Advanced Logic

http://www-sop.inria.fr/members/Martin.Avanzini/teaching/2021/AL/

Martin Avanzini

Summer Semester 2021

Last Lecture

Presburger Arithmetic refers to the first-order theory over $(\mathbb{N}, \{0, +, <\})$

$$s, t ::= 0 \mid x \mid s + t$$

$$\phi, \psi ::= \top \mid \bot \mid s = t \mid s < t \mid \phi \wedge \psi \mid \neg\psi \mid \exists x.\phi$$

Theorem

Satisfiability and Validity are decidable for Presburger Arithmetic.

Theorem

For any formula ϕ, the constructed DFA recognizing $\hat{L}(\phi)$ has size $O(2^{2^n})$.

- ⋆ this bound can be reached

Today's Lecture

- non-determinism
- alternative finite automata
- relationship with regular languages

Non-Determinism

Angelican vs Demonic Non-Determinism

What is a non-deterministic machine?

- ⋆ a machine which admits several executions on the same input
- ⋆ put otherwise, during processing, several choices are possible

Angelican vs Demonic Non-Determinism

What is a non-deterministic machine?

- a machine which admits several executions on the same input
- put otherwise, during processing, several choices are possible
- such choices can be resolved in favor (anglican non-determinism) or against (demonic non-determinism) a positive outcome (e.g. acceptance, termination, etc)
  - Anglican: an angel resolves choices

    $\Longrightarrow$ it is sufficient to have one "good" execution path, to have a positive outcome

Angelican vs Demonic Non-Determinism

What is a non-deterministic machine?

- ⋆ a machine which admits several executions on the same input
- ⋆ put otherwise, during processing, several choices are possible
- ⋆ such choices can be resolved in favor (anglican non-determinism) or against (demonic non-determinism) a positive outcome (e.g. acceptance, termination, etc)
  - Anglican: an angel resolves choices

    $\Longrightarrow$ it is sufficient to have one "good" execution path, to have a positive outcome
  - Demonic: a demon resolves choices

    $\Longrightarrow$ all execution paths must be good "good", to have a positive outcome

Angelican vs Demonic Non-Determinism

What is a non-deterministic machine?

- ⋆ a machine which admits several executions on the same input
- ⋆ put otherwise, during processing, several choices are possible
- ⋆ such choices can be resolved in favor (anglican non-determinism) or against (demonic non-determinism) a positive outcome (e.g. acceptance, termination, etc)
  - Anglican: an angel resolves choices

    $\Longrightarrow$ it is sufficient to have one “good” execution path, to have a positive outcome
  - Demonic: a demon resolves choices

    $\Longrightarrow$ all execution paths must be good “good”, to have a positive outcome

Example

- ⋆ NFAs are based on anglican non-determinism
- ⋆ worst-case complexity analysis assumes demonic non-determinism

NFAs with Demonic Choice

- NFAs incorporate angelic non-determinism because, in order for $w \in L(\mathcal{A})$, only one accepting run of w has to exists

NFAs with Demonic Choice

- NFAs incorporate angelic non-determinism because, in order for $w \in \mathsf{L}(\mathcal{A})$, only one accepting run of w has to exists

- demonic non-determinism introduced by re-formulating the acceptance condition

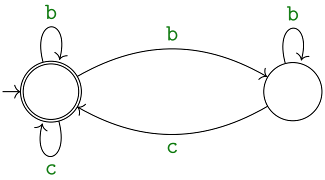
$$\mathsf{L}^{-}(\mathcal{A}) \triangleq \{w \mid \text{all runs on } w \text{ are accepting}\}$$

NFAs with Demonic Choice

- ⋆ NFAs incorporate angelic non-determinism because, in order for $w \in L(\mathcal{A})$, only one accepting run of w has to exists
- ⋆ demonic non-determinism introduced by re-formulating the acceptance condition

$$L^{-}(\mathcal{A}) \triangleq \{w \mid \text{all runs on } w \text{ are accepting}\}$$

Example



- ⋆ $L(\mathcal{A}) = (\mathtt{b} \cup \mathtt{c})^*$
- ⋆ $L^{-}(\mathcal{A}) = \epsilon \cup (\mathtt{b} \cup \mathtt{c})^* \cdot \mathtt{c}$

Duality of Non-Determinism

- recall that for each NFA $\mathcal{A}$, its dual $\overline{\mathcal{A}}$ is given by complementing final states
- in general, only when $\mathcal{A}$ is deterministic, then $\mathsf{L}(\overline{\mathcal{A}}) = \overline{\mathsf{L}(\mathcal{A})}$

Duality of Non-Determinism

- recall that for each NFA $\mathcal{A}$, its dual $\overline{\mathcal{A}}$ is given by complementing final states
- in general, only when $\mathcal{A}$ is deterministic, then $\mathsf{L}(\overline{\mathcal{A}}) = \overline{\mathsf{L}(\mathcal{A})}$

Proposition

$$w \in \mathsf{L}(\mathcal{A}) \quad \iff \quad w \notin \mathsf{L}^{-}(\overline{\mathcal{A}})$$

Duality of Non-Determinism

- recall that for each NFA $\mathcal{A}$, its dual $\overline{\mathcal{A}}$ is given by complementing final states
- in general, only when $\mathcal{A}$ is deterministic, then $\mathsf{L}(\overline{\mathcal{A}}) = \overline{\mathsf{L}(\mathcal{A})}$

Proposition

$$w \in \mathsf{L}(\mathcal{A}) \quad \Longleftrightarrow \quad w \notin \mathsf{L}^{-}(\overline{\mathcal{A}})$$

- regime to resolve non-determinism has no effect on expressiveness of NFAs
- although potentially on the conciseness of the language description through NFAs

Duality of Non-Determinism

- recall that for each NFA $\mathcal{A}$, its dual $\overline{\mathcal{A}}$ is given by complementing final states
- in general, only when $\mathcal{A}$ is deterministic, then $L(\overline{\mathcal{A}}) = \overline{L(\mathcal{A})}$

Proposition

$$w \in L(\mathcal{A}) \quad \iff \quad w \notin L^{-}(\overline{\mathcal{A}})$$

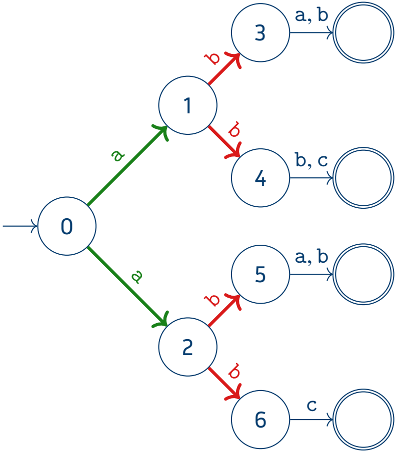
- regime to resolve non-determinism has no effect on expressiveness of NFAs
- although potentially on the conciseness of the language description through NFAs

what happens if we leave regime internal to the automata?

Alternating Finite Automata

Alternating Finite Automata

- General Idea: mix Anglican an Demonic choice on the level of individual transitions
  - a player resolves Anglican choice
  - an oppenent resolves Demonic choice



$$\delta(0, a) = 1 \vee 2$$
$$\delta(1, b) = 3 \wedge 4$$
$$\delta(2, b) = 5 \wedge 6$$
$$\vdots$$

$$\begin{aligned} L(\mathcal{A}) &= a\big(b(a \cup b) \cap b(b \cup c)\big) \\ &\quad \cup a\big(b(a \cup b) \cap bc\big) \\ &= abb \cup \varnothing \\ &= abb \end{aligned}$$

Alternating Finite Automata, Formally

Positive Boolean Formulas

- let $A = \{a, b, \dots\}$ be a set of atoms
- the positive Boolean formulas $\mathbb{B}^+(A)$ over atoms A are given by the following grammar:

$$\phi, \psi ::= a \;\Big|\; \phi \wedge \psi \;\Big|\; \phi \vee \psi$$

  - such formulas are called positive because negation is missing

Alternating Finite Automata, Formally

Positive Boolean Formulas

- let $A = \{a, b, \dots\}$ be a set of atoms
- the positive Boolean formulas $\mathbb{B}^+(A)$ over atoms A are given by the following grammar:

$$\phi, \psi ::= a \;\Big|\; \phi \wedge \psi \;\Big|\; \phi \vee \psi$$

  - such formulas are called positive because negation is missing

- a set $M \subseteq A$ is a model of ϕ if $M \vDash \phi$ where

$$M \vDash a :\Longleftrightarrow a \in M \qquad M \vDash \phi \wedge \psi :\Longleftrightarrow M \vDash \phi \textbf{ and } M \vDash \psi \qquad M \vDash \phi \vee \psi :\Longleftrightarrow M \vDash \phi \textbf{ or } M \vDash \psi$$

Alternating Finite Automata, Formally

Positive Boolean Formulas

- let $A = \{a, b, \dots\}$ be a set of atoms
- the positive Boolean formulas $\mathbb{B}^+(A)$ over atoms A are given by the following grammar:

$$\phi, \psi ::= a \mid \phi \wedge \psi \mid \phi \vee \psi$$

  - such formulas are called positive because negation is missing

- a set $M \subseteq A$ is a model of ϕ if $M \models \phi$ where

$$M \models a :\Longleftrightarrow a \in M \qquad M \models \phi \wedge \psi :\Longleftrightarrow M \models \phi \textbf{ and } M \models \psi \qquad M \models \phi \vee \psi :\Longleftrightarrow M \models \phi \textbf{ or } M \models \psi$$

Example

consider $\phi = a \wedge (b \vee c)$, then

$$\{a, b\} \models \phi \qquad \{a, c\} \models \phi \qquad \{a\} \not\models \phi \qquad \{b, c\} \not\models \phi$$

Alternating Finite Automata, Formally (II)

an alternating finite automata (AFA) is a tuple $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$ where all components are identical to an NFA except that

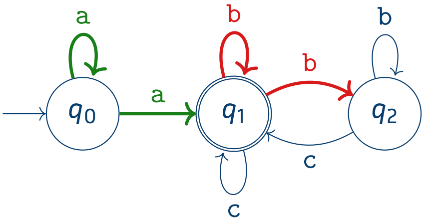
$$\delta : Q \times \Sigma \to \mathbb{B}^+(Q)$$

Alternating Finite Automata, Formally (II)

an alternating finite automata (AFA) is a tuple $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$ where all components are identical to an NFA except that

$$\delta : Q \times \Sigma \to \mathbb{B}^+(Q)$$

Example



δ	a	b	c
q_0	$q_0 \vee q_1$	$q_\perp$	$q_\perp$
q_1	$q_\perp$	$q_1 \wedge q_2$	q_1
q_2	$q_\perp$	q_2	q_1
$q_\perp$	$q_\perp$	$q_\perp$	$q_\perp$

Runs in an AFA

let $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$ be an AFA

- an execution for a word $w = \mathtt{a}_1 \ldots \mathtt{a}_n \in \Sigma^*$ is a tree T_w whose nodes are labeled by states Q s.t.:
  1. the root node of T_w is labeled by the initial state q_I
  2. for all nodes v on the ith layer ($i = 0, \ldots, n-1$) with successors $v_1, \ldots, v_k$ on layer $i+1$, labeled by $q_1, \ldots, q_k$, respectively:

$$\{q_1, \ldots, q_k\} \models \delta(q, \mathtt{a}_{\mathtt{i+1}})$$

Runs in an AFA

let $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$ be an AFA

- ⋆ an execution for a word $w = \mathtt{a}_1 \dots \mathtt{a}_n \in \Sigma^*$ is a tree T_w whose nodes are labeled by states Q s.t.:
  1. the root node of T_w is labeled by the initial state q_I
  2. for all nodes v on the ith layer ($i = 0, \dots, n-1$) with successors $v_1, \dots, v_k$ on layer $i+1$, labeled by $q_1, \dots, q_k$, respectively:

$$\{q_1, \dots, q_k\} \models \delta(q, \mathtt{a}_{i+1})$$

- ⋆ an execution is accepting if all leafs are labeled by final states

Runs in an AFA

let $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$ be an AFA

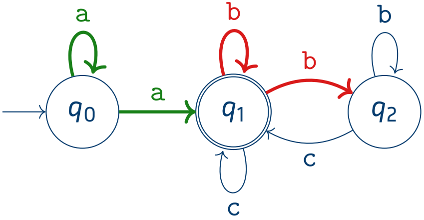
- ⋆ an execution for a word $w = \mathtt{a}_1 \dots \mathtt{a}_n \in \Sigma^*$ is a tree T_w whose nodes are labeled by states Q s.t.:
  1. the root node of T_w is labeled by the initial state q_I
  2. for all nodes v on the ith layer ($i = 0, \dots, n-1$) with successors $v_1, \dots, v_k$ on layer $i+1$, labeled by $q_1, \dots, q_k$, respectively:

$$\{q_1, \dots, q_k\} \models \delta(q, \mathtt{a}_{i+1})$$

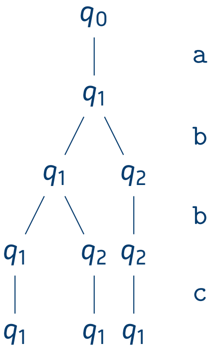
- ⋆ an execution is accepting if all leafs are labeled by final states
- ⋆ the language recognized by $\mathcal{A}$ is given by

$$L(\mathcal{A}) \triangleq \{w \mid \text{there exists an accepting execution } T_w \text{ for } w\}$$

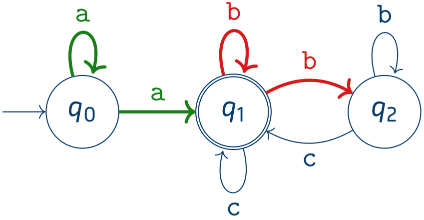
Example of Accepting Execution for w = abbc



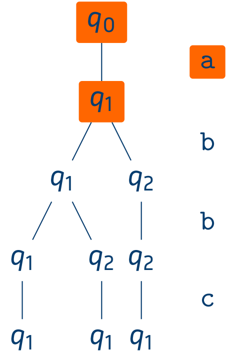
δ	a	b	c
q_0	$q_0 \vee q_1$	$q_\perp$	$q_\perp$
q_1	$q_\perp$	$q_1 \wedge q_2$	q_1
q_2	$q_\perp$	q_2	q_1
$q_\perp$	$q_\perp$	$q_\perp$	$q_\perp$



Example of Accepting Execution for w = abbc

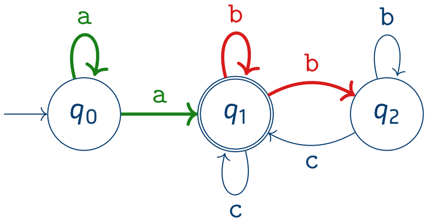


δ	a	b	c
q_0	$q_0 \vee q_1$	$q_\perp$	$q_\perp$
q_1	$q_\perp$	$q_1 \wedge q_2$	q_1
q_2	$q_\perp$	q_2	q_1
$q_\perp$	$q_\perp$	$q_\perp$	$q_\perp$

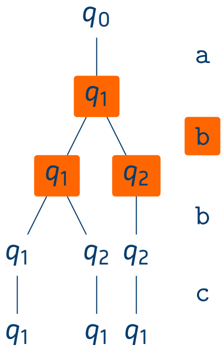


$\{q_1\} \models q_0 \vee q_1$

Example of Accepting Execution for w = abbc

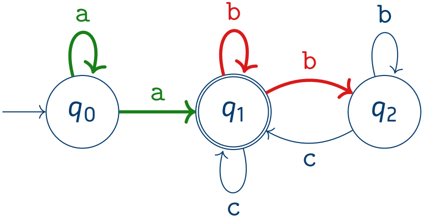


δ	a	b	c
q_0	$q_0 \vee q_1$	$q_\perp$	$q_\perp$
q_1	$q_\perp$	$q_1 \wedge q_2$	q_1
q_2	$q_\perp$	q_2	q_1
$q_\perp$	$q_\perp$	$q_\perp$	$q_\perp$

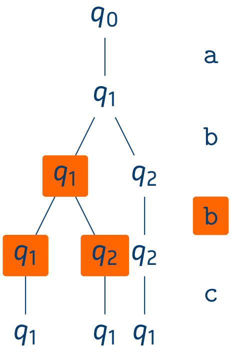


$\{q_1, q_2\} \models q_1 \wedge q_2$

Example of Accepting Execution for w = abbc

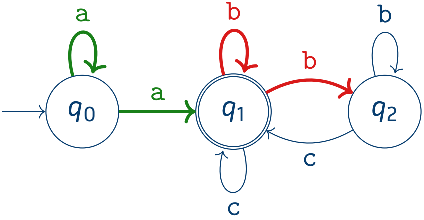


δ	a	b	c
q_0	$q_0 \vee q_1$	$q_\perp$	$q_\perp$
q_1	$q_\perp$	$q_1 \wedge q_2$	q_1
q_2	$q_\perp$	q_2	q_1
$q_\perp$	$q_\perp$	$q_\perp$	$q_\perp$

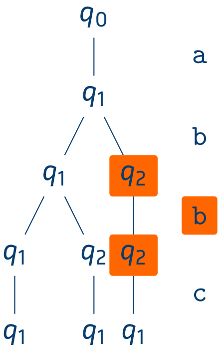


$\{q_1, q_2\} \models q_1 \wedge q_2$

Example of Accepting Execution for w = abbc

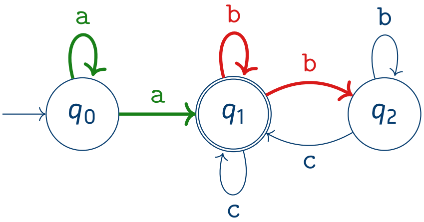


δ	a	b	c
q_0	$q_0 \vee q_1$	$q_\perp$	$q_\perp$
q_1	$q_\perp$	$q_1 \wedge q_2$	q_1
q_2	$q_\perp$	$\boxed{q_2}$	q_1
$q_\perp$	$q_\perp$	$q_\perp$	$q_\perp$

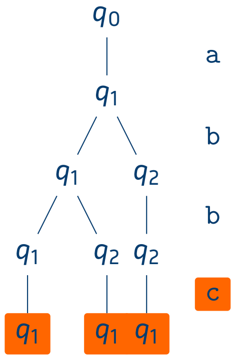


$\boxed{\{q_2\} \vDash q_2}$

Example of Accepting Execution for $w = \mathtt{abbc}$

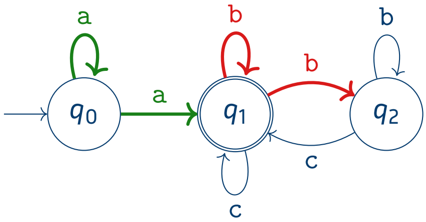


δ	a	b	c
q_0	$q_0 \vee q_1$	$q_\perp$	$q_\perp$
q_1	$q_\perp$	$q_1 \wedge q_2$	$\boxed{q_1}$
q_2	$q_\perp$	q_2	$\boxed{q_1}$
$q_\perp$	$q_\perp$	$q_\perp$	$q_\perp$

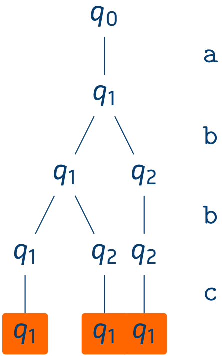


$\boxed{\{q_1\} \models q_1}$

Example of Accepting Execution for w = abbc



δ	a	b	c
q_0	$q_0 \vee q_1$	$q_\perp$	$q_\perp$
q_1	$q_\perp$	$q_1 \wedge q_2$	q_1
q_2	$q_\perp$	q_2	q_1
$q_\perp$	$q_\perp$	$q_\perp$	$q_\perp$



$\{q_1, q_1, q_1\} \subseteq F$

Extended Transition Function

the extended transition function

$$\hat{\delta} : \mathbb{B}^+(Q) \times \Sigma^* \to \mathbb{B}^+(Q)$$

is recursively defined by:

$$\hat{\delta}(q, \epsilon) \triangleq q \qquad \hat{\delta}(\phi \vee \psi, w) = \hat{\delta}(\phi, w) \vee \hat{\delta}(\psi, w)$$

$$\hat{\delta}(q, \mathtt{a} \cdot w) \triangleq \hat{\delta}(\delta(q, \mathtt{a}), w) \qquad \hat{\delta}(\phi \wedge \psi, w) = \hat{\delta}(\phi, w) \wedge \hat{\delta}(\psi, w)$$

Extended Transition Function

the extended transition function

$$\hat{\delta} : \mathbb{B}^+(Q) \times \Sigma^* \to \mathbb{B}^+(Q)$$

is recursively defined by:

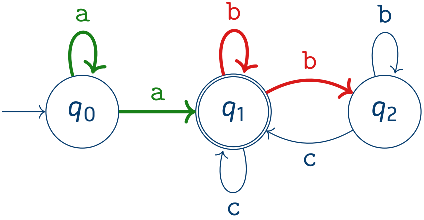
$$\hat{\delta}(q, \epsilon) \triangleq q$$
$$\hat{\delta}(q, \mathrm{a} \cdot w) \triangleq \hat{\delta}(\delta(q, \mathrm{a}), w)$$

$$\hat{\delta}(\phi \vee \psi, w) = \hat{\delta}(\phi, w) \vee \hat{\delta}(\psi, w)$$
$$\hat{\delta}(\phi \wedge \psi, w) = \hat{\delta}(\phi, w) \wedge \hat{\delta}(\psi, w)$$

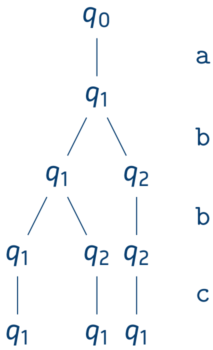
Lemma

$$\mathrm{L}(\mathcal{A}) = \{w \mid F \models \hat{\delta}(q_I, w)\}$$

Example of Accepting Execution for w = abbc (II)



δ	a	b	c
q_0	$q_0 \vee q_1$	$q_\perp$	$q_\perp$
q_1	$q_\perp$	$q_1 \wedge q_2$	q_1
q_2	$q_\perp$	q_2	q_1
$q_\perp$	$q_\perp$	$q_\perp$	$q_\perp$



$$
\begin{aligned}
\hat{\delta}(q_0, \mathtt{abbc}) &= \hat{\delta}(q_0 \vee q_1, \mathtt{bbc}) \\
&= \hat{\delta}(q_0, \mathtt{bbc}) \vee \hat{\delta}(q_1, \mathtt{bbc}) \\
&= \hat{\delta}(q_\perp, \mathtt{bc}) \vee (\hat{\delta}(q_1, \mathtt{bc}) \wedge \hat{\delta}(q_2, \mathtt{bc})) \\
&= \hat{\delta}(q_\perp, \mathtt{c}) \vee (\hat{\delta}(q_1, \mathtt{c}) \wedge \hat{\delta}(q_2, \mathtt{c})) \\
&= \hat{\delta}(q_\perp, \epsilon) \vee \hat{\delta}(q_1, \epsilon) \\
&= q_\perp \vee q_1
\end{aligned}
$$

$$\{q_1\} \models q_\perp \vee q_1$$

Comparison to NFAs and DFAs

- AFAs generalise NFAs
  - every DFA is a NFA is an AFA

Comparison to NFAs and DFAs

- ⋆ AFAs generalise NFAs
  - every DFA is a NFA is an AFA
- ⋆ AFAs allow often more succinct encoding / automata constructions

Comparison to NFAs and DFAs

- AFAs generalise NFAs
  - every DFA is a NFA is an AFA
- AFAs allow often more succinct encoding / automata constructions

Example

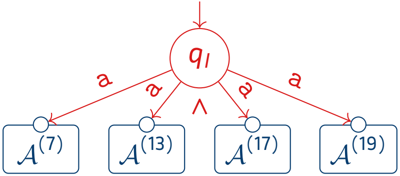
- let $\mathcal{A}^{(m)}m = (Q^{(m)}, \{\mathrm{a}\}, \delta^{(m)}, q_I^{(m)}, F^{(m)})$ be an NFA such that $\mathrm{L}(\mathcal{A}_i) = \{w \mid |w| = 0 \bmod m\}$
  - this NFA has at least m states

Comparison to NFAs and DFAs

- AFAs generalise NFAs
  - every DFA is a NFA is an AFA
- AFAs allow often more succinct encoding / automata constructions

Example

- let $\mathcal{A}^{(m)}m = (Q^{(m)}, \{a\}, \delta^{(m)}, q_I^{(m)}, F^{(m)})$ be an NFA such that $L(\mathcal{A}_i) = \{w \mid |w| = 0 \bmod m\}$
  - this NFA has at least m states
- consider the AFA $\mathcal{A}$ defined from $\mathcal{A}^{(m)}$ for primes $m = 7, 13, 17, 19$ by

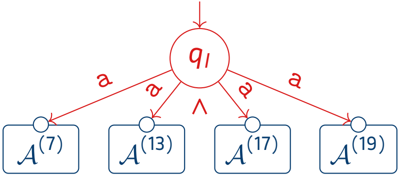


Comparison to NFAs and DFAs

- AFAs generalise NFAs
  - every DFA is a NFA is an AFA
- AFAs allow often more succinct encoding / automata constructions

Example

- let $\mathcal{A}^{(m)}m = (Q^{(m)}, \{\mathrm{a}\}, \delta^{(m)}, q_I^{(m)}, F^{(m)})$ be an NFA such that $\mathrm{L}(\mathcal{A}_i) = \{w \mid |w| = 0 \bmod m\}$
  - this NFA has at least m states
- consider the AFA $\mathcal{A}$ defined from $\mathcal{A}^{(m)}$ for primes $m = 7, 13, 17, 19$ by



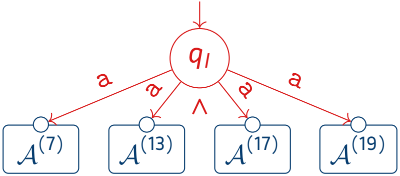
  - $\mathrm{L}(\mathcal{A}) = \{w \mid |w| = 1 \bmod 29393\}$ since $29393 = 7 \cdot 13 \cdot 17 \cdot 19$

Comparison to NFAs and DFAs

- AFAs generalise NFAs
  - every DFA is a NFA is an AFA
- AFAs allow often more succinct encoding / automata constructions

Example

- let $\mathcal{A}^{(m)}m = (Q^{(m)}, \{\mathrm{a}\}, \delta^{(m)}, q_I^{(m)}, F^{(m)})$ be an NFA such that $\mathrm{L}(\mathcal{A}_i) = \{w \mid |w| = 0 \bmod m\}$
  - this NFA has at least m states
- consider the AFA $\mathcal{A}$ defined from $\mathcal{A}^{(m)}$ for primes $m = 7, 13, 17, 19$ by



  - $\mathrm{L}(\mathcal{A}) = \{w \mid |w| = 1 \bmod 29393\}$ since $29393 = 7 \cdot 13 \cdot 17 \cdot 19$
  - AFA $\mathcal{A}$ has $57 = 1 + 7 + 13 + 17 + 19$, whereas a corresponding NFA needs 29393 states

Complementation

- recall: NFA-complementation may blow-up automata sizes by an exponential

Lemma

For every AFA $\mathcal{A}$ there exists an AFA $\overline{\mathcal{A}}$ of equal size such that $\mathsf{L}(\overline{\mathcal{A}}) = \overline{\mathsf{L}(\mathcal{A})}$

Complementation

- recall: NFA-complementation may blow-up automata sizes by an exponential

Lemma

For every AFA $\mathcal{A}$ there exists an AFA $\overline{\mathcal{A}}$ of equal size such that $\mathsf{L}(\overline{\mathcal{A}}) = \overline{\mathsf{L}(\mathcal{A})}$

Proof Outline.

- let $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$
- define the dual formula $\overline{\phi}$ of $\phi \in \mathbb{B}^+(Q)$ following De Morgans rules

$$\overline{q} \triangleq q \qquad \overline{\phi \vee \psi} \triangleq \overline{\phi} \wedge \overline{\psi} \qquad \overline{\phi \wedge \psi} \triangleq \overline{\phi} \vee \overline{\psi}$$

  - morally, $q \in Q$ re-used for their "negation"; we have (i) $M \vDash \phi$ iff $Q \backslash M \nvDash \overline{\phi}$

Complementation

- recall: NFA-complementation may blow-up automata sizes by an exponential

Lemma

For every AFA $\mathcal{A}$ there exists an AFA $\overline{\mathcal{A}}$ of equal size such that $\mathsf{L}(\overline{\mathcal{A}}) = \overline{\mathsf{L}(\mathcal{A})}$

Proof Outline.

- let $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$
- define the dual formula $\overline{\phi}$ of $\phi \in \mathbb{B}^+(Q)$ following De Morgans rules

$$\overline{q} \triangleq q \qquad \overline{\phi \vee \psi} \triangleq \overline{\phi} \wedge \overline{\psi} \qquad \overline{\phi \wedge \psi} \triangleq \overline{\phi} \vee \overline{\psi}$$

  - morally, $q \in Q$ re-used for their "negation"; we have (i) $M \models \phi$ iff $Q \setminus M \not\models \overline{\phi}$
- we now define $\overline{\mathcal{A}} \triangleq (Q, \Sigma, \overline{\delta}, q_I, Q \setminus F)$ where $\overline{\delta}(q, \mathrm{a}) \triangleq \overline{\delta(q, \mathrm{a})}$ for all $q \in Q$, $\mathrm{a} \in \Sigma$

Complementation

- recall: NFA-complementation may blow-up automata sizes by an exponential

Lemma

For every AFA $\mathcal{A}$ there exists an AFA $\overline{\mathcal{A}}$ of equal size such that $\mathsf{L}(\overline{\mathcal{A}}) = \overline{\mathsf{L}(\mathcal{A})}$

Proof Outline.

- let $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$
- define the dual formula $\overline{\phi}$ of $\phi \in \mathbb{B}^+(Q)$ following De Morgans rules

$$\overline{q} \triangleq q \qquad \overline{\phi \vee \psi} \triangleq \overline{\phi} \wedge \overline{\psi} \qquad \overline{\phi \wedge \psi} \triangleq \overline{\phi} \vee \overline{\psi}$$

  - morally, $q \in Q$ re-used for their "negation"; we have (i) $M \vDash \phi$ iff $Q \backslash M \nvDash \overline{\phi}$
- we now define $\overline{\mathcal{A}} \triangleq (Q, \Sigma, \overline{\delta}, q_I, Q \backslash F)$ where $\overline{\delta}(q, \mathrm{a}) \triangleq \overline{\delta(q, \mathrm{a})}$ for all $q \in Q$, $\mathrm{a} \in \Sigma$
  - by induction on $|w|$ it can now be shown that (ii) $\hat{\overline{\delta}}(q_I, w) = \overline{\hat{\delta}(q, w)}$

Complementation

- recall: NFA-complementation may blow-up automata sizes by an exponential

Lemma

For every AFA $\mathcal{A}$ there exists an AFA $\overline{\mathcal{A}}$ of equal size such that $\mathsf{L}(\overline{\mathcal{A}}) = \overline{\mathsf{L}(\mathcal{A})}$

Proof Outline.

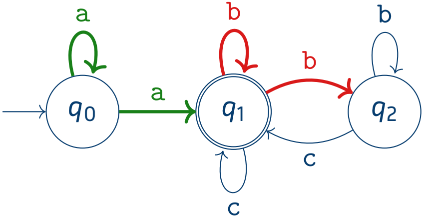
- let $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$
- define the dual formula $\overline{\phi}$ of $\phi \in \mathbb{B}^+(Q)$ following De Morgans rules

$$\overline{q} \triangleq q \qquad \overline{\phi \vee \psi} \triangleq \overline{\phi} \wedge \overline{\psi} \qquad \overline{\phi \wedge \psi} \triangleq \overline{\phi} \vee \overline{\psi}$$

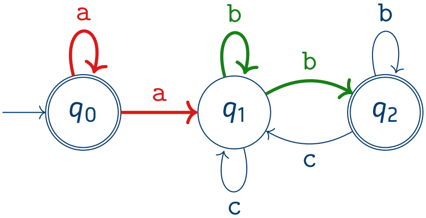
  - morally, $q \in Q$ re-used for their "negation"; we have (i) $M \vDash \phi$ iff $Q \backslash M \nvDash \overline{\phi}$
- we now define $\overline{\mathcal{A}} \triangleq (Q, \Sigma, \overline{\delta}, q_I, Q \backslash F)$ where $\overline{\delta}(q, \mathrm{a}) \triangleq \overline{\delta(q, \mathrm{a})}$ for all $q \in Q$, $\mathrm{a} \in \Sigma$
  - by induction on $|w|$ it can now be shown that (ii) $\hat{\overline{\delta}}(q_I, w) = \overline{\hat{\delta}(q, w)}$
  - overall, we have

$$w \notin \mathsf{L}(\mathcal{A}) \overset{def.}{\iff} F \nvDash \hat{\delta}(q_I, w) \overset{(i)}{\iff} Q \backslash F \vDash \overline{\hat{\delta}(q_I, w)} \overset{(ii)}{\iff} Q \backslash F \vDash \hat{\overline{\delta}}(q_I, w) \overset{def.}{\iff} w \in \mathsf{L}(\overline{\mathcal{A}})$$

Example



$\Updownarrow$ complement



Relationship with Regular Languages

AFAs Recognize *REG*

Theorem

For every AFA $\mathcal{A}$ *there exist a DFA* $\mathcal{B}$ *with* $O(2^{2^{|\mathcal{A}|}})$ *states such that* $L(\mathcal{A}) = L(\mathcal{B})$.

AFAs Recognize *REG*

Theorem

For every AFA $\mathcal{A}$ there exist a DFA $\mathcal{B}$ with $O(2^{2^{|\mathcal{A}|}})$ states such that $L(\mathcal{A}) = L(\mathcal{B})$.

Proof Outline.

let $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$

Idea:

- the states of $\mathcal{B}$ are formulas
- $\phi \xrightarrow{a} \psi$ in $\mathcal{B}$ if $\hat{\delta}(\phi, a) = \psi$
  - Example: $\delta(p, a) = q \wedge r$ and $\delta(q, a) = r \quad \Longrightarrow \quad p \vee q \xrightarrow{a} (q \wedge r) \vee r$
  - a run $q_I \xrightarrow{a_1} \cdots \xrightarrow{a_n} \phi$ thus models $\hat{\delta}(q_I, a_1 \ldots a_n) = \phi$
- the formula q_I is the initial state
- the formulas modeled by F are final

AFAs Recognize *REG*

Theorem

For every AFA $\mathcal{A}$ there exist a DFA $\mathcal{B}$ with $O(2^{2^{|\mathcal{A}|}})$ states such that $L(\mathcal{A}) = L(\mathcal{B})$.

Proof Outline.

let $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$

Idea:

- the states of $\mathcal{B}$ are formulas
- $\phi \xrightarrow{a} \psi$ in $\mathcal{B}$ if $\hat{\delta}(\phi, a) = \psi$
  - Example: $\delta(p, a) = q \wedge r$ and $\delta(q, a) = r \implies p \vee q \xrightarrow{a} (q \wedge r) \vee r$
  - a run $q_I \xrightarrow{a_1} \cdots \xrightarrow{a_n} \phi$ thus models $\hat{\delta}(q_I, a_1 \ldots a_n) = \phi$
- the formula q_I is the initial state
- the formulas modeled by F are final
- to keep the construction finite, we'll identify equivalent formulas

AFAs Recognize *REG*

Theorem

For every AFA $\mathcal{A}$ there exist a DFA $\mathcal{B}$ with $O(2^{2^{|\mathcal{A}|}})$ states such that $L(\mathcal{A}) = L(\mathcal{B})$.

Proof Outline.

let $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$

Formally:

- the equivalence $\sim$ on $\mathbb{B}^+(Q)$ is given by $\phi \sim \psi$ if $\{M \mid M \models \phi\} = \{M \mid M \models \psi\}$
  - $q \sim q \vee q \sim q \wedge q$ but $q \not\sim p \vee q \not\sim p \wedge q$

AFAs Recognize *REG*

Theorem

For every AFA $\mathcal{A}$ there exist a DFA $\mathcal{B}$ with $O(2^{2^{|\mathcal{A}|}})$ states such that $L(\mathcal{A}) = L(\mathcal{B})$.

Proof Outline.

let $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$

Formally:

- the equivalence $\sim$ on $\mathbb{B}^+(Q)$ is given by $\phi \sim \psi$ if $\{M \mid M \models \phi\} = \{M \mid M \models \psi\}$
  - $q \sim q \vee q \sim q \wedge q$ but $q \not\sim p \vee q \not\sim p \wedge q$
- the equivalence class $[\phi]_\sim$ can be simply conceived as the formula ϕ, with equivalent formulas $\phi \sim \psi$ identified
  - $[q \vee q]_\sim = \{q, q \vee q, q \wedge q, \ldots\}$

AFAs Recognize *REG*

Theorem

For every AFA $\mathcal{A}$ there exist a DFA $\mathcal{B}$ with $O(2^{2^{|\mathcal{A}|}})$ states such that $L(\mathcal{A}) = L(\mathcal{B})$.

Proof Outline.

let $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$

Formally:

- the equivalence $\sim$ on $\mathbb{B}^+(Q)$ is given by $\phi \sim \psi$ if $\{M \mid M \models \phi\} = \{M \mid M \models \psi\}$
  - $q \sim q \vee q \sim q \wedge q$ but $q \not\sim p \vee q \not\sim p \wedge q$
- the equivalence class $[\phi]_\sim$ can be simply conceived as the formula ϕ, with equivalent formulas $\phi \sim \psi$ identified
  - $[q \vee q]_\sim = \{q, q \vee q, q \wedge q, \dots\}$
- the set of all such equivalence classes $\mathbb{B}^+(Q)/\sim$ contains $O(2^{2^{|Q|}})$ elements

AFAs Recognize *REG*

Theorem

For every AFA $\mathcal{A}$ there exist a DFA $\mathcal{B}$ with $\mathrm{O}(2^{2^{|\mathcal{A}|}})$ states such that $\mathrm{L}(\mathcal{A}) = \mathrm{L}(\mathcal{B})$.

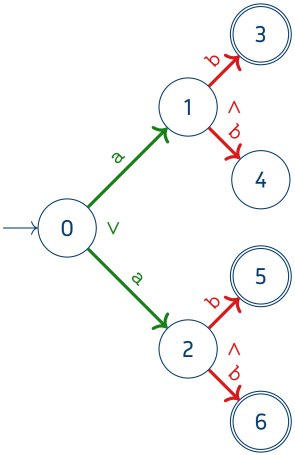
Proof Outline.

let $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$

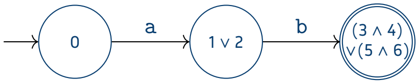
Formally:

- ⋆ the equivalence $\sim$ on $\mathbb{B}^+(Q)$ is given by $\phi \sim \psi$ if $\{M \mid M \vDash \phi\} = \{M \mid M \vDash \psi\}$
  - $q \sim q \vee q \sim q \wedge q$ but $q \not\sim p \vee q \not\sim p \wedge q$
- ⋆ the equivalence class $[\phi]_\sim$ can be simply conceived as the formula ϕ, with equivalent formulas $\phi \sim \psi$ identified
  - $[q \vee q]_\sim = \{q, q \vee q, q \wedge q, \ldots\}$
- ⋆ the set of all such equivalence classes $\mathbb{B}^+(Q)/\sim$ contains $\mathrm{O}(2^{2^{|Q|}})$ elements
- ⋆ $\mathcal{B} \triangleq (\mathbb{B}^+(Q)/\sim, \Sigma, q_I, \delta_\sim, \{[\phi]_\sim \mid F \vDash \phi\})$ where $\delta_\sim([\phi]_\sim, \mathrm{a}) \triangleq [\hat{\delta}(\phi, \mathrm{a})]_\sim$ recognises $\mathrm{L}(\mathcal{A})$

Example



the initial AFA



the translated DFA

From AFAs to NFA

Theorem

For every AFA $\mathcal{A}$ there exist a NFA $\mathcal{B}$ with $O(2^{|\mathcal{A}|})$ states such that $L(\mathcal{A}) = L(\mathcal{B})$.

Proof Outline.

- let $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$
- idea: rather then “recording” to be validated formulas as in the DFA construction, the corresponding NFA “records” valuations
  - the construction is simpler, at the expense of non-determinism

From AFAs to NFA

Theorem

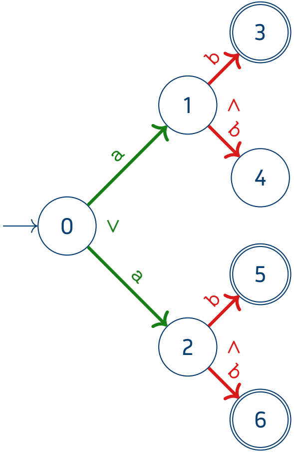
For every AFA $\mathcal{A}$ there exist a NFA $\mathcal{B}$ with $\mathrm{O}(2^{|\mathcal{A}|})$ states such that $\mathrm{L}(\mathcal{A}) = \mathrm{L}(\mathcal{B})$.

Proof Outline.

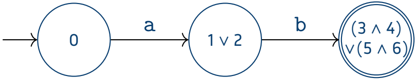
- let $\mathcal{A} = (Q, \Sigma, q_I, \delta, F)$
- idea: rather then "recording" to be validated formulas as in the DFA construction, the corresponding NFA "records" valuations
  - the construction is simpler, at the expense of non-determinism
- the NFA is given by $\mathcal{B} \triangleq (2^Q, \Sigma, \{q_I\}, \delta', \{M \mid M \subseteq F\})$ where

$$N \in \delta'(M, \mathrm{a}) \quad :\Longleftrightarrow \quad N \vDash \bigwedge_{q \in M} \delta(q, \mathrm{a})$$

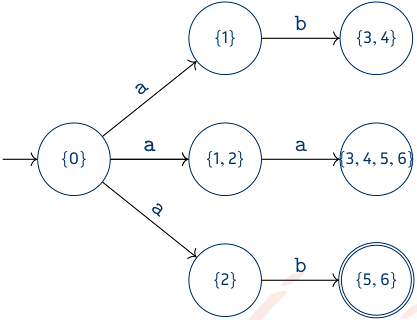
Example (II)



the initial AFA



the translated DFA



the translated NFA

Discussion

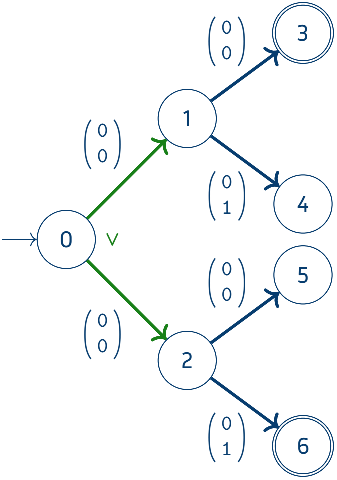
- What if we translate wMSO formulas to AFAs?
  - for basic formulas $x < y$ and $X(y)$, the construction is as seen previously
  - Boolean connectives are reflected directly in the transition
  - Quantifier elimination through projection homomorphisms

Discussion

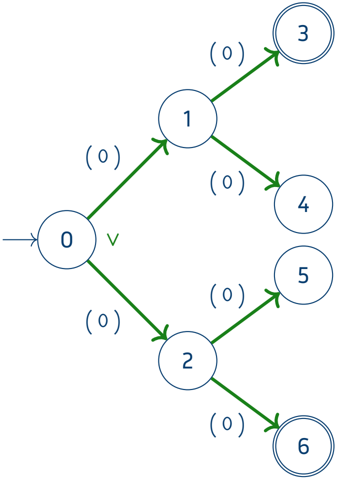
- ⋆ What if we translate wMSO formulas to AFAs?
  - for basic formulas $x < y$ and $X(y)$, the construction is as seen previously
  - Boolean connectives are reflected directly in the transition
  - Quantifier elimination through projection homomorphisms
- ⋆ this suggests resulting automaton is linear in size of formula

$\Longrightarrow$ wMSO model-checking in exponential time, contradicting the lower-bound result!

Projections and AFAs



$\Longrightarrow$



$L(\mathcal{A}) = \varnothing$

$L(\mathrm{del}_{2,2}(\mathcal{A})) = \{00\}$

Discussion

- What if we translate wMSO formulas to AFAs?
  - for basic formulas $x < y$ and $X(y)$, the construction is as seen previously
  - Boolean connectives are reflected directly in the transition
  - Quantifier elimination through projection homomorphisms
- this suggests resulting automaton is linear in size of formula

  $\Longrightarrow$ wMSO model-checking in exponential time, contradicting the lower-bound result!

Problem:

We do not have a polytime algorithm for homorphism applications on AFAs