TIMED AUTOMATA

LECTURE 18

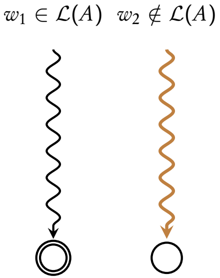
Theorem

Deterministic timed automata are **closed under complement**

Theorem

Deterministic timed automata are **closed under complement**

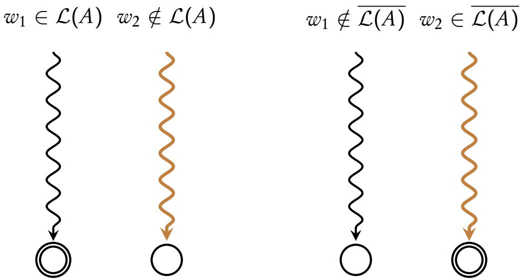
1. **Unique** run for every timed word



Theorem

Deterministic timed automata are **closed under complement**

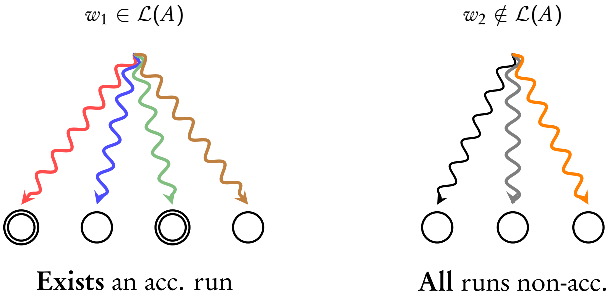
1. **Unique** run for every timed word
2. **Complementation: Interchange** acc. and non-acc. states



Theorem (Lecture 1)

Non-deterministic timed automata are **not closed under complement**

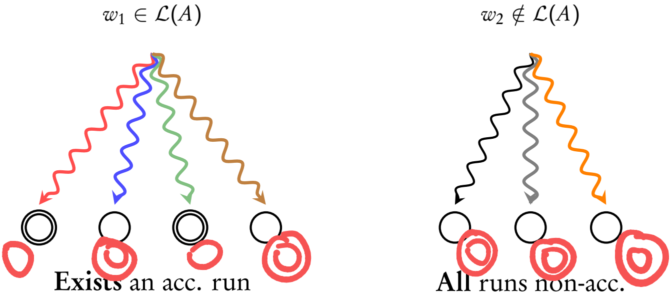
Many runs for a timed word



Theorem (Lecture 1)

Non-deterministic timed automata are **not closed under complement**

Many runs for a timed word



Complementation: **interchange** acc/non-acc + ask are **all runs acc.** ?

A timed automaton model with **existential** and **universal** semantics for acceptance

Alternating timed automata

Lasota and Walukiewicz. *FoSSaCS'05*, *ACM TOCL'2008*

Section 1:

Introduction to ATA

- X : set of **clocks**
- $\Phi(X)$: set of clock constraints σ (**guards**)

$$\sigma : \; x < c \mid x \leq c \mid \sigma_1 \wedge \sigma_2 \mid \neg\sigma$$

c is a non-negative **integer**

- Timed automaton A: $(Q, Q_0, \Sigma, X, T, F)$

$$T \; \subseteq \; Q \times \Sigma \times \Phi(X) \times Q \times \mathcal{P}(X)$$

$$T \subseteq Q \times \Sigma \times \Phi(X) \times Q \times \mathcal{P}(X)$$

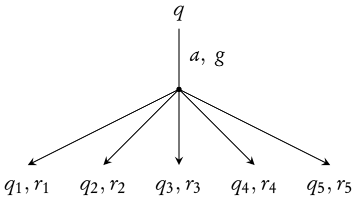
(q_1, a, g, q_1, r_1)

(q_1, a, g, q_2, r_2)

$\vdots$

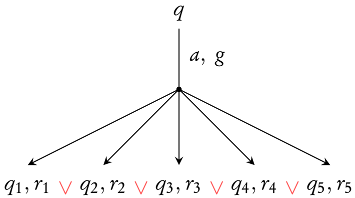
$(q_1, a, g, q_5, r_5$

$$T : Q \times \Sigma \times \Phi(X) \mapsto \mathcal{P}(Q \times \mathcal{P}(X))$$



$$T \subseteq Q \times \Sigma \times \Phi(X) \times Q \times \mathcal{P}(X)$$

$$T : Q \times \Sigma \times \Phi(X) \mapsto \mathcal{P}(Q \times \mathcal{P}(X))$$



$$T : Q \times \Sigma \times \Phi(X) \mapsto \mathcal{P}(Q \times \mathcal{P}(X))$$

$$T : Q \times \Sigma \times \Phi(X) \mapsto \mathcal{P}(Q \times \mathcal{P}(X))$$

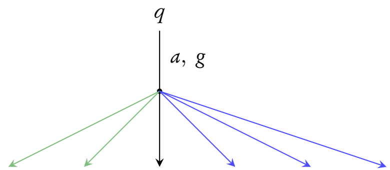
$\downarrow$ $\mathcal{B}^+(S)$ is all $\phi ::= S \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2$

$$T : Q \times \Sigma \times \Phi(X) \ \mapsto\ \mathcal{B}^+(Q \times \mathcal{P}(X))$$

$$T : Q \times \Sigma \times \Phi(X) \mapsto \mathcal{P}(Q \times \mathcal{P}(X))$$

$\mathcal{B}^+(S)$ is all $\phi ::= S \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2$

$$T : Q \times \Sigma \times \Phi(X) \mapsto \mathcal{B}^+(Q \times \mathcal{P}(X))$$



$$(q_1, r_1 \wedge q_2, r_2) \vee (q_3, r_3) \vee (q_4, r_4 \wedge q_5, r_5 \wedge q_6, r_6)$$

Alternating Timed Automata

An **ATA** is a tuple $A = (Q, q_0, \Sigma, X, T, F)$ where:

$$T : Q \times \Sigma \times \Phi(X) \mapsto \mathcal{B}^{+}(Q \times \mathcal{P}(X))$$

is a **finite partial function.**

Alternating Timed Automata

An **ATA** is a tuple $A = (Q, q_0, \Sigma, X, T, F)$ where:

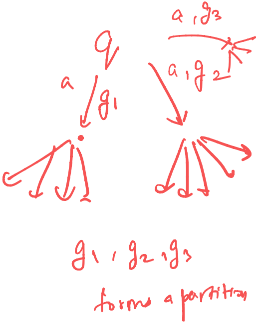
$$T : Q \times \Sigma \times \Phi(X) \mapsto \mathcal{B}^+(Q \times \mathcal{P}(X))$$

is a **finite partial function.**

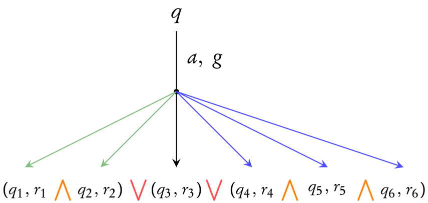
Partition: For every q, a the set

$$\{ \, [\sigma] \mid T(q, a, \sigma) \text{ is defined } \}$$

gives a finite partition of $\mathbb{R}_{\geq 0}^X$

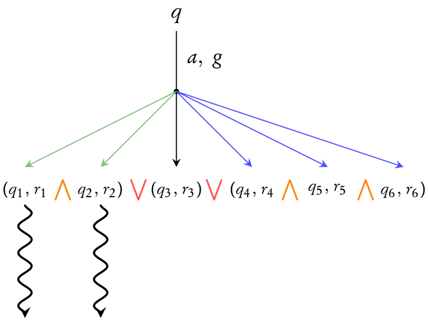


Acceptance



Accepting run from q iff:

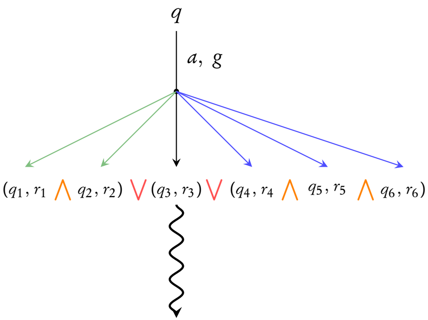
Acceptance



Accepting run from q iff:

- accepting run from q_1 **and** q_2,

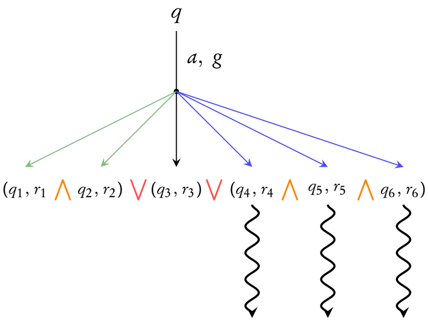
Acceptance



Accepting run from q iff:

- accepting run from q_1 **and** q_2,
- **or** accepting run from q_3,

Acceptance

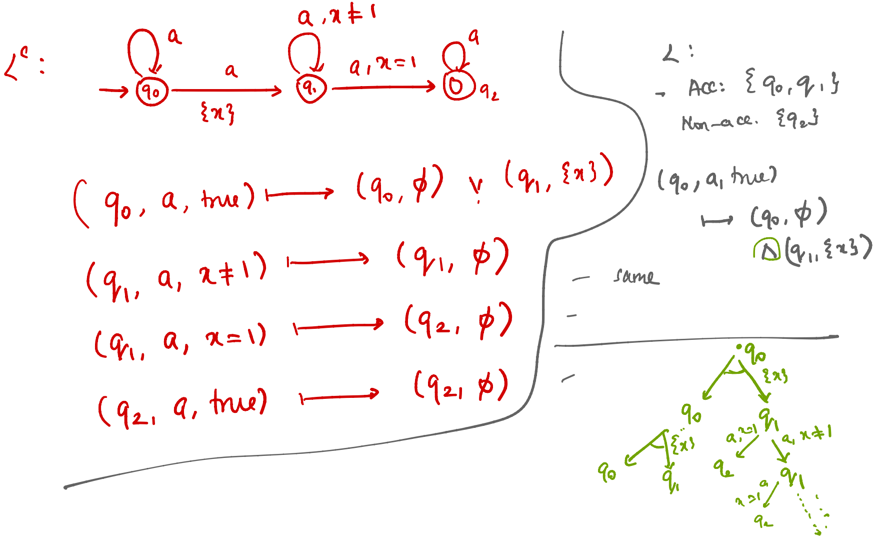


Accepting run from q iff:

- accepting run from q_1 **and** q_2,
- **or** accepting run from q_3,
- **or** accepting run from q_4 **and** q_5 **and** q_6

L : timed words over $\{a\}$ containing **no two** a's at distance 1

(Not expressible by non-deterministic TA)



L : timed words over $\{a\}$ containing **no two** $a's$ at distance 1

(Not expressible by non-deterministic TA)

ATA:

$$q_0, a, tt \;\mapsto\; (q_0, \emptyset) \wedge (q_1, \{x\})$$

$$q_1, a, x = 1 \;\mapsto\; (q_2, \emptyset)$$

$$q_1, a, x \neq 1 \;\mapsto\; (q_1, \emptyset)$$

$$q_2, a, tt \;\mapsto\; (q_2, \emptyset)$$

q_0, q_1 are acc., q_2 is non-acc.

$(a, 0)\ (a, 0.5)\ (a, 0.8)\ (a, 1)\ (a, 1.2) \in L$?

$(q_0, 0)$ $(q_1, 0)$ $(q_1, 0.5)$ $(q_1, 0.8)$ $(q_2, 1)$ $(q_2, 1.2)$

0 - $(q_0, \phi) \wedge (q_1, \{x\})$ 0.5 0.8 1

↪ Adam wins.

$\Rightarrow w \notin L$.

Eve. — Exists

Adam — For All

- Eve wins if the final state reached in the play is accepting.

$(q_0, a, true) \longmapsto (q_0, \phi) \wedge (q_1, \{x\})$

$(q_1, a, x \neq 1) \longmapsto (q_1, \phi)$

$(q_1, a, x = 1) \longmapsto (q_2, \phi)$

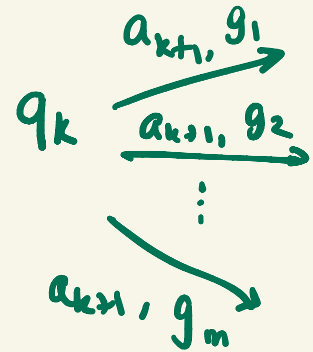
$(q_2, a, true) \longmapsto (q_2, \phi)$

Acceptance Game: $\mathcal{G}_{A,w}$

$$w := (a_0, t_0)\,(a_1, t_1)\,(a_2, t_2)\,\dots\,(a_{k+1}, t_{k+1})\,\dots\,(a_n, t_n)$$

Phase 0 — Phase 1 — … — (q_k, v_k) — Phase k+1

$$\bar{v} = v_k + t_{k+1} - t_k$$



- Let σ be unique constraint s.t. $\bar{v}$ satisfies σ

$$b = \delta(q_{k+1}, a_{k+1}, \sigma)$$

- $b = b_1 \wedge b_2$: Adam chooses a subformula and game continues with the subformula.

- $b = b_1 \vee b_2$: Eve

- $b = (q, r) \in Q \times P(C)$

- Phase ends with

$$(q_{k+1}, v_{k+1}) := (q, \bar{v}[r := 0])$$

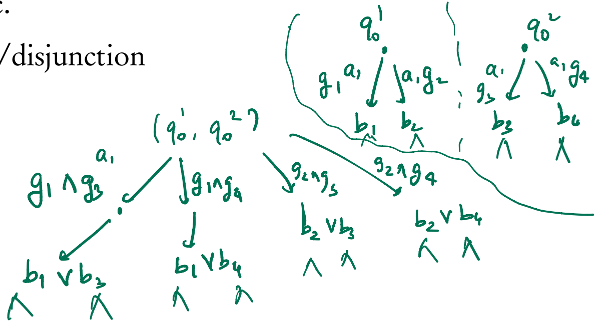
→ Play ends with (q_{n+1}, v_{n+1})

- Eve wins the play if q_{n+1} is accepting.

- $w \in \mathcal{L}(A)$ if Eve has a strategy to win $G_{A,w}$. Else $w \notin \mathcal{L}(A)$.

Closure properties

- Union, intersection: use disjunction/conjunction
- Complementation: **interchange**
  1. acc./non-acc.
  2. conjunction/disjunction



Closure properties

- Union, intersection: use disjunction/conjunction
- Complementation: **interchange**
  1. acc./non-acc.
  2. conjunction/disjunction

No change in the number of clocks!

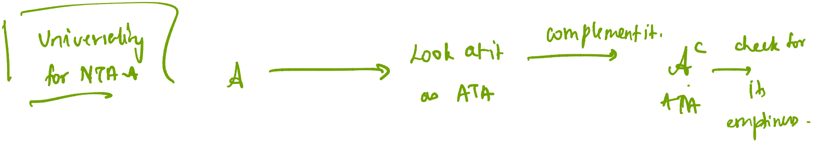
Section 2:

The 1-clock restriction

- Emptiness: given A, is $\mathcal{L}(A)$ empty
- Universality: given A, does $\mathcal{L}(A)$ contain all timed words
- Inclusion: given A, B, is $\mathcal{L}(A) \subseteq \mathcal{L}(B)$

- Emptiness: given A, is $\mathcal{L}(A)$ empty
- Universality: given A, does $\mathcal{L}(A)$ contain all timed words
- Inclusion: given A, B, is $\mathcal{L}(A) \subseteq \mathcal{L}(B)$

Undecidable for **two clocks or more** (~~via Lecture 3~~)



- Emptiness: given A, is $\mathcal{L}(A)$ empty
- Universality: given A, does $\mathcal{L}(A)$ contain all timed words
- Inclusion: given A, B, is $\mathcal{L}(A) \subseteq \mathcal{L}(B)$

Undecidable for **two clocks or more** (via Lecture 3)

Decidable for **one clock** ~~(via Lecture 4)~~

- Emptiness: given A, is $\mathcal{L}(A)$ empty
- Universality: given A, does $\mathcal{L}(A)$ contain all timed words
- Inclusion: given A, B, is $\mathcal{L}(A) \subseteq \mathcal{L}(B)$

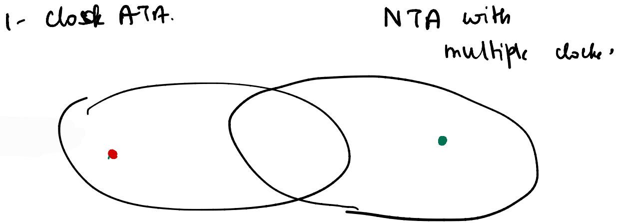
Undecidable for **two clocks or more** (~~via Lecture 3~~)

Decidable for **one clock** (~~via Lecture 4~~)

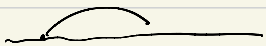
Restrict to one-clock ATA

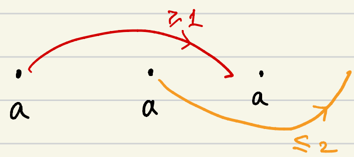
Theorem

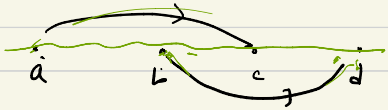
Languages recognizable by 1-clock ATA and (many clock) TA are **incomparable**



Alternation Vs. Multiple clocks:

Alternation:  For every point, ∃ another at distance c.

Multiple clocks: 

Interleaving. 

Need multiple clocks

Example L: no a's at distance 1. → 1-clock ATA. but no NTA.

Interleaving eg. need multiple clocks, but no 1-ATA

→ proper proof in the next class.

- Emptiness for 1-clock ATA is decidable.
  ↳ Proof goes along similar lines as proof of decidability of universality of 1-clock NTA.

Section 3:

Complexity

- $\mathcal{L}(B) \subseteq \mathcal{L}(A)$
  ↳ A is 1-clock ATA, B is an NTA.

↳ decidable.

Lower bound

Complexity of emptiness of **purely universal** 1-clock ATA is **not** bounded by a **primitive recursive** function

Lower bound

Complexity of emptiness of **purely universal** 1-clock ATA is **not** bounded by a **primitive recursive** function

$\Rightarrow$ complexity of Ouaknine-Worrell algorithm for **universality** of 1-clock TA is **non-primitive recursive**

Summary:

- ATA
  - ↳ closed under union, intersection, complement

- Emptiness is undecidable in general for ATA.

- Restricting to 1-clock ATA make emptiness & universality decidable.
  - further: $\mathcal{L}(B) \subseteq \mathcal{L}(A)$ when B is NTA, A is 1-clock ATA is decidable.

Proof similar to Ouaknine-Worrell algorithm.

- 1-Clock ATA and (many clock) NTA are incomparable w.r.t. expressive power.

Next class:

- Proof of incomparable expressive power
- Primitive recursive functions / Non-primitive recursive complexity.