Uniform satisfiability problem for local temporal logics over Mazurkiewicz traces

Paul Gastin[1] Dietrich Kuske[2]

[1]LSV, ENS de Cachan & CNRS, FRANCE

[2]Institut für Informatik, Universität Leipzig, GERMANY

VERSYDIS, 17/10/2005

Outline

Overview

Definitions and some known temporal logics

Uniform satisfiability: upper bound

Uniform satisfiability: lower bound

Open problems

Motivation

Model Checking

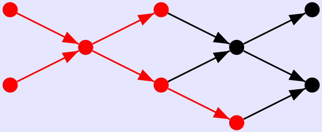
Concurrent system $\models$ Concurrent specification

- Concurrent behaviors: partial orders, Mazurkiewicz traces
- Concurrent specifications: local temporal logics for partial orders
- Main problems: expressivity and complexity

Global versus Local logics

Global logics

A global temporal logic formula is evaluated at global configurations.



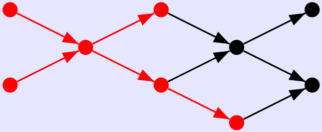
Local logics

A local temporal logic formula is evaluated at events or local configurations.

Global versus Local logics

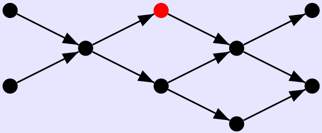
Global logics

A global temporal logic formula is evaluated at global configurations.



Local logics

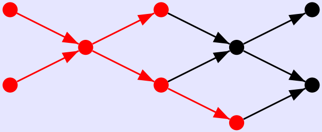
A local temporal logic formula is evaluated at events or local configurations.



Global versus Local logics

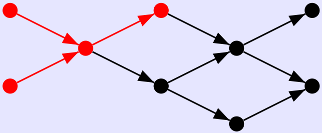
Global logics

A global temporal logic formula is evaluated at global configurations.



Local logics

A local temporal logic formula is evaluated at events or local configurations.



Global logics: known results

Existential until

ISTL: Interleaving Set Temporal Logic, Katz, Peled (PODC'87, TCS'90).
CCTL: Concurrent CTL, Penczek (CSL'89).
POL: Partial Order Logic, Penczek (IPL'92).

But . . . the satisfiability problem is undecidable Penczek (IPL'92).

Universal until

$\mathrm{FO} = \mathrm{LTL}$: Universal until and since, finite traces only Ebinger (PhD'94).
$\mathrm{FO} = \mathrm{LTrL}$: Universal until and past constants Thiagarajan, Walukiewicz (LICS'97).
$\mathrm{FO} = \mathrm{LTL}$: Universal until only Diekert, Gastin (ICALP'00, JCSS'02).

But . . . the satisfiability problem is non elementary Walukiewicz (ICALP'98).

Global logics: known results

Existential until

ISTL: Interleaving Set Temporal Logic, Katz, Peled (PODC'87, TCS'90).
CCTL: Concurrent CTL, Penczek (CSL'89).
POL: Partial Order Logic, Penczek (IPL'92).

But . . . the satisfiability problem is undecidable Penczek (IPL'92).

Universal until

$\mathrm{FO} = \mathrm{LTL}$: Universal until and since, finite traces only Ebinger (PhD'94).
$\mathrm{FO} = \mathrm{LTrL}$: Universal until and past constants Thiagarajan, Walukiewicz (LICS'97).
$\mathrm{FO} = \mathrm{LTL}$: Universal until only Diekert, Gastin (ICALP'00, JCSS'02).

But . . . the satisfiability problem is non elementary Walukiewicz (ICALP'98).

Local logics: known results

- TrPTL: process based Logic — Thiagarajan (LICS'94).
  Satisfiability is decidable in EXPTIME. — Gossip automata
  $\mathrm{TrPTL} \subseteq \mathrm{FO}$.
- TLC: Existential until — Alur, Peled, Penczek (LICS'95).
  Satisfiability is PSPACE-complete. — Tableau construction
  TLC incomparable with FO in general.
- $\mathrm{LocTL}(\mathsf{EX}, \mathsf{U})$: Universal until — Diekert, Gastin (LPAR'01, IC'04).
  Satisfiability is PSPACE-complete. — Alternating automata
  $\mathrm{LocTL}(\mathsf{EX}, \mathsf{U}) = \mathrm{FO}$. — Diekert, Gastin (LATIN'04).

Gastin & Kuske (CONCUR'03)

Theorem

All MSO-definable local temporal logics over Mazurkiewicz traces are decidable in PSPACE (both satisfiability and model checking).

Restriction

In all previous results,

the architecture of the concurrent system was not part of the input

but, when we are given a model checking problem

Concurrent system $\models$ concurrent specification

the concurrent system is abstracted in

- an architecture, e.g., a dependence alphabet (Σ, D)
- some (distributed) transition system

Hence, complexity with respect to the architecture is also important.

Gastin & Kuske (CONCUR'03)

Theorem

All MSO-definable local temporal logics over Mazurkiewicz traces are decidable in PSPACE (both satisfiability and model checking).

Restriction

In all previous results,

the architecture of the concurrent system was not part of the input

but, when we are given a model checking problem

Concurrent system $\models$ concurrent specification

the concurrent system is abstracted in

- an architecture, e.g., a dependence alphabet (Σ, D)
- some (distributed) transition system

Hence, complexity with respect to the architecture is also important.

Gastin & Kuske (CONCUR'03)

Theorem

All MSO-definable local temporal logics over Mazurkiewicz traces are decidable in PSPACE (both satisfiability and model checking).

Restriction

In all previous results,

the architecture of the concurrent system was not part of the input

but, when we are given a model checking problem

$$\text{Concurrent system} \models \text{concurrent specification}$$

the concurrent system is abstracted in

- an architecture, e.g., a dependence alphabet (Σ, D)
- some (distributed) transition system

Hence, complexity with respect to the architecture is also important.

Gastin & Kuske (CONCUR'03)

Theorem

All MSO-definable local temporal logics over Mazurkiewicz traces are decidable in PSPACE (both satisfiability and model checking).

Restriction

In all previous results,

the architecture of the concurrent system was not part of the input

but, when we are given a model checking problem

Concurrent system $\models$ concurrent specification

the concurrent system is abstracted in

- an architecture, e.g., a dependence alphabet (Σ, D)
- some (distributed) transition system

Hence, complexity with respect to the architecture is also important.

Gastin & Kuske (CONCUR'05)

Uniform satisfiability problem

Input: the architecture and the formula.

Question: Is there a behavior on this architecture satisfying the formula.

Theorem 1

If the modalities of the local temporal logics are definable in $\mathrm{M}\Delta^1_n(\lessdot)$ then the uniform satisfiability problem is decidable in n-**EXPSPACE**, or more precisely in space

$$\texttt{poly}(|\text{specification}|) \cdot \texttt{tower}(n, \texttt{poly}(|\text{architecture}|)).$$

Proposition

All known local temporal logics over traces are definable in $\mathrm{M}\Delta^1_1(\lessdot)$.
Hence, their uniform satisfiability problem is decidable in 1-**EXPSPACE**.

Theorem 2

There is local temporal logic definable in $\mathrm{M}\Pi^1_n(\lessdot)$ for which the uniform satisfiability problem is $(n-1)$-**EXPSPACE** hard.

Gastin & Kuske (CONCUR'05)

Uniform satisfiability problem

Input: the architecture and the formula.

Question: Is there a behavior on this architecture satisfying the formula.

Theorem 1

If the modalities of the local temporal logics are definable in $\mathrm{M}\Delta^1_n(\lessdot)$ then the uniform satisfiability problem is decidable in n-$\mathbf{EXPSPACE}$, or more precisely in space

$$\mathtt{poly}(|\text{specification}|) \cdot \mathtt{tower}(n, \mathtt{poly}(|\text{architecture}|)).$$

Proposition

All known local temporal logics over traces are definable in $\mathrm{M}\Delta^1_1(\lessdot)$.
Hence, their uniform satisfiability problem is decidable in 1-$\mathbf{EXPSPACE}$.

Theorem 2

There is local temporal logic definable in $\mathrm{M}\Pi^1_n(\lessdot)$ for which the uniform satisfiability problem is $(n-1)$-$\mathbf{EXPSPACE}$ hard.

Gastin & Kuske (CONCUR'05)

Uniform satisfiability problem

Input: the architecture and the formula.

Question: Is there a behavior on this architecture satisfying the formula.

Theorem 1

If the modalities of the local temporal logics are definable in $\mathrm{M}\Delta_n^1(\lessdot)$ then the uniform satisfiability problem is decidable in n-**EXPSPACE**, or more precisely in space

$$\texttt{poly}(|\text{specification}|) \cdot \texttt{tower}(n, \texttt{poly}(|\text{architecture}|)).$$

Proposition

All known local temporal logics over traces are definable in $\mathrm{M}\Delta_1^1(\lessdot)$.
Hence, their uniform satisfiability problem is decidable in 1-**EXPSPACE**.

Theorem 2

There is local temporal logic definable in $\mathrm{M}\Pi_n^1(\lessdot)$ for which the uniform satisfiability problem is $(n-1)$-**EXPSPACE** hard.

Outline

Overview

Definitions and some known temporal logics

Uniform satisfiability: upper bound

Uniform satisfiability: lower bound

Open problems

Architecture = Dependence alphabet

shared variable

$$\begin{aligned} a: &\quad x := x + y \\ b: &\quad y := x \times z \\ c: &\quad z := y + 1 \end{aligned}$$

- a and c are independent.
- a and b are dependent.

Dependence alphabet: $(\Sigma, D) = a — b — c$

- Σ is a finite set of actions
- $D \subseteq \Sigma \times \Sigma$ is a reflexive and symmetric dependence relation

Dependence alphabet are convenient abstractions of architectures for

- programs with shared variables,
- 1-safe Petri nets,
- ...

Architecture = Dependence alphabet

shared variable

$$\begin{aligned} a: &\quad x := x + y \\ b: &\quad y := x \times z \\ c: &\quad z := y + 1 \end{aligned}$$

- a and c are independent.
- a and b are dependent.

Dependence alphabet: $(\Sigma, D) = a — b — c$

- Σ is a finite set of actions
- $D \subseteq \Sigma \times \Sigma$ is a reflexive and symmetric dependence relation

Dependence alphabet are convenient abstractions of architectures for

- programs with shared variables,
- 1-safe Petri nets,
- ...

Architecture = Dependence alphabet

shared variable

$$\begin{aligned} a:&\quad x := x + y \\ b:&\quad y := x \times z \\ c:&\quad z := y + 1 \end{aligned}$$

- a and c are independent.
- a and b are dependent.

Dependence alphabet: $(\Sigma, D) = a - b - c$

- Σ is a finite set of actions
- $D \subseteq \Sigma \times \Sigma$ is a reflexive and symmetric dependence relation

Dependence alphabet are convenient abstractions of architectures for

- programs with shared variables,
- 1-safe Petri nets,
- ...

Mazurkiewicz Traces

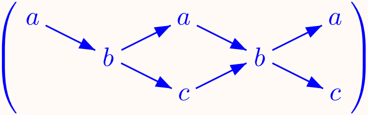
Mazurkiewicz traces over (Σ, D)

Trace = labelled partial order $t = [V, \leq, \lambda]$ such that

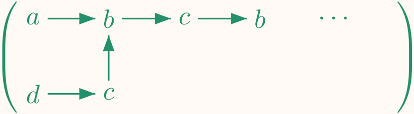
- $(\lambda(x), \lambda(y)) \in D \Rightarrow x \leq y$ or $y \leq x$, — Dependent events must be ordered.
- $x \lessdot y \Rightarrow (\lambda(x), \lambda(y)) \in D$, — Consecutive events must be dependent.
- $\downarrow x = \{y \in V \mid y \leq x\}$ is finite. — Each event has a finite past.

Example: $(\Sigma, D) = a — b — c — d$

A finite trace $t =$



An infinite trace $t =$



Local temporal logics

Example: $\mathrm{TL}_\Sigma(\neg, \vee, \mathsf{EX}, \mathsf{U})$

$$\varphi ::= a \ (a \in \Sigma) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{EX}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi$$

Definition: $\mathrm{TL}_N(B)$

- N: set of action names
- B: set of modality names
- $\texttt{arity} : B \to \mathbb{N}$

Syntax

$$\varphi ::= \sum_{a \in N} a \quad + \quad \sum_{M \in B} M(\underbrace{\varphi, \ldots, \varphi}_{\texttt{arity}(M)})$$

Local temporal logics: Semantics

Example: $\mathrm{TL}_\Sigma(\neg, \vee, \mathsf{EX}, \mathsf{U})$

$$\varphi ::= a \ (a \in \Sigma) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{EX}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi$$

Semantics: $t = [V, \leq, \lambda]$ with $\lambda : V \to \Sigma$ labelled partial order

$t, x \models a$	if	$\lambda(x) = a$
$t, x \models \mathsf{EX}\,\varphi$	if	$\exists y.\ x < y \wedge t, y \models \varphi$
$t, x \models \varphi\,\mathsf{U}\,\psi$	if	$\exists z.\ x \leq z \wedge t, z \models \psi \wedge \forall y.\ (x \leq y < z) \to t, y \models \varphi$

Example

Local temporal logics: Semantics

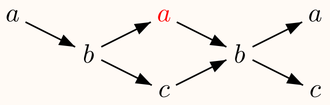
Example: $\mathrm{TL}_\Sigma(\neg, \vee, \mathsf{EX}, \mathsf{U})$

$$\varphi ::= a \ (a \in \Sigma) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{EX}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi$$

Semantics: $t = [V, \leq, \lambda]$ with $\lambda : V \to \Sigma$ labelled partial order

$t, x \models a$	if	$\lambda(x) = a$
$t, x \models \mathsf{EX}\,\varphi$	if	$\exists y.\ x \lessdot y \wedge t, y \models \varphi$
$t, x \models \varphi\,\mathsf{U}\,\psi$	if	$\exists z.\ x \leq z \wedge t, z \models \psi \wedge \forall y.\ (x \leq y < z) \rightarrow t, y \models \varphi$

Example



Local temporal logics: Semantics

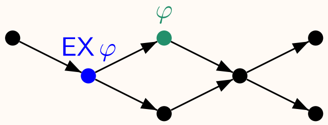
Example: $\mathrm{TL}_\Sigma(\neg, \vee, \mathsf{EX}, \mathsf{U})$

$$\varphi ::= a \ (a \in \Sigma) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{EX}\,\varphi \mid \varphi \,\mathsf{U}\, \varphi$$

Semantics: $t = [V, \leq, \lambda]$ with $\lambda : V \to \Sigma$ labelled partial order

$t, x \models a$	if	$\lambda(x) = a$
$t, x \models \mathsf{EX}\,\varphi$	if	$\exists y.\ x \lessdot y \wedge t, y \models \varphi$
$t, x \models \varphi \,\mathsf{U}\, \psi$	if	$\exists z.\ x \leq z \wedge t, z \models \psi \wedge \forall y.\ (x \leq y < z) \to t, y \models \varphi$

Example



$[\![\mathsf{EX}]\!]$ is definable in $\mathrm{FO}(\lessdot)$ or $\mathrm{FO}(\leq)$

Local temporal logics: Semantics

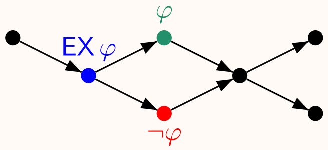
Example: $\mathrm{TL}_\Sigma(\neg, \vee, \mathsf{EX}, \mathsf{U})$

$$\varphi ::= a \ (a \in \Sigma) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{EX}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi$$

Semantics: $t = [V, \leq, \lambda]$ with $\lambda : V \to \Sigma$ labelled partial order

$t, x \models a$	if	$\lambda(x) = a$
$t, x \models \mathsf{EX}\,\varphi$	if	$\exists y.\ x \lessdot y \wedge t, y \models \varphi$
$t, x \models \varphi\,\mathsf{U}\,\psi$	if	$\exists z.\ x \leq z \wedge t, z \models \psi \wedge \forall y.\ (x \leq y < z) \to t, y \models \varphi$

Example There may be several next events



$[\![\mathsf{EX}]\!]$ is definable in $\mathrm{FO}(\lessdot)$ or $\mathrm{FO}(\leq)$

Local temporal logics: Semantics

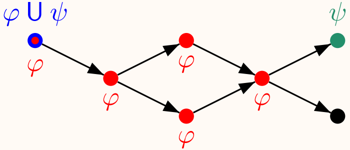
Example: $\mathrm{TL}_\Sigma(\neg, \vee, \mathsf{EX}, \mathsf{U})$

$$\varphi ::= a \ (a \in \Sigma) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{EX}\,\varphi \mid \varphi \,\mathsf{U}\, \varphi$$

Semantics: $t = [V, \leq, \lambda]$ with $\lambda : V \to \Sigma$ labelled partial order

$t, x \models a$	if	$\lambda(x) = a$
$t, x \models \mathsf{EX}\,\varphi$	if	$\exists y.\ x \lessdot y \wedge t, y \models \varphi$
$t, x \models \varphi \,\mathsf{U}\, \psi$	if	$\exists z.\ x \leq z \wedge t, z \models \psi \wedge \forall y.\ (x \leq y < z) \to t, y \models \varphi$

Example



$[\![\mathsf{U}]\!]$ is definable in $\mathrm{FO}(\leq)$

Local temporal logics: Semantics

Example: $\mathrm{TL}_\Sigma(\neg, \vee, \mathsf{EX}, \mathsf{U})$

$$\varphi ::= a \ (a \in \Sigma) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{EX}\,\varphi \mid \varphi \,\mathsf{U}\, \varphi$$

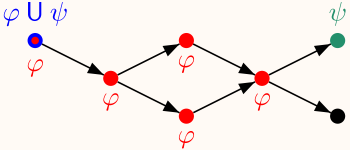
Semantics: $t = [V, \leq, \lambda]$ with $\lambda : V \to \Sigma$ labelled partial order

$$t, x \models a \quad \text{if} \quad \lambda(x) = a$$

$$t, x \models \mathsf{EX}\,\varphi \quad \text{if} \quad \exists y.\ x \lessdot y \wedge t, y \models \varphi$$

$$t, x \models \varphi \,\mathsf{U}\, \psi \quad \text{if} \quad \exists z.\ x \leq z \wedge t, z \models \psi \wedge \forall y.\ (x \leq y < z) \to t, y \models \varphi$$

Example



$\mathrm{TL}_\Sigma(\neg, \vee, \mathsf{EX}, \mathsf{U})$ is $\mathrm{FO}(\leq)$-definable

Local temporal logics: Definability

Definition

Let $\mathcal{L}$ be some logic (MSO, FO, $\mathrm{M}\Sigma_n^1$, ...).

$\mathrm{TL}_N(B)$ is $\mathcal{L}$-definable if the semantics of each modality $M \in B$ is given by some formula $[\![M]\!]$ in $\mathcal{L}$.

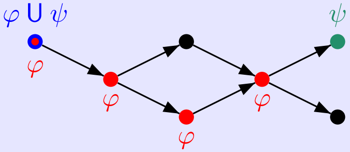
Various logics $\mathcal{L}$

- $\mathrm{FO}(\lessdot)$ or $\mathrm{FO}(\leq)$
- $\mathrm{MSO}(\lessdot) = \mathrm{MSO}(\leq)$
- $\mathrm{M}\Sigma_n^1(\lessdot)$: formulas logically equivalent with one of the form
$$\exists \overrightarrow{X_1} \forall \overrightarrow{X_2} \ldots \exists/\forall \overrightarrow{X_n} \varphi$$
where φ does not contain any second-order quantification.
- $\mathrm{M}\Pi_n^1(\lessdot)$: formulas whose negation are in $\mathrm{M}\Sigma_n^1(\lessdot)$, i.e., which are logically equivalent with one of the form
$$\forall \overrightarrow{X_1} \exists \overrightarrow{X_2} \ldots \exists/\forall \overrightarrow{X_n} \varphi$$
where φ does not contain any second-order quantification.
- $\mathrm{M}\Delta_n^1(\lessdot) = \mathrm{M}\Sigma_n^1(\lessdot) \cap \mathrm{M}\Pi_n^1(\lessdot)$

Existential until (TLC, Alur et al., LICS'95)

Semantics

$$t, x \models \varphi \,\mathsf{EU}\, \psi \quad \text{if} \quad \exists x = x_0 \lessdot x_1 \lessdot \cdots \lessdot x_n, \quad t, x_n \models \psi \wedge \forall 0 \leq i < n, t, x_i \models \varphi$$



EU is $\mathrm{M}\Sigma_1^1(\lessdot)$-definable over traces

$$t, x \models \varphi \,\mathsf{EU}\, \psi \quad \text{iff} \quad \exists Z, \quad \forall z \in Z, \; ((t, z \models \varphi \vee \psi) \wedge (z = x \vee \exists y \in Z, y \lessdot z)) \wedge \exists z \in Z, \; t, z \models \psi$$

Correct if any event in t has a finite past.

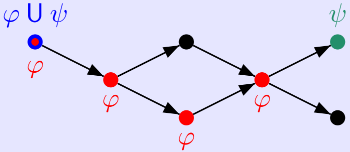
EU is $\mathrm{M}\Delta_1^1(\lessdot)$-definable over traces

$$t, x \not\models \varphi \,\mathsf{EU}\, \psi \quad \text{iff} \quad \exists Y, \quad \forall y, \; (y \in Y \leftrightarrow t, y \models \varphi \wedge (y = x \vee \exists z \in Y, \; z \lessdot y)) \wedge \forall z, \; t, z \models \psi \rightarrow (z \neq x \wedge \neg \exists y \in Y, \; y \lessdot z)$$

Existential until (TLC, Alur et al., LICS'95)

Semantics

$$t, x \models \varphi\,\mathsf{EU}\,\psi \quad \text{if} \quad \exists x = x_0 \lessdot x_1 \lessdot \cdots \lessdot x_n,$$
$$t, x_n \models \psi \land \forall 0 \leq i < n, t, x_i \models \varphi$$



EU is $\mathrm{M}\Sigma^1_1(\lessdot)$-definable over traces

$$t, x \models \varphi\,\mathsf{EU}\,\psi \quad \text{iff} \quad \exists Z, \quad \forall z \in Z,\ ((t, z \models \varphi \lor \psi) \land (z = x \lor \exists y \in Z, y \lessdot z))$$
$$\land\, \exists z \in Z,\ t, z \models \psi$$

Correct if any event in t has a finite past.

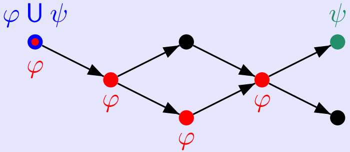
EU is $\mathrm{M}\Delta^1_1(\lessdot)$-definable over traces

$$t, x \not\models \varphi\,\mathsf{EU}\,\psi \quad \text{iff} \quad \exists Y, \quad \forall y,\ (y \in Y \leftrightarrow t, y \models \varphi \land (y = x \lor \exists z \in Y,\ z \lessdot y))$$
$$\land\, \forall z,\ t, z \models \psi \to (z \neq x \land \neg \exists y \in Y,\ y \lessdot z)$$

Existential until (TLC, Alur et al., LICS'95)

Semantics

$$t, x \models \varphi \, \mathsf{EU} \, \psi \quad \text{if} \quad \exists x = x_0 \lessdot x_1 \lessdot \cdots \lessdot x_n, \quad t, x_n \models \psi \wedge \forall 0 \leq i < n, t, x_i \models \varphi$$



EU is $\mathrm{M}\Sigma_1^1(\lessdot)$-definable over traces

$$t, x \models \varphi \, \mathsf{EU} \, \psi \quad \text{iff} \quad \exists Z, \quad \forall z \in Z, \; ((t, z \models \varphi \vee \psi) \wedge (z = x \vee \exists y \in Z, y \lessdot z)) \wedge \exists z \in Z, \; t, z \models \psi$$

Correct if any event in t has a finite past.

EU is $\mathrm{M}\Delta_1^1(\lessdot)$-definable over traces

$$t, x \not\models \varphi \, \mathsf{EU} \, \psi \quad \text{iff} \quad \exists Y, \quad \forall y, \; (y \in Y \leftrightarrow t, y \models \varphi \wedge (y = x \vee \exists z \in Y, \; z \lessdot y)) \wedge \forall z, \; t, z \models \psi \rightarrow (z \neq x \wedge \neg \exists y \in Y, \; y \lessdot z)$$

Known local temporal logics

Proposition

All known local temporal logics over traces are definable in $\mathrm{M}\Delta^1_1(\lessdot, \mathrm{finite})$.

Auxiliary formulas

$$(X = \uparrow x) \quad := \quad \forall y (y \in X \leftrightarrow y = x \vee \exists z (z \in X \wedge z \lessdot y))$$
$$(X = \downarrow x) \quad := \quad \forall y (y \in X \leftrightarrow y = x \vee \exists z (z \in X \wedge y \lessdot z)) \wedge \mathrm{finite}(X)$$

Correct if any event in t has a finite past.

Semantics of Eco

$$t, x \models \mathsf{Eco}\, \varphi \quad \text{if} \quad \exists y \, (x \not\leq y \wedge y \not\leq x \wedge t, y \models \varphi)$$

Eco is definable in $\mathrm{M}\Delta^1_1(\lessdot)$

$$t, x \models \mathsf{Eco}\, \varphi \quad \text{iff} \quad \exists X \exists Y \, (X = \uparrow x) \wedge (Y = \downarrow x) \wedge \exists z \, (z \notin X \cup Y \wedge t, z \models \varphi)$$
$$t, x \not\models \mathsf{Eco}\, \varphi \quad \text{iff} \quad \exists X \exists Y \, (X = \uparrow x) \wedge (Y = \downarrow x) \wedge \forall z \, (z \notin X \cup Y \rightarrow t, z \not\models \varphi)$$

$\mathrm{TL}_\Sigma(\mathsf{EX}, \mathsf{Eco}, \mathsf{EU})$ is $\mathrm{M}\Delta^1_1(\lessdot)$-definable (also TLC from Alur et al.)

Known local temporal logics

Proposition

All known local temporal logics over traces are definable in $\mathrm{M}\Delta^1_1(\lessdot, \mathrm{finite})$.

Auxiliary formulas

$$
\begin{aligned}
(X = {\uparrow}x) &:= \forall y(y \in X \leftrightarrow y = x \vee \exists z(z \in X \wedge z \lessdot y)) \\
(X = {\downarrow}x) &:= \forall y(y \in X \leftrightarrow y = x \vee \exists z(z \in X \wedge y \lessdot z)) \wedge \mathrm{finite}(X)
\end{aligned}
$$

Correct if any event in t has a finite past.

Semantics of Eco

$$t, x \models \mathsf{Eco}\,\varphi \quad \text{if} \quad \exists y\ (x \not\leq y \wedge y \not\leq x \wedge t, y \models \varphi)$$

Eco is definable in $\mathrm{M}\Delta^1_1(\lessdot)$

$$t, x \models \mathsf{Eco}\,\varphi \quad \text{iff} \quad \exists X \exists Y\ (X = {\uparrow}x) \wedge (Y = {\downarrow}x) \wedge \exists z\ (z \notin X \cup Y \wedge t, z \models \varphi)$$

$$t, x \not\models \mathsf{Eco}\,\varphi \quad \text{iff} \quad \exists X \exists Y\ (X = {\uparrow}x) \wedge (Y = {\downarrow}x) \wedge \forall z\ (z \notin X \cup Y \rightarrow t, z \not\models \varphi)$$

$\mathrm{TL}_\Sigma(\mathsf{EX}, \mathsf{Eco}, \mathsf{EU})$ is $\mathrm{M}\Delta^1_1(\lessdot)$-definable (also TLC from Alur et al.)

Known local temporal logics

Proposition

All known local temporal logics over traces are definable in $\mathrm{M}\Delta^1_1(\lessdot, \mathrm{finite})$.

Auxiliary formulas

$$\begin{array}{lcl} (X = \uparrow x) & := & \forall y (y \in X \leftrightarrow y = x \vee \exists z (z \in X \wedge z \lessdot y)) \\ (X = \downarrow x) & := & \forall y (y \in X \leftrightarrow y = x \vee \exists z (z \in X \wedge y \lessdot z)) \wedge \mathrm{finite}(X) \end{array}$$

Correct if any event in t has a finite past.

Semantics of Eco

$$t, x \models \mathsf{Eco}\, \varphi \quad \text{if} \quad \exists y \ (x \not\leq y \wedge y \not\leq x \wedge t, y \models \varphi)$$

Eco is definable in $\mathrm{M}\Delta^1_1(\lessdot)$

$$t, x \models \mathsf{Eco}\, \varphi \quad \text{iff} \quad \exists X \exists Y \ (X = \uparrow x) \wedge (Y = \downarrow x) \wedge \exists z \ (z \notin X \cup Y \wedge t, z \models \varphi)$$

$$t, x \not\models \mathsf{Eco}\, \varphi \quad \text{iff} \quad \exists X \exists Y \ (X = \uparrow x) \wedge (Y = \downarrow x) \wedge \forall z \ (z \notin X \cup Y \rightarrow t, z \not\models \varphi)$$

$\mathrm{TL}_\Sigma(\mathsf{EX}, \mathsf{Eco}, \mathsf{EU})$ is $\mathrm{M}\Delta^1_1(\lessdot)$-definable (also TLC from Alur et al.)

Known local temporal logics

Proposition

All known local temporal logics over traces are definable in $\mathrm{M}\Delta_1^1(\lessdot, \mathrm{finite})$.

Auxiliary formulas

$$(X = \uparrow x) \quad := \quad \forall y(y \in X \leftrightarrow y = x \vee \exists z(z \in X \wedge z \lessdot y))$$
$$(X = \downarrow x) \quad := \quad \forall y(y \in X \leftrightarrow y = x \vee \exists z(z \in X \wedge y \lessdot z)) \wedge \mathrm{finite}(X)$$

Correct if any event in t has a finite past.

Semantics of Eco

$$t, x \models \text{Eco}\,\varphi \quad \text{if} \quad \exists y\ (x \not\leq y \wedge y \not\leq x \wedge t, y \models \varphi)$$

Eco is definable in $\mathrm{M}\Delta_1^1(\lessdot)$

$$t, x \models \text{Eco}\,\varphi \quad \text{iff} \quad \exists X \exists Y\ (X = \uparrow x) \wedge (Y = \downarrow x) \wedge \exists z\ (z \notin X \cup Y \wedge t, z \models \varphi)$$
$$t, x \not\models \text{Eco}\,\varphi \quad \text{iff} \quad \exists X \exists Y\ (X = \uparrow x) \wedge (Y = \downarrow x) \wedge \forall z\ (z \notin X \cup Y \rightarrow t, z \not\models \varphi)$$

$\mathrm{TL}_\Sigma(\text{EX}, \text{Eco}, \text{EU})$ is $\mathrm{M}\Delta_1^1(\lessdot)$-definable (also TLC from Alur et al.)

Universal (strict) until

Semantics of SU

$$t, x \models \varphi \, \mathsf{SU} \, \psi \quad \text{if} \quad \exists z \, (x < z \wedge t, z \models \psi \wedge \forall y \, (x < y < z \rightarrow t, y \models \varphi)$$

Note that $\varphi \, \mathsf{U} \, \psi = \psi \vee (\varphi \wedge (\varphi \, \mathsf{SU} \, \psi))$.

SU is definable in $\mathrm{M}\Delta^1_1(\lessdot)$

$$\begin{aligned} t, x \models \varphi \, \mathsf{SU} \, \psi \quad \text{iff} \quad & \exists X \exists Y \, (X = \uparrow x) \\ & \wedge \forall y \, (y \in Y \leftrightarrow y \in X \setminus \{x\} \wedge \forall z \in X \setminus \{x\} \\ & \qquad (z \lessdot y \rightarrow z \in Y \wedge t, z \models \varphi) \\ & \wedge \exists y \, (y \in Y \wedge t, y \models \psi) \end{aligned}$$

$$\begin{aligned} t, x \not\models \varphi \, \mathsf{SU} \, \psi \quad \text{iff} \quad & \exists X \exists Y \, (X = \uparrow x) \\ & \wedge \forall y \, (y \in Y \leftrightarrow y \in X \setminus \{x\} \wedge \forall z \in X \setminus \{x\} \\ & \qquad (z \lessdot y \rightarrow z \in Y \wedge t, z \models \varphi) \\ & \wedge \forall y \, (y \in Y \rightarrow t, y \not\models \psi) \end{aligned}$$

$\mathrm{TL}_\Sigma(\mathsf{EX}, \mathsf{U})$ is $\mathrm{M}\Delta^1_1(\lessdot)$-definable

Universal (strict) until

Semantics of SU

$$t, x \models \varphi \,\mathsf{SU}\, \psi \quad \text{if} \quad \exists z \; (x < z \wedge t, z \models \psi \wedge \forall y \; (x < y < z \rightarrow t, y \models \varphi)$$

Note that $\varphi \,\mathsf{U}\, \psi = \psi \vee (\varphi \wedge (\varphi \,\mathsf{SU}\, \psi))$.

SU is definable in $\mathrm{M}\Delta^1_1(\lessdot)$

$$\begin{aligned} t, x \models \varphi \,\mathsf{SU}\, \psi \quad \text{iff} \quad & \exists X \exists Y \; (X = \uparrow x) \\ & \wedge \forall y \; (y \in Y \leftrightarrow y \in X \setminus \{x\} \wedge \forall z \in X \setminus \{x\} \\ & \qquad (z \lessdot y \rightarrow z \in Y \wedge t, z \models \varphi) \\ & \wedge \exists y \; (y \in Y \wedge t, y \models \psi) \end{aligned}$$

$$\begin{aligned} t, x \not\models \varphi \,\mathsf{SU}\, \psi \quad \text{iff} \quad & \exists X \exists Y \; (X = \uparrow x) \\ & \wedge \forall y \; (y \in Y \leftrightarrow y \in X \setminus \{x\} \wedge \forall z \in X \setminus \{x\} \\ & \qquad (z \lessdot y \rightarrow z \in Y \wedge t, z \models \varphi) \\ & \wedge \forall y \; (y \in Y \rightarrow t, y \not\models \psi) \end{aligned}$$

$\mathrm{TL}_\Sigma(\mathsf{EX}, \mathsf{U})$ is $\mathrm{M}\Delta^1_1(\lessdot)$-definable

Universal (strict) until

Semantics of SU

$$t, x \models \varphi \,\mathsf{SU}\, \psi \quad \text{if} \quad \exists z \; (x < z \wedge t, z \models \psi \wedge \forall y \; (x < y < z \rightarrow t, y \models \varphi)$$

Note that $\varphi \,\mathsf{U}\, \psi = \psi \vee (\varphi \wedge (\varphi \,\mathsf{SU}\, \psi))$.

SU is definable in $\mathrm{M}\Delta^1_1(\lessdot)$

$$\begin{aligned} t, x \models \varphi \,\mathsf{SU}\, \psi \quad \text{iff} \quad & \exists X \exists Y \; (X = \uparrow x) \\ & \wedge \forall y \; (y \in Y \leftrightarrow y \in X \setminus \{x\} \wedge \forall z \in X \setminus \{x\} \\ & \qquad (z \lessdot y \rightarrow z \in Y \wedge t, z \models \varphi) \\ & \wedge \exists y \; (y \in Y \wedge t, y \models \psi) \end{aligned}$$

$$\begin{aligned} t, x \not\models \varphi \,\mathsf{SU}\, \psi \quad \text{iff} \quad & \exists X \exists Y \; (X = \uparrow x) \\ & \wedge \forall y \; (y \in Y \leftrightarrow y \in X \setminus \{x\} \wedge \forall z \in X \setminus \{x\} \\ & \qquad (z \lessdot y \rightarrow z \in Y \wedge t, z \models \varphi) \\ & \wedge \forall y \; (y \in Y \rightarrow t, y \not\models \psi) \end{aligned}$$

$\mathrm{TL}_\Sigma(\mathsf{EX}, \mathsf{U})$ is $\mathrm{M}\Delta^1_1(\lessdot)$-definable

Outline

Overview

Definitions and some known temporal logics

Uniform satisfiability: upper bound

Uniform satisfiability: lower bound

Open problems

Uniform satisfiability problem: upper bound

Uniform satisfiability for $\mathrm{TL}_N(B)$

Input: A dependence alphabet (Σ, D) and a formula $\varphi \in \mathrm{TL}_\Sigma(B)$.

Question: Is there a trace $t = [V, \leq, \lambda]$ over (Σ, D) and a vertex $x \in V$ such that $t, x \models \varphi$?

Theorem 1

If $\mathrm{TL}_N(B)$ is $\mathrm{M}\Delta^1_n(\lessdot)$-definable then the uniform satisfiability problem over traces for $\mathrm{TL}_N(B)$ is decidable in n-$\mathbf{EXPSPACE}$, or more precisely in space

$$\mathtt{poly}(|\varphi|) \cdot \mathtt{tower}(n, \mathtt{poly}(|\Sigma|))$$

Corollary

The uniform satisfiability problem over traces for all known local temporal logics is decidable in 1-$\mathbf{EXPSPACE}$, or more precisely in space

$$\mathtt{poly}(|\varphi|) \cdot 2^{\mathtt{poly}(|\Sigma|))}$$

Uniform satisfiability problem: upper bound

Uniform satisfiability for $\mathrm{TL}_N(B)$

Input: A dependence alphabet (Σ, D) and a formula $\varphi \in \mathrm{TL}_\Sigma(B)$.

Question: Is there a trace $t = [V, \leq, \lambda]$ over (Σ, D) and a vertex $x \in V$ such that $t, x \models \varphi$?

Theorem 1

If $\mathrm{TL}_N(B)$ is $\mathrm{M}\Delta^1_n(\lessdot)$-definable then the uniform satisfiability problem over traces for $\mathrm{TL}_N(B)$ is decidable in n-$\mathbf{EXPSPACE}$, or more precisely in space

$$\mathtt{poly}(|\varphi|) \cdot \mathtt{tower}(n, \mathtt{poly}(|\Sigma|))$$

Corollary

The uniform satisfiability problem over traces for all known local temporal logics is decidable in 1-$\mathbf{EXPSPACE}$, or more precisely in space

$$\mathtt{poly}(|\varphi|) \cdot 2^{\mathtt{poly}(|\Sigma|)}$$

Uniform satisfiability problem: upper bound

Uniform satisfiability for $\mathrm{TL}_N(B)$

Input: A dependence alphabet (Σ, D) and a formula $\varphi \in \mathrm{TL}_\Sigma(B)$.

Question: Is there a trace $t = [V, \leq, \lambda]$ over (Σ, D) and a vertex $x \in V$ such that $t, x \models \varphi$?

Theorem 1

If $\mathrm{TL}_N(B)$ is $\mathrm{M}\Delta^1_n(\lessdot)$-definable then the uniform satisfiability problem over traces for $\mathrm{TL}_N(B)$ is decidable in n-**EXPSPACE**, or more precisely in space

$$\mathtt{poly}(|\varphi|) \cdot \mathtt{tower}(n, \mathtt{poly}(|\Sigma|))$$

Corollary

The uniform satisfiability problem over traces for all known local temporal logics is decidable in 1-**EXPSPACE**, or more precisely in space

$$\mathtt{poly}(|\varphi|) \cdot 2^{\mathtt{poly}(|\Sigma|)}$$

Sketch of proof

Notations

A word $w = a_0 a_1 a_2 \cdots \in \Sigma^\infty$ gives rise to a trace $[w] = [V, \leq, \lambda]$ over (Σ, D) with $V = \{i \in \mathbb{N} \mid 0 \leq i < |w|\}$, $\lambda(i) = a_i$ and

$$\leq \, = E^* \text{ where } (i,j) \in E \text{ if } i < j \text{ and } (a_i, a_j) \in D.$$

Idea of the proof

Given (Σ, D) and $\varphi \in \mathrm{TL}_\Sigma(B)$, we build a word automaton $\mathcal{A}$ over the alphabet $\Sigma \times \{0,1\}^{\mathrm{Sub}(\varphi)}$ which accepts exactly the words $(w, (X_\xi)_{\xi \in \mathrm{Sub}(\varphi)})$ such that

$$\forall \xi \in \mathrm{Sub}(\varphi), \quad X_\xi = \{i < |w| \mid [w], i \models \xi\}.$$

Then, we check whether $\mathcal{A}$ accepts some word whose φ-line X_φ is nonempty.

Modality automata

Given (Σ, D) and a formula $\xi = M(\xi_1, \ldots, \xi_m)$, we build a modality automaton $\mathcal{A}_\xi$ (depending on M only) over the alphabet $\Sigma \times \{0,1\}^{1+m}$ which accepts exactly the words $(w, X_0, X_1, \ldots, X_m)$ such that

$$X_0 = \{i < |w| \mid [w], i \models M(X_1, \ldots, X_m)\}.$$

Then, $\mathcal{A}$ is obtain as a product of the $(\mathcal{A}_\xi)_{\xi \in \mathrm{Sub}(\varphi)}$.

Sketch of proof

Notations

A word $w = a_0 a_1 a_2 \cdots \in \Sigma^\infty$ gives rise to a trace $[w] = [V, \leq, \lambda]$ over (Σ, D) with $V = \{i \in \mathbb{N} \mid 0 \leq i < |w|\}$, $\lambda(i) = a_i$ and

$$\leq\, = E^* \text{ where } (i,j) \in E \text{ if } i < j \text{ and } (a_i, a_j) \in D.$$

Idea of the proof

Given (Σ, D) and $\varphi \in \mathrm{TL}_\Sigma(B)$, we build a word automaton $\mathcal{A}$ over the alphabet $\Sigma \times \{0,1\}^{\mathrm{Sub}(\varphi)}$ which accepts exactly the words $(w, (X_\xi)_{\xi \in \mathrm{Sub}(\varphi)})$ such that

$$\forall \xi \in \mathrm{Sub}(\varphi), \quad X_\xi = \{i < |w| \mid [w], i \models \xi\}.$$

Then, we check whether $\mathcal{A}$ accepts some word whose φ-line X_φ is nonempty.

Modality automata

Given (Σ, D) and a formula $\xi = M(\xi_1, \ldots, \xi_m)$, we build a modality automaton $\mathcal{A}_\xi$ (depending on M only) over the alphabet $\Sigma \times \{0,1\}^{1+m}$ which accepts exactly the words $(w, X_0, X_1, \ldots, X_m)$ such that

$$X_0 = \{i < |w| \mid [w], i \models M(X_1, \ldots, X_m)\}.$$

Then, $\mathcal{A}$ is obtain as a product of the $(\mathcal{A}_\xi)_{\xi \in \mathrm{Sub}(\varphi)}$.

Sketch of proof

Notations

A word $w = a_0a_1a_2\cdots \in \Sigma^\infty$ gives rise to a trace $[w] = [V, \leq, \lambda]$ over (Σ, D) with $V = \{i \in \mathbb{N} \mid 0 \leq i < |w|\}$, $\lambda(i) = a_i$ and

$$\leq = E^* \text{ where } (i,j) \in E \text{ if } i < j \text{ and } (a_i, a_j) \in D.$$

Idea of the proof

Given (Σ, D) and $\varphi \in \mathrm{TL}_\Sigma(B)$, we build a word automaton $\mathcal{A}$ over the alphabet $\Sigma \times \{0,1\}^{\mathrm{Sub}(\varphi)}$ which accepts exactly the words $(w, (X_\xi)_{\xi\in\mathrm{Sub}(\varphi)})$ such that

$$\forall \xi \in \mathrm{Sub}(\varphi), \quad X_\xi = \{i < |w| \mid [w], i \models \xi\}.$$

Then, we check whether $\mathcal{A}$ accepts some word whose φ-line X_φ is nonempty.

Modality automata

Given (Σ, D) and a formula $\xi = M(\xi_1, \ldots, \xi_m)$, we build a modality automaton $\mathcal{A}_\xi$ (depending on M only) over the alphabet $\Sigma \times \{0,1\}^{1+m}$ which accepts exactly the words $(w, X_0, X_1, \ldots, X_m)$ such that

$$X_0 = \{i < |w| \mid [w], i \models M(X_1, \ldots, X_m)\}.$$

Then, $\mathcal{A}$ is obtain as a product of the $(\mathcal{A}_\xi)_{\xi\in\mathrm{Sub}(\varphi)}$.

Modality automata

Theorem

Given (Σ, D) and a modality M definable in $\mathrm{M}\Delta^1_n(\lessdot)$, we can build a modality automaton for M over (Σ, D) in space $\texttt{tower}(n+1, \texttt{poly}(|\Sigma|))$.

Naive approach

The modality automaton for M over (Σ, D) is a Büchi automaton for

$$\alpha_M(X_0, X_1, \ldots, X_m) = \forall x(x \in X_0 \leftrightarrow M(X_1, \ldots, X_m, x)$$

In α_M, we find atomic subformulas $x \lessdot y$. Hence, reading a word w, we have to check whether two positions $i, j < |w|$ are consecutive ($i \lessdot j$) in the trace $[w]$. This can be done by keeping a subset of Σ in memory.

Then, use the classical constructions on automata: projection ($\exists$), complement ($\neg$) and union ($\vee$).

Note that $\forall = \neg\exists\neg$ needs 2 determinizations hence yields 2 exponentials.

But, this yields an exponential tower even for $\mathrm{FO}(\lessdot)$-modalities, e.g., when

$$\alpha_M = \forall x \exists z_1 \forall z_2 \ldots \ \gamma_M.$$

Modality automata

Theorem

Given (Σ, D) and a modality M definable in $\mathrm{M}\Delta^1_n(\lessdot)$, we can build a modality automaton for M over (Σ, D) in space $\texttt{tower}(n+1, \texttt{poly}(|\Sigma|))$.

Naive approach

The modality automaton for M over (Σ, D) is a Büchi automaton for

$$\alpha_M(X_0, X_1, \ldots, X_m) = \forall x (x \in X_0 \leftrightarrow M(X_1, \ldots, X_m, x))$$

In α_M, we find atomic subformulas $x \lessdot y$. Hence, reading a word w, we have to check whether two positions $i, j < |w|$ are consecutive ($i \lessdot j$) in the trace $[w]$. This can be done by keeping a subset of Σ in memory.

Then, use the classical constructions on automata: projection ($\exists$), complement ($\neg$) and union ($\vee$).

Note that $\forall = \neg\exists\neg$ needs 2 determinizations hence yields 2 exponentials.

But, this yields an exponential tower even for $\mathrm{FO}(\lessdot)$-modalities, e.g., when

$$\alpha_M = \forall x \exists z_1 \forall z_2 \ldots\ \gamma_M.$$

Building the modality automaton

- The modality automaton for M over (Σ, D) is a Büchi automaton for
  $$\alpha_M(X_0, X_1, \ldots, X_m) = \forall x (x \in X_0 \leftrightarrow M(X_1, \ldots, X_m, x))$$
- If M is $\mathrm{M}\Delta^1_n(\lessdot)$-definable, then α_M is in $\mathrm{M}\Pi^1_n(\lessdot)$:
  $$\alpha_M = \forall \overrightarrow{Y_1} \exists \overrightarrow{Y_2} \ldots \exists/\forall \overrightarrow{Y_n} \beta_M(X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n})$$
  where β_M does not contain set quantifications.
- Using Schwentick and Bartelmann's theorem, we get
  $$\beta_M \sim \exists x_1 \ldots \exists x_\ell \; \forall y \; \gamma_M(X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n}, x_1, \ldots, x_\ell, y)$$
  where quantifications in γ_M are restricted to the $(r, \lessdot)$-sphere around y.

Theorem

Given (Σ, D), we can build in space $2^{|\Sigma|^{O(r)}}$ a sphere automaton which accepts exactly the words $(w, X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n}, x_1, \ldots, x_\ell)$ satisfying $\forall y \; \gamma_M$.

- By projection, we get an automaton for β_M. space $2^{|\Sigma|^{O(r)}}$
- Using projection and complement, we get a modality automaton for α_M.
  space $\mathtt{tower}(n+1, \mathtt{poly}(|\Sigma|))$

Building the modality automaton

- The modality automaton for M over (Σ, D) is a Büchi automaton for
$$\alpha_M(X_0, X_1, \ldots, X_m) = \forall x (x \in X_0 \leftrightarrow M(X_1, \ldots, X_m, x))$$
- If M is $\mathrm{M}\Delta^1_n(\lessdot)$-definable, then α_M is in $\mathrm{M}\Pi^1_n(\lessdot)$:
$$\alpha_M = \forall \overrightarrow{Y_1} \exists \overrightarrow{Y_2} \ldots \exists/\forall \overrightarrow{Y_n} \beta_M(X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n})$$
where β_M does not contain set quantifications.
- Using Schwentick and Bartelmann's theorem, we get
$$\beta_M \sim \exists x_1 \ldots \exists x_\ell \; \forall y \; \gamma_M(X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n}, x_1, \ldots, x_\ell, y)$$
where quantifications in γ_M are restricted to the $(r, \lessdot)$-sphere around y.

Theorem

Given (Σ, D), we can build in space $2^{|\Sigma|^{O(r)}}$ a sphere automaton which accepts exactly the words $(w, X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n}, x_1, \ldots, x_\ell)$ satisfying $\forall y \; \gamma_M$.

- By projection, we get an automaton for β_M. space $2^{|\Sigma|^{O(r)}}$
- Using projection and complement, we get a modality automaton for α_M.
space $\mathtt{tower}(n+1, \mathtt{poly}(|\Sigma|))$

Building the modality automaton

- The modality automaton for M over (Σ, D) is a Büchi automaton for
$$\alpha_M(X_0, X_1, \ldots, X_m) = \forall x(x \in X_0 \leftrightarrow M(X_1, \ldots, X_m, x))$$
- If M is $\mathrm{M}\Delta^1_n(\lessdot)$-definable, then α_M is in $\mathrm{M}\Pi^1_n(\lessdot)$:
$$\alpha_M = \forall \overrightarrow{Y_1} \exists \overrightarrow{Y_2} \ldots \exists/\forall \overrightarrow{Y_n} \beta_M(X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n})$$
where β_M does not contain set quantifications.
- Using Schwentick and Bartelmann's theorem, we get
$$\beta_M \sim \exists x_1 \ldots \exists x_\ell \, \forall y \, \gamma_M(X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n}, x_1, \ldots, x_\ell, y)$$
where quantifications in γ_M are restricted to the $(r, \lessdot)$-sphere around y.

Theorem

Given (Σ, D), we can build in space $2^{|\Sigma|^{O(r)}}$ a sphere automaton which accepts exactly the words $(w, X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n}, x_1, \ldots, x_\ell)$ satisfying $\forall y \, \gamma_M$.

- By projection, we get an automaton for β_M. space $2^{|\Sigma|^{O(r)}}$
- Using projection and complement, we get a modality automaton for α_M.
space $\mathtt{tower}(n+1, \mathtt{poly}(|\Sigma|))$

Building the modality automaton

- The modality automaton for M over (Σ, D) is a Büchi automaton for
$$\alpha_M(X_0, X_1, \ldots, X_m) = \forall x(x \in X_0 \leftrightarrow M(X_1, \ldots, X_m, x))$$
- If M is $\mathrm{M}\Delta^1_n(\lessdot)$-definable, then α_M is in $\mathrm{M}\Pi^1_n(\lessdot)$:
$$\alpha_M = \forall \overrightarrow{Y_1} \exists \overrightarrow{Y_2} \ldots \exists/\forall \overrightarrow{Y_n} \beta_M(X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n})$$
where β_M does not contain set quantifications.
- Using Schwentick and Bartelmann's theorem, we get
$$\beta_M \sim \exists x_1 \ldots \exists x_\ell \; \forall y \; \gamma_M(X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n}, x_1, \ldots, x_\ell, y)$$
where quantifications in γ_M are restricted to the $(r, \lessdot)$-sphere around y.

Theorem

Given (Σ, D), we can build in space $2^{|\Sigma|^{O(r)}}$ a sphere automaton which accepts exactly the words $(w, X_0, \ldots, X_m, \overrightarrow{Y_1}, \ldots, \overrightarrow{Y_n}, x_1, \ldots, x_\ell)$ satisfying $\forall y \; \gamma_M$.

- By projection, we get an automaton for β_M. space $2^{|\Sigma|^{O(r)}}$
- Using projection and complement, we get a modality automaton for α_M.
space $\texttt{tower}(n+1, \texttt{poly}(|\Sigma|))$

Outline

Uniform satisfiability problem: lower bound

Uniform satisfiability for $\mathrm{TL}_N(B)$

Input: A dependence alphabet (Σ, D) and a formula $\varphi \in \mathrm{TL}_\Sigma(B)$.

Question: Is there a trace $t = [V, \leq, \lambda]$ over (Σ, D) and a vertex $x \in V$ such that $t, x \models \varphi$?

Theorem 2

There is local temporal logic $\mathrm{TL}_N(\neg, \vee, \mathsf{SU}, M)$ where the modality M is definable in $\mathrm{M\Pi}^1_n(\lessdot)$ for which the uniform satisfiability problem is $(n-1)$-$\mathbf{EXPSPACE}$ hard.

Proof idea

The modality M ensures that the trace encode the computation of a turing machine that works in $(n-1)$-$\mathbf{EXPSPACE}$.

Details in the paper.

Uniform satisfiability problem: lower bound

Uniform satisfiability for $\mathrm{TL}_N(B)$

Input: A dependence alphabet (Σ, D) and a formula $\varphi \in \mathrm{TL}_\Sigma(B)$.

Question: Is there a trace $t = [V, \leq, \lambda]$ over (Σ, D) and a vertex $x \in V$ such that $t, x \models \varphi$?

Theorem 2

There is local temporal logic $\mathrm{TL}_N(\neg, \vee, \mathsf{SU}, M)$ where the modality M is definable in $\mathrm{M\Pi}^1_n(\lessdot)$ for which the uniform satisfiability problem is $(n-1)$-$\mathbf{EXPSPACE}$ hard.

Proof idea

The modality M ensures that the trace encode the computation of a turing machine that works in $(n-1)$-$\mathbf{EXPSPACE}$.

Details in the paper.

Uniform satisfiability problem: lower bound

Uniform satisfiability for $\mathrm{TL}_N(B)$

Input: A dependence alphabet (Σ, D) and a formula $\varphi \in \mathrm{TL}_\Sigma(B)$.

Question: Is there a trace $t = [V, \leq, \lambda]$ over (Σ, D) and a vertex $x \in V$ such that $t, x \models \varphi$?

Theorem 2

There is local temporal logic $\mathrm{TL}_N(\neg, \vee, \mathsf{SU}, M)$ where the modality M is definable in $\mathrm{M\Pi}^1_n(\lessdot)$ for which the uniform satisfiability problem is $(n-1)$-**EXPSPACE** hard.

Proof idea

The modality M ensures that the trace encode the computation of a turing machine that works in $(n-1)$-**EXPSPACE**.

Details in the paper.

Outline

Open problems

- Can we cope similarly with the uniform model checking problem?
- Can we close the gap between the lower and upper bounds for the uniform satisfiability problem?
- Can we give some non trivial upper bound if we use $\leq$ instead of $\lessdot$?
- Can we extend these results to other labelled partial orders such as MSC's?