Automata and Communication

Georg Schnitger

Institute of Computer Science
Goethe Universität Frankfurt am Main

Outline

The Goal

Predict the minimal size of automata,
i.e., number of states or number of transitions.

Outline

The Goal

Predict the minimal size of automata,
i.e., number of states or number of transitions.

- The size of deterministic finite automata (DFA) and of Las Vegas Automata:
  Tight characterizations with Communication Complexity.

Outline

The Goal

Predict the minimal size of automata,
i.e., number of states or number of transitions.

- The size of deterministic finite automata (DFA) and of Las Vegas Automata:
  Tight characterizations with Communication Complexity.
- The size of nondeterministic finite automata. (NFA)
  - Approximation Complexity.

Outline

The Goal

Predict the minimal size of automata,
i.e., number of states or number of transitions.

- The size of deterministic finite automata (DFA) and of Las Vegas Automata:
  Tight characterizations with Communication Complexity.
- The size of nondeterministic finite automata. (NFA)
  - Approximation Complexity.
  - Multi-party communication.

Outline

The Goal

Predict the minimal size of automata,
i.e., number of states or number of transitions.

- The size of deterministic finite automata (DFA) and of Las Vegas Automata:
  Tight characterizations with Communication Complexity.
- The size of nondeterministic finite automata. (NFA)
  - Approximation Complexity.
  - Multi-party communication.
  - The size of NFA with limited ambiguity.

Outline

The Goal

Predict the minimal size of automata,
i.e., number of states or number of transitions.

- The size of deterministic finite automata (DFA) and of Las Vegas Automata:
  Tight characterizations with Communication Complexity.
- The size of nondeterministic finite automata. (NFA)
  - Approximation Complexity.
  - Multi-party communication.
  - The size of NFA with limited ambiguity.
- Two-Way automata: The size of deterministic sweeping automata and nondeterministic communication.

DFA and One-Way Communication

DFA and One-Way Communication

If A is a DFA with input $x = x_1 x_2$ and $|x_1| = |x_2|$:

DFA and One-Way Communication

If A is a DFA with input $x = x_1 x_2$ and $|x_1| = |x_2|$: Alice simulates A on prefix x_1 and sends the final state to Bob.

DFA and One-Way Communication

If A is a DFA with input $x = x_1 x_2$ and $|x_1| = |x_2|$: Alice simulates A on prefix x_1 and sends the final state to Bob. Bob completes the simulation of A on suffix x_2.

DFA and One-Way Communication

If A is a DFA with input $x = x_1x_2$ and $|x_1| = |x_2|$: Alice simulates A on prefix x_1 and sends the final state to Bob. Bob completes the simulation of A on suffix x_2.

- A DFA with q states is simulated by a one-way deterministic protocol with at most q messages.

DFA and One-Way Communication

If A is a DFA with input $x = x_1 x_2$ and $|x_1| = |x_2|$: Alice simulates A on prefix x_1 and sends the final state to Bob. Bob completes the simulation of A on suffix x_2.

- A DFA with q states is simulated by a one-way deterministic protocol with at most q messages.
- Consider the language

$$L_n = \{0^{2^n} x_1 x_2 : |x_1| = |x_2| = n, x_1 = x_2\}.$$

DFA and One-Way Communication

If A is a DFA with input $x = x_1 x_2$ and $|x_1| = |x_2|$: Alice simulates A on prefix x_1 and sends the final state to Bob. Bob completes the simulation of A on suffix x_2.

- A DFA with q states is simulated by a one-way deterministic protocol with at most q messages.
- Consider the language

$$L_n = \{0^{2n} x_1 x_2 : |x_1| = |x_2| = n, x_1 = x_2\}.$$

Two messages suffice, but a minimal DFA has at least 2^n states.

DFA and One-Way Communication

If A is a DFA with input $x = x_1 x_2$ and $|x_1| = |x_2|$: Alice simulates A on prefix x_1 and sends the final state to Bob. Bob completes the simulation of A on suffix x_2.

- A DFA with q states is simulated by a one-way deterministic protocol with at most q messages.
- Consider the language

$$L_n = \{0^{2n} x_1 x_2 : |x_1| = |x_2| = n, x_1 = x_2\}.$$

Two messages suffice, but a minimal DFA has at least 2^n states.

Is one-way communication inherently too powerful?

DFA and Uniform Communication

Require that one-way protocols are uniform:
require that the communication protocol “works” for any input partition $x = x_1 x_2$ with an arbitrary prefix x_1 and an arbitrary suffix x_2.

DFA and Uniform Communication

Require that one-way protocols are uniform:
require that the communication protocol “works” for any input partition $x = x_1 x_2$ with an arbitrary prefix x_1 and an arbitrary suffix x_2.

- The minimal number of messages equals the number of different rows in the (infinite) communication matrix.

DFA and Uniform Communication

Require that one-way protocols are uniform:
require that the communication protocol "works" for any input partition $x = x_1 x_2$ with an arbitrary prefix x_1 and an arbitrary suffix x_2.

- The minimal number of messages equals the number of different rows in the (infinite) communication matrix.
- Within the communication matrix: the rows for prefix u and prefix v coincide iff u and v are Nerode equivalent.

DFA and Uniform Communication

Require that one-way protocols are uniform:
require that the communication protocol "works" for any input partition $x = x_1 x_2$ with an arbitrary prefix x_1 and an arbitrary suffix x_2.

- The minimal number of messages equals the number of different rows in the (infinite) communication matrix.
- Within the communication matrix: the rows for prefix u and prefix v coincide iff u and v are Nerode equivalent.

DFA and Communication

The minimal number of states of a DFA for a language L coincides with the minimal number of messages of a uniform one-way protocol for L.

Las Vegas Automata and Uniform Communication

A Las Vegas automaton never errs and may output a question mark with probability at most $\frac{1}{2}$.

Las Vegas Automata and Uniform Communication

A Las Vegas automaton never errs and may output a question mark with probability at most $\frac{1}{2}$.

- If a Las Vegas one-way protocol sends at most m messages, then there is an equivalent deterministic protocol sending at most $O(m^2)$ messages (Hromkovic and S, 2001).

Las Vegas Automata and Uniform Communication

A Las Vegas automaton never errs and may output a question mark with probability at most $\frac{1}{2}$.

- If a Las Vegas one-way protocol sends at most m messages, then there is an equivalent deterministic protocol sending at most $O(m^2)$ messages (Hromkovic and S, 2001).
- The minimal number of states of a Las Vegas automaton is at most quadratic in the minimal number of states of a deterministic finite automaton.

Las Vegas Automata and Uniform Communication

A Las Vegas automaton never errs and may output a question mark with probability at most $\frac{1}{2}$.

- If a Las Vegas one-way protocol sends at most m messages, then there is an equivalent deterministic protocol sending at most $O(m^2)$ messages (Hromkovic and S, 2001).
- The minimal number of states of a Las Vegas automaton is at most quadratic in the minimal number of states of a deterministic finite automaton.

Las Vegas Automata and Communication

The minimal number of states of a Las Vegas automaton for a language L is at most quadratic in the minimal number of messages of a uniform Las Vegas protocol for L.

The Size of NFA: Approximation Complexity

Given an NFA of size n, determine the size of an equivalent minimal NFA:

The Size of NFA: Approximation Complexity

Given an NFA of size n, determine the size of an equivalent minimal NFA:

Unless P $\neq$ PSPACE, no efficient approximation algorithm reaches approximation factor $o(n)$.

The Size of NFA: Approximation Complexity

Given an NFA of size n, determine the size of an equivalent minimal NFA:

Unless P $\neq$ PSPACE, no efficient approximation algorithm reaches approximation factor $o(n)$.

Given a DFA of size n, determine the size of an equivalent minimal NFA:

The Size of NFA: Approximation Complexity

Given an NFA of size n, determine the size of an equivalent minimal NFA:

Unless P $\neq$ PSPACE, no efficient approximation algorithm reaches approximation factor $o(n)$.

Given a DFA of size n, determine the size of an equivalent minimal NFA:

If strong pseudo-random functions exist in non-uniform NC^1, then no efficient approximation algorithm reaches approximation factor $O(\sqrt{n}/\mathrm{poly}(log_2 n))$.

The Size of NFA: Approximation Complexity

Given an NFA of size n, determine the size of an equivalent minimal NFA:

Unless $P \neq PSPACE$, no efficient approximation algorithm reaches approximation factor $o(n)$.

Given a DFA of size n, determine the size of an equivalent minimal NFA:

If strong pseudo-random functions exist in non-uniform NC^1, then no efficient approximation algorithm reaches approximation factor $O(\sqrt{n}/\mathrm{poly}(log_2 n))$.
If size is measured as the number of transitions, then approximation factor $O(n/\mathrm{poly}(log_2 n))$ cannot be reached. (Gramlich and S, 2007)

The Size of NFA and Multi-Party Communication

There are languages L_n such that NFA for L_n have at least $2^{\Omega(n)}$ states. However uniform nondeterministic protocols with $O(n^2)$ messages exist.

The Size of NFA and Multi-Party Communication

There are languages L_n such that NFA for L_n have at least $2^{\Omega(n)}$ states. However uniform nondeterministic protocols with $O(n^2)$ messages exist.

Two-Party protocols fail. What about uniform multi-party protocols with k players?

The Size of NFA and Multi-Party Communication

There are languages L_n such that NFA for L_n have at least $2^{\Omega(n)}$ states. However uniform nondeterministic protocols with $O(n^2)$ messages exist.

Two-Party protocols fail. What about uniform multi-party protocols with k players?

- Partition an input $x = x_1 \cdots x_k$ into k arbitrary substrings.

The Size of NFA and Multi-Party Communication

There are languages L_n such that NFA for L_n have at least $2^{\Omega(n)}$ states. However uniform nondeterministic protocols with $O(n^2)$ messages exist.

Two-Party protocols fail. What about uniform multi-party protocols with k players?

- Partition an input $x = x_1 \cdots x_k$ into k arbitrary substrings.
- Player i receives x_i.

The Size of NFA and Multi-Party Communication

There are languages L_n such that NFA for L_n have at least $2^{\Omega(n)}$ states. However uniform nondeterministic protocols with $O(n^2)$ messages exist.

Two-Party protocols fail. What about uniform multi-party protocols with k players?

- Partition an input $x = x_1 \cdots x_k$ into k arbitrary substrings.
- Player i receives x_i. Upon receiving a message m_{i-1} from its predecessor player i sends a nondeterministic message m_i to player $i + 1$.

The Size of NFA and Multi-Party Communication

There are languages L_n such that NFA for L_n have at least $2^{\Omega(n)}$ states. However uniform nondeterministic protocols with $O(n^2)$ messages exist.

Two-Party protocols fail. What about uniform multi-party protocols with k players?

- Partition an input $x = x_1 \cdots x_k$ into k arbitrary substrings.
- Player i receives x_i. Upon receiving a message m_{i-1} from its predecessor player i sends a nondeterministic message m_i to player $i+1$.
- Player k decides.

The Size of NFA and Multi-Party Communication

There are languages L_n such that NFA for L_n have at least $2^{\Omega(n)}$ states. However uniform nondeterministic protocols with $O(n^2)$ messages exist.

Two-Party protocols fail. What about uniform multi-party protocols with k players?

- Partition an input $x = x_1 \cdots x_k$ into k arbitrary substrings.
- Player i receives x_i. Upon receiving a message m_{i-1} from its predecessor player i sends a nondeterministic message m_i to player $i + 1$.
- Player k decides.
- The communication complexity of a protocol is the maximum, over all players i, of

The Size of NFA and Multi-Party Communication

There are languages L_n such that NFA for L_n have at least $2^{\Omega(n)}$ states. However uniform nondeterministic protocols with $O(n^2)$ messages exist.

Two-Party protocols fail. What about uniform multi-party protocols with k players?

- Partition an input $x = x_1 \cdots x_k$ into k arbitrary substrings.
- Player i receives x_i. Upon receiving a message m_{i-1} from its predecessor player i sends a nondeterministic message m_i to player $i + 1$.
- Player k decides.
- The communication complexity of a protocol is the maximum, over all players i, of the number of different messages m_i sent by player i.

How many players have to be considered?

Assume that a language L has a minimal NFA with q states. Do multi-party protocols with, say, q players have communication complexity $\Omega(q)$?

How many players have to be considered?

Assume that a language L has a minimal NFA with q states. Do multi-party protocols with, say, q players have communication complexity $\Omega(q)$?

Consider the unary language $L_n = \{1^m : m \neq n\}$.

- The minimal NFA for L_n has $\Theta(\sqrt{n})$ states.

How many players have to be considered?

Assume that a language L has a minimal NFA with q states. Do multi-party protocols with, say, q players have communication complexity $\Omega(q)$?

Consider the unary language $L_n = \{1^m : m \neq n\}$.

- The minimal NFA for L_n has $\Theta(\sqrt{n})$ states.
- There are uniform k-party protocols for L_n exchanging at most $\log_2^2(kn)$ messages.

How many players have to be considered?

Assume that a language L has a minimal NFA with q states. Do multi-party protocols with, say, q players have communication complexity $\Omega(q)$?

Consider the unary language $L_n = \{1^m : m \neq n\}$.

- The minimal NFA for L_n has $\Theta(\sqrt{n})$ states.
- There are uniform k-party protocols for L_n exchanging at most $\log_2^2(kn)$ messages.
- Even for $k = 2^{\Omega(n^{1/4})}$ players, communication complexity is smaller than the number of states.

How many players have to be considered?

Assume that a language L has a minimal NFA with q states. Do multi-party protocols with, say, q players have communication complexity $\Omega(q)$?

Consider the unary language $L_n = \{1^m : m \neq n\}$.

- The minimal NFA for L_n has $\Theta(\sqrt{n})$ states.
- There are uniform k-party protocols for L_n exchanging at most $\log_2^2(kn)$ messages.
- Even for $k = 2^{\Omega(n^{1/4})}$ players, communication complexity is smaller than the number of states. (Hromkovic and S, 2008)

How to predict the size of NFA?

How many players have to be considered?

Assume that a language L has a minimal NFA with q states. Do multi-party protocols with, say, q players have communication complexity $\Omega(q)$?

Consider the unary language $L_n = \{1^m : m \neq n\}$.

- The minimal NFA for L_n has $\Theta(\sqrt{n})$ states.
- There are uniform k-party protocols for L_n exchanging at most $\log_2^2(kn)$ messages.
- Even for $k = 2^{\Omega(n^{1/4})}$ players, communication complexity is smaller than the number of states. (Hromkovic and S, 2008)

How to predict the size of NFA? No idea!

THE Open Problem for Two-Way Automata

2-DFA versus 2-NFA

Are there languages L_n with two-way NFA of size $O(n)$ such that any two-way DFA for L_n requires more than poly(n) states?

THE Open Problem for Two-Way Automata

2-DFA versus 2-NFA

Are there languages L_n with two-way NFA of size $O(n)$ such that any two-way DFA for L_n requires more than poly(n) states?

- Communication fails, since automata with q states seem to require simulating protocols with $q^{\Omega(q)}$ messages:

THE Open Problem for Two-Way Automata

2-DFA versus 2-NFA

Are there languages L_n with two-way NFA of size $O(n)$ such that any two-way DFA for L_n requires more than poly(n) states?

- Communication fails, since automata with q states seem to require simulating protocols with $q^{\Omega(q)}$ messages: crossing sequences may be of length $\Omega(q)$.

THE Open Problem for Two-Way Automata

2-DFA versus 2-NFA

Are there languages L_n with two-way NFA of size $O(n)$ such that any two-way DFA for L_n requires more than poly(n) states?

- Communication fails, since automata with q states seem to require simulating protocols with $q^{\Omega(q)}$ messages: crossing sequences may be of length $\Omega(q)$.
- A rotating automaton scans its input from left to right. After reaching the right end of the input it stops or starts a new left-to-right sweep.

THE Open Problem for Two-Way Automata

2-DFA versus 2-NFA

Are there languages L_n with two-way NFA of size $O(n)$ such that any two-way DFA for L_n requires more than poly(n) states?

- Communication fails, since automata with q states seem to require simulating protocols with $q^{\Omega(q)}$ messages: crossing sequences may be of length $\Omega(q)$.
- A rotating automaton scans its input from left to right. After reaching the right end of the input it stops or starts a new left-to-right sweep.

If uniform nondeterministic protocols for the complement of L require at least s messages, then any deterministic rotating automaton for $(L\$)^*$ has to have at least $\Omega(\sqrt{s})$ states (Hromkovic and S, 2008).

And Another Open Problem

- Given is an arbitrary subset $S \subseteq \{1, \ldots, n\}$.

And Another Open Problem

- Given is an arbitrary subset $S \subseteq \{1, \ldots, n\}$.
- Alice receives $x \in \{1, \ldots, n\}$, Bob receives $y \in \{1, \ldots, n\}$.

And Another Open Problem

- Given is an arbitrary subset $S \subseteq \{1, \ldots, n\}$.
- Alice receives $x \in \{1, \ldots, n\}$, Bob receives $y \in \{1, \ldots, n\}$.
- They have to determine whether $x + y \in S$ by using a probabilistic, bounded error one-way protocol.

And Another Open Problem

- Given is an arbitrary subset $S \subseteq \{1, \ldots, n\}$.
- Alice receives $x \in \{1, \ldots, n\}$, Bob receives $y \in \{1, \ldots, n\}$.
- They have to determine whether $x + y \in S$ by using a probabilistic, bounded error one-way protocol.

If at least $\log_2 n \cdot |S|^{\Omega(1)}$ messages are required for **any** set S, then

And Another Open Problem

- Given is an arbitrary subset $S \subseteq \{1, \ldots, n\}$.
- Alice receives $x \in \{1, \ldots, n\}$, Bob receives $y \in \{1, \ldots, n\}$.
- They have to determine whether $x + y \in S$ by using a probabilistic, bounded error one-way protocol.

If at least $\log_2 n \cdot |S|^{\Omega(1)}$ messages are required for **any** set S, then unary bounded-error automata have a **normal form** which is optimal up to a polynomial.