Weak Alternating Timed Automata

Pawel Parys and Igor Walukiewicz

CNRS, LaBRI, Bordeaux

IFIP, September 2009

Timed languages [Alur & Dill'94]

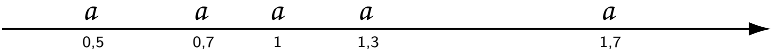
Infinite sequences

$$abaacb\ldots$$

(Infinite) timed words

$$(a,t_0),(b,t_1)(a,t_2)\ldots$$

The sequence $\{t_i\}_{i=0,1,\ldots}$ is strictly increasing and unbounded (nonZeno).



Languages of timed words

- There are two a's that appear at interval 1.
- No two a's appear at interval 1.

Timed automata

Clock and guards

$$(x > 2), \quad (x \leq 3), \quad (x > 2) \wedge (x \leq 3)$$

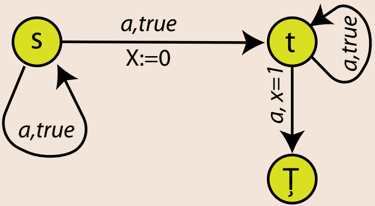
We will use only one clock x.

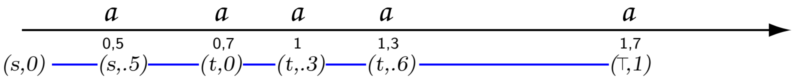
Timed automata

$$\mathcal{A} = \langle Q, q_0, \Sigma, \delta, F \rangle$$

- $F \subseteq Q$ is the set of final states.
- $\delta : Q \times \Sigma \times \text{Guards} \xrightarrow{\cdot} \mathcal{P}(Q \times \{\texttt{nop}, \texttt{reset}\})$

Example *(two a's at distance 1)*





Timed automata: properties

Properties

- Emptiness is decidable. (region construction)
- Universality is undecidable. (Π^1_1-hard)
- Not closed under complement. (No to $a's$ at distance 1.)
- Deterministic version not closed under disjunction.

Current state

- No good class of regular timed languages.
- Development of logics independent from automata (MTL, TLTL).

Alternating timed automata (ATA)

$$\mathcal{A} = \langle Q, q_0, \Sigma, \delta, F \rangle$$

- $F \subseteq Q$ is the set of final states.
- $\delta : Q \times \Sigma \times \text{Guards} \dot{\to} \mathcal{B}^+(Q \times \{\mathtt{nop}, \mathtt{reset}\})$

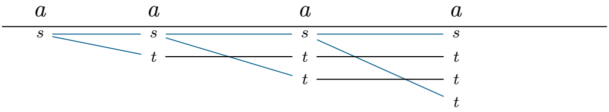
Example (No two a's at distance 1)

An alternating automaton for L:

$$s, a, tt \mapsto (s, \mathtt{nop}) \wedge (t, \mathtt{reset})$$

$$t, a, x \neq 1 \mapsto (t, \mathtt{nop}) \qquad t, a, x = 1 \mapsto (\bot, \mathtt{nop})$$

All states but $\bot$ are accepting.



Properties

Closure properties

ATA are effectively closed under boolean operations.

Expressibility

The class of languages recognized by 1-clock ATA is incomparable with the class of languages recognized by timed automata (with many clocks).

Decidability

The emptiness problem over finite words for 1-clock ATA is decidable.

Undecidability

The emptiness problem over infinite words for 1-clock ATA is undecidable.

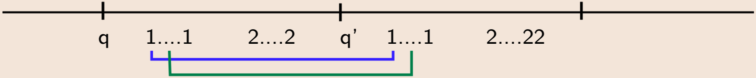
The problem with infinite words

Theorem (Lasota & W., Ouaknine & Worell)

The emptiness problem for ATA with Buchi acceptance conditions is undecidable.

Proof sketch

- We encode the problem of existence of an accepting computation of a 2-counter machine. We can assume that after reaching an accepting state the machine restarts in the initial conf.
- Each configuration is put in one unit interval.



- We can easily simulate "gainy" machines: counters can increase without our control.

Infinite computations problem for "gainy" machines

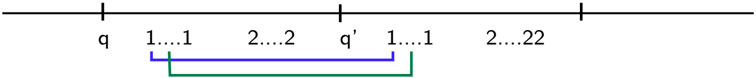
Does a given 5 counter "gainy" machine has a run where an accepting state appears infinitely often.

Theorem (Ouaknine & Worell)

The above problem is undecidable.

Theorem (Mayr)

It is undecidable whether there is an uniform bound on the size of all reachable configurations of a 4*-counter lossy machine.*



Coding infinite computations of "gainy" counter machines

We need do say that q_{acc} appears infinitely often.
We express it as GFq_{acc} (at every moment there is q_{acc} in the future)

Acceptance conditions

An infinite run

$$q_1\, q_2\, q_3\, q_2\, q_3 \ldots$$

Parity condition: $\Omega : Q \to \mathbb{N}$

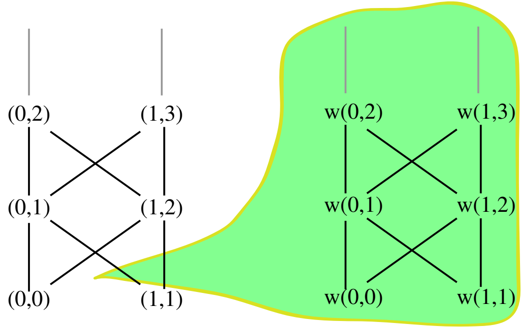
- Strong condition: a run is accepting if

  $\min\{\Omega(q) : q$ appears infinitely often in the run$\}$ is even

- Weak condition: a sequence is accepting if

  $\min\{\Omega(q) : q$ appears at least once in the run$\}$ is even

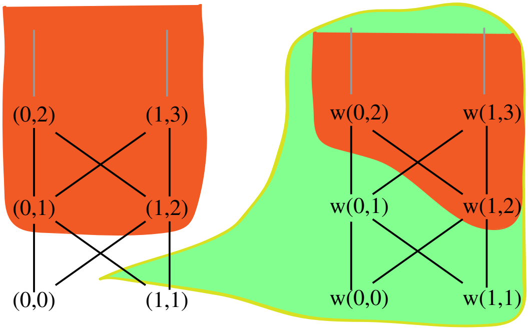
Hierarchies of acceptance conditions



Index hierarchies

- Interesting ranges: $(0, i)$, $(1, i)$ for $i = 0, 1, \dots$.
- Strong condition with range $(0, 1)$ corresponds to a Büchi condition, and $(1, 2)$ to a coBüchi condition.
- With a range $(0, i + 1)$ we can accept more than with $(0, i)$ and the sets of languages accepted by $(0, i)$ and $(1, i + 1)$ are incomparable.
- Expressing GFq_{acc}: alternation + range w$(1, 2)$.

Hierarchies of acceptance conditions (2)



The emptiness problem for universal timed automata

- Decidable for level $(1, 1)$ (finite words).
- Undecidable for level w$(1, 2)$.

Question

What about levels $(0, 0)$ and w$(0, 1)$?

Results

Theorem (Parys & W.)

The emptiness problem, over nonZeno words, is decidable for ATA with index w$(0,1)$ (hence for $(0,0)$ too).

Theorem (Parys & W.)

The emptiness problem is undecidable for ATA with index w$(1,2)$ even when only tests for the interval interval $[0,1)$ are used.

Corollary

In this setting relaxing punctuality (à la MITL) does not pay.
Equality constraints are not need to force complicated behaviours.

History

- Abdulla & Jonson TACAS'98 (PN's with one clock).
- Ouaknine & Worrell LICS'04 (Universality for one clock is decidable).
- Lasota & W. FOSSACS'05 (ATA, emptiness is nonelementary, undecidability over infinite words).
- Ouaknine & Worrell LICS'05 (decidability for MTL over finite words).
- Ouaknine & Worell FOSSACS'06 (undecidability of MTL over infinite words).
- Ouaknine & Worell TACAS'06 (decidability of restricted ATA without acceptance conditions over infinite words).
- Bouyer & Markey & Ouaknine & Worrell LICS07, ICALP08 (decidable extensions of MTL).

The case of finite words

Powerset construction: transition system $\mathcal{T}$.

- Macro state: $\{(q_1, t_1), \ldots, (q_n, t_n)\}$
- Transition relation

$$\{(q_1, t_1), \ldots, (q_n, t_n)\}$$

$$(a, t)$$

$$\{(q_{1,1}, t_1 + t), \ldots, (q_{1,k}, 0), \ldots, (q_{n,1}, 0), \ldots, (q_{n,l}, t_n + t)\}$$

Nonemptiness $\equiv$ reachability

- Final macro state: all the states accepting.
- Non-emptiness $\equiv$ reachability of a final state in $\mathcal{T}$.

Well quasi-order

- In every infinite sequence $c_1, c_2, \ldots$ there exist indexes $i < j$ with (c_i, c_j) in the relation.
- If a final state is reachable from $\{(q_1, t_1), \ldots, (q_n, t_n)\}$ and the it is reachable from every its subset $\{(q_{i_1}, t_{i_1}), \ldots, (q_{i_k}, t_{i_k})\}$.
- **Problem:** This relation is not a well quasi-order.

Where is the challenge

The case of finite words

Construct appropriate WQO and do reachability tree.

Detecting existence of an infinite computation

- We need to take care of nonZeno.
- The reachability tree argument does not work.
- We calculate the set of configurations from which every computation is finite. (This set is upwards closed in some WQO).
- Effectiveness is very specific to our model. For example for lossy channel systems it is undecidable if from every channel contents all computations terminate.

Applications to logics

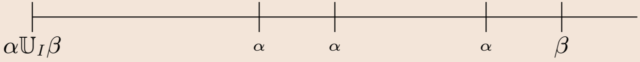
MTL

MTL

$$p \mid \neg p \mid \alpha \vee \beta \mid \alpha \wedge \beta \mid \alpha \mathbb{U}_I \beta \mid \alpha \tilde{\mathbb{U}}_I \beta$$

- I is an interval, eg., $(0,1)$, $[5,\infty]$.

Pointwise semantics



Fragments

- Bounded MTL (BMTL): All intervals bounded.
- Metric Interval TL (MITL): no singleton intervals.

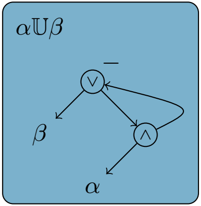
Translating MTL to automata (I)

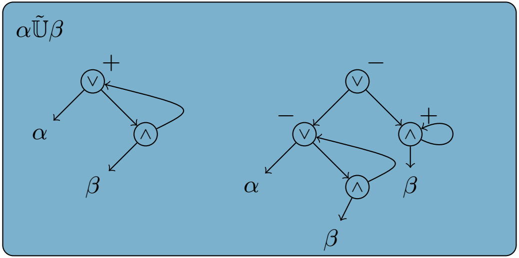
Theorem (Parys & W.)

Emptiness over nonZeno words is decidable for ATA with index w$(0,1)$.

Remark

A Buchi automaton is w$(0,1)$ if there is no transition going from accepting state to non accepting state.





Positive MTL

Positive formulas: $p \mid \neg p \mid \alpha \vee \beta \mid \alpha \wedge \beta \mid \alpha\tilde{\mathbb{U}}_I\beta \mid \alpha\mathbb{U}_J\beta$ $\quad J$ bounded.

PMTL: $\alpha \vee \beta \mid \alpha \wedge \beta \mid \alpha\mathbb{U}_I\beta \mid \alpha\tilde{\mathbb{U}}_J\psi$ $\quad \psi$ positive, or J bounded.

Translating MTL to automata (II)

Theorem (Parys & W.)

Emptiness over nonZeno words is decidable for ATA with index w$(0,1)$.

Remark

A Buchi automaton is w$(0,1)$ if there is no transition going from accepting state to non accepting state.

Theorem (Parys & W.)

Emptiness over nonZeno words is decidable for ATA with index w$(0,1)$.

Corollary

The satisfiability problem for Positive-MITL over nonZeno words is decidable.

Flat vs positive

Positive MTL

Positive formulas: $p \mid \neg p \mid \alpha \vee \beta \mid \alpha \wedge \beta \mid \alpha \tilde{\mathbb{U}}_I \beta \mid \alpha \mathbb{U}_J \beta \quad J$ bounded.

PMTL: $\alpha \vee \beta \mid \alpha \wedge \beta \mid \alpha \mathbb{U}_I \beta \mid \alpha \tilde{\mathbb{U}}_J \psi \quad \psi$ positive, or J bounded.

Flat-MTL

- In $\alpha \mathbb{U}_I \beta$ either $\alpha \in MITL$ or I bounded.
- In $\alpha \tilde{\mathbb{U}}_I \beta$ either $\beta \in MITL$ or I bounded.

Remarks

- In positive-MTL the restriction is only one sided. One can express eventuality properties. (In flat only invariance of MITL properties).
- We cannot admit MITL in the positive fragment, as any infinitely-often property will lead to undecidability (in automata, not clear if in logic).

TPTL

TPTL (with one clock)

$$p \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \mathbb{U} \psi \mid \varphi \tilde{\mathbb{U}}\, \psi \mid x \sim c \mid x.\varphi$$

Semantics in a sequence $w = (a_0, t_0)(a_1, t_1), \ldots$

$w, i, v \vDash p$	if $a_i = p$
$w, i, v \vDash x \sim c$	if $t_i - v \sim c$
$w, i, v \vDash x.\varphi$	if $w, i, t_i \vDash \varphi$
$w, i, v \vDash \varphi \mathbb{U} \psi$	if $\exists_{j>i} w, j, v \vDash \psi$ and $\forall_{k \in (i,j)} w, k, v \vDash \varphi$
$w, i, v \vDash \varphi \tilde{\mathbb{U}} \psi$	if $\forall_{j>i} w, j, v \vDash \psi$ or $\exists_{k \in (i,j)} w, k, v \vDash \varphi$

Theorem (Bouyer & Chevalier & Markey)

The formula $x.(F(b \wedge F(c \wedge x \leq 2)))$ *is not expressible in MTL.*

Remark

One can define positive TPTL the same way as positive MTL. The resulting logic can be encoded into w$(1, 2)$ ATA.

Conclusions

1. We have established the decidability frontier, with respect to the index, for ATA over infinite words.
2. Restricting to non-singular intervals does not make the problem easier.
3. The decidability result gives a new decidable fragment of MTL (Positive MTL).
4. In a similar way we can also obtain a decidable fragment of TPTL with one clock.

- So what is the class of languages accepted by ATA?
- Reduce the use of resets to get more decidability?