The Complexity of Intersecting Finite Automata Having Few Final States

Michael Blondin   Pierre McKenzie

Département d'informatique et de recherche opérationnelle,
Université de Montréal, Québec

July 7, 2012

Definition

An *automaton* is a 5-tuple:

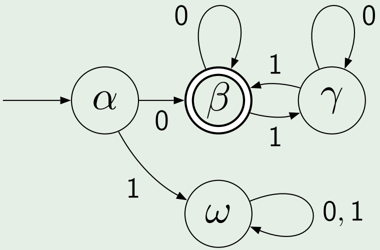
- Ω (finite set of *states*)
- Σ (finite *alphabet*)
- $\delta : \Omega \times \Sigma \to \Omega$ (*transition function*)
- $\alpha \in \Omega$ (*initial state*)
- $F \subseteq \Omega$ (*final states*)

Definition

Transition monoid $\mathcal{M}(A)$ of A:

$$\langle\{T_\sigma \ : \ \sigma \in \Sigma\}\rangle \text{ where } T_\sigma(\gamma) = \delta(\gamma, \sigma).$$

Example



$$T_{011} = \begin{pmatrix} \alpha & \beta & \gamma & \omega \\ \beta & \beta & \gamma & \omega \end{pmatrix}$$

Definition

$\mathsf{AutoInt}_b(X)$ (Automata nonemptiness intersection problem)

Input: Automata $A_1, \ldots, A_k$ on alphabet Σ with $\mathcal{M}(A_i) \in X$ and at most b final states.

Question: $\bigcap_{i=1}^{k} \mathrm{Language}(A_i) \neq \emptyset$?

Kozen 77

AutoInt and AutoInt_1 are PSPACE−complete.

Galil 76

AutoInt is NP−complete when $\Sigma = \{a\}$.

AutoInt interesting because generalizes:

Definition

Memb(X) (Membership problem)

Input: $g, g_1, \ldots, g_k : [m] \to [m]$ such that $\langle g_1, \ldots, g_k \rangle \in X$.
Question: $g \in \langle g_1, \ldots, g_k \rangle$?

AutoInt interesting because generalizes:

Definition

Memb(X) (Membership problem)

Input: $g, g_1, \ldots, g_k : [m] \to [m]$ such that $\langle g_1, \ldots, g_k \rangle \in X$.
Question: $g \in \langle g_1, \ldots, g_k \rangle$?

Connections with graph isomorphism led to deep results on group problems. It is known that Memb(Groups) $\in$ NC.

Definition

AC^k: languages accepted by Boolean circuits of poly size and depth $O(\log^k n)$. NC^k: similar with gates of indegree 2.

$$NC = AC = \bigcup_{k \geq 0} NC^k$$

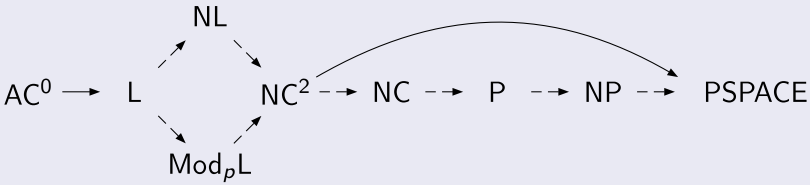
Definition

L: languages accepted by log-space Turing machines.

NL: languages accepted by log-space non deterministic Turing machines.

Mod_pL: languages S s.t. $w \in S$ iff $\#$ accept paths $\equiv 0 \pmod{p}$ for some NL machine.

Inclusion chain of complexity classes



Main result: completeness results for $\text{AutoInt}_b(X)$

	Maximum number of final states		
	1	2	3+
$\Sigma = \{a\}$	L	L	NP
$\mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2$	$\oplus$L	$\oplus$L	NP
$\mathbb{Z}_p \times \cdots \times \mathbb{Z}_p$	Mod_pL	NP	NP
Abelian groups	$\in \text{NC}^3, \text{FL}^{\text{ModL}}/\text{poly}$	NP	NP
Groups	$\in$ NC	NP	NP
$\mathbf{J_1}$	$\in \text{AC}^0$	NP	NP

White: Our classification.

Green: Will appear in journal version (Blondin, Krebs & McKenzie).

Orange: Beaudry 88.

Main result: completeness results for $\text{AutoInt}_b(X)$

	Maximum number of final states		
	1	2	3+
$\Sigma = \{a\}$	L	L	NP
$\mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2$	$\oplus$L	$\oplus$L	NP
$\mathbb{Z}_p \times \cdots \times \mathbb{Z}_p$	Mod_pL	NP	NP
Abelian groups	$\in \text{NC}^3, \text{FL}^{\text{ModL}}/\text{poly}$	NP	NP
Groups	$\in$ NC	NP	NP
$\mathbf{J_1}$	$\in \text{AC}^0$	NP	NP

Theorem

$\text{AutoInt}_2(X)$ *is hard for* NP *for any* X *beyond* $\mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2$.

Proof sketch

$X \not\subseteq \mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2$ implies aperiodic monoid or cyclic group $\mathbb{Z}_q$, $q > 2$, in X.

Reduction from CIRCUIT–SAT to $\text{AutoInt}_2(X)$ in both cases.

Theorem

$\mathsf{AutoInt}_2(X)$ *is hard for* NP *for any* X *beyond* $\mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2$*.*

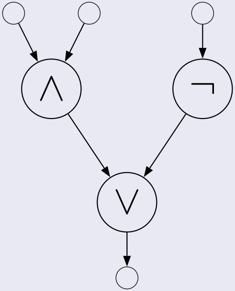
Proof sketch

$X \not\subseteq \mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2$ implies aperiodic monoid or cyclic group $\mathbb{Z}_q$, $q > 2$, in X.

Reduction from CIRCUIT–SAT to $\mathsf{AutoInt}_2(X)$ in both cases.

Proof sketch: CIRCUIT–SAT reduces to AutoInt$_2(\mathbb{Z}_q)$

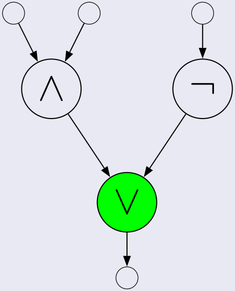
Given a circuit, we let Σ be the set of gates.



$$\Sigma = \{\circ_0, \circ_1, \circ_2, \wedge_0, \neg_0, \vee_0, \circ_3\}$$

Proof sketch: CIRCUIT–SAT reduces to AutoInt$_2(\mathbb{Z}_q)$

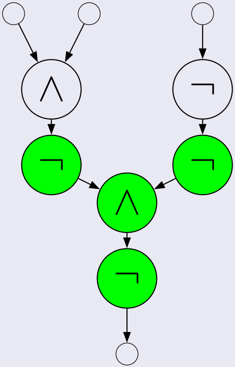
Given a circuit, we let Σ be the set of gates.



$$\Sigma = \{\circ_0, \circ_1, \circ_2, \wedge_0, \neg_0, \vee_0, \circ_3\}$$

Proof sketch: CIRCUIT–SAT reduces to AutoInt$_2(\mathbb{Z}_q)$

Given a circuit, we let Σ be the set of gates.



$\Sigma = \{\circ_0, \circ_1, \circ_2, \wedge_0, \neg_0, \neg_1, \neg_2, \wedge_2 \circ_3\}$

Proof sketch: CIRCUIT–SAT reduces to AutoInt$_2(\mathbb{Z}_q)$

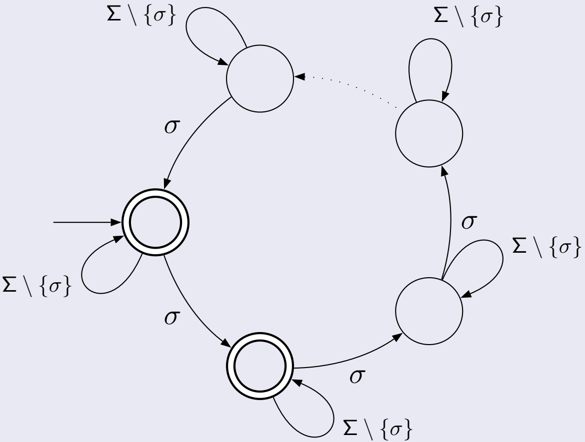
For each gate σ, we build automata A such that $\mathcal{M}(A) = \mathbb{Z}_p$.

Strategy:

- Occurrences of σ mod p encode assignment to σ (0 or 1),
- Automata verify soundness locally,
- Intersection represents satisfying assignments.

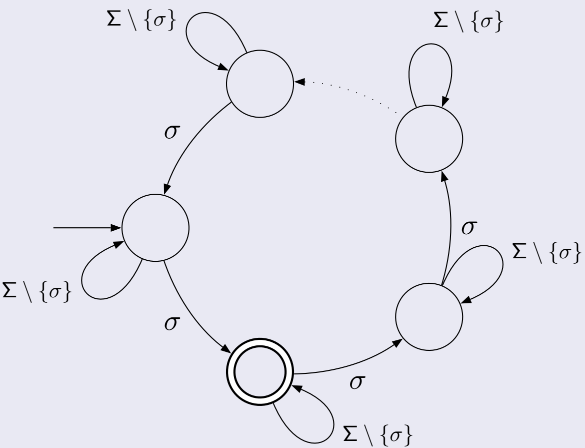
Proof sketch: CIRCUIT–SAT reduces to AutoInt$_2(\mathbb{Z}_q)$

For each $\sigma \in \Sigma$, we accept words w such that $|w|_\sigma \equiv 0, 1 \pmod{q}$.



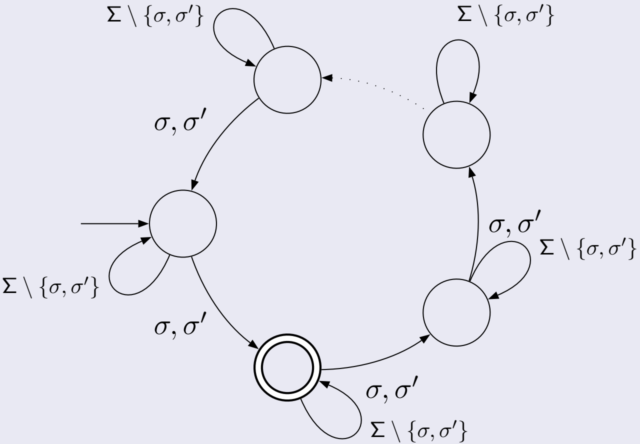
Proof sketch: CIRCUIT–SAT reduces to AutoInt$_2(\mathbb{Z}_q)$

For output gate σ, we accept words w such that $|w|_\sigma \equiv 1 \pmod{q}$.



Proof sketch: CIRCUIT–SAT reduces to $\mathrm{AutoInt}_2(\mathbb{Z}_q)$

For each ¬-gate σ with input σ', we accept words w such that $|w|_\sigma + |w|_{\sigma'} \equiv 1 \pmod q$.



Proof sketch: CIRCUIT–SAT reduces to $\mathsf{AutoInt}_2(\mathbb{Z}_q)$

For each $\wedge$-gate σ with inputs σ', σ'', we accept words w such that $|w|_{\sigma'} + |w|_{\sigma''} - 2\,|w|_{\sigma} \equiv 0, 1 \pmod{q}$.

$\sigma'\sigma''\sigma$	$\sigma' \wedge \sigma'' = \sigma$	$\sigma' + \sigma'' - 2\sigma$
000	1	0
001	0	-2
010	1	1
011	0	-1
100	1	1
101	0	-1
110	0	2
111	1	0

Proof sketch: CIRCUIT–SAT reduces to $\mathrm{AutoInt}_2(\mathbb{Z}_q)$

Problems when $q = 3$ since $-2 \equiv 1 \pmod 3$.

$\sigma'\sigma''\sigma$	$\sigma' \wedge \sigma'' = \sigma$	$\sigma' + \sigma'' - 2\sigma$
000	1	0
001	0	-2
010	1	1
011	0	-1
100	1	1
101	0	-1
110	0	2
111	1	0

Proof sketch: CIRCUIT–SAT reduces to $\mathrm{AutoInt}_2(\mathbb{Z}_q)$

When $q = 3$, we also build $|w|_{\sigma'} + |w|_{\sigma''} - |w|_{\sigma} \equiv 0, 1 \pmod 3$.

$\sigma'\sigma''\sigma$	$\sigma' \wedge \sigma'' = \sigma$	$\sigma' + \sigma'' - 2\sigma$	$\sigma' + \sigma'' - \sigma$
000	1	0	0
001	0	1	2
010	1	1	1
011	0	2	0
100	1	1	1
101	0	2	0
110	0	2	2
111	1	0	1

Proof sketch: CIRCUIT–SAT reduces to $\text{AutoInt}_2(\mathbb{Z}_q)$

$\Rightarrow$) A satisfying assignment yields a word $\sigma_1^{b_1} \cdots \sigma_s^{b_s}$ accepted by the automata.

$\Leftarrow$) A word w accepted by the intersection yields a sastisfying assignment $\sigma_i \leftarrow |w|_{\sigma_i} \bmod p$. □

Complexity of AutoInt$_2(\mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2)$

	Maximum number of final states		
	1	2	3+
$\Sigma = \{a\}$	L	L	NP
$\mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2$	$\oplus$L	$\oplus$L	NP
$\mathbb{Z}_p \times \cdots \times \mathbb{Z}_p$	Mod$_p$L	NP	NP
Abelian groups	$\in \mathsf{NC}^3, \mathsf{FL}^{\mathsf{ModL}}/\mathsf{poly}$	NP	NP
Groups	$\in$ NC	NP	NP
J_1	$\in \mathsf{AC}^0$	NP	NP

Hint for AutoInt$_2(\mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2) \in \oplus$L

We solve AutoInt$_1$(Abelian groups) with congruences. Extending it to AutoInt$_2(\mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2)$ yields systems of the form:

$$\exists \vec{x} \quad B\vec{x} \equiv b \pmod 2 \;\vee\; B\vec{x} \equiv b' \pmod 2.$$

It is equivalent to

$$\exists \vec{x}, y, y' \quad \begin{pmatrix} \vec{0} & 1 & 1 \\ B & b & b' \end{pmatrix} \begin{pmatrix} \vec{x} \\ y \\ y' \end{pmatrix} \equiv \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \pmod 2.$$

Gap from $\text{AutoInt}_2(\mathbb{Z}_2)$ to $\text{AutoInt}_2(\mathbb{Z}_q)$

	Maximum number of final states		
	1	2	3+
$\Sigma = \{a\}$	L	L	NP
$\mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2$	$\oplus$L	$\oplus$L	NP
$\mathbb{Z}_p \times \cdots \times \mathbb{Z}_p$	Mod_pL	NP	NP
Abelian groups	$\in \text{NC}^3, \text{FL}^{\text{ModL}}/\text{poly}$	NP	NP
Groups	$\in$ NC	NP	NP
$\mathbf{J_1}$	$\in \text{AC}^0$	NP	NP

- Relationships between algebraic problems and $\mathrm{AutoInt}_b(X)$
- Extensive classification of $\mathrm{AutoInt}_b$
- Close relationship between complexity of Memb and $\mathrm{AutoInt}_1$
- Surprising gap from $\mathrm{AutoInt}_2(\mathbb{Z}_2)$ to $\mathrm{AutoInt}_2(\mathbb{Z}_3)$

What is the complexity of $\mathrm{AutoInt}_1(X)$ for other X such that $\mathrm{Memb}(X)$ is in between P and NP?

Спасибо! Thank you! Merci!