DETERMINISABILITY & DETERMINISTIC SEPARABILITY of TIMED AUTOMATA

Lorenzo Clemente, joint work with
Sławek Lasota & Radek Piórkowski.
University of Warsaw (Poland)

SUMMARY

- Introduction to timed automata.
- Determinisability & undecidability.
- Deterministic separability.
  → reduction to timed games.
- Timed Büchi-Landweber games: decidability & undecidability.
  → Techical novelty: constant synthesis.
- Future work.

TIMED AUTOMATA (NTA)

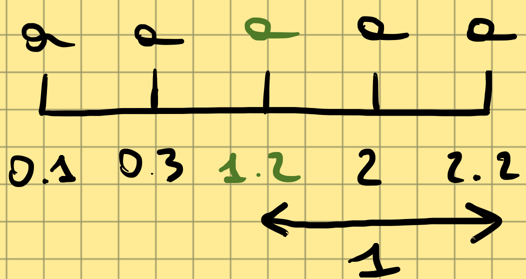
- Finite automata extended with clocks $\{x, y\}$ which can be reset $x := 0$ and tested $x > 2$, $y = 1$, ...
- They recognise monotonic timed languages $L \subseteq (\Sigma \times \mathbb{R}_{\geq 0})^*$

TIMED AUTOMATA (NTA)

- Finite automata extended with clocks $\{x, y\}$ which can be reset $x := 0$ and tested $x > 2$, $y = 1$, ...
- They recognise monotonic timed languages $L \subseteq (\Sigma \times \mathbb{R}_{\geq 0})^*$

EXAMPLE $L =$ "last letter occurred 1 t.u. earlier", $\Sigma = \{a\}$.

$w =$

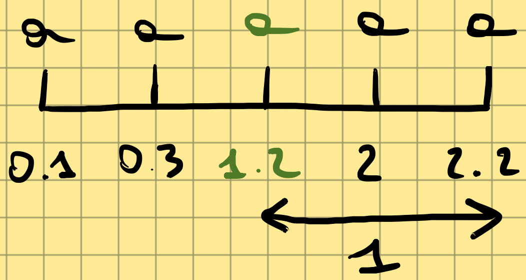


$\in L$

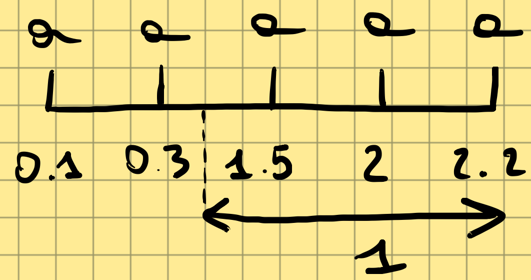
TIMED AUTOMATA (NTA)

- Finite automata extended with clocks $\{x, y\}$ which can be reset $x := 0$ and tested $x > 2$, $y = 1$, ...
- They recognise monotonic timed languages $L \subseteq (\Sigma \times \mathbb{R}_{\geq 0})^*$

EXAMPLE $L =$ "last letter occurred 1 t.u. earlier", $\Sigma = \{a\}$.

$w =$ 

$\in L$

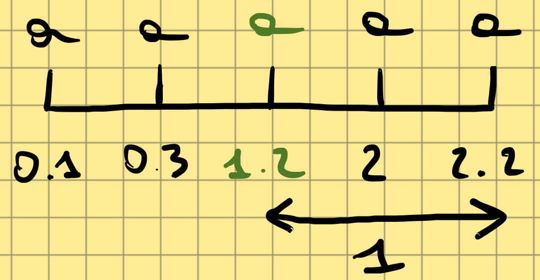
$w =$ 

$\notin L$

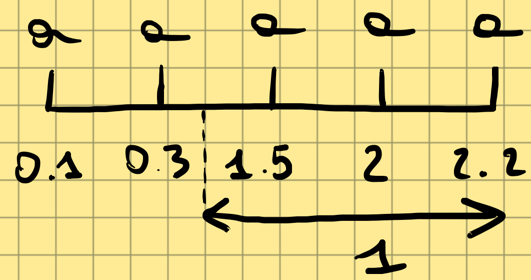
TIMED AUTOMATA (NTA)

- Finite automata extended with clocks $\{x, y\}$ which can be reset $x := 0$ and tested $x > 2$, $y = 1$, ...
- They recognise monotonic timed languages $L \subseteq (\Sigma \times \mathbb{R}_{\geq 0})^*$

EXAMPLE L = "last letter occurred 1 t.u. earlier", $\Sigma = \{a\}$.

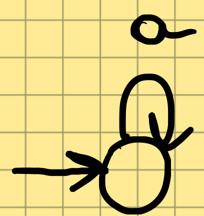
w = 

$\in L$

w = 

$\notin L$

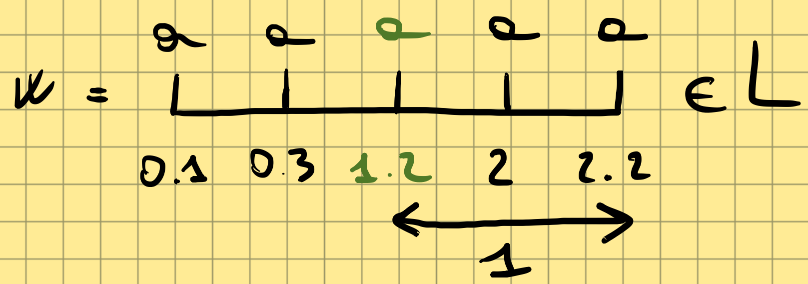
NTA:

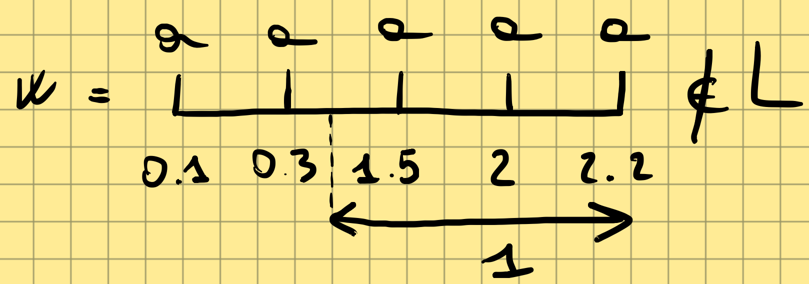


TIMED AUTOMATA (NTA)

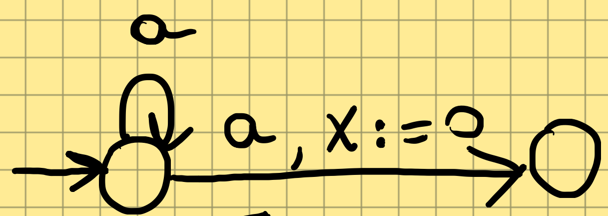
- Finite automata extended with clocks $\{x,y\}$ which can be reset $x:=0$ and tested $x>2$, $y=1$, ...
- They recognise monotonic timed languages $L \subseteq (\Sigma \times \mathbb{R}_{\geq 0})^*$

EXAMPLE $L =$ "last letter occurred 1 t.u. earlier", $\Sigma = \{a\}$.





NTA:

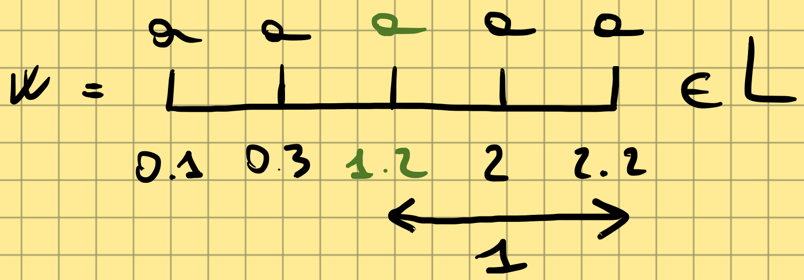


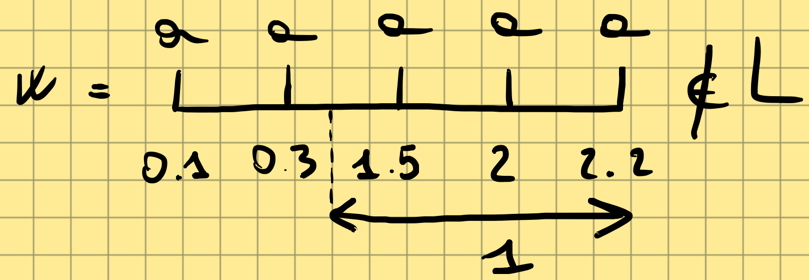
guess occurrence of a,
reset the clock x

TIMED AUTOMATA (NTA)

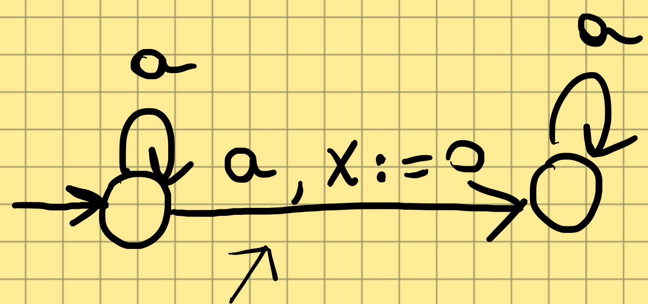
- Finite automata extended with clocks $\{x,y\}$ which can be reset $x:=0$ and tested $x>2$, $y=1$, ...
- They recognise monotonic timed languages $L \subseteq (\Sigma \times \mathbb{R}_{\geq 0})^*$

EXAMPLE L = "last letter occurred 1 t.u. earlier", $\Sigma = \{a\}$.





NTA:

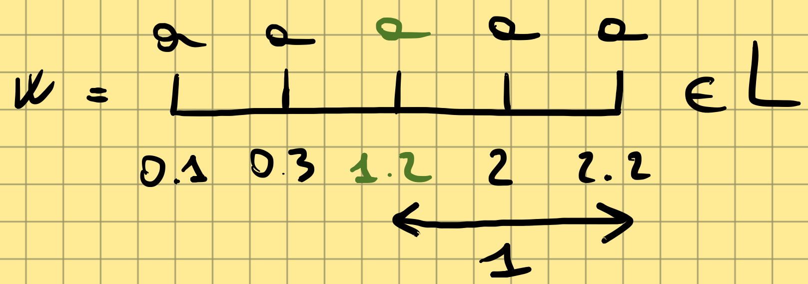


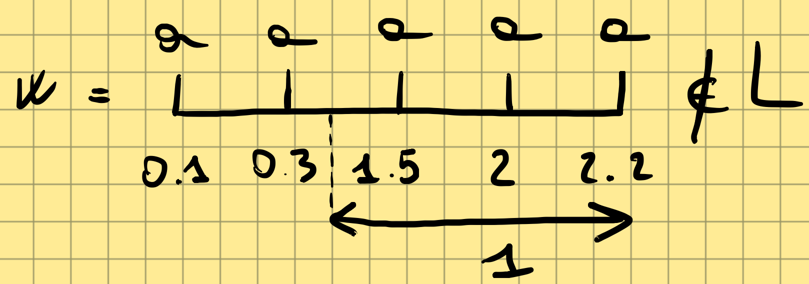
guess occurrence of a,
reset the clock x

TIMED AUTOMATA (NTA)

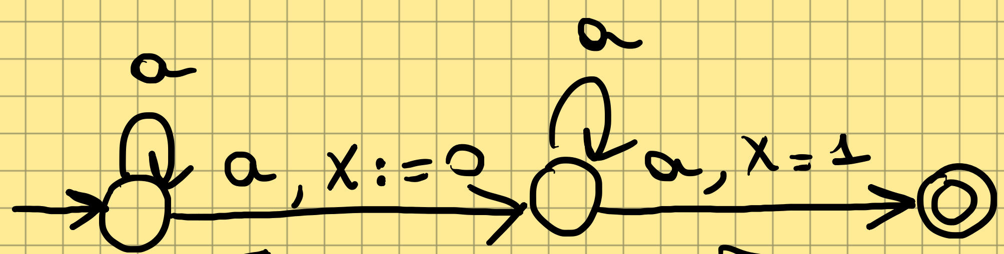
- Finite automata extended with clocks $\{x,y\}$ which can be reset $x:=0$ and tested $x>2$, $y=1$, ...
- They recognise monotonic timed languages $L \subseteq (\Sigma \times \mathbb{R}_{\geq 0})^*$

EXAMPLE $L =$ "last letter occurred 1 t.u. earlier", $\Sigma = \{a\}$.





NTA:



guess occurrence of a, reset the clock x

last occurrence of a

DETERMINISTIC TIMED AUTOMATA LESS EXPRESSIVE

L = "last letter occurred 1 t.u. earlier", over $\Sigma = \{a\}$

- NTA for L with 1-clock:

guess the second-last occurrence and measure 1 t.u.

- No DTA for L: deterministically ∞-many clocks are required to store unboundedly many timestamps.

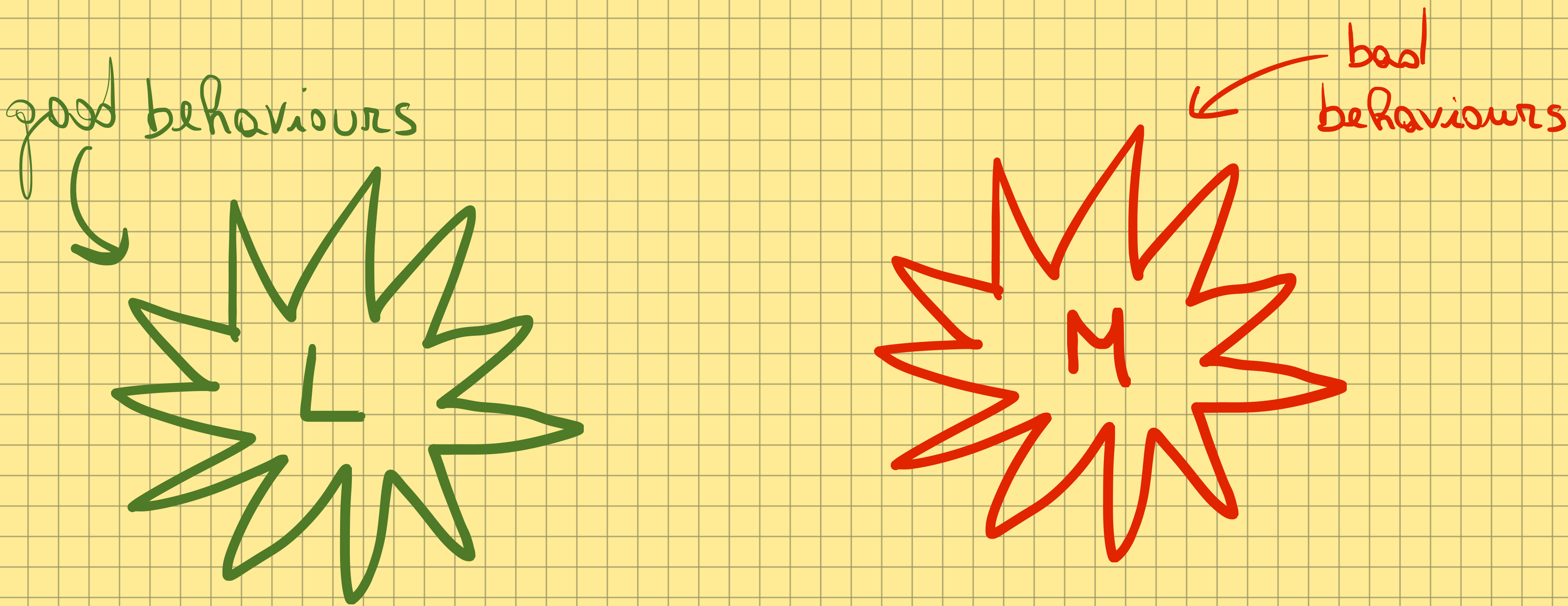
  → any DTA with k clocks can be "fooled" since some important timestamp must be forgotten.

DETERMINISABILITY

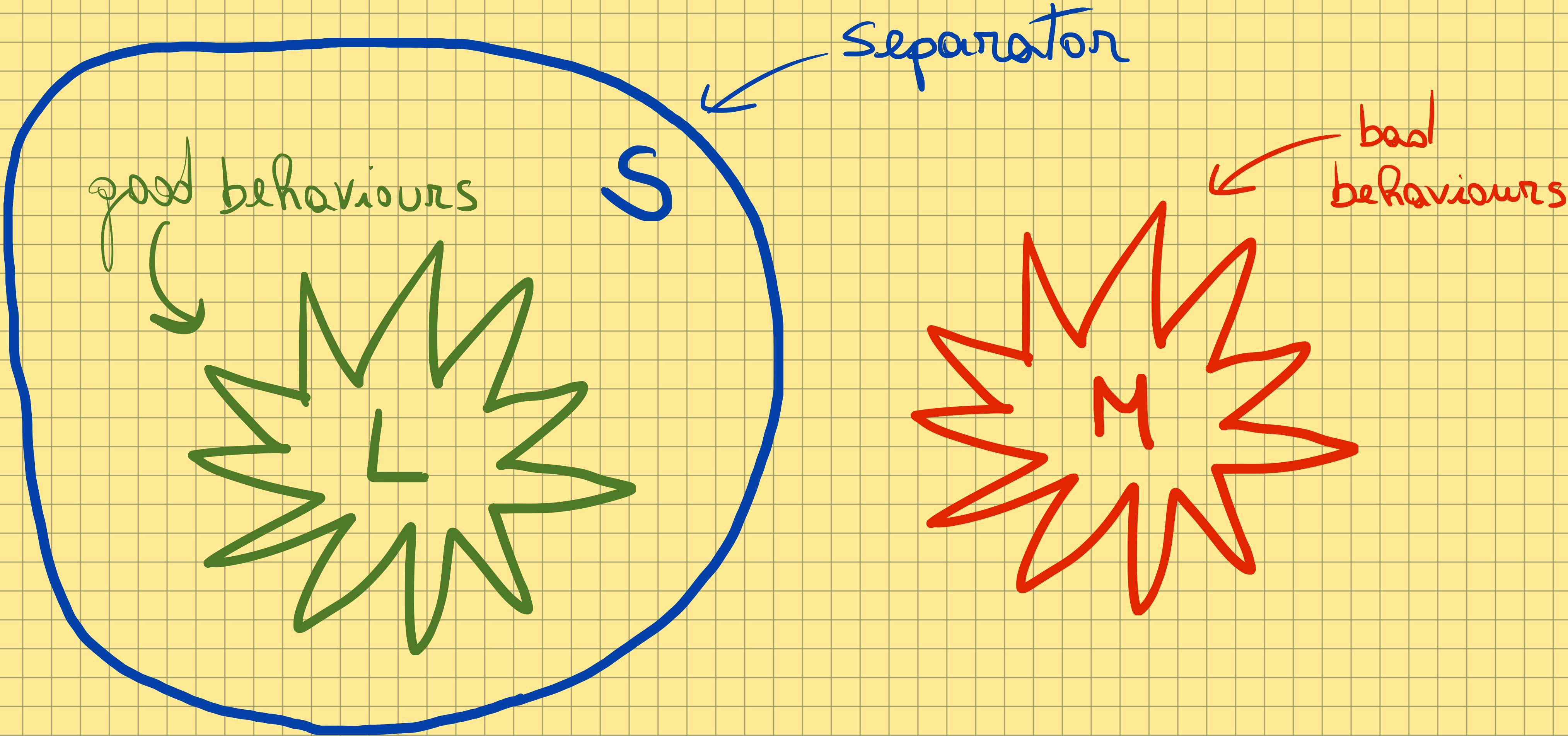
INPUT \ OUTPUT	DTA, K clocks, constants ≤ M	DTA, K clocks	DTA
NTA, ≥ 2 clocks	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)
NTA, 1 clock, ε-transitions	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)

(1) [Finkel '06], [Tripakis'06]

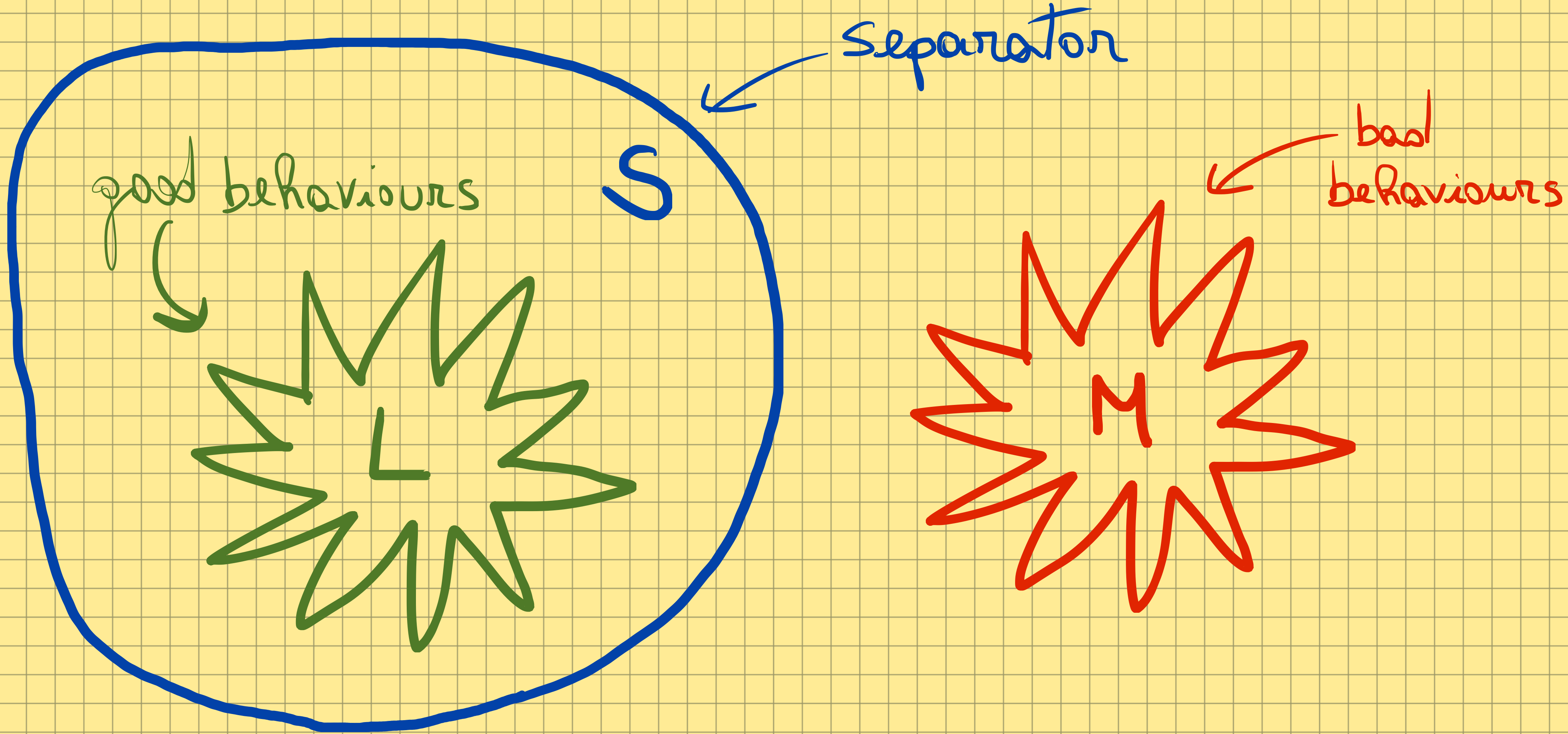
DETERMINISTIC SEPARABILITY



DETERMINISTIC SEPARABILITY



DETERMINISTIC SEPARABILITY



L, M are **complex** : NTA. S is **simple** : DTA.

WHY SEPARABILITY ?

checkable !

- Simple explanation of disjointness.
  → Disjointness undecidable, but separability decidable.
  Example: CFL and piecewise-testable separability.
  [Czerwiński, Martens, van Rooijen, Zeitoun, Zetzsche '15-17]

WHY SEPARABILITY ?

checkable !

- Simple explanation of disjointness.
  → Disjointness undecidable, but separability decidable.
  Example : CFL and piecewise-testable separability.
  [Czerwiński, Martens, van Rooijen, Zeitoun, Zetzsche '15 - 17]

- Generalisation of learning from positive & negative examples.

WHY SEPARABILITY ?

- Simple explanation of disjointness. → checkable!
  → Disjointness undecidable, but separability decidable.
  Example: CFL and piecewise-testable separability.
  [Czerwiński, Martens, van Rooijen, Zeitoun, Zetzsche '15 - 17]

- Generalisation of learning from positive & negative examples.
- Approximate determinisability with guarantees.
  → Determinisability undecidable, but det. sep. decidable.
  Example: this work!

WHY SEPARABILITY ?

checkable !

- Simple explanation of disjointness.
  → Disjointness undecidable, but separability decidable.
  Example : CFL and piecewise-testable separability.
  [Czerwiński, Martens, van Rooijen, Zeitoun, Zetzsche '15 - 17]

- Generalisation of learning from positive & negative examples.
- Approximate determinisability with guarantees.
  → Determinisability undecidable, but det. sep. decidable.
  Example : this work !
- Deeper understanding of nondeterminism vs. determinism.

DETERMINISTIC SEPARABILITY NONTRIVIAL

L = "last letter occurred 1 t.u. earlier" (1-clock NTA, not DTA).

M = complement of L (2-clock NTA).

DETERMINISTIC SEPARABILITY NONTRIVIAL

L = "last letter occurred 1 t.u. earlier" (1-clock NTA, not DTA).

M = complement of L (2-clock NTA).

$w \in M$ iff either: 1) $|w| = 1$, or

DETERMINISTIC SEPARABILITY NONTRIVIAL

L = "last letter occurred 1 t.u. earlier" (1-clock NTA, not DTA).

M = complement of L (2-clock NTA).

$w \in M$ iff either: 1) $|w| = 1$, or

2) w elapses < 1 time unit, or

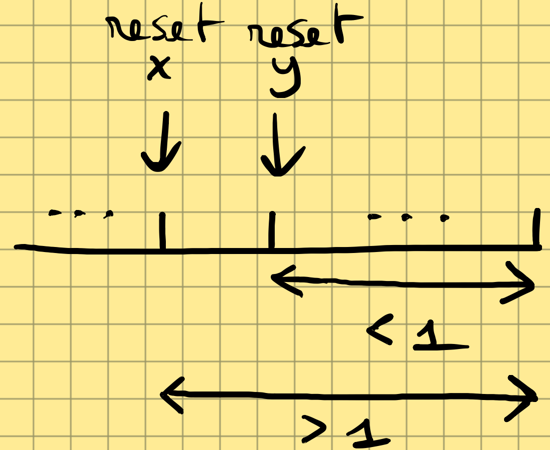
DETERMINISTIC SEPARABILITY NONTRIVIAL

L = "last letter occurred 1 t.u. earlier" (1-clock NTA, not DTA).

M = complement of L (2-clock NTA).

$w \in M$ iff either: 1) $|w| = 1$, or

2) w elapses < 1 time unit, or

3) 

(2 clocks)

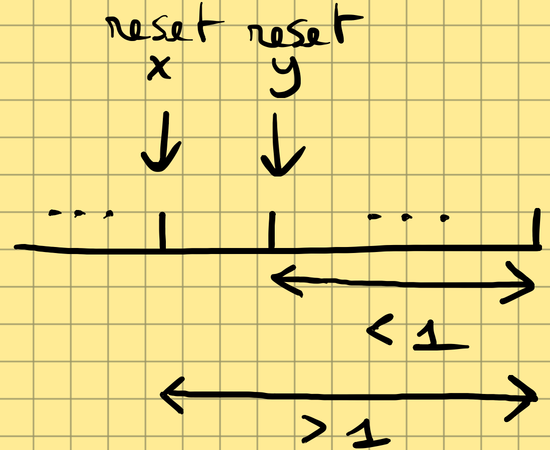
DETERMINISTIC SEPARABILITY NONTRIVIAL

L = "last letter occurred 1 t.u. earlier" (1-clock NTA, not DTA).

M = complement of L (2-clock NTA).

$w \in M$ iff either: 1) $|w| = 1$, or

2) w elapses < 1 time unit, or

3) 

(2 clocks)

→ L, M not DTA separable.

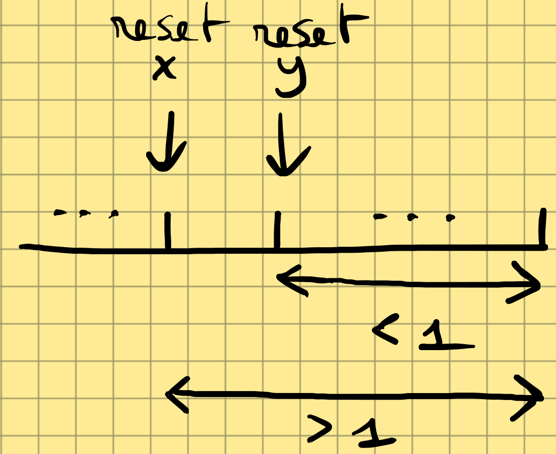
DETERMINISTIC SEPARABILITY NONTRIVIAL

L = "last letter occurred 1 t.u. earlier" (1-clock NTA, not DTA).

M = complement of L (2-clock NTA).

$w \in M$ iff either: 1) $|w| = 1$, or

2) w elapses < 1 time unit, or

3) 

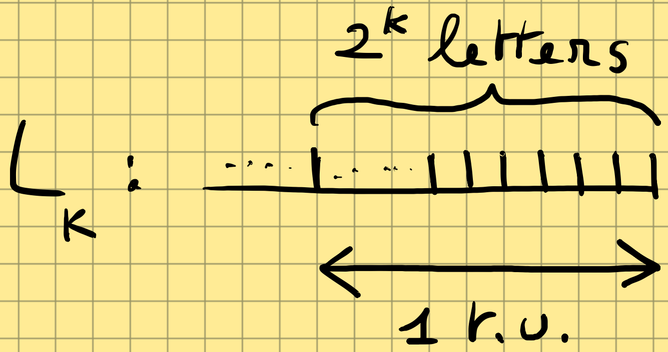
(2 clocks)

→ L, M not DTA separable.

→ 2 clocks necessary: 1-NTA ∩ 1-coNTA ⊆ DTA (unpublished).

(Stronger conjecture: disjoint 1-NTAs always DTA-separable)

SEPARATOR NEEDS MANY CLOCKS



NTA with $2K+2$ clocks :

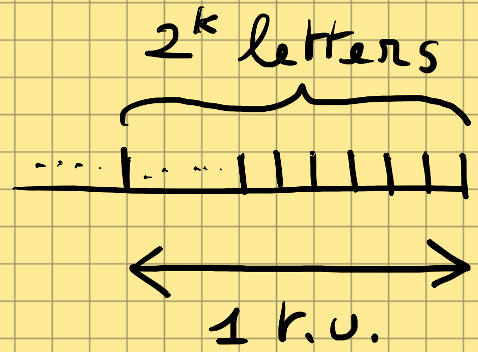
1 clock for measuring 1 t.u. (as before).

1 clock for strict monotonicity.

$2K$ clocks for counting 2^k letters in binary.

SEPARATOR NEEDS MANY CLOCKS

L_k :



NTA with $2k+2$ clocks :

1 clock for measuring 1 t.u. (as before).

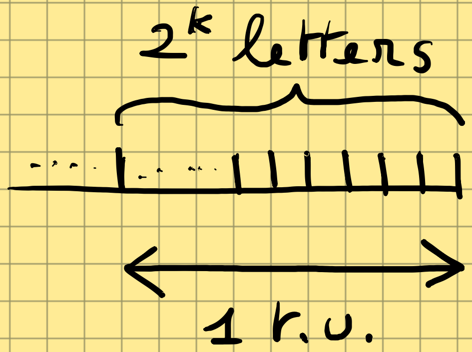
1 clock for strict monotonicity.

$2k$ clocks for counting 2^k letters in binary.

M_k = complement of L_k, also NTA with $2k+2$ clocks.

SEPARATOR NEEDS MANY CLOCKS

L_K :



NTA with $2K+2$ clocks :

1 clock for measuring 1 t.u. (as before).

1 clock for strict monotonicity.

$2K$ clocks for counting 2^K letters in binary.

M_K = complement of L_K, also NTA with $2k+2$ clocks.

A DTA for L_K needs 2^K clocks to store the last 2^K timestamps.

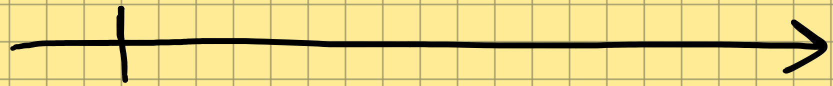
DETERMINISTIC SEPARABILITY GAME

INPUT : = w timed actions

SEPARATOR : instantaneous untimed response

DETERMINISTIC SEPARABILITY GAME

INPUT : $(a, 0.2)$ $= w$ timed actions



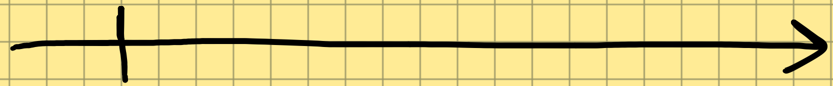
SEPARATOR :

instantaneous
untimed response

DETERMINISTIC SEPARABILITY GAME

INPUT : $(a, 0.2) = w$

timed actions



instantaneous untimed response

SEPARATOR : reject

DETERMINISTIC SEPARABILITY GAME

INPUT: $(a, 0.2) \quad (b, 1) \quad = w$

timed actions

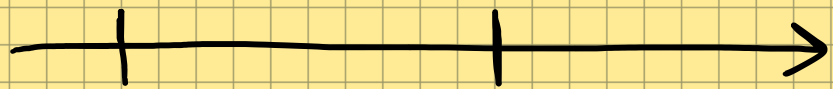
SEPARATOR: reject

instantaneous untimed response

DETERMINISTIC SEPARABILITY GAME

INPUT : (a, 0.2) (b, 1) ... = w

timed actions



SEPARATOR : reject accept ...

instantaneous untimed response

DETERMINISTIC SEPARABILITY GAME

INPUT : $(a, 0.2)$ $(b, 1)$ ... $= w$ timed actions

SEPARATOR : reject accept ... instantaneous untimed response

Winning condition for INPUT :

1) prefix of $w \in L$ & SEPARATOR rejects, or
2) prefix of $w \in M$ & SEPARATOR accepts.

DETERMINISTIC SEPARABILITY GAME

INPUT : (a, 0.2) (b, 1) ... $= w$ — timed actions

SEPARATOR : reject accept ... — instantaneous untimed response

Winning condition W for INPUT :

1) prefix of $w \in L$ & SEPARATOR rejects, or
2) prefix of $w \in M$ & SEPARATOR accepts.

W: NTA language over $\{a, b\} \times \{\text{accept}, \text{reject}\}$

DETERMINISTIC SEPARABILITY GAME

INPUT: $(a, 0.2)$ $(b, 1)$ ... $= w$ — timed actions

SEPARATOR: reject accept ... — instantaneous untimed response

Winning condition W for INPUT:
1) prefix of $w \in L$ & SEPARATOR rejects, or
2) prefix of $w \in M$ & SEPARATOR accepts.

W: NTA language over $\{a, b\} \times \{accept, reject\}$

S DTA, k clocks, const. $\leq M$	$\xleftrightarrow{\text{bijection}}$	Fin. mem. controller for Sep. k clocks, const. $\leq M$

RESULTS ON DETERMINISTIC SEPARABILITY

SEPARATOR / INPUT	(A) DTA, K clocks, constants $\leq M$	(B) DTA, K clocks	(C) DTA
NTA with ε-transitions	DECIDABLE	DECIDABLE	?

RESULTS ON DETERMINISTIC SEPARABILITY

SEPARATOR \ INPUT	(A) DTA k clocks constants $\leq M$	(B) DTA k clocks	(C) DTA
NTA with ε-transitions	DECIDABLE	DECIDABLE	?

	REDUCTION ↓	REDUCTION ↓	REDUCTION ↓

Synthesis of a finite-memory winning strategy with...	DECIDABLE	DECIDABLE	UNDECIDABLE
	k clocks constants $\leq M$	k clocks	arbitrary
	(A)	(B)	(C)

BÜCHI-LANDWEBER GAMES

INPUT :

OUTPUT :

BÜCHI-LANDWEBER GAMES

INPUT : i_0

OUTPUT :

BÜCHI-LANDWEBER GAMES

INPUT : i_0

OUTPUT : o_0

BÜCHI-LANDWEBER GAMES

INPUT : i_0 i_1

OUTPUT : o_0

BÜCHI-LANDWEBER GAMES

INPUT : i_0 i_1

OUTPUT : o_0 o_1

BÜCHI-LANDWEBER GAMES

finite input alphabet

INPUT : $i_0 \quad i_1 \quad \dots \quad i_k \in I$

finite output alphabet

OUTPUT : $o_0 \quad o_1 \quad \dots \quad o_k \in O$

BÜCHI-LANDWEBER GAMES

finite input alphabet

INPUT : $i_0 \quad i_1 \quad \dots \quad i_k \in I$

finite output alphabet

OUTPUT : $o_0 \quad o_1 \quad \dots \quad o_k \in O$

$$\underbrace{\text{play } \pi = (i_0, o_0)\,(i_1, o_1) \dots \overset{?}{\in} W \subseteq (I \times O)^{\omega}}_{\text{GALE-STEWART game}}$$

BÜCHI-LANDWEBER GAMES

finite input alphabet

finite output alphabet

INPUT : i_0 i_1 ... $i_k \in I$

OUTPUT : o_0 o_1 ... $o_k \in O$

$$\underbrace{\text{play } \pi = (i_0, o_0)\ (i_1, o_1)\ \ldots \ \overset{?}{\in} W \subseteq (I \times O)^\omega}_{\text{GALE-STEWART game}}$$

INPUT's regular winning condition

Proposed by Alonzo Church, solved by Richard Büchi & Lawrence Landweber in '69:

- Determined (exactly one player wins).
- Decidable.
- The winning player has a $\underbrace{\text{finite-memory winning strategy}}_{= \text{finite-state Mealy machine}}$.

TIMED BÜCHI-LANDWEBER GAMES

TIMED INPUT :

(UNTIMED) OUTPUT :

TIMED BÜCHI-LANDWEBER GAMES

TIMED INPUT : (i_0, t_0)

(UNTIMED) OUTPUT :

TIMED BÜCHI-LANDWEBER GAMES

TIMED INPUT : (i_0, t_0)

(UNTIMED) OUTPUT : o_0

TIMED BÜCHI-LANDWEBER GAMES

TIMED INPUT : (i_0, t_0) (i_1, t_1)

(UNTIMED) OUTPUT : o_0

TIMED BÜCHI-LANDWEBER GAMES

TIMED INPUT : (i_0, t_0) (i_1, t_1)

(UNTIMED) OUTPUT : o_0 o_1

TIMED BÜCHI-LANDWEBER GAMES

NEW!

TIMED INPUT : (i_0, t_0) (i_1, t_1) $\cdots$ $i_k \in I$, $t_k \in \mathbb{R}_{\geq 0}$

(UNTIMED) OUTPUT : o_0 o_1 $\cdots$ $o_k \in O$

TIMED BÜCHI-LANDWEBER GAMES

NEW!

TIMED INPUT : (i_0, t_0) (i_1, t_1) $\cdots$ $i_k \in \mathcal{I}$, $t_k \in \mathbb{R}_{\geq 0}$

(UNTIMED) OUTPUT : o_0 o_1 $\cdots$ $o_k \in \mathcal{O}$

play $\pi = (i_0 o_0 t_0)$ $(i_1 o_1 t_1) \cdots \in \mathcal{W}$? : NTA winning condition

TIMED BÜCHI-LANDWEBER GAMES

NEW!

TIMED INPUT : (i_0, t_0) (i_1, t_1) $\cdots$ $i_k \in I$, $t_k \in \mathbb{R}_{\geq 0}$

(UNTIMED) OUTPUT : o_0 o_1 $\cdots$ $o_k \in O$

play $\pi = (i_0 o_0 t_0)\ (i_1 o_1 t_1) \cdots \in W$? : NTA winning condition

Strategy for OUTPUT : Timed Mealy machine (= DTA + output).

TIMED BÜCHI-LANDWEBER GAMES

NEW!

TIMED INPUT : (i_0, t_0) (i_1, t_1) $\cdots$ $i_k \in I$, $t_k \in \mathbb{R}_{\geq 0}$

(UNTIMED) OUTPUT : o_0 o_1 $\cdots$ $o_k \in O$

play $\pi = (i_0 o_0 t_0)$ $(i_1 o_1 t_1) \cdots \in W$? : NTA winning condition

Strategy for OUTPUT : Timed Mealy machine (= DTA + output).

Decision problems : Does OUTPUT have a finite-memory winning strategy

(A) with K clocks & constants $\leq M$?

TIMED BÜCHI-LANDWEBER GAMES

NEW!

TIMED INPUT : $(i_0, t_0)\ (i_1, t_1) \cdots \quad i_k \in I,\ t_k \in \mathbb{R}_{\geq 0}$

(UNTIMED) OUTPUT : $o_0 \quad o_1 \cdots \quad o_k \in O$

play $\pi = (i_0 o_0 t_0)\ (i_1 o_1 t_1) \cdots \in W$? : NTA winning condition

Strategy for OUTPUT : Timed Mealy machine (= DTA + output).

Decision problems : Does OUTPUT have a finite-memory winning strategy

(A) with K clocks & constants $\leq M$?

(B) with K clocks ?

TIMED BÜCHI-LANDWEBER GAMES

NEW!

TIMED INPUT : $(i_0, t_0)\ (i_1, t_1) \cdots \quad i_k \in I,\ t_k \in \mathbb{R}_{\geq 0}$

(UNTIMED) OUTPUT : $o_0 \quad o_1 \cdots \quad o_k \in O$

play $\pi = (i_0 o_0 t_0)\ (i_1 o_1 t_1) \cdots \in W$? : NTA winning condition

Strategy for OUTPUT : Timed Mealy machine (= DTA + output).

Decision problems : Does OUTPUT have a finite-memory winning strategy

(A) with K clocks & constants $\leq M$?

(B) with K clocks ?

(C) arbitrary ?

RESULTS ON TIMED BÜCHI-LANDWEBER GAMES

winning condition W \ OUTPUT's win. strat.	(A) K clocks, constants ≤ M	(B) K clocks	(C) arbitrary
for OUTPUT	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)

(1) from universality of NTA ≥ 2 clocks: $O = \{a\}$, INPUT tries to find $\pi \notin W$.

RESULTS ON TIMED BÜCHI-LANDWEBER GAMES

winning condition W \ OUTPUT's win. strat.	(A) K clocks, constants $\leq M$	(B) K clocks	(C) arbitrary
for OUTPUT	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)
for INPUT	DECIDABLE (2)	DECIDABLE (3)	UNDECIDABLE (4)

(1) from universality of NTA ≥ 2 clocks: $O = \{a\}$, INPUT tries to find $\pi \notin W$.

RESULTS ON TIMED BÜCHI-LANDWEBER GAMES

winning condition W \ OUTPUT's win. strat.	(A) k clocks, constants $\leq M$	(B) k clocks	(C) arbitrary
for OUTPUT	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)
for INPUT	DECIDABLE (2)	DECIDABLE (3)	UNDECIDABLE (4)

(1) from universality of NTA ≥ 2 clocks: $O = \{a\}$, INPUT tries to find $\pi \notin W$.

(2) [D'Souza & Madhusudan '02]. We give a construction yielding (3) as well.

RESULTS ON TIMED BÜCHI-LANDWEBER GAMES

OUTPUT's win. strat. / winning condition W	(A) K clocks, constants $\leq M$	(B) K clocks	(C) arbitrary
for OUTPUT	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)
for INPUT	DECIDABLE (2)	DECIDABLE (3)	UNDECIDABLE (4)

(1) from universality of NTA ≥ 2 clocks: $O = \{a\}$, INPUT tries to find $\pi \notin W$.

(2) [D'Souza & Madhusudan '02]. We give a construction yielding (3) as well.

(3) NEW: synthesis of max constant.

RESULTS ON TIMED BÜCHI-LANDWEBER GAMES

winning condition W \ OUTPUT's win. strat.	(A) K clocks, constants $\leq M$	(B) K clocks	(C) arbitrary
for OUTPUT	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)
for INPUT	DECIDABLE (2)	DECIDABLE (3)	UNDECIDABLE (4)

(1) from universality of NTA ≥ 2 clocks: $O = \{a\}$, INPUT tries to find $\pi \notin W$.

(2) [D'Souza & Madhusudan'02]. We give a construction yielding (3) as well.

(3) NEW: synthesis of max constant.

(4) Claimed with no proof in [D'Souza & Madhusudan'02].
We show a reduction from finiteness of lossy counter machines,
W NTA with 1 clock suffices.

TIMED → UNTIMED BÜCHI-LANDWEBER GAMES

- Reductions to untimed Büchi-Landweber games.
- Idea: INPUT plays K-clock regions. (details omitted)
  OUTPUT plays clock resets.

TIMED → UNTIMED BÜCHI-LANDWEBER GAMES

- Reductions to untimed Büchi-Landweber games.
- Idea: INPUT plays K-clock regions. (details omitted)
  OUTPUT plays clock resets.
- Correctness for (A):

ORIGINAL TIMED GAME		UNTIMED GAME
OUTPUT finite-memory winning strategies with K-clocks & const. $\leq M$	$\xleftrightarrow{\text{bijection}}$	OUTPUT finite memory winning strategies

TIMED → UNTIMED BÜCHI-LANDWEBER GAMES

- Reductions to untimed Büchi-Landweber games.
- Idea: INPUT plays K-clock regions. (details omitted)
  OUTPUT plays clock resets.
- Correctness for (A):

ORIGINAL TIMED GAME		UNTIMED GAME
OUTPUT finite-memory winning strategies with K-clocks & const. ≤ M	$\xleftrightarrow{\text{bijection}}$	OUTPUT finite memory winning strategies

1) Const. ≤ M hardcoded in the winning condition.

TIMED → UNTIMED BÜCHI-LANDWEBER GAMES

- Reductions to untimed Büchi-Landweber games.
- Idea: INPUT plays K-clock regions. (details omitted)
  OUTPUT plays clock resets.
- Correctness for (A):

ORIGINAL TIMED GAME

OUTPUT finite-memory winning strategies with K-clocks & const. ≤ M

$\xleftrightarrow{\text{bijection}}$

UNTIMED GAME

OUTPUT finite memory winning strategies

1) Const. ≤ M hardcoded in the winning condition.

2) K & M fixed

⇒ ω-regular winning condition!

TIMED → UNTIMED BÜCHI-LANDWEBER GAMES

- We design reductions to untimed Büchi-Landweber games.
- Idea: INPUT plays K-clock regions. (details omitted)
  OUTPUT plays clock resets.
- Correctness for (B):

ORIGINAL TIMED GAME		UNTIMED GAME
OUTPUT finite-memory winning strategies with K-clocks & bounded constants	$\xleftrightarrow{\text{bijection}}$	OUTPUT finite memory winning strategies

TIMED → UNTIMED BÜCHI-LANDWEBER GAMES

- We design reductions to untimed Büchi-Landweber games.
- Idea : INPUT plays K-clock regions. (details omitted)
  OUTPUT plays clock resets.
- Correctness for (B) :

ORIGINAL TIMED GAME		UNTIMED GAME
OUTPUT finite-memory winning strategies with K-clocks & bounded constants	$\xleftrightarrow{\text{bijection}}$	OUTPUT finite memory winning strategies

1) no infinite request chain.

TIMED → UNTIMED BÜCHI-LANDWEBER GAMES

- We design reductions to untimed Büchi-Landweber games.
- Idea: INPUT plays K-clock regions. (details omitted)
  OUTPUT plays clock resets.
- Correctness for (B):

ORIGINAL TIMED GAME

OUTPUT finite-memory winning strategies with K-clocks & bounded constants

bijection ⟷

UNTIMED GAME

OUTPUT finite memory winning strategies

1) no infinite request chain.
2) finite memory determinacy
⇒ uniformly bounded request chains (!)

DETERMINISABILITY

(1) [Finkel '06], [Tripakis '06]

OUTPUT / INPUT	DTA, k clocks, constants $\leq M$	DTA, k clocks	DTA
NTA ≥ 2 clocks	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)
NTA 1 clock ε-transitions	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)

DETERMINISABILITY

(1) [Finkel '06], [Tripakis'06]

OUTPUT / INPUT	DTA, k clocks, constants ≤ M	DTA, k clocks	DTA
NTA, ≥ 2 clocks	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)
NTA, 1 clock, ε-transitions	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)
NTA, 1 clock, no ε-tr.	DECIDABLE (2)	DECIDABLE (3)	UNDECIDABLE (4)

DETERMINISABILITY

(1) [Finkel '06], [Tripakis '06]

INPUT \ OUTPUT	DTA k clocks constants ≤ M	DTA k clocks	DTA
NTA ≥ 2 clocks	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)
NTA 1 clock ε-transitions	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)
NTA 1 clock no ε-tr.	DECIDABLE (2)	DECIDABLE (3)	UNDECIDABLE (4)

NEW : (2) by computable bound on # control locations.

DETERMINISABILITY

(1) [Finkel '06], [Tripakis'06]

INPUT \ OUTPUT	DTA k clocks constants ≤ M	DTA k clocks	DTA
NTA ≥ 2 clocks	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)
NTA 1 clock ε-transitions	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)
NTA 1 clock no ε-tr.	DECIDABLE (2)	DECIDABLE (3)	UNDECIDABLE (4)

NEW : (2) by computable bound on # control locations.

(3) If L NTA with const. ≤ M is DTA, then it is DTA with const. ≤ M.

DETERMINISABILITY

(1) [Finkel '06], [Tripakis '06]

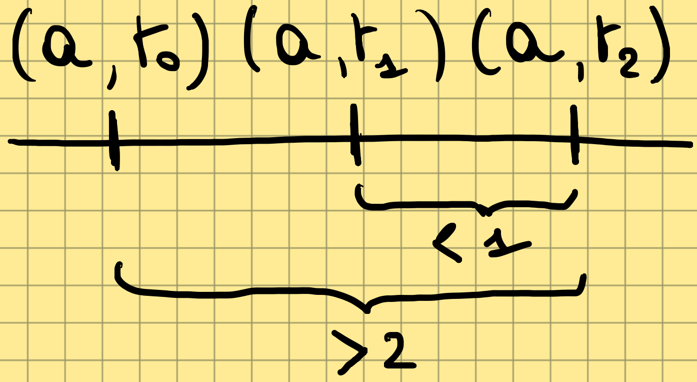
OUTPUT / INPUT	DTA k clocks constants ≤ M	DTA k clocks	DTA
NTA ≥ 2 clocks	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)
NTA 1 clock ε-transitions	UNDECIDABLE (1)	UNDECIDABLE (1)	UNDECIDABLE (1)
NTA 1 clock no ε-tr.	DECIDABLE (2)	DECIDABLE (3)	UNDECIDABLE (4)

NEW : (2) by computable bound on # control locations.
(3) If L NTA with const. ≤ M is DTA, then it is DTA with const. ≤ M.
(4) reduction from finiteness of lossy counter machines [Mayr '03].

ALWAYS RESETTING NTA with 2 CLOCKS

- Every transition resets at least one clock $\approx$ "$1\frac{1}{2}$ clocks".
- Intermediate expressive power between 1 and 2 clocks:

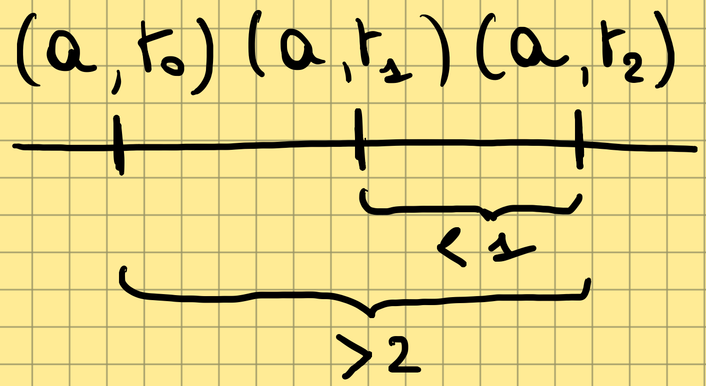
- L :



rec. by NTA with $1\frac{1}{2}$ clocks, but not by NTA with 1 clock.

ALWAYS RESETTING NTA with 2 CLOCKS

- Every transition resets at least one clock ≈ "$1\frac{1}{2}$ clocks".
- Intermediate expressive power between 1 and 2 clocks:
- L :



rec. by NTA with $1\frac{1}{2}$ clocks, but not by NTA with 1 clock.

- Similar decidability properties as NTA with 1 clock:
  - Universality decidable.
  - Determinisability with K clocks and constants $\leq M$ decidable.
  - Determinisability with K clocks decidable.

FUTURE WORK

- Deterministic separability decidable?
  - Current reduction is to an undecidable problem.

    $\Rightarrow$ need different ideas

FUTURE WORK

- Deterministic separability decidable?
  - Current reduction is to an undecidable problem.

    $\Rightarrow$ need different ideas
- Is determinisability of L, M decidable assuming $L = M^c$?

FUTURE WORK

- Deterministic separability decidable?
  - Current reduction is to an undecidable problem.

    ⇒ need different ideas
- Is determinisability of L, M decidable assuming $L = M^c$?
- Determinisability / Deterministic separability of unambiguous timed automata. Decidable?

FUTURE WORK

- Deterministic separability decidable?
  - Current reduction is to an undecidable problem.

    ⇒ need different ideas
- Is determinisability of L, M decidable assuming $L = M^c$?
- Determinisability / Deterministic separability of unambiguous timed automata. Decidable?
- Unambiguous separability of timed automata:
  - Is it always true? If not, is it decidable?

FUTURE WORK

- Deterministic separability decidable?
  - Current reduction is to an undecidable problem.
    $\Rightarrow$ need different ideas
- Is determinisability of L, M decidable assuming $L = M^c$?
- Determinisability / Deterministic separability of unambiguous timed automata. Decidable?
- Unambiguous separability of timed automata:
  - Is it always true? If not, is it decidable?
- Complexity of separability (lower bounds)?

FUTURE WORK

- Deterministic separability decidable?
  - Current reduction is to an undecidable problem.
    $\Rightarrow$ need different ideas
- Is determinisability of L, M decidable assuming $L = M^c$?
- Determinisability / Deterministic separability of unambiguous timed automata. Decidable?
- Unambiguous separability of timed automata:
  - Is it always true? If not, is it decidable?
- Complexity of separability (lower bounds)?
- Similar results for register automata over $(\mathbb{A}, =)$, $(\mathbb{Q}, \leq)$, ...

FROM CONSTANTS $\leq M$ to CONSTANTS ≤ 1

Observation : In NTA with ε-transitions we can always make constants ≤ 1. Gadget $\text{"}x = 3\text{"} \xrightarrow{\varepsilon,\ x=1,\ x:=0} \text{"}x = 4\text{"}$.

Do the same with OUTPUT's controller ?

Obstacle : No ε-transitions in OUTPUT's controller !

Solution : Force INPUT to provide clock expiry information.

ORIGINAL GAME
$I = \{a\}$, $O = \{<2, =2, >2\}$

INPUT : $(a,0)$ $(a,2)$ $(a,7)$ ...

OUTPUT : <2 $=2$ >2

max constant $M = 2$ required.

extra control symbols

NEW GAME
$I = \{a, \checkmark\}$, $O = \{<2, =2, >2, \varepsilon\} \times \{R, \square\}$

INPUT : $(a,0)$ $(\checkmark,1)$ $(a,2)$ $\underbrace{(\checkmark,3)\ (\checkmark,4)}^{2}$ $(a,7)$ ...

OUTPUT : $<2,R$ ε,R $=2,R$ ε,R $\varepsilon,\square$ $>2,R$

- max constant $M' = 1$ suffices in the new game.
- max constant M in the original game is the length of request chains (2).
- the new winning condition ensures that
  - INPUT does not cheat.
  - M given $\Rightarrow$ request chains have length $\leq M$.
  - M not given $\Rightarrow$ no infinite request chain.