

Automata and Logic

Aart Middeldorp and Samuel Frontull

Theorem

every DBA M can be effectively transformed into NBA M' such that $L(M') = \sim L(M)$

Theorem

ω -regular sets are closed under complement

Notation

for NBA $M = (Q, \Sigma, \Delta, S, F)$ and states $p, q \in Q$

$$L_{pq} = \{x \in \Sigma^* \mid q \in \hat{\Delta}(\{p\}, x)\}$$

$$L_{pq}^f = \bigcup_{f \in F} L_{pf} \cdot L_{fq}$$

Definition

relation $\sim_M$ on Σ^* for NBA $M = (Q, \Sigma, \Delta, S, F)$: $u \sim_M v$ if for all $p, q \in Q$

$$u \in L_{pq} \iff v \in L_{pq} \quad \text{and} \quad u \in L_{pq}^f \iff v \in L_{pq}^f$$

Outline

1. Summary of Previous Lecture
2. Büchi Automata
3. Intermezzo
4. Model Checking
5. Linear-Time Temporal Logic (LTL)
6. Further Reading

Lemma

- ① $\sim_M$ is equivalence relation of finite index
- ② each equivalence class of $\sim_M$ is regular

Lemma

for all $x \in U \cdot V^\omega$ with equivalence classes U and V of $\sim_M$

$$x \in L(M) \implies U \cdot V^\omega \subseteq L(M) \quad x \notin L(M) \implies U \cdot V^\omega \subseteq \sim L(M)$$

Lemma

for all $x \in \Sigma^\omega$ there exist equivalence classes U and V of $\sim_M$ such that $x \in U \cdot V^\omega$

Corollary

$$\sim L(M) = \bigcup \{U \cdot V^\omega \mid U \text{ and } V \text{ are equivalence classes of } \sim_M \text{ such that } U \cdot V^\omega \cap L(M) = \emptyset\}$$

Definitions

- ▶ first-order variables $V_1 = \{x, y, \dots\}$ ranging over natural numbers
- ▶ second-order variables $V_2 = \{X, Y, \dots\}$ ranging over **sets of natural numbers**
- ▶ formulas of **monadic second-order logic (MSO)**

$$\varphi ::= \perp \mid x < y \mid X(x) \mid \neg \varphi \mid \varphi_1 \vee \varphi_2 \mid \exists x. \varphi \mid \exists X. \varphi$$

with $x, y \in V_1$ and $X \in V_2$

- ▶ assignment α is mapping from variables $x \in V_1$ to $\mathbb{N}$ and $X \in V_2$ to **subsets of $\mathbb{N}$**
- ▶ assignment α satisfies formula φ ($\alpha \models \varphi$):

$$\alpha \not\models \perp$$

$$\alpha \models x < y \iff \alpha(x) < \alpha(y) \quad \alpha \models \neg \varphi \iff \alpha \not\models \varphi$$

$$\alpha \models X(x) \iff \alpha(x) \in \alpha(X) \quad \alpha \models \varphi_1 \vee \varphi_2 \iff \alpha \models \varphi_1 \text{ or } \alpha \models \varphi_2$$

$$\alpha \models \exists x. \varphi \iff \alpha[x \mapsto n] \models \varphi \text{ for some } n \in \mathbb{N}$$

$$\alpha \models \exists X. \varphi \iff \alpha[X \mapsto N] \models \varphi \text{ for some subset } N \subseteq \mathbb{N}$$

Automata

- ▶ (deterministic, nondeterministic, alternating) finite automata
- ▶ regular expressions
- ▶ (alternating) **Büchi automata**

Logic

- ▶ (weak) monadic second-order logic
- ▶ Presburger arithmetic
- ▶ **linear-time temporal logic**

Definitions

- ▶ assignment α for MSO formula φ with $FV(\varphi) = (x_1, \dots, x_m, X_1, \dots, X_n)$ is encoded as infinite string $\underline{\alpha} \in (\{0, 1\}^{m+n})^\omega$:

$$\alpha(x_i) = j \quad \text{if } i\text{-th entry of } j\text{-th symbol in } \underline{\alpha} \text{ is } 1$$

$$\alpha(X_i) = \{j \mid (m+i)\text{-th entry of } j\text{-th symbol in } \underline{\alpha} \text{ is } 1\}$$

- ▶ infinite string over $\{0, 1\}^{m+n}$ is **m-admissible** if first m rows contain exactly one 1 each
- ▶ $L_a(\varphi) = \{x \in (\{0, 1\}^{m+n})^\omega \mid x \text{ is } m\text{-admissible and } \underline{x} \models \varphi\}$

Theorem

$L_a(\varphi)$ is ω -regular for every MSO formula φ

Theorem

set $A \subseteq \Sigma^\omega$ is ω -regular if and only if A is MSO definable

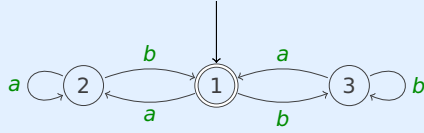
Outline

1. Summary of Previous Lecture
2. Büchi Automata
3. Intermezzo
4. Model Checking
5. Linear-Time Temporal Logic (LTL)
6. Further Reading

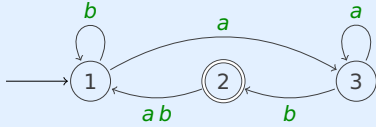
Remark

minimal DBAs are not unique

- ▶ $L = \{x \in \{a, b\}^\omega \mid |x|_a = \infty \text{ and } |x|_b = \infty\}$
- ▶ $L = L(M_1)$ for DBA M_1



- ▶ $L = L(M_2)$ for DBA M_2



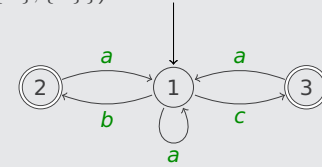
- ▶ M_1 and M_2 are not isomorphic
- ▶ no DBA with 2 states accepts L

Definition

- ▶ **generalized Buchi automaton (GBA)** is automaton $M = (Q, \Sigma, \Delta, S, \{F_1, \dots, F_k\})$ with $F_1, \dots, F_k \subseteq Q$
- ▶ $\infty(r) = \{q \in Q \mid q \text{ occurs infinitely often in run } r\}$
- ▶ run r of GBA is **accepting** if $\infty(r) \cap F_i \neq \emptyset$ for all $1 \leq i \leq k$

Example

GBA $M = (\{1, 2, 3\}, \{a, b, c\}, \Delta, \{1\}, \{\{2\}, \{3\}\})$



$L(M) = \{x \in \{a, b, c\}^\omega \mid |x|_b = |x|_c = \infty \text{ and each } b \text{ and } c \text{ in } x \text{ is followed by } a\}$

Remark

NBA $(Q, \Sigma, \Delta, S, F)$ is equivalent to GBA $(Q, \Sigma, \Delta, S, \{F\})$

Lemma

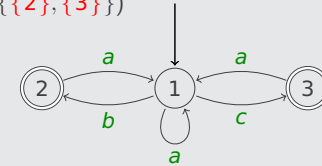
every GBA can be transformed into equivalent NBA

Proof

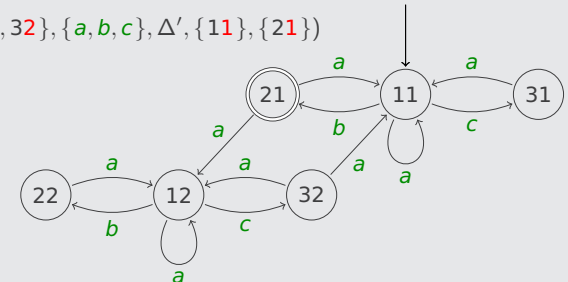
- ▶ GBA $M = (Q, \Sigma, \Delta, S, \{F_1, \dots, F_k\})$
- ▶ $L(M) = L(M')$ for NBA $M' = (Q', \Sigma, \Delta', S', F')$ with
 - ▶ $Q' = Q \times \{1, \dots, k\}$
 - ▶ $S' = S \times \{1\}$
 - ▶ $\Delta'((q, i), a) = \Delta(q, a) \times \{j\}$ with $j = \begin{cases} i & \text{if } q \notin F_i \\ (i \bmod k) + 1 & \text{if } q \in F_i \end{cases}$
 - ▶ $F' = F_1 \times \{1\}$

Example

GBA $M = (\{1, 2, 3\}, \{a, b, c\}, \Delta, \{1\}, \{\{2\}, \{3\}\})$



NBA $M' = (\{11, 21, 31, 12, 22, 32\}, \{a, b, c\}, \Delta', \{11\}, \{21\})$



Outline

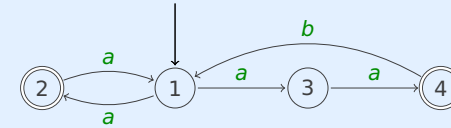
1. Summary of Previous Lecture
2. Büchi Automata
3. Intermezzo
4. Model Checking
5. Linear-Time Temporal Logic (LTL)
6. Further Reading

Outline

1. Summary of Previous Lecture
2. Büchi Automata
3. Intermezzo
4. Model Checking
5. Linear-Time Temporal Logic (LTL)
6. Further Reading

Question

Which statements about the following GBA $M = (\{1, 2, 3, 4\}, \{a, b\}, \Delta, \{1\}, F)$ are true ?



- A $L(M) = \{aa, aab\}^\omega$ if $F = \{\{2\}, \{3\}\}$
- B $L(M) = \{aa, aab\}^\omega$ if $F = \{\{2, 3\}\}$
- C it does not matter whether state 3 or 4 is final
- D there is only one possible choice for F such that $L(M) = \{aa, aab\}^\omega$



Formal Verification comprises

- ▶ framework for modeling systems (description language)
- ▶ specification language for describing properties to be verified
- ▶ verification method to establish whether description of system satisfies specification

Model Checking

automatic formal verification approach for concurrent systems based on temporal logic

Temporal Logic

- ▶ formulas are not statically true or false in model
- ▶ models of temporal logic contain several states and truth is dynamic
- ▶ formula can be true in some states and false in others

Model Checking

- ▶ models are transition systems $\mathcal{M}$
- ▶ properties are formulas φ in temporal logic
- ▶ model checker determines whether $\mathcal{M} \models \varphi$ is true or not

Two Temporal Logics

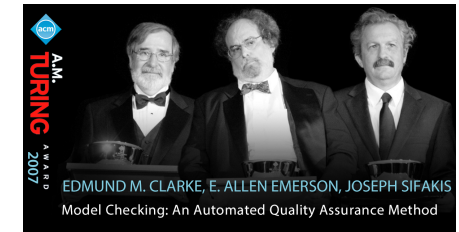
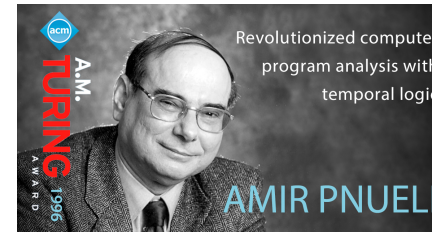
- ▶ computation tree logic (CTL)
- ▶ linear-time temporal logic (LTL)

Impact

both logics have been proven to be **extremely fruitful** in verifying hardware and communication protocols, and are increasingly applied to software verification

Outline

1. Summary of Previous Lecture
2. Büchi Automata
3. Intermezzo
4. Model Checking
5. Linear-Time Temporal Logic (LTL)
Syntax Semantics Adequacy
6. Further Reading



ACM Turing Awards

1996 Amir Pnueli

2007 Edmund M. Clarke, E. Allen Emerson, Joseph Sifakis

Definitions

- ▶ **LTL (linear-time temporal logic)** formulas are built from

- ▶ atoms $p, q, r, p_1, p_2, \dots$
- ▶ logical connectives $\perp, \top, \neg, \wedge, \vee, \rightarrow$
- ▶ **temporal connectives** **X, F, G, U, W, R**

according to following BNF grammar:

$$\varphi ::= \perp \mid \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \mathbf{X}\varphi \mid \mathbf{F}\varphi \mid \mathbf{G}\varphi \mid \varphi \mathbf{U}\varphi \mid \varphi \mathbf{W}\varphi \mid \varphi \mathbf{R}\varphi$$

- ▶ notational conventions:

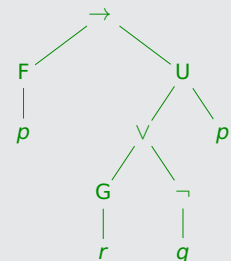
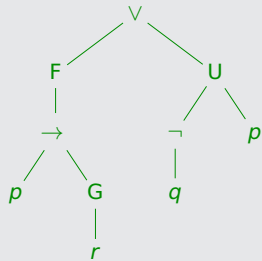
- ▶ binding precedence $\neg, \mathbf{X}, \mathbf{F}, \mathbf{G} > \mathbf{U}, \mathbf{W}, \mathbf{R} > \wedge, \vee > \rightarrow$
- ▶ omit outer parentheses
- ▶ $\rightarrow, \wedge, \vee$ are right-associative

Example

formula $F(p \rightarrow Gr) \vee \neg qUp$

formula $Fp \rightarrow (Gr \vee \neg q)Up$

parse tree



X next state

F $\exists$ future state

W weak until

U until

G $\forall$ states globally

R release

Outline

1. Summary of Previous Lecture

2. Büchi Automata

3. Intermezzo

4. Model Checking

5. Linear-Time Temporal Logic (LTL)

Syntax

Semantics

Adequacy

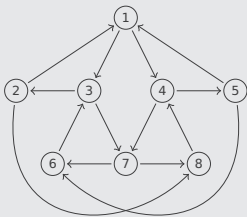
6. Further Reading

Definition

transition system (model) is triple $\mathcal{M} = (S, \rightarrow, L)$ with

- ① set of **states** S
- ② **transition relation** $\rightarrow \subseteq S \times S$ such that $\forall s \in S \exists t \in S$ with $s \rightarrow t$ ("no deadlock")
- ③ **labeling function** $L: S \rightarrow 2^{\text{atoms}}$

Example



model $\mathcal{M} = (S, \rightarrow, L)$

$S = \{1, 2, 3, 4, 5, 6, 7, 8\}$

$L(1) = \{I_A, I_B\}$

$L(5) = \{I_A, P_B\}$

$L(2) = \{P_A, I_B\}$

$L(6) = \{R_A, P_B\}$

$L(3) = \{R_A, I_B\}$

$L(7) = \{R_A, R_B\}$

$L(4) = \{I_A, R_B\}$

$L(8) = \{P_A, R_B\}$

Definition

- ▶ **path** in model $\mathcal{M} = (S, \rightarrow, L)$ is infinite sequence $s_1 \rightarrow s_2 \rightarrow \dots$
- ▶ $\forall$ paths $\pi = s_1 \rightarrow s_2 \rightarrow \dots \quad \forall i \geq 1 \quad \pi^i = s_i \rightarrow s_{i+1} \rightarrow \dots$

Definition

satisfaction of LTL formula φ with respect to path $\pi = s_1 \rightarrow s_2 \rightarrow \dots$ in model $\mathcal{M} = (S, \rightarrow, L)$

$\pi \models \varphi$

is defined by induction on φ :

$\pi \models \top$

$\pi \not\models \perp$

$\pi \models \varphi \wedge \psi$

$\iff \pi \models \varphi$ and $\pi \models \psi$

$\pi \models p \iff p \in L(s_1)$

$\pi \models \varphi \vee \psi$

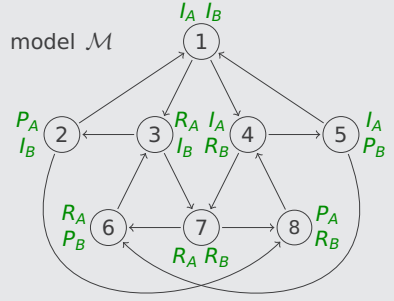
$\iff \pi \models \varphi$ or $\pi \models \psi$

$\pi \models \neg \varphi \iff \pi \not\models \varphi$

$\pi \models \varphi \rightarrow \psi$

$\iff \pi \not\models \varphi$ or $\pi \models \psi$

Example



$$\pi_1 = 1 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow \dots = (132)^\omega$$

$$\pi_2 = 7 \rightarrow 6 \rightarrow 3 \rightarrow 7 \rightarrow 6 \rightarrow 3 \rightarrow \dots = (763)^\omega$$

$$\pi_1 \models I_A$$

$$\pi_1 \not\models R_A \wedge I_B$$

$$\pi_1 \not\models I_B \rightarrow P_A \vee R_B$$

$$\pi_2 \not\models I_A$$

$$\pi_2^6 \models R_A \wedge I_B$$

$$\pi_2 \models I_B \rightarrow P_A \vee R_B$$

Definition

satisfaction of LTL formula φ with respect to path $\pi = s_1 \rightarrow s_2 \rightarrow \dots$ in model $\mathcal{M} = (S, \rightarrow, L)$

$$\pi \models \varphi$$

is defined by induction on φ :

$$\pi \models X\varphi \iff \pi^2 \models \varphi$$

$$\pi \models F\varphi \iff \exists i \geq 1 \pi^i \models \varphi$$

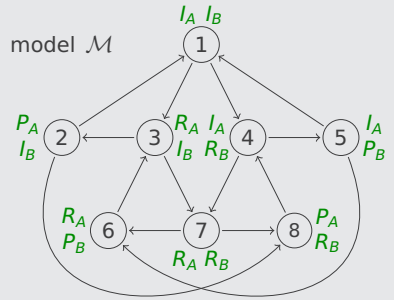
$$\pi \models G\varphi \iff \forall i \geq 1 \pi^i \models \varphi$$

$$\pi \models \varphi U \psi \iff \exists i \geq 1 \pi^i \models \psi \text{ and } \forall j < i \pi^j \models \varphi$$

$$\pi \models \varphi W \psi \iff (\exists i \geq 1 \pi^i \models \psi \text{ and } \forall j < i \pi^j \models \varphi) \text{ or } \forall i \geq 1 \pi^i \models \varphi$$

$$\pi \models \varphi R \psi \iff (\exists i \geq 1 \pi^i \models \varphi \text{ and } \forall j \leq i \pi^j \models \psi) \text{ or } \forall i \geq 1 \pi^i \models \psi$$

Example



$$\pi_1 = (132)^\omega \quad \pi_2 = (763)^\omega$$

$$\pi_1 \models X(R_A \vee R_B)$$

$$\pi_1 \models FP_A$$

$$\pi_1 \not\models XXP_B$$

$$\pi_2 \not\models FP_A$$

$$\pi_2 \models G\neg I_A$$

$$\pi_2 \models GF P_B$$

$$\pi_1 \not\models I_A U P_A$$

$$\pi_2 \models \neg I_A W P_A$$

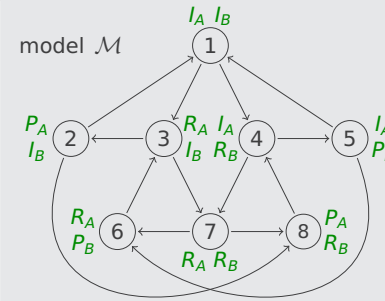
$$\pi_2 \not\models P_B R R_B$$

Definition

model $\mathcal{M} = (S, \rightarrow, L)$, state $s \in S$, LTL formula φ

$$\mathcal{M}, s \models \varphi \iff \forall \text{ paths } \pi = s \rightarrow \dots \quad \pi \models \varphi \quad \text{"formula } \varphi \text{ holds in state } s \text{ of model } \mathcal{M}"$$

Example



$$\mathcal{M}, 1 \not\models G(R_A \rightarrow FP_A)$$

$$\mathcal{M}, 4 \not\models \neg(R_B U P_B)$$

$$\mathcal{M}, 4 \not\models R_B U P_B$$

$$\mathcal{M}, 6 \models X(FI_B \wedge ((X\neg P_B) R R_A))$$

Definition

LTL formulas φ and ψ are **semantically equivalent** ($\varphi \equiv \psi$) if

$\forall$ models $\mathcal{M} = (S, \rightarrow, L)$

$\forall$ paths π in $\mathcal{M}$

$$\pi \models \varphi \iff \pi \models \psi$$

Theorem

$\neg X\varphi \equiv X\neg\varphi$	$\varphi U \psi \equiv \neg(\neg\psi U (\neg\varphi \wedge \neg\psi)) \wedge F\psi$
$\neg F\varphi \equiv G\neg\varphi$	$F(\varphi \vee \psi) \equiv F\varphi \vee F\psi$
$\neg G\varphi \equiv F\neg\varphi$	$G(\varphi \wedge \psi) \equiv G\varphi \wedge G\psi$
$\neg(\varphi U \psi) \equiv \neg\varphi R \neg\psi$	$F\varphi \equiv T U \varphi$
$\neg(\varphi R \psi) \equiv \neg\varphi U \neg\psi$	$G\varphi \equiv \perp R \varphi$
$\varphi U \psi \equiv \varphi W \psi \wedge F\psi$	$\varphi W \psi \equiv \psi R (\varphi \vee \psi)$
$\varphi W \psi \equiv \varphi U \psi \vee G\varphi$	$\varphi R \psi \equiv \psi W (\varphi \wedge \psi)$

Theorem

$$\varphi U \psi \equiv \neg(\neg\psi U (\neg\varphi \wedge \neg\psi)) \wedge F\psi$$

see overlay version for proof

Outline

1. Summary of Previous Lecture

2. Büchi Automata

3. Intermezzo

4. Model Checking

5. Linear-Time Temporal Logic (LTL)

Syntax

Semantics

Adequacy

6. Further Reading

Theorem

$\{X, U\}$, $\{X, W\}$ and $\{X, R\}$ are **adequate** sets of temporal connectives for LTL

Proof

▶ $F\varphi \equiv T U \varphi$	▶ $\varphi R \psi \equiv \psi W (\varphi \wedge \psi)$	▶ $\varphi U \psi \equiv \neg(\neg\varphi R \neg\psi)$
▶ $G\varphi \equiv \neg F\neg\varphi$	▶ $\varphi U \psi \equiv \neg(\neg\varphi R \neg\psi)$	▶ $F\varphi \equiv T U \varphi$
▶ $\varphi R \psi \equiv \neg(\neg\varphi U \neg\psi)$	▶ $F\varphi \equiv T U \varphi$	▶ $G\varphi \equiv \neg F\neg\varphi$
▶ $\varphi W \psi \equiv \varphi U \psi \vee G\varphi$	▶ $G\varphi \equiv \neg F\neg\varphi$	▶ $\varphi W \psi \equiv \varphi U \psi \vee G\varphi$

Outline

- 1. Summary of Previous Lecture
- 2. Büchi Automata
- 3. Intermezzo
- 4. Model Checking
- 5. Linear-Time Temporal Logic (LTL)
- 6. Further Reading

Esparza and Blondin

► Sections 11.1 and 13.2 of [Automata Theory: An Algorithmic Approach](#) (MIT Press 2023)

Important Concepts

- | | | |
|-------------------------------|------------------------------|-----|
| ► adequacy | ► GBA | ► U |
| ► generalized Büchi automaton | ► linear-time temporal logic | ► W |
| ► F | ► LTL | ► X |
| ► G | ► R | |

homework for January 9