theorem

Theorem ([Parikh '66])

For each context-free language $L \subseteq \Sigma^*$, there exists a Parikh equivalent regular language $R \subseteq \Sigma^*$.

Example:

$$L = \{a^n b^n \mid n \geq 0\} \quad \text{and} \quad R = (ab)^*$$

have the same Parikh image, namely the set

$$\psi(L) = \psi(R) = \{(n, n) \mid n \geq 0\}$$

Parikh equivalence: motivations

- Interesting theoretical properties:
  wrt Parikh equivalence regular and context-free languages are indistinguishable [Parikh '66]
- Connections with:
  - Semilinear sets
  - Presburger arithmetics [Ginsburg&Spanier '66]
  - Petri nets [Esparza '97]
  - Logical formulas [Verma&Seidl&Schwentick '05]
  - Formal verification
    [Dang&Ibarra&Bultan&Kemmerer&Su '00, Göller&Mayr&To '09]
  - ...

Our Goal: from a *descriptional complexity* point of view

We investigate the conversion of one-way nondeterministic automata (1NFAs) and context-free grammars (CFGs) into *Parikh equivalent* one-way and two-way deterministic automata (1DFAs and 2DFAs)

Problem (1NFAs to 1DFAs and 2DFAs)

1NFA *n states*	$\Longrightarrow_\pi$	1DFA, 2DFA *how many states?*

Problem (CFGs to 1DFAs and 2DFAs)

CFG *in Chomsky normal form* *h variables*	$\Longrightarrow_\pi$	1DFA, 2DFA *how many states?*

From 1NFAs to Parikh equivalent 1DFAs

Our first contribution:

Problem (1NFAs to 1DFAs)

1NFA		1DFA
n states	$\Longrightarrow_\pi$	*how many states?*
L_1		L_2

- Upper bound: 2^n
  by subset construction [Rabin&Scott '59]
- Lower bound: $e^{\sqrt{n \ln n}}$
  This bound derives from the *unary case*:
  the state cost of the conversion of unary n-state 1NFAs
  into equivalent 1DFAs is $e^{\Theta(\sqrt{n \ln n})}$ [Chrobak '86]

Converting 1NFAs accepting only nonunary strings

A preliminary step:

Problem (1NFAs to 1DFAs, restricted)

1NFA *s.t. each accepted string is nonunary* *n states* L_1	$\Longrightarrow_\pi$	1DFA *how many states?* L_2

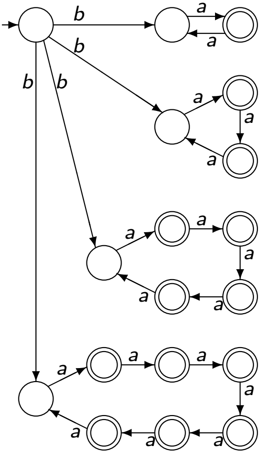
Quite surprisingly, we can obtain a 1DFA with a number of states *polynomial* in n,

i.e., this conversion is less expensive than the conversion in the unary case, which costs $e^{\Theta(\sqrt{n \ln n})}$

Converting 1NFAs accepting only nonunary strings

Let us consider the following language

$$L = \{ba^n \mid n \bmod 210 \neq 0\}$$



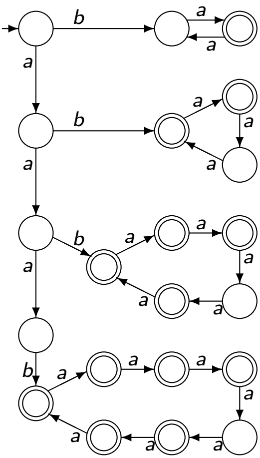
- L does not contain any unary word.
- $L_1 = \{ba^n \mid n \bmod 2 \neq 0\}$,
- $L_2 = \{ba^n \mid n \bmod 3 \neq 0\}$,
- $L_3 = \{ba^n \mid n \bmod 5 \neq 0\}$,
- $L_4 = \{ba^n \mid n \bmod 7 \neq 0\}$,
- $L = L_1 \cup L_2 \cup L_3 \cup L_4$.

- L is accepted by the 18-state 1NFA A.
- The smallest 1DFA for L uses 211 states.

Converting 1NFAs accepting only nonunary strings

We can build a complete 1DFA A' with only 22 states, accepting a language L' Parikh equivalent to L.

- $L_i =_\pi L'_i$, $i = 1, \ldots, 4$
- Words in L'_i begin with the prefix $a^{i-1}b$
- $L'_1 = \{ba^n \mid n \bmod 2 \neq 0\} = L_1$,
- $L'_2 = \{aba^{n-1} \mid n \bmod 3 \neq 0\}$,
- $L'_3 = \{a^2ba^{n-2} \mid n \bmod 5 \neq 0\}$,
- $L'_4 = \{a^3ba^{n-3} \mid n \bmod 7 \neq 0\}$,
- $L' = L'_1 \cup L'_2 \cup L'_3 \cup L'_4$.



Converting 1NFAs accepting only nonunary strings

The conversion uses a modification of the following result:

Theorem ([Kopczyński&To '10])

Given $\Sigma = \{a_1, \ldots, a_m\}$, there is a polynomial p s.t. for each n-state 1NFA A over Σ,

$$\psi(L(A)) = \bigcup_{i \in I} Z_i$$

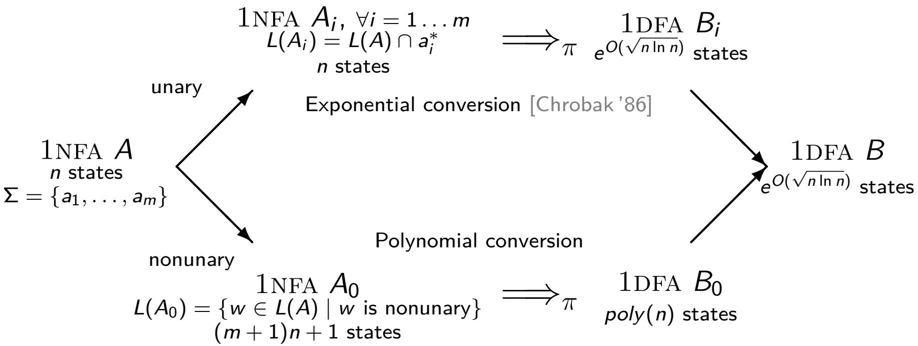
where:

- *I is a set of at most $p(n)$ indices*
- *for $i \in I$, $Z_i \subseteq \mathbb{N}^m$ is a linear set of the form:*

  $$Z_i = \{\alpha_0 + n_1\alpha_1 + \cdots + n_k\alpha_k \mid n_1, \ldots, n_k \in \mathbb{N}\}$$

  with
  - $0 \leq k \leq m$
  - *the components of α_0 are bounded by $p(n)$*
  - *$\alpha_1, \ldots, \alpha_k$ are linearly independent vectors from $\{0, 1, \ldots, n\}^m$*

From 1NFAs to Parikh equivalent 1DFAs: general case



Theorem

For each n-state 1NFA *over* Σ*, there exists a Parikh equivalent* 1DFA *with* $e^{O(\sqrt{n \ln n})}$ *states. Furthermore, this cost is optimal.*

From CFGs to Parikh equivalent 1DFAs

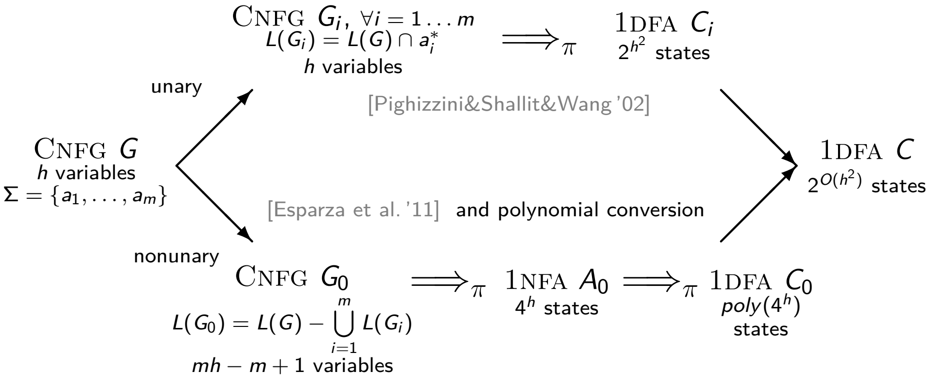
Our second contribution:

Problem (CFGs to 1DFAs)

CNFG CFG *in Chomsky normal form* h *variables* L_1	$\Longrightarrow_\pi$	1DFA *how many states?* L_2

- Upper bound: $2^{O(4^h)}$
  by subset construction
  and [Esparza&Ganty&Kiefer&Luttenberger '11]
- Lower bound: 2^{ch^2}
  This bound derives from the *unary case*:
  the state cost of the conversion of unary h-variable CNFGs into equivalent 1DFAs is less than 2^{h^2}
  [Pighizzini&Shallit&Wang '02]

From CFGs to Parikh equivalent 1DFAs



Theorem

For each h-variable CNFG over Σ, there exists a Parikh equivalent 1DFA with at most $2^{O(h^2)}$ states. Furthermore, this cost is optimal.

Conversions into Parikh equivalent 2DFAs

	1DFA	2DFA
1NFA n states		
CNFG h variables		

Conversions into Parikh equivalent 2DFAs

	1DFA	2DFA
1NFA n states	$e^{O(\sqrt{n \ln n})}$	
CNFG h variables	$2^{O(h^2)}$	

Conversions into Parikh equivalent 2DFAs

	1DFA	2DFA
1NFA n states	$e^{O(\sqrt{n \ln n})}$	$poly(n)$
CNFG h variables	$2^{O(h^2)}$	$2^{O(h)}$

The idea is to split the accepted language into its unary and nonunary parts:

- The state cost of the conversion of unary n-state 1NFAs into equivalent 2DFAs is $O(n^2)$ [Chrobak '86]
- The state cost of the conversion of unary h-variable CNFGs into equivalent 1NFAs is at most $2^{2h-1} + 1$ [Pighizzini&Shallit&Wang '02]

Conclusion

- Quite surprisingly, the costs into Parikh equivalent 1DFA is due to the unary parts of the languages.
- The conversion of the parts consisting of nonunary strings is less expensive.
- Conversions into Parikh equivalent 2DFA are less expensive than the corresponding ones into 1DFA.

Further works

- Minimization problem: given a deterministic automaton A, is there a deterministic automaton A' such that
  - A' is Parikh equivalent to A
  - number of states of A' is less than number of states of A?
- State complexity of operations (concatenation, union, intersection and so on) under Parikh equivalence.
- It could be interesting to study computational complexity aspects of Parikh equivalent conversions.

Gracias por la atención!