TIMED AUTOMATA

LECTURE 20

Alternating Timed Automata:

- What we have seen so far?

  - Model is closed under union, intersection, complement
  - Emptiness is undecidable for general ATA
  - Consider 1-clock ATA
    - ↦ Expressive power incomparable to many clock NTA.

Today:

- Emptiness is decidable for 1-clock ATA (idea of proof)
- Complexity of the emptiness problem

Algorithm for the emptiness problem for 1-ATA:

Given a 1-clock ATA $\mathcal{A}$, is $\mathcal{L}(\mathcal{A})$ empty?

- Algorithm similar to Ouaknine-Worell algorithm for universality of 1-NTA
- Now we need to handle both universal and existential transitions.

Assumption:

- boolean combinations in the transitions are in

  disjunctive normal form

$$(\cdot \wedge \cdot \wedge \cdot \cdots) \vee (\cdot \wedge \cdot \wedge \cdot \wedge \cdots) \vee \cdots \vee (\cdot \wedge \cdot \wedge \cdots \wedge)$$

Labelled transition system: $T(A)$

Configuration P: $\{ (q_1, v_1)\ (q_2, v_2), \dots, (q_k, v_k) \}$

a set of states

(location of automaton, value of clock)

Transitions between configurations:

$$P \xrightarrow{t, a} P'$$

For each $(q, v) \in P$

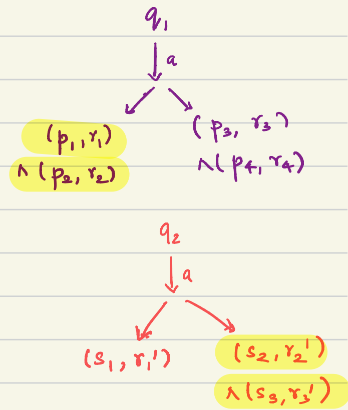
- let $v' = v + t$
- let $b = \delta(q, a, \sigma)$ for the uniquely determined σ satisfied by v'
- choose one of the disjuncts of b : $(q_1, r_1) \wedge (q_2, r_2) \wedge \dots \wedge (q_k, r_k)$
- $Next_{(q,v)} := \{ (q_i, v'[r_i := 0]) \mid i = 1, \dots, k \}$

Then, $P' = \bigcup_{(q,v) \in P} Next_{(q,v)}$

$$\{ (q_1, v_1) \,,\, (q_2, v_2) \}$$

$$\Big\downarrow a, t$$

$$\{ (p_1, \cdot)(p_2, \cdot) \,,\, (s_2, \cdot), (s_3, \cdot) \}.$$



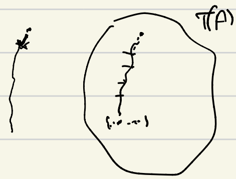
One possible transition in $T(\mathcal{A})$

Good node: All states are accepting

Theorem: $\mathcal{L}(\mathcal{A})$ is non-empty

iff

$T(\mathcal{A})$ has a path to a good node from the initial configuration



Rest of the algorithm similar to OW-05.

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

NTA: (pure existential). → checking universality of NTA $\mathcal{A}$

$\equiv$ emptiness of $\mathcal{A}^c$ (purely universal ATA)

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

emptiness of purely universal 1-ATA $\xrightarrow[\text{to}]{\text{reduced}}$ universality of purely existential 1-ATA (1-NTA)

Primitive recursive functions

Functions $f : \cancel{\mathbb{N} \mapsto \mathbb{N}}$ $\underline{\mathbb{N}^k} \longmapsto \underline{\mathbb{N}^\ell}$ $k \geq 0$

Basic primitive recursive functions:

- **Zero function:** $Z() = 0$
- **Successor function:** $Succ(n) = n + 1$
- **Projection function:** $P_i(x_1, \ldots, x_n) = x_i$

Operations:

$g : \ell_1 \longrightarrow \ell_2$ $h(g(\ldots))$

$h : \ell_2 \longrightarrow \ell_3$ $: \ell_1 \longrightarrow \ell_3$

- **Composition**
- **Primitive recursion:** if f and g are p.r. of arity k and $k + 2$, there is a p.r. h of arity $k + 1$:

$$
\begin{aligned}
h(0, x_1, \ldots, x_k) &= f(x_1, \ldots, x_k) \\
h(n + 1, x_1, \ldots, x_k) &= \underline{g}(\underbrace{h(n, x_1, \ldots, x_k)}, \underline{n}, \underbrace{x_1, \ldots, x_k})
\end{aligned}
$$

$h(n, x_1 \ldots x_k)$

Addition:

$$
\begin{aligned}
Add(0, y) &= y \\
Add(\underline{n} + 1, \underline{y}) &= Succ(Add(n, y))
\end{aligned}
$$

2

$Succ(P_1(Add(n,y), n, y))$

Addition:

$$\begin{aligned} Add(\mathtt{0}, y) &= y \\ Add(n+1, y) &= Succ(Add(n, y)) \end{aligned}$$

Multiplication:

$$\begin{aligned} Mult(\mathtt{0}, y) &= Z() \\ Mult(n+1, y) &= Add(Mult(n, y), y) \end{aligned}$$

Addition:

$$
\begin{aligned}
Add(0, y) &= y \\
Add(n+1, y) &= Succ(Add(n, y))
\end{aligned}
$$

Multiplication:

$$
\begin{aligned}
Mult(0, y) &= Z() \\
Mult(n+1, y) &= Add(Mult(n, y), y)
\end{aligned}
$$

Exponentiation 2^n:

$$
\begin{aligned}
Exp(0) &= Succ(Z()) \\
Exp(n+1) &= Mult(Exp(n), 2)
\end{aligned}
$$

$2^{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}$ tower n 2's

Addition:

$$
\begin{aligned}
Add(0, y) &= y \\
Add(n+1, y) &= Succ(Add(n, y))
\end{aligned}
$$

Multiplication:

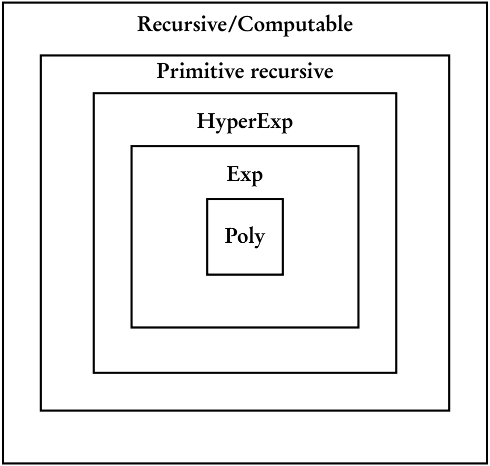
$$
\begin{aligned}
Mult(0, y) &= Z() \\
Mult(n+1, y) &= Add(Mult(n, y), y)
\end{aligned}
$$

Exponentiation 2^n:

$$
\begin{aligned}
Exp(0) &= Succ(Z()) \\
Exp(n+1) &= Mult(Exp(n), 2)
\end{aligned}
$$

Hyper-exponentiation (tower of n two-s):

$$
\begin{aligned}
HyperExp(0) &= Succ(Z()) \\
HyperExp(n+1) &= Exp(HyperExp(n))
\end{aligned}
$$

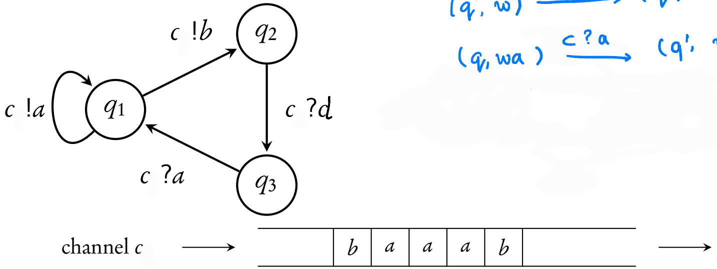


Recursive but not primitive rec.: Ackermann function, Sudan function

Coming next: a problem that has complexity non-primitive recursive

Channel systems

$$(q, w) \xrightarrow{c!a} (q', aw)$$
$$(q, wa) \xrightarrow{c?a} (q', w)$$



Finite state description of communication protocols
G. von Bochmann. 1978

On communicating finite-state machines
D. Brand and P. Zafiropulo. 1983

Theorem [BZ'83]

Reachability in channel systems is **undecidable**

Coming next: modifying the model for decidability

Lossy channel systems

Finkel'94, Abdulla and Jonsson'96

Messages stored in channel can be **lost** during transition

$(q, w) \xrightarrow{c!a} (q', w')$ where w' is a subword of aw

$(q, wa) \xrightarrow{c?a} (q', w'')$ where w'' is a subword of w

Lossy channel systems

Finkel'94, Abdulla and Jonsson'96

Messages stored in channel can be **lost** during transition

Theorem [Schnoebelen'2002]

Reachability for **lossy one-channel** systems is **non-primitive recursive**

Reachability problem for **lossy one-channel** systems can be reduced to emptiness problem for **purely universal 1-clock ATA**

1-clock ATA

- **closed** under boolean operations
- **decidable** emptiness problem
- expressivity **incomparable** to many clock TA
- **non-primitive recursive** complexity for emptiness

1-clock ATA

- **closed** under boolean operations
- **decidable** emptiness problem
- expressivity **incomparable** to many clock TA
- **non-primitive recursive** complexity for emptiness

- Other results: **Undecidability** of:
  - 1-clock ATA + ε-transitions
  - 1-clock ATA over infinite words

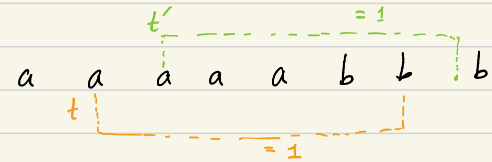
Exercise:

- Construct an ATA for the language consisting of words $a^n b^m$ such that:

  for every 'b' there is an 'a' at exactly one time unit before its occurrence

$$\mathcal{L} = \{ (a^n b^m, \tau_1 \tau_2 \dots \tau_{n+m}) \mid n, m \geq 1,$$
$$\forall j : n+1 \leq j \leq n+m \Rightarrow \exists i \leq n \text{ s.t. } \tau_j - \tau_i = 1 \}$$

Idea:



1. For every pair of consecutive a's occurring, say at t and t':

   there is no 'b' in the open interval $(t+1, t'+1)$

2. If the first 'a' occurs at t_f, there is no 'b' in (t_f, t_f+1)

3. If the last 'a' occurs at t_ℓ, there is no 'b' in $(t_\ell+1, \infty)$

Claim: A word $w = (a^n b^m, \tau)$ satisfies all the above properties,

iff

$w \in \mathcal{L}$.

Proof: Let $w = \begin{array}{cccccccc} a & a & \cdots & a & b & b & \cdots & b \\ \tau_1 & \tau_2 & & \tau_n & \tau_{n+1} & & \cdots & \tau_{n+m} \end{array}$

Due to (2) and (3) : $\tau_1 + 1 \leq \tau_{n+j} \leq \tau_n + 1 \quad \forall j \in \{1 \ldots m\}$

From (1) : $\tau_{n+j} \notin (\tau_i + 1, \tau_{i+1} + 1) \quad \forall i \in \{1 \ldots n\}$
$\forall j \in \{1 \ldots m\}$

Therefore each $\tau_{n+j} \in \{\tau_1 + 1, \tau_2 + 1, \ldots, \tau_n + 1\}$

Construct ATA for each, (1), (2), (3) and take intersection.

For every pair of consecutive a's occurring, say at t and t':

there is no 'b' in the open interval $(t+1, t'+1)$

$q_0 \xrightarrow{a} (q_x, \{x\}) \wedge (q_0, \phi)$ $\qquad$ $q_0 \xrightarrow{b} q_0$

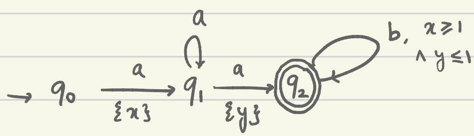
$q_x \xrightarrow{a} (q_y, \{y\})$ $\qquad$ $q_x \xrightarrow{b} (q_x, \phi)$

$q_y \xrightarrow{a} (q_y, \phi)$

$q_y \xrightarrow{y<1<x,\ b} (q_{reject}, \phi)$ $\qquad$ $q_y \xrightarrow{\neg(y<1<x),\ b} (q_y, \phi)$

Accepting states: $\{q_0, q_x, q_y\}$ $\qquad$ Reject states: $\{q_{reject}\}$

- If the first 'a' occurs at t_f, there is no 'b' in (t_f, t_f+1)
- If the last 'a' occurs at t_ℓ, there is no 'b' in $(t_\ell+1, \infty)$



$\cup$