Model Checking

René Thiemann

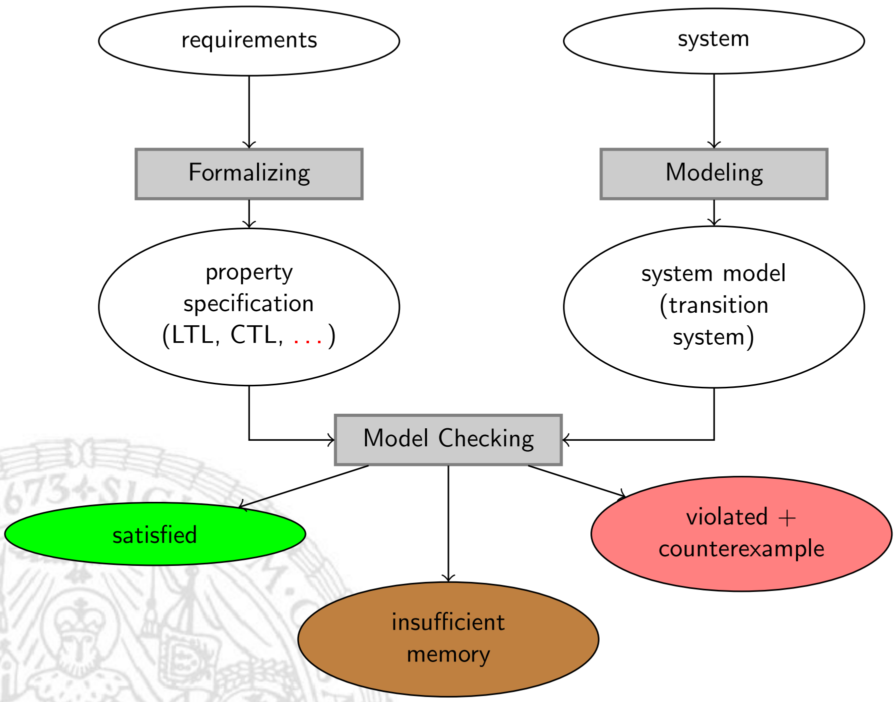
Institute of Computer Science
University of Innsbruck

SS 2008

Outline

- First-order logic of 1 Successor
- Second-order logic of 1 Successor
- Complementation of NBAs

Model Checking Overview



Limitations of LTL

- No LTL-operator that allows to look in the past
  (all LTL-operators only look in the future)

  At some moment we see red and some moment before we see orange

- No possibility to fix and compare several moments in time

  Green holds until red appears, and at sometime in the future orange is satisfied. Moreover, the red is later than the orange.

Statements can be encoded to LTL, but formalization is not obvious and error-prone

- $\text{orange}\,\mathsf{U}\,\text{red}$ or $(\mathsf{F}\,\text{orange})\,\mathsf{U}\,\mathsf{X}\,\text{red}$ or $\mathsf{F}\,(\text{orange} \wedge \mathsf{X}\,\mathsf{F}\,\text{red})$
- $(\text{green}\,\mathsf{U}\,\text{red}) \wedge \mathsf{F}\,\text{orange} \wedge ?$ or $(\text{green}\,\mathsf{U}\,\text{red}) \wedge \mathsf{F}\,(\text{orange} \wedge \mathsf{X}\,\mathsf{F}\,\text{red})$ or $\text{green}\,\mathsf{U}\,(\text{green} \wedge \text{orange} \wedge \mathsf{X}\,(\text{green}\,\mathsf{U}\,\text{red}))$

Solution: Use first-order logic which speaks about points in time

- $\exists x : \text{red}(x) \wedge \exists y : y < x \wedge \text{orange}(y)$
- $\exists x : \text{red}(x) \wedge \forall y : (y < x \Rightarrow \text{green}(y)) \wedge \exists z : \text{orange}(z) \wedge z < x$

First-order logic of 1 Successor (F1S)

F1S is like first-order (predicate) logic with the following differences

- The universe is fixed to $\mathbb{N}$
- Two predefined binary predicates: $=$ and $<$ with the obvious semantics
- Two function symbols: $'$ is the successor function and 0 the constant for the number 0
- For each atomic proposition a (of the transition system) there is a unary predicate symbol a
  Semantic of these predicates are specified by input word:

$$\text{For } w = \begin{matrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & \dots & \text{green} \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & \dots & \text{red} \end{matrix}$$

obtain $\mathsf{green}^w = \{0, 2, 4, 6, \dots\}$ and $\mathsf{red}^w = \{2, 3, 5, 7, \dots\}$

F1S Syntax

Let $\mathcal{V} = \{x, y, \dots\}$ be a set of variables (for time-points)
Let $\mathcal{S} = \{\mathsf{a}_1, \dots, \mathsf{a}_n\}$ be a set of unary predicate symbols

The set of F1S-terms over $\mathcal{V}$ is the smallest set such that

- every variable of $\mathcal{V}$ is a term
- 0 is a term
- If t is a term then t' is also a term

The set of F1S-formulas over $\mathcal{V}$ and $\mathcal{S}$ is the smallest set such that

- $t_1 < t_2$ and $t_1 = t_2$ are formulas for every two terms t_1 and t_2
- $\mathsf{a}_i(t)$ is a formula for every term t and $1 \leqslant i \leqslant n$
- If φ and ψ are formulas and $x \in \mathcal{V}$ then $\varphi \wedge \psi$,$\neg\varphi$, $\exists x : \varphi$, and $\forall x : \varphi$ are formulas (connectives $\vee, \Rightarrow, \dots$ are derived as usual)

Binding priority: $\{=, <\} \sqsupset \{\neg\} \sqsupset \{\wedge, \vee\} \sqsupset \{\Rightarrow, \Leftrightarrow\} \sqsupset \{\forall, \exists\}$

F1S Semantics

Use interpretations $\alpha : \mathcal{V} \to \mathbb{N}$ to map variables to numbers and sets $P_i \subseteq \mathbb{N}$ to interpret unary predicates $\mathsf{a_i}$.
α is extended to a mapping from terms to $\mathbb{N}$ in the usual way:

- $\alpha(0) = 0$
- $\alpha(t') = 1 + \alpha(t)$

Then $\mathcal{P} = P_1, \ldots, P_n$ and α satisfies φ, written $\mathcal{P} \models_\alpha \varphi$ iff

- $\varphi = t_1 = t_2$ and $\alpha(t_1) = \alpha(t_2)$
- $\varphi = t_1 < t_2$ and $\alpha(t_1) < \alpha(t_2)$
- $\varphi = \mathsf{a_i}(t)$ and $\alpha(t) \in P_i$
- $\varphi = \neg\psi$ and $\mathcal{P} \not\models_\alpha \psi$
- $\varphi = \varphi_1 \wedge \varphi_2$ and $\mathcal{P} \models_\alpha \varphi_1$ and $\mathcal{P} \models_\alpha \varphi_2$
- $\varphi = \exists x : \psi$ and $\mathcal{P} \models_{\alpha[x:=n]} \psi$ for some $n \in \mathbb{N}$
- $\varphi = \forall x : \psi$ and $\mathcal{P} \models_{\alpha[x:=n]} \psi$ for all $n \in \mathbb{N}$

F1S Semantics continued

φ is closed F1S-formula iff φ does not contain free variables

For closed formula φ and infinite word $w \in (2^n)^\omega$ define

$$w \models \varphi \text{ iff } \mathcal{P}^w \models \varphi \qquad \text{where}$$

- $\mathcal{P}_w = P_1^w, \ldots, P_n^w$
- $P_i^w = \{m \in \mathbb{N} \mid w[m][i] = 1\}$
- $w[m]$ is the the letter (vector) A at the m-th position of w
- $A[i]$ is the i-th element of vector A

Moreover, the language of φ is

$$\mathcal{L}(\varphi) = \{w \mid w \models \varphi\}$$

Relating LTL and F1S

Theorem

For every LTL-formula φ there is a closed F1S-formula $\mathsf{ltl_2f1s}(\varphi)$ such that $\mathcal{L}(\varphi) = \mathcal{L}(\mathsf{ltl_2f1s}(\varphi))$.

Proof.

Use mapping $\mathsf{ltl_2f1s}$: LTL-formulas $\times$ F1S-terms $\to$ closed F1S-formulas and define $\mathsf{ltl_2f1s}(\varphi) = \mathsf{ltl_2f1s}(\varphi, 0)$

- $\mathsf{ltl_2f1s}(a_i, t) = \mathsf{a_i}(t)$
- $\mathsf{ltl_2f1s}(\neg\varphi, t) = \neg\mathsf{ltl_2f1s}(\varphi, t)$
- $\mathsf{ltl_2f1s}(\varphi \wedge \psi, t) = \mathsf{ltl_2f1s}(\varphi, t) \wedge \mathsf{ltl_2f1s}(\varphi, t)$
- $\mathsf{ltl_2f1s}(\mathsf{X}\,\varphi, t) = \mathsf{ltl_2f1s}(\varphi, t')$
- $\mathsf{ltl_2f1s}(\varphi \,\mathsf{U}\, \psi, t) = \exists x : t \leqslant x \wedge \mathsf{ltl_2f1s}(\psi, x) \wedge$
  $\forall y : t \leqslant y \wedge y < x \Rightarrow \mathsf{ltl_2f1s}(\varphi, y)$

(x and y must be fresh in last step and $t \leqslant y$ abbrev. $t = y \vee t < y$) ■

Proof of Correctness of Construction

Example

Consider $\varphi = \mathsf{X}\,(\neg a\,\mathsf{U}\,\mathsf{X}\,(b \wedge c))$. Then the translation yields

Relating LTL and F1S

Theorem

For every closed F1S-formula φ there is an LTL-formula $\mathsf{f1s_2ltl}(\varphi)$ such that $\mathcal{L}(\varphi) = \mathcal{L}(\mathsf{f1s_2ltl}(\varphi))$.

⇒ LTL and F1S have same expressive power

⇒ write readable, straight-forward specifications in F1S and perform LTL-model checking afterwards

Theorem (Considering sizes)

- *If closed F1S-formula has size m then equivalent LTL-formula can be constructed which has size $\mathcal{O}(2^{2^{\cdot^{\cdot^{\cdot^{2^m}}}}})$ (height of tower is m)*
- *The bound is strict*

⇒ Optimization for special cases strongly required

⇒ sometimes hand-written LTL-specifications may be better

Relating F1S and NBAs

Recall relation between LTL and NBAs

NBAs are strictly more powerful than LTL

($\mathcal{L} = \{(00)^n 1^\omega \mid n \in \mathbb{N}\}$ is NBA-definable, but not by LTL)

With previous results directly achieve

NBAs are strictly more powerful than F1S

Question: Is there a logic which has same expressiveness as NBAs?

Yes, extension of F1S to second-order (S1S)

$$\varphi = \exists \mathsf{even} : \mathsf{even}(0) \wedge \forall x : \mathsf{even}(x) \Leftrightarrow \neg \mathsf{even}(x') \wedge \\ \exists y : \forall z : (z < y \Rightarrow \neg \mathsf{a}(z)) \wedge (y \leqslant z \Rightarrow \mathsf{a}(z)) \wedge \mathsf{even}(y)$$

Syntax and Semantic of S1S

Syntax

S1S-formulas are extension of F1S-formulas where now $\mathcal{S} = \{\mathsf{a}_1, \mathsf{a}_2, \dots\}$ are second order variables ranging over subsets of $\mathbb{N}$.

S1S-formulas are built like F1S-formulas with the following extension:

- If φ is a formula then $\exists \mathsf{a}_i : \varphi$ is also a formula

Write $\varphi(\mathsf{a}_1, \dots, \mathsf{a}_n)$ to denote that $\mathsf{a}_1, \dots, \mathsf{a}_n$ are the free second-order variables of φ.

Semantic

Extend F1S-semantic as follows:

- $P_1, \dots, P_{n-1} \models_\alpha \exists \mathsf{a}_n : \varphi$ iff exists $P_n \subseteq \mathbb{N}$ with $P_1, \dots, P_n \models_\alpha \varphi$
- If $\varphi(\mathsf{a}_1, \dots, \mathsf{a}_n)$ has no free first-order variables then

$$\mathcal{L}(\varphi) = \{w \in (2^n)^\omega \mid \mathcal{P}^w \models \varphi\}$$

Expressiveness of S1S

All-Quantifier

- One can extend S1S by construct $\forall \mathsf{a} : \varphi$ with obvious semantics
- This does not increase power since

$$\forall \mathsf{a} : \varphi \qquad \equiv \qquad \neg\neg\forall \mathsf{a} : \varphi \qquad \equiv \qquad \neg\exists \mathsf{a} : \neg\varphi$$

Comparison to NBAs

Theorem (Equivalence of S1S and NBAs, Büchi)

- *For every NBA $\mathcal{A}$ there is S1S-formula $\varphi_{\mathcal{A}}$ such that* $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\varphi_{\mathcal{A}})$
- *For every S1S-formula φ there is NBA $\mathcal{A}_\varphi$ such that* $\mathcal{L}(\varphi) = \mathcal{L}(\mathcal{A}_\varphi)$

Consequences of Büchi's Theorem

Model Checking for S1S possible as for LTL

- Construct NBA $\mathcal{A}_{\neg\varphi}$ and check $\mathcal{L}(TS \otimes \mathcal{A}_{\neg\varphi}) = \varnothing$

Satisfiability of S1S-formulas is decidable

(Given φ, construct NBA $\mathcal{A}_\varphi$ and check $\mathcal{L}(\mathcal{A}_\varphi) \neq \varnothing$)

- $\varphi_1 = \forall x : \exists y : x < y$
- $\varphi_2 = \forall x : \exists y : y < x$
- $\varphi_3 = \forall \mathsf{a} : \mathsf{a}(0) \wedge (\forall x : \mathsf{a}(x) \Rightarrow \mathsf{a}(x')) \Rightarrow \forall x : \mathsf{a}(x)$
- $\varphi_4 = \forall \mathsf{a} : \mathsf{a}(0) \wedge (\forall x : (\forall y : y < x \Rightarrow \mathsf{a}(y)) \Rightarrow \mathsf{a}(x)) \Rightarrow \forall x : \mathsf{a}(x)$

Short Distance to Undecidability

Theorem (Undecidability of first-order arithmetic, Gödel)

If one extends F1S by addition ($+$) and multiplication ($\cdot$) then satisfiability of formulas is undecidable.

Corollary

If one allows second-order quantification over relations (and not only over sets as in S1S), then satisfiability of formulas is undecidable.

Proving Büchi's Theorem, 1. Direction: NBA to S1S

Let $\mathcal{A} = (\mathcal{Q} = \{q_0, \ldots, q_m\}, \Sigma = 2^n, q_0, \delta, F)$

Main ideas:

- $\varphi_\mathcal{A}$ guesses accepting run ρ for input variables $\mathsf{a}_1, \ldots, \mathsf{a}_\mathsf{n}$
- To this end for each q_i a second-order variable b_i is used

$$\rho \text{ should visit } q_i \text{ at moment } x \text{ iff } \mathsf{b}_\mathsf{i}(x)$$

- $\varphi_\mathcal{A}$ has to make sure that
  - the sets $\mathsf{b}_0, \ldots, \mathsf{b}_\mathsf{m}$ form a partition of $\mathbb{N}$
  - infinitely often a final state is visited
  - the partition corresponds to a run w.r.t. δ

First Direction: From NBA to S1S

Let $\mathcal{A} = (\mathcal{Q} = \{q_0, \ldots, q_m\}, \Sigma = 2^n, q_0, \delta, F)$

Define $\varphi_\mathcal{A}$ as the follows:

- $\varphi_\mathcal{A}(\mathrm{a}_1, \ldots, \mathrm{a}_\mathrm{n}) = \exists \mathrm{b}_0 : \ldots \exists \mathrm{b}_\mathrm{m} : \psi(\mathrm{a}_1, \ldots, \mathrm{a}_\mathrm{n}, \mathrm{b}_0, \ldots, \mathrm{b}_\mathrm{m})$ where

$$\begin{aligned}
\psi = \quad & \forall x : \left(\bigvee\nolimits_{0 \leqslant i \leqslant m} \mathrm{b_i}(x)\right) \wedge \neg \bigvee\nolimits_{i \neq j} (\mathrm{b_i}(x) \wedge \mathrm{b_j}(x)) \quad \wedge && \text{(partition)} \\
& \forall x : \exists y : x < y \wedge \bigvee\nolimits_{q_i \in F} \mathrm{b_i}(y) \quad \wedge && \text{(accepting)} \\
& \mathrm{b_0}(0) \wedge \forall x : \bigvee\nolimits_{q_j \in \delta(q_i, A)} \left(\mathrm{b_i}(x) \wedge \mathrm{input}_{A,x} \wedge \mathrm{b_j}(x')\right) && \text{(run)}
\end{aligned}$$

- Here, $\mathrm{input}_{A,x} = (\neg)\mathrm{a}_1(x) \wedge \ldots \wedge (\neg)\mathrm{a}_\mathrm{n}(x)$ where the i-th $\neg$ is present iff $A[i] = 0$. Example:

$$\mathrm{input}_{(0,1,0)^T,x} = \neg \mathrm{a}_1(x) \wedge \mathrm{a}_2(x) \wedge \neg \mathrm{a}_3(x)$$

Example

Second Direction: From S1S to NBA

Perform 3-step translation

1. From S1S to simplified logic $S1S_0$
2. From $S1S_0$ to NBA
   (assuming that NBAs are closed under union and complement)
3. Show that NBAs are closed under union and complement

$S1S_0$

$S1S_0$: simplified version of S1S

- No first-order quantification, no $0, ', <, =$
- sing predicate, $\mathcal{P} \models \mathsf{sing}(\mathsf{a})$ iff $|P_a| = 1$
- succ predicate, $\mathcal{P} \models \mathsf{succ}(\mathsf{a}, \mathsf{b})$ iff $P_a = \{n\}$ and $P_b = \{n+1\}$ for some $n \in \mathbb{N}$
- $\subseteq$ predicate, $\mathcal{P} \models \mathsf{a} \subseteq \mathsf{b}$ iff $P_a \subseteq P_b$

Lemma

For each S1S-formula there is an equivalent $S1S_0$-formula.

First Step: From S1S to S1S$_0$

Proof.

1. Eliminate 0: $\varphi[0] \rightsquigarrow \exists x : (\varphi[x] \land \neg\exists y : y' = x)$
2. Eliminate iteration of $'$: $\varphi[t''] \rightsquigarrow \exists x : (t' = x \land \varphi[x'])$
3. Eliminate $<$: $s < t \rightsquigarrow \forall \mathsf{b} : (\mathsf{b}(s') \land \forall x : \mathsf{b}(x) \Rightarrow \mathsf{b}(x')) \Rightarrow \mathsf{b}(t)$
   (t is in successor closure of s')
4. Eliminate $'$ in b: $\mathsf{b}(t') \rightsquigarrow \exists x : t' = x \land \mathsf{b}(x)$

Obtain S1S-formula with atomic formulas $x' = y$, $x = y$, and $\mathsf{b}(x)$ only
From this obtain S1S$_0$-formula by replacing x by a_x:

- $\exists x : \varphi \rightsquigarrow \exists \mathsf{a}_x : \mathsf{sing}(\mathsf{a}_x) \land \varphi$
- $\forall x : \varphi \rightsquigarrow \forall \mathsf{a}_x : \mathsf{sing}(\mathsf{a}_x) \Rightarrow \varphi$
- $x' = y \rightsquigarrow \mathsf{succ}(\mathsf{a}_x, \mathsf{a}_y)$
- $x = y \rightsquigarrow \mathsf{a}_x \subseteq \mathsf{a}_y \land \mathsf{a}_y \subseteq \mathsf{a}_x$
- $\mathsf{b}(x) \rightsquigarrow \mathsf{a}_x \subseteq \mathsf{b}$

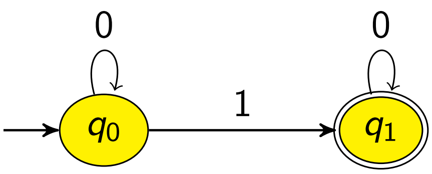
Second Step: From S1S$_0$ to NBA

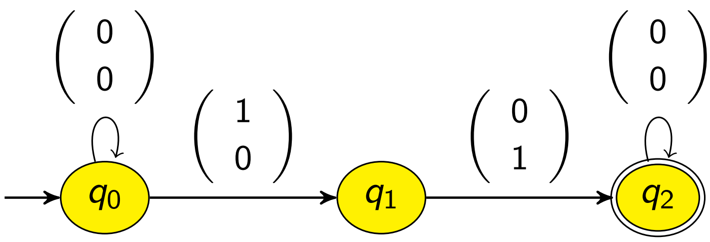
Lemma

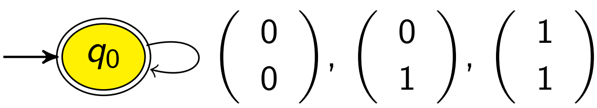
For each S1S$_0$-formula $\varphi(\mathsf{a}_1, \ldots, \mathsf{a}_n)$ there is an equivalent NBA $\mathcal{A}_\varphi$.

Proof

We use induction on φ. W.l.o.g. the only connectives are $\lor, \exists, \neg$.

- $\varphi = \mathsf{sing}(\mathsf{a})$: $\mathcal{A}_\varphi =$ 

- $\varphi = \mathsf{succ}(\mathsf{a}_1, \mathsf{a}_2)$: $\mathcal{A}_\varphi =$ 

- $\varphi = \mathsf{a}_1 \subseteq \mathsf{a}_2$: $\mathcal{A}_\varphi =$ 

Proof Continued

- $\varphi = \exists \mathsf{a}_{n+1} : \psi$: By induction obtain $\mathcal{A}_\psi = (\mathcal{Q}, \Sigma' = 2^{n+1}, q_0, \delta', F)$. Obtain $\mathcal{A}_\varphi = (\mathcal{Q}, \Sigma = 2^n, q_0, \delta, F)$ by dropping last component of input letters:

$$\delta(q, (b_1, \ldots, b_n)^T) = \delta'(q, (b_1, \ldots, b_n, 0)) \cup \delta'(q, (b_1, \ldots, b_n, 1))$$

- $\varphi = \neg\psi$: By induction obtain $\mathcal{A}_\psi$. NBA complementation yields $\mathcal{A}_\varphi$.
- $\varphi = \psi_1 \vee \psi_2$: By induction obtain $\mathcal{A}_{\psi_1}$ and $\mathcal{A}_{\psi_2}$. First enlarge the input alphabets of both NBAs to have the same input letters. (Obtain $\mathcal{B}_{\psi_1}$ and $\mathcal{B}_{\psi_2}$ with same input alphabet). Then $\mathcal{A}_\varphi$ is the union NBA for $\mathcal{B}_{\psi_1}$ and $\mathcal{B}_{\psi_2}$.
  Enlargement: Let ψ_1 have free variables $\mathsf{a}_\mathsf{i}, \ldots, \mathsf{a}_\mathsf{n}$ and ψ_2 has free variables $\mathsf{a}_1, \ldots, \mathsf{a}_\mathsf{k}$ then $\mathcal{A}_{\psi_2} = (\mathcal{Q}, 2^k, q_0, \delta', F)$. Define $\mathcal{B}_{\psi_2} = (\mathcal{Q}, 2^n, q_0, \delta, F)$ where $(c_1, \ldots, c_n)^T \in \delta(q, (b_1, \ldots, b_n)^T)$ iff $(c_1, \ldots, c_k)^T \in \delta'(q, (b_1, \ldots, b_k)^T)$. $\mathcal{B}_{\psi_1}$ is defined in the same way.

Illustration of $\exists$

Illustration of Enlargement

Third Step: NBA operations

Union

Let $\mathcal{A}_1 = (\mathcal{Q}_1, \Sigma, q_{0,1}, \delta_1, F_1)$ and $\mathcal{A}_2 = (\mathcal{Q}_2, \Sigma, q_{0,2}, \delta_2, F_2)$ be given where $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are disjoint.

Idea for union NBA $\mathcal{A}$: Copy both NBAs and add new starting state which chooses between $\mathcal{A}_1$ and $\mathcal{A}_2$.

Formally: $\mathcal{A} = (\mathcal{Q}_1 \uplus \mathcal{Q}_2 \uplus \{q_0\}, \Sigma, \delta, q_0, F_1 \cup F_2)$ where δ is defined as follows:

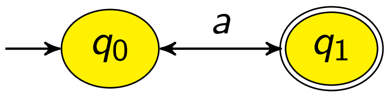
- $\delta(q_0, A) = \delta_1(q_{0,1}, A) \cup \delta_2(q_{0,2}, A)$
- $\delta(q, A) = \delta_1(q, A)$ if $q \in \mathcal{Q}_1$
- $\delta(q, A) = \delta_2(q, A)$ if $q \in \mathcal{Q}_2$

Obviously: $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$

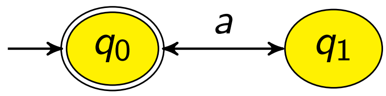
Complementation of NFAs

Non-deterministic Finite Automata can be complemented in two steps:

1. Construct equivalent deterministic finite automaton (DFA)
2. Exchange final and non-final states of DFA



odd nr of a's



even nr of a's

Both steps do not work with NBAs

1. Not every NBA has a corresponding det. Büchi automaton (DBA)
2. Exchanging final and non-final states of DBA does not yield complement

Notations

- $a, b, \ldots \in \Sigma$ — letters
- $u, v \in \Sigma^*$ — finite words
- $U, V \subseteq \Sigma^*$ — sets of finite words
- $w \in \Sigma^\omega$ — infinite word
- $W \subseteq \Sigma^\omega$ — set of infinite words
- $p, q \in Q$ — states of NFA or NBA
- $U \cdot W = \{uw \in \Sigma^\omega \mid u \in U, w \in W\}$ — concatenation
- $U^\omega = \{u_0\, u_1\, u_2 \ldots \in \Sigma^\omega \mid \text{all } u_i \in U\}$ — infinite concatenation
- δ extended to function $\hat{\delta} : Q \times \Sigma^* \to 2^Q$

$$\begin{aligned} \hat{\delta}(q, \varepsilon) &= \{q\} \\ \hat{\delta}(q, a\, u) &= \bigcup_{p \in \delta(q,a)} \hat{\delta}(p, u) \end{aligned}$$

$\hat{\delta}(q, u)$ are all states which are reachable when reading u starting in q

Overview NBA Complementation

Complementing Büchi Automaton $\mathcal{A}$ with $\mathcal{L} = \mathcal{L}(\mathcal{A})$

1. Find family of finitely many sets $U_1, \ldots, U_n \subseteq \Sigma^*$ such that
2. for all i, j: $U_i \cdot U_j^\omega \subseteq \mathcal{L}$ or $U_i \cdot U_j^\omega \cap \mathcal{L} = \varnothing$
   $U_i \cdot U_j^\omega$ is completely in $\mathcal{L}$ or not at all
3. every infinite word is contained in some $U_i \cdot U_j^\omega$.
4. Then assemble $2^\omega \setminus \mathcal{L} = \bigcup_{U_i \cdot U_j^\omega \cap \mathcal{L} = \varnothing} U_i \cdot U_j^\omega$.
5. Finally, show that everything can be encoded into one NBA.

Note that NBA $\mathcal{A} = (\mathcal{Q}, \Sigma, q_0, \delta, F)$ is fixed in remainder of this lecture.

Towards Step 1: Transition Profiles

Definition (Transition profile)

Transition profiles are subsets of

$$\{p \twoheadrightarrow q \mid p, q \in \mathcal{Q}\} \cup \{p \twoheadrightarrow_F q \mid p, q \in \mathcal{Q}\}.$$

The transition profile of a finite word u is

$$\begin{aligned} tp(u) &= \{p \twoheadrightarrow q \mid q \in \hat{\delta}(p, u)\} \\ &\cup \{p \twoheadrightarrow_F q \mid q \in \hat{\delta}(p, u),\ \text{run from } p \text{ to } q \text{ contains final state}\} \end{aligned}$$

Definition ($\mathcal{A}$-equivalence)

We define $\mathcal{A}$-equivalence as a relation $\sim_\mathcal{A} \subseteq \Sigma^* \times \Sigma^*$:

$$u \sim_\mathcal{A} v \qquad \text{iff} \qquad tp(u) = tp(v)$$

Step 1: A Family of Sets $U_1, \ldots, U_n$

Lemma

$\sim_{\mathcal{A}}$ is an equivalence relation (reflexive, symmetric, transitive).

Lemma

$\sim_{\mathcal{A}}$ has only finitely many equivalence classes.

Proof.

Obvious, since there are only finitely many transition profiles. ∎

Let there be n equivalence classes of $\sim_{\mathcal{A}}$.

Define $U_1, \ldots, U_n$ as the equivalence classes of $\sim_{\mathcal{A}}$.

Example

Step 2: $U_i \cdot U_j^\omega$ is Completely in $\mathcal{L}$ or Not At All

Lemma

If $w \in U_i \cdot U_j^\omega \cap \mathcal{L}$ then $U_i \cdot U_j^\omega \subseteq \mathcal{L}$.

Step 3: Every Word is Contained in Some $U_i \cdot U_j^\omega$

Recall the following result from graph theory:

Theorem (Infinite version of Ramsey's Theorem)

Let $\mathcal{G}$ be an undirected graph with infinitely many nodes N which is fully connected, and where every edge is colored with a color between 1 and n. Then there is an infinite subset M of nodes where all edges between these nodes have the same color.

Step 4: Assembling $2^\omega \setminus \mathcal{L}$ from $U_1, \ldots, U_n$

Obviously,

$$W_1 := \bigcup_{U_i \cdot U_j^\omega \cap \mathcal{L} = \varnothing} U_i \cdot U_j^\omega \subseteq 2^\omega \setminus \mathcal{L}$$

With Step 2 we know: $U_i \cdot U_j^\omega$ is completely in $\mathcal{L}$ or not at all. Hence,

$$W_2 := \bigcup_{U_i \cdot U_j^\omega \cap \mathcal{L} \neq \varnothing} U_i \cdot U_j^\omega \subseteq \mathcal{L}$$

Step 3 proves that $W_1 \cup W_2 = 2^\omega$. Hence

$$2^\omega \setminus \mathcal{L} = W_1 = \bigcup_{U_i \cdot U_j^\omega \cap \mathcal{L} = \varnothing} U_i \cdot U_j^\omega$$

Step 5: $2^\omega \setminus \mathcal{L}$ can be encoded as NBA

$$2^\omega \setminus \mathcal{L} = \bigcup_{U_i \cdot U_j^\omega \cap \mathcal{L} = \varnothing} U_i \cdot U_j^\omega$$

For encoding of NBA need the following components:

1. Construct NFAs for each U_i
2. Construct NBA for infinite concatenation U_j^ω given NFA for U_j
3. Construct NBA for concatenation $U_i \cdot U_j^\omega$ given NFA for U_i and NBA for U_j^ω
4. Construct NBA for intersection of $U_i \cdot U_j^\omega \cap \mathcal{L}$ and check resulting NBA on emptyness
5. Construct final NBA as union $\bigcup_{\ldots} U_i \cdot U_j^\omega$

Union and emptyness-check of NBAs have already been presented.
Intersection of NBAs can also easily be done (IMC)

Step 5.1: Construction of NFA for U_i

Recall that $U_i = \{u \mid tp(u) = TP\}$ for some transition profile TP.

$$\begin{array}{rcl} tp(u) & = & \{p \twoheadrightarrow q \mid q \in \hat{\delta}(p,u)\} \\ & \cup & \{p \twoheadrightarrow_F q \mid q \in \hat{\delta}(p,u), \text{ run from } p \text{ to } q \text{ contains final state}\} \end{array}$$

- Define $V_{pq} = \{u \mid q \in \hat{\delta}(p,u)\}$ and $V^F_{pq} = \{u \mid q \in \hat{\delta}(p,u)$, run from p to q contains final state$\}$
- Obviously, each V_{pq} is regular (accepted by NFA $(\mathcal{Q}, \Sigma, p, \delta, \{q\})$)

$\Rightarrow$ each V^F_{pq} is regular, since $V^F_{pq} = \bigcup_{r \in F} V_{pr} \cdot V_{rq}$

$\Rightarrow$ U_i is regular since

$$\begin{array}{rcllcl} U_i & = & \bigcap_{p \twoheadrightarrow q \in TP} V_{pq} & \cap & \bigcap_{p \twoheadrightarrow_F q \in TP} V^F_{pq} \\ & \cap & \bigcap_{p \twoheadrightarrow q \notin TP} \Sigma^* \setminus V_{pq} & \cap & \bigcap_{p \twoheadrightarrow_F q \notin TP} \Sigma^* \setminus V^F_{pq} \end{array}$$

Here, we used the well-known result that regular languages are closed under concatenation, union, complement, and intersection

Step 5.2: Construction of NBA for U^ω

Let $\mathcal{B}' = (\mathcal{Q}', \Sigma, q_0', \delta', F')$ be NFA with $\mathcal{L}(\mathcal{B}') = U$.

Main ideas:

- Add new state q_0'' which will be visited between u_i and u_{i+1} in infinite word $u_0\, u_1\, u_2 \ldots \in U^\omega$ where each $u_i \in U$
- One can start in q_0'' and read words as in q_0'
- Whenever final state is reached one can jump back to q_0''

In detail: Let $\mathcal{B} = (\mathcal{Q}' \uplus \{q_0''\}, \Sigma, q_0'', \delta'', \{q_0''\})$ where δ'' is defined as:

- $\delta''(q_0'', a) = \delta'(q_0', a) \cup \{q_0'' \mid \text{if } F' \cap \delta'(q_0', a) \neq \varnothing\}$
- $\delta''(q\ \ , a) = \delta'(q\ \ , a) \cup \{q_0'' \mid \text{if } F' \cap \delta'(q, a) \neq \varnothing\}$ if $q \in \mathcal{Q}'$

Lemma

$\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{B}')^\omega = U^\omega$.

Proof of Lemma

Step 5.3: Construction of NBA for $U_i \cdot U_j^\omega$

Let NFA $\mathcal{A}_1 = (\mathcal{Q}_1, \Sigma, q_{0,1}, \delta_1, F_1)$ and NBA $\mathcal{A}_2 = (\mathcal{Q}_2, \Sigma, q_{0,2}, \delta_2, F_2)$ be given such that $\mathcal{L}(\mathcal{A}_1) = U_i$ and $\mathcal{L}(\mathcal{A}_2) = U_j^\omega$.
Main ideas:

- Copy both automata
- One can switch from $\mathcal{A}_1$ to $\mathcal{A}_2$ for every final state of $\mathcal{A}_1$

In detail: Let $\mathcal{B} = (\mathcal{Q}_1 \uplus \mathcal{Q}_2, \Sigma, q_{0,1}, \delta', F_2)$ where δ' is defined as:

- If $q' \in \delta_1(q, a) \cup \delta_2(q, a)$ then $q' \in \delta'(q, a)$
- If $q \in F_1$ and $q' \in \delta_2(q_{0,2}, a)$ then $q' \in \delta'(q, a)$
- No other states are in $\delta'(q, a)$

Lemma

$\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{A}_1) \cdot \mathcal{L}(\mathcal{A}_2) = U_i \cdot U_j^\omega$