A Quadratic Lower Bound for Simulation

Jan Friso Groote
Jan Martens

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Sjouke Mauw

Technische Universiteit
Eindhoven
University of Technology

Sjouke Mauw

TU/e Technische Universiteit Eindhoven University of Technology

Sjouke Mauw

Process Algebra as a Tool for the Specification and Verification of CIM-architectures

Technische Universiteit **Eindhoven** University of Technology

Sjouke Mauw

Process Algebra as a Tool for the Specification and Verification of CIM-architectures

Frank Biemans and Sjoerd Sjoerdsma
Pieter Blonk and Axel Belinfante

TU/e Technische Universiteit Eindhoven University of Technology

Sjouke Mauw

Process Algebra as a Tool for the Specification and Verification of CIM-architectures

Frank Biemans and Sjoerd Sjoerdsma
Pieter Blonk and Axel Belinfante

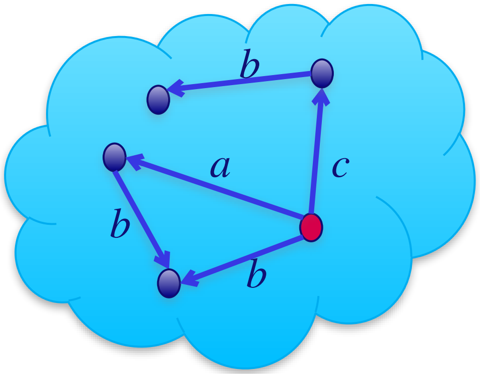
Now the complete definition of the Workcell (W) is

$$W = \tau_I \partial_H (WC \| T^\lambda \| WA \| WB)$$

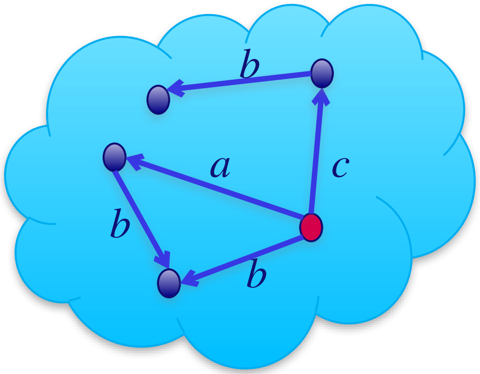
Technische Universiteit
Eindhoven
University of Technology

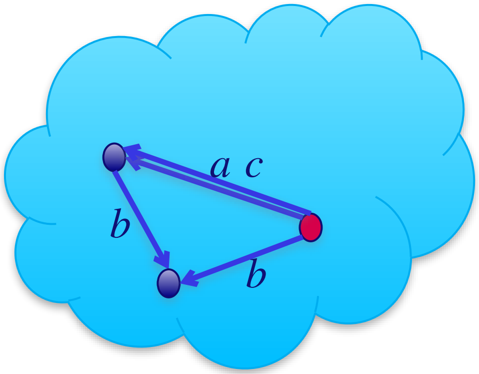
Process equivalences.

Process equivalences.

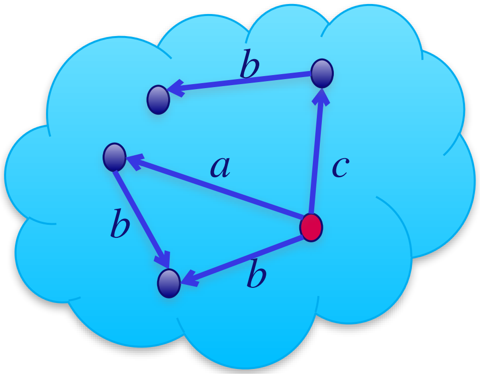


Process equivalences.

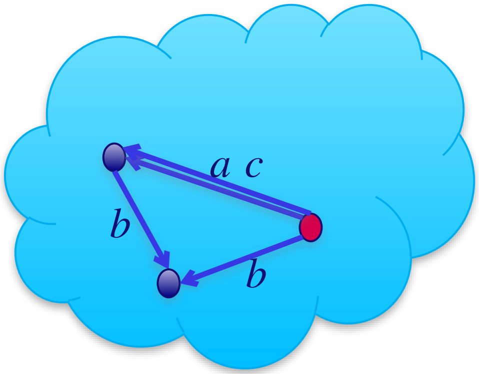




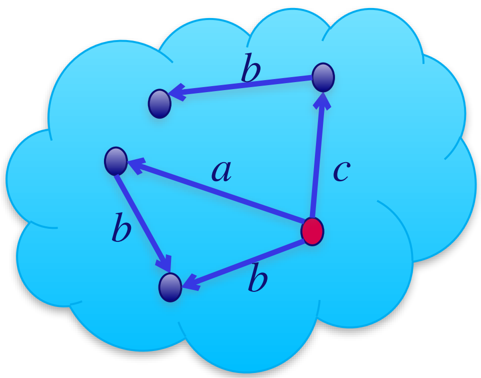
Process equivalences.



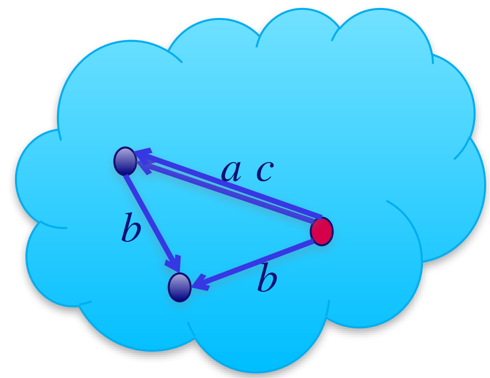
equal?



Process equivalences.

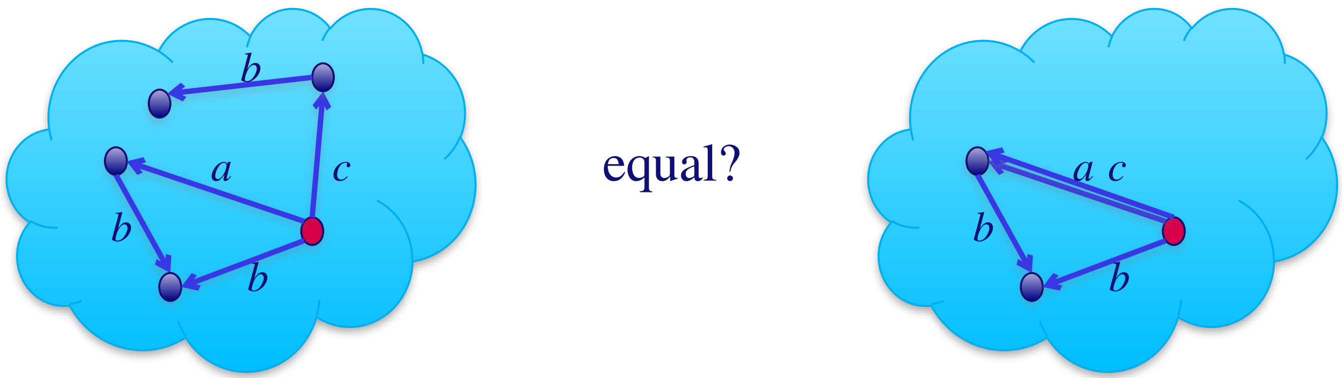


equal?



Equivalences:
Strong bisimulation
Branching bisimulation
Simulation equivalence
Simulation preorder

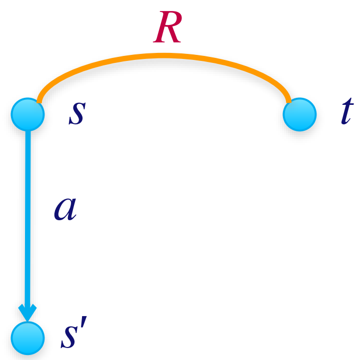
Process equivalences.



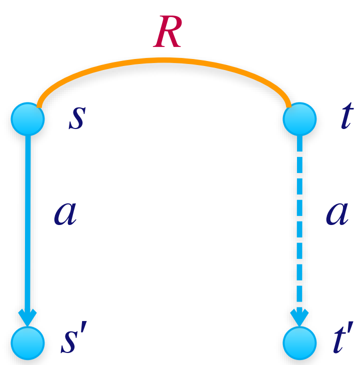
n: number of states
m: number or transitions

Equivalences:
Strong bisimulation
Branching bisimulation
Simulation equivalence
Simulation preorder

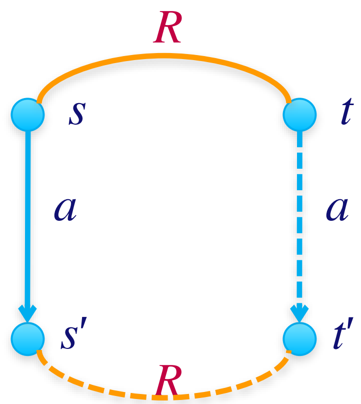
Simulation preorder and equivalence



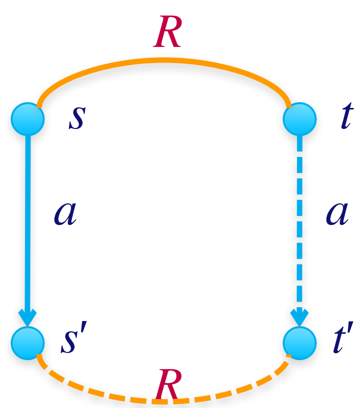
Simulation preorder and equivalence



Simulation preorder and equivalence

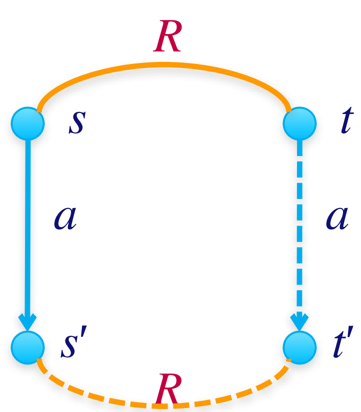


Simulation preorder and equivalence



A state $t \in S$ *simulates* a state $s \in S$, notation $s \subsetneqq t$, iff there is a simulation relation R such that sRt.

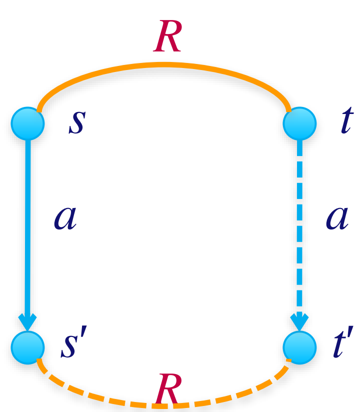
Simulation preorder and equivalence



A state $t \in S$ *simulates* a state $s \in S$, notation $s \subsetneq t$, iff there is a simulation relation R such that sRt.

Two states $s,t \in S$ are *simulation equivalent*, notation $s \sim t$, iff $s \subsetneq t$ and $t \subsetneq s$.

Simulation preorder and equivalence



A state $t \in S$ *simulates* a state $s \in S$, notation $s \subsetneq t$, iff there is a simulation relation R such that sRt.

Two states $s,t \in S$ are *simulation equivalent*, notation $s \sim t$, iff $s \subsetneq t$ and $t \subsetneq s$.

An FSA B *simulates* an FSA A, notation $A \subsetneq B$, iff the initial state of B simulates the initial state of A.

Complexities to calculate equivalences on regular transition systems.

Strong bisimulation	$O(m \log n)$	Paige/Tarjan, 1987
- in parallel	$O(n)$	Martens/Groote/Haak/Hijma/Wijs, 2022
Branching bisimulation	$O(m \log n)$	Groote/Jansen/Keiren/Wijs, 2020
- in parallel	$O(n)$	Martens/Groote/Haak/Hijma/Wijs, 2023
Simulation preorder/equivalence	$O(mn)$	Bloom/Paige, 1995, and others.
Trace based equivalences	Exponential	

Complexities to calculate equivalences on FSAs.

Strong bisimulation	$O(m \log n)$	Paige/Tarjan, 1987
- in parallel	$O(n)$	Martens/Groote/Haak/Hijma/Wijs, 2022
Optimal using partition refinement Groote/Martens 2023		
Branching bisimulation	$O(m \log n)$	Groote/Jansen/Keiren/Wijs, 2020
- in parallel	$O(n)$	Martens/Groote/Haak/Hijma/Wijs, 2023
Optimal using partition refinement Groote/Martens 2023		
Simulation preorder/equivalence	$O(mn)$	Bloom/Paige, 1995, and others
Trace based equivalences	Exponential	
Pspace hard		

Complexities to calculate equivalences on FSAs.

Strong bisimulation	$O(m \log n)$	Paige/Tarjan, 1987
- in parallel	$O(n)$	Martens/Groote/Haak/Hijma/Wijs, 2022
Optimal using partition refinement Groote/Martens 2023		
Branching bisimulation	$O(m \log n)$	Groote/Jansen/Keiren/Wijs, 2020
- in parallel	$O(n)$	Martens/Groote/Haak/Hijma/Wijs, 2023
Optimal using partition refinement Groote/Martens 2023		
Simulation preorder/equivalence		
	$O(mn)$	Bloom/Paige, 1995, and others
?????????????		
Trace based equivalences	Exponential	
Pspace hard		

(Strong) Exponential Time Hypothesis
Impagliazza & Paturi, 1999

ETH: Satisfiability for 3-CNF requires time $O(2^{\varepsilon n})$, for some $\varepsilon>0$; n the number of proposition letters

SETH: Satisfiability for CNF cannot be solved in time $O(2^{\delta n})$, for any $\delta<1$

(Strong) Exponential Time Hypothesis Impagliazza & Paturi, 1999

ETH: Satisfiability for 3-CNF requires time $O(2^{\varepsilon n})$,
for some $\varepsilon>0$; n the number of proposition letters

SETH: Satisfiability for CNF cannot be solved in time $O(2^{\ n})$,
for any <1

$$\text{ETH} \Rightarrow \text{SETH} \Rightarrow \text{P} \neq \text{NP}$$

Intersection of regular automata.

(Non-)emptiness of language intersection for FSAs.

$$L(A_1) \cap L(A_2) = \varnothing \quad ????$$

Algorithm for NEI: $O(n^2)$.

Intersection of regular automata.

(Non-)emptiness of language intersection for FSAs.

$$L(A_1) \cap L(A_2) = \varnothing \quad ????$$

Algorithm for NEI: $O(n^2)$.

THEOREM:

SETH $\Rightarrow$ NEI is not solvable in $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$

Intersection of regular automata.

(Non-)emptiness of language intersection for FSAs.

$$L(A_1) \cap L(A_2) = \varnothing \quad ????$$

Algorithm for NEI: $O(n^2)$.

THEOREM:

SETH $\Rightarrow$ NEI is not solvable in $O(n^{2-\varepsilon})$ for any $\varepsilon>0$

This even holds if A_1 and A_2 are deterministic and have only finite traces.

Intersection of regular automata.

(Non-)emptiness of language intersection for FSAs.

$$L(A_1) \cap L(A_2) = \varnothing \quad ????$$

Algorithm for NEI: $O(n^2)$.

THEOREM:

SETH $\Rightarrow$ NEI is not solvable in $O(n^{2-\varepsilon})$ for any $\varepsilon>0$

This even holds if A_1 and A_2 are deterministic and have only finite traces.

SETH $\Rightarrow$

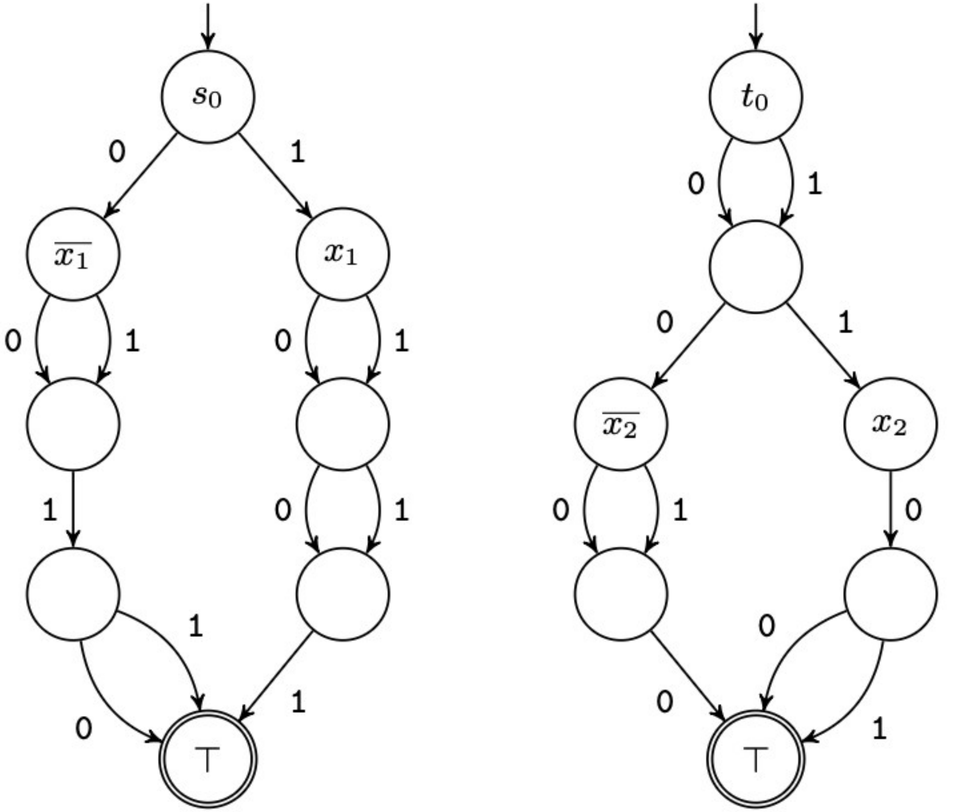
NEI for k automata is not solvable in $O(n^{k-\varepsilon})$ for any $\varepsilon>0$

Proof sketch that SETH implies NEI not solvable in $O(n^{2-\varepsilon})$

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2)$$

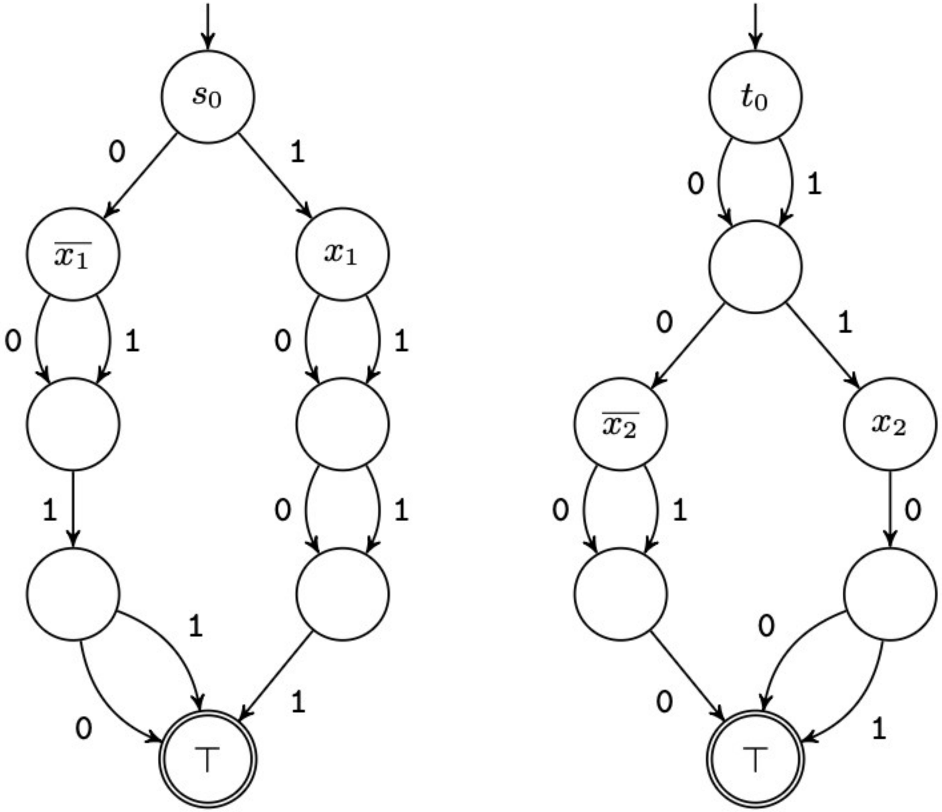
Proof sketch that SETH implies NEI not solvable in $O(n^{2-\varepsilon})$

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2)$$



Proof sketch that SETH implies NEI not solvable in $O(n^{2-\varepsilon})$

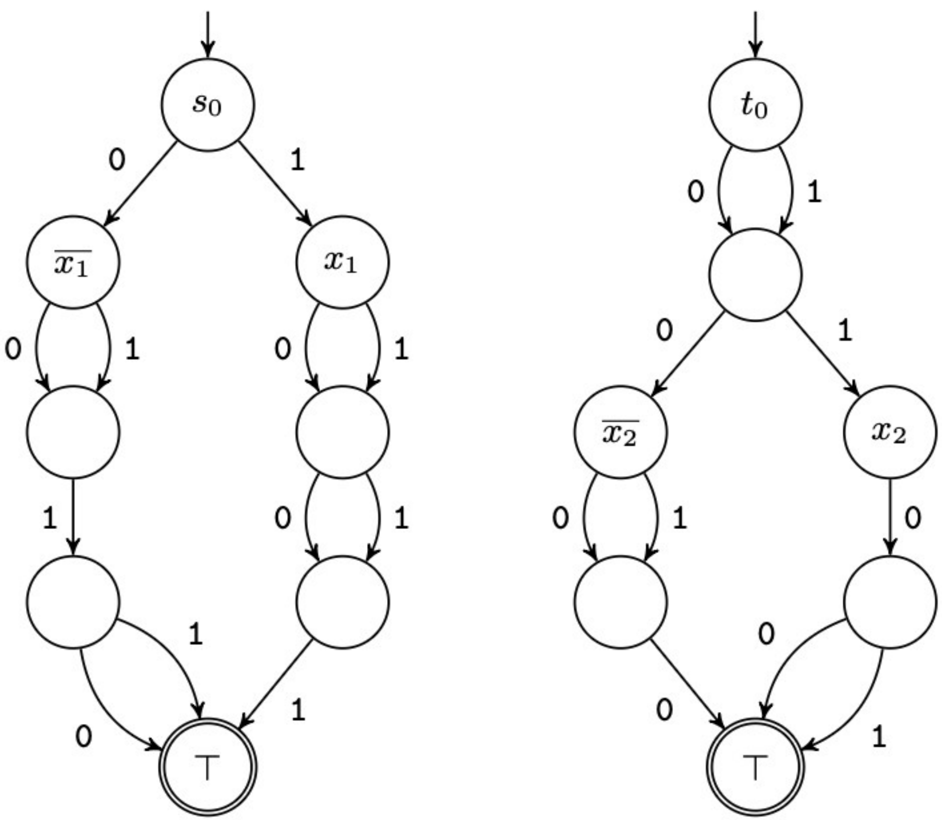
$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2)$$



x_1

Proof sketch that SETH implies NEI not solvable in $O(n^{2-\varepsilon})$

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2)$$

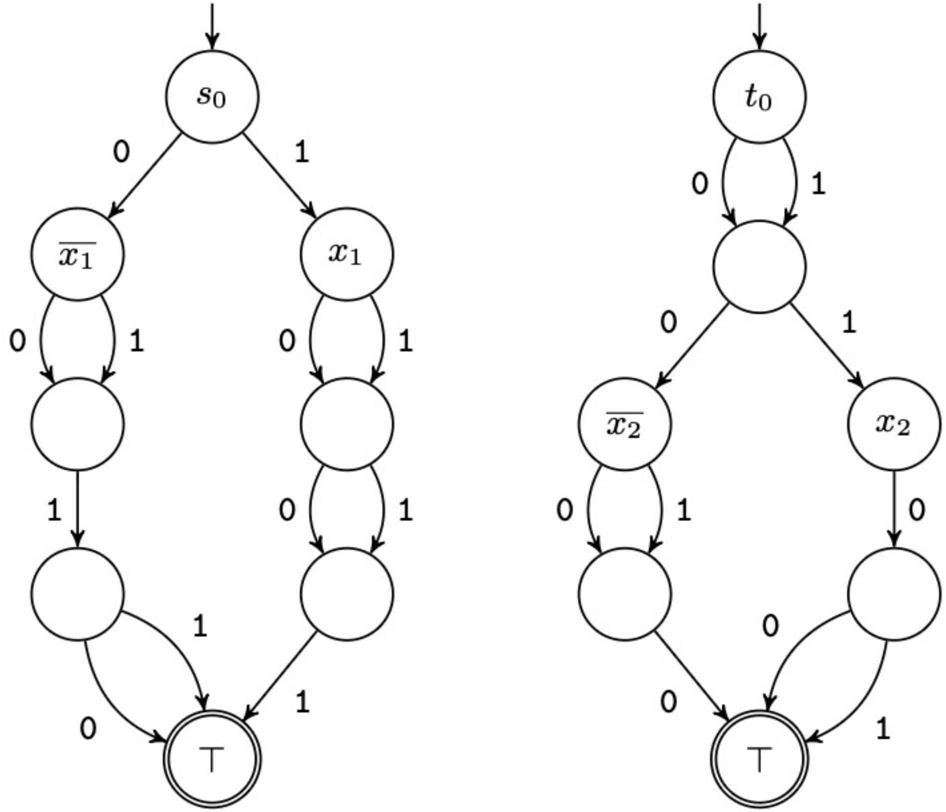


x_1

x_2

Proof sketch that SETH implies NEI not solvable in $O(n^{2-\varepsilon})$

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2)$$



x_1

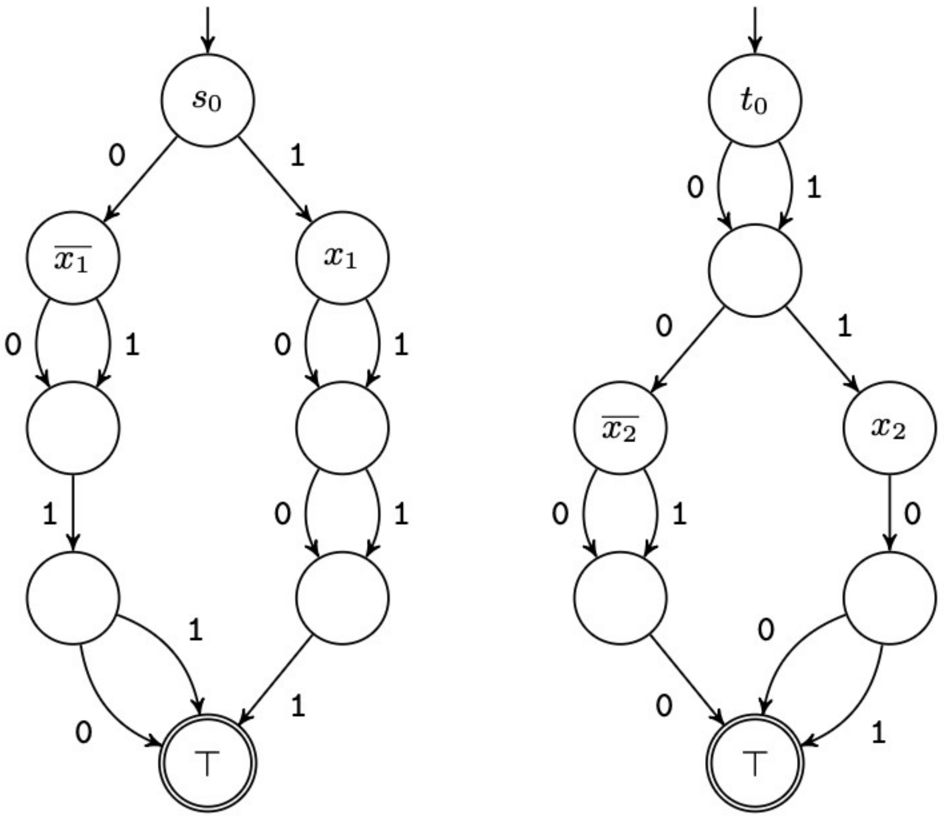
x_2

0 if x_1 satisfies first clause

1 if x_2 satisfies first clause

Proof sketch that SETH implies NEI not solvable in $O(n^{2-\varepsilon})$

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2)$$



x_1

x_2

0 if x_1 satisfies first clause
1 if x_2 satisfies first clause

idem for second clause.

Proof sketch that SETH implies NEI not solvable in $O(n^{2-\varepsilon})$

$$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2)$$

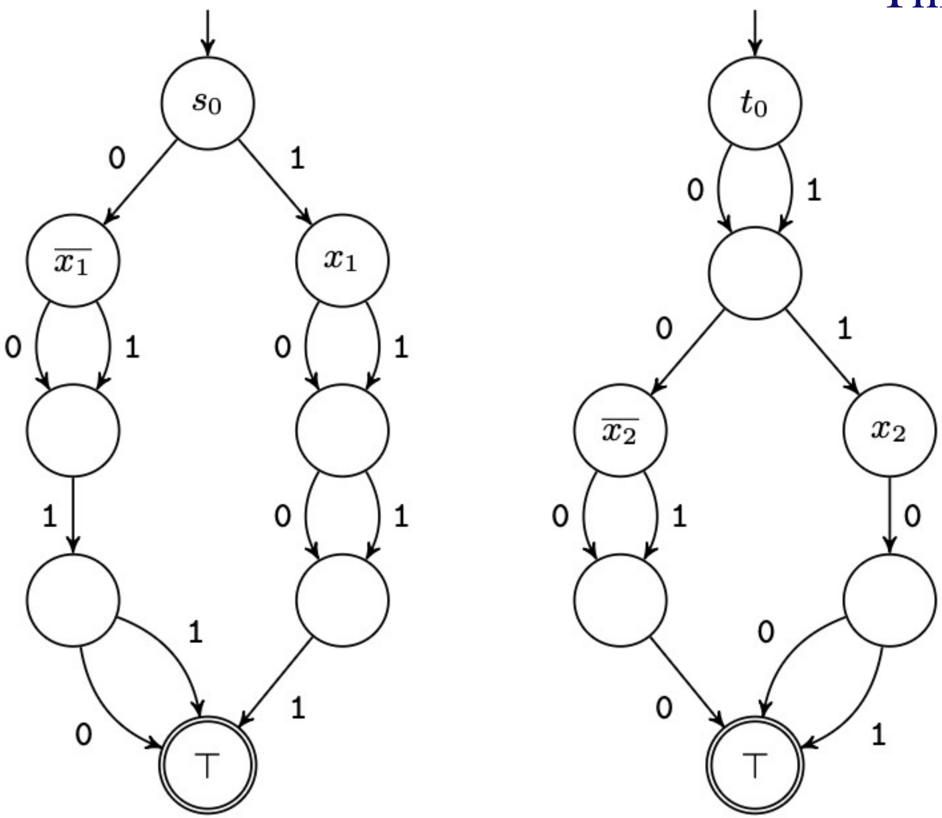
Suppose NEI is solvable in $O(N^{2-\varepsilon})$

$N = mn2^{\frac{1}{2}n}$ is the # of states of each DFA.

Then SAT is solvable in:

$O((mn2^{\frac{1}{2}n})^{2-\varepsilon}) = O(m^{2-\varepsilon}n^{2-\varepsilon}2^{n(1-\frac{1}{2}\varepsilon)})$

This contradicts SETH.



x_1

x_2

0 if x_1 satisfies first clause

1 if x_2 satisfies first clause

idem for second clause.

Back to simulation equivalence.

Lemma. Consider deterministic regular automata (with explicit termination transitions) A and B:

$$L(A) \subseteq L(B) \qquad \text{iff} \qquad A \subsetsim B$$

Back to simulation equivalence.

Lemma. Consider deterministic regular automata (with explicit termination transitions) A and B:

$$L(A) \subseteq L(B) \quad \text{iff} \quad A \subsetneqq B$$

Theorem. SETH implies similarity ($\subsetneqq$) cannot be decided in $O(n^{2-\varepsilon})$ for any $\varepsilon>0$:

Back to simulation equivalence.

Lemma. Consider deterministic regular automata (with explicit termination transitions) A and B:

$$L(A) \subseteq L(B) \quad \text{iff} \quad A \precsim B$$

Theorem. SETH implies similarity ($\precsim$) cannot be decided in $O(n^{2-\varepsilon})$ for any $\varepsilon>0$:

Proof. Deciding

$$L(A) \cap L(B) = \varnothing$$

is equivalent to deciding:

$$L(A) \subseteq \Sigma^* \backslash L(B)$$

Simulation equivalence.

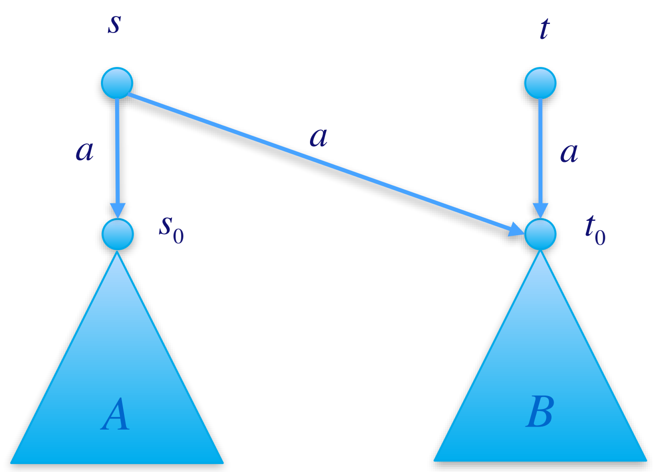
Theorem. SETH implies that simulation equivalence cannot be decided in $O(n^{2-\varepsilon})$ for any $\varepsilon>0$. FSAs need to be nondeterministic, but can have finite behaviour.

Simulation equivalence.

Theorem. SETH implies that simulation equivalence cannot be decided in $O(n^{2-\varepsilon})$ for any $\varepsilon>0$. FSAs need to be nondeterministic, but can have finite behaviour.

Proof. Consider DFAs A and B. Then

$$s_0 \subsetneq t_0 \quad \text{iff} \quad s \sim t.$$



Conclusion

Assuming SETH (which seems very reasonable)

- Calculating simulation preorder is essentially quadratic in the number of states for DFAs.
- Calculating simulation equivalence is essentially quadratic on NFAs.
- These results even hold if all traces are finite.

Conclusion

Assuming SETH (which seems very reasonable)

- Calculating simulation preorder is essentially quadratic in the number of states for DFAs.
- Calculating simulation equivalence is essentially quadratic on NFAs.
- These results even hold if all traces are finite.

Remarkably, these results cannot be generalized easily to obtain other lower bounds.

Conclusion

Assuming SETH (which seems very reasonable)

- Calculating simulation preorder is essentially quadratic in the number of states for DFAs.
- Calculating simulation equivalence is essentially quadratic on NFAs.
- These results even hold if all traces are finite.

Remarkably, these results cannot be generalized easily to obtain other lower bounds.