

Specifications for distributed systems

Paul Gastin and Marc Zeitoun

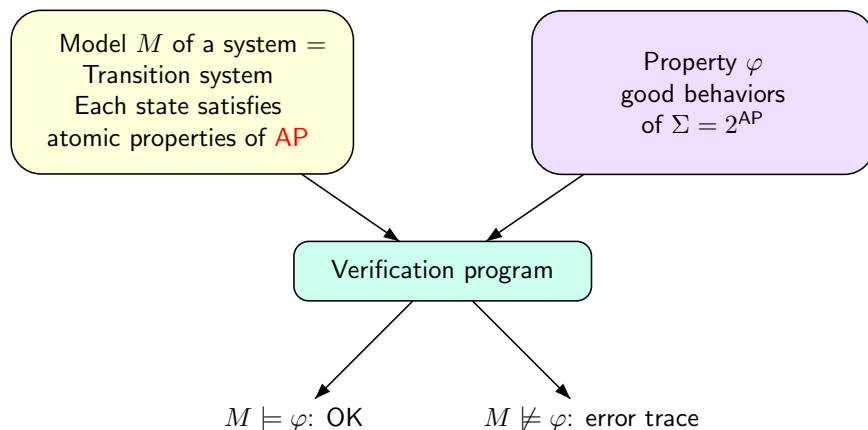
LIAFA, Université Paris 7, UMR CNRS 7089
 Paul.Gastin,Marc.Zeitoun@liafa.jussieu.fr

EPIT Concurrency 26-30/04/2004

Outline

- 1 Distributed behaviors and specifications
- 2 First order logic
- 3 Monadic second order logic
- 4 Linear temporal logic for sequences
- 5 Global temporal logics
- 6 Local temporal logics
- 7 MSO-definable local TL
- 8 Local μ -calculus

Context: the model checking

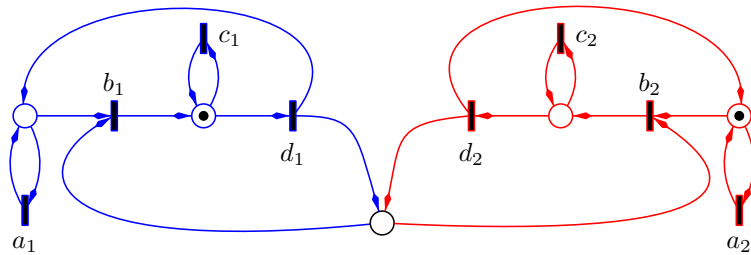


$M \models \varphi$ means **all behaviors of M** satisfy φ (linear time semantics)

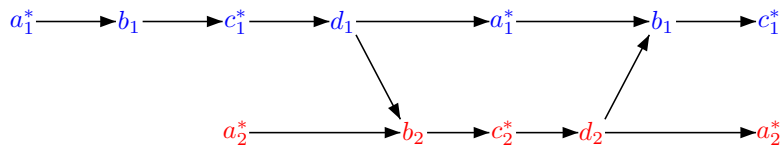
Concurrent models and their behaviors

Sequential setting	
System	Behaviors
Automaton, Kripke structure	Word
Distributed setting	
System	Behaviors: labeled DAGs
Event structure	
Petri Net	
1-safe Petri Net	
Asynchronous automaton	Mazurkiewicz trace
Communicating automaton	Mazurkiewicz trace
Netchart	Message Sequence Chart (MSC)
HMSC	MSC
	MSC

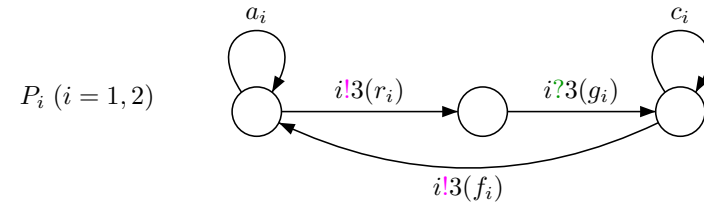
A 1-safe Petri net



Behavior: a Mazurkiewicz trace

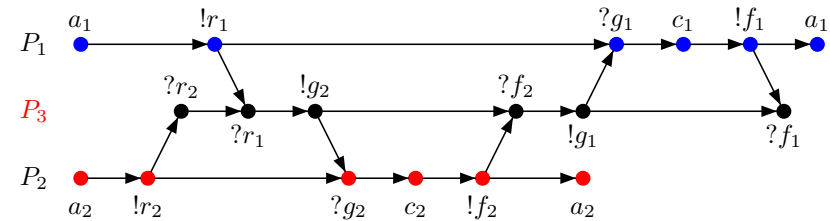


A communicating automaton



- P_3 grants requests of P_1 and P_2 with a FIFO policy.

Behavior: a MSC



Representing behaviors

Recap

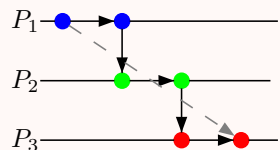
Behaviors = labeled DAGs. Not sets of (equivalent) words.

Behaviors

- Labeled DAGs $t = [V, E, \lambda]$ with $E \subseteq V \times V$ and
 - $\lambda : V \rightarrow \Sigma = 2^{AP}$ labels each vertex (action, event) of t .
 - E depicts causality we are interested in.
- $\leq = E^*$ is a partial order
- $< \text{ is } (< \setminus <^2)$, i.e., the immediate predecessor relation (Hasse).

Example: MSC

- $x E_m y$ if $(x, y) = \text{message}$, or $x < y$ on the same process p .



- $\leq = E_m^*$: visual order.
- Here, $< \subsetneq E_m \subsetneq <$.

Mazurkiewicz Traces

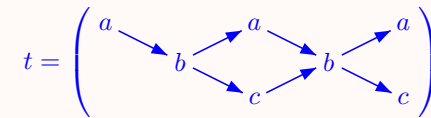
- Dependence alphabet $(\Sigma, D) = a - b - c - d$
- $D \subseteq \Sigma \times \Sigma$ reflexive and symmetric dependence relation.

Mazurkiewicz traces

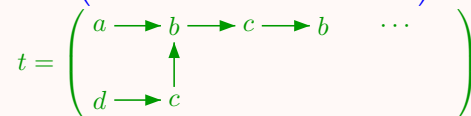
Trace = labeled partial order $t = [V, \leq, \lambda]$ such that

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

A finite trace



An infinite trace



Specification languages and criteria

Logical formalisms used to specify properties of systems.

Practical criteria

- How easy is it to write a specification?
- How easy is it to read a specification?

Theoretical criteria

- Decidability of SAT and MC.
- Complexity.
- Expressiveness.
- Conciseness.

Specification benchmark

Specifications: all executions of the system satisfy...

- Safety: All executions remain in good (local/global) states.
- MutEx: Safety of a global state.
- Liveness: Process p is activated infinitely often.
- Response: Every request will eventually be granted.
- Response': Same + between 2 requests, there is a grant.
- Release: Alarm remains active as long as not switched off.

Satisfiability and Model checking

We are given

- a universe of behaviors \mathbb{B} ,
- a class of models \mathbb{M} ,
- a specification language \mathbb{L} .

SAT

Input A formula $\varphi \in \mathbb{L}$.

Problem Does there exist $t \in \mathbb{B}$ such that $t \models \varphi$?

MC

Input A formula $\varphi \in \mathbb{L}$,

A model $M \in \mathbb{M}$ with behaviors $\mathcal{B}(M) (\subseteq \mathbb{B})$.

Problem Does $t \models \varphi$ hold for all $t \in \mathcal{B}(M)$?

Satisfiability & model checking: reductions

Reduction SAT \leq MC

Assume we have in \mathbb{M} a universal model $MU: \mathcal{B}(MU) = \mathbb{B}$.

$$\varphi \text{ satisfiable} \iff MU \not\models \neg\varphi$$

Reduction MC \leq SAT

Assume that any model can be effectively encoded by a formula, that is

$$M \rightsquigarrow \varphi_M : \forall t \in \mathbb{B}, t \models \varphi_M \iff t \in \mathcal{B}(M)$$

$$M \models \varphi \iff (\varphi_M \Rightarrow \varphi) \text{ is a tautology}$$

$$\iff (\varphi_M \wedge \neg\varphi) \text{ is not satisfiable}$$

Model checking in practice

Build from φ an automaton \mathcal{A}_φ recognizing behaviors satisfying φ :

$$\forall t \in \mathbb{B}, t \models \varphi \iff t \in \mathcal{L}(\mathcal{A}_\varphi)$$

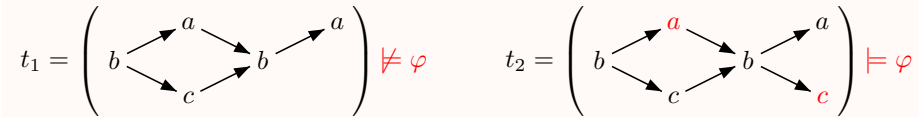
Then, $\mathcal{L}(M \times \mathcal{A}_{\neg\varphi}) = \emptyset \iff M \models \varphi$

Interleaving approach

Instead of verifying properties of distributed behaviors, check them on linearizations

Distributed model $M \rightsquigarrow$ Sequential transition system
 Distributed behavior \rightsquigarrow Set of equivalent linearizations (observations)

Example: checking a property on Mazurkiewicz traces



- Linearizations of t_1 : $bacba, bcaba$.
 - Linearizations of t_2 : $bacbac, bcabac, bacbca, bcabca$.
- $\varphi = \text{there is an } a \text{ before a } c = \exists x \exists y (x \leq y) \wedge (\lambda(x) = a) \wedge (\lambda(y) = c)$

Equivalent formula on linearizations, with $(\Sigma, D) = a - b - c?$

$$\Phi = \exists x \exists y (x \leq_w y) \wedge (\lambda(x) = a) \wedge (\lambda(y) = c) \wedge \exists z (x < z) \wedge (z < y) \wedge \lambda(z) = b$$

Uses the fact that φ holds on $t \in (\Sigma, D)$ iff there exist $a < c$ ordered via some b .

Interleaving approach: + & -

Advantage

One may use sequential specification languages & existing model checking tools.

Drawbacks

- Need of "robust" specifications: φ robust if $u \sim v \implies (u \models \varphi \iff v \models \varphi)$.
- Distributed specifications translated for linearizations often depend on the underlying dependencies or model under check.
- Translations on linearizations complex and cumbersome to obtain.
- Not usable if several distributed behaviors correspond to a single linearization.

Examples

Traces MutEx: $\exists x \exists y \neg(x \leq y) \wedge \neg(y \leq x) \wedge (\lambda(x) = c_1) \wedge (\lambda(y) = c_2)$

linearizations $x \leq y : \exists x = x_0 \dots \exists x_{|\Sigma|} = y, \bigwedge [x_i \leq_w x_{i+1} \wedge (\lambda(x_i) D \lambda(x_{i+1}))]$

MSCs E_m not expressible in $\text{MSO}(<_w)$ on linearizations.

Conclusion: Logics on strings (linearizations) are not adequate

Properties of distributed systems don't depend on ordering of independent actions.

Outline

- 1 Distributed behaviors and specifications
- 2 First order logic
- 3 Monadic second order logic
- 4 Linear temporal logic for sequences
- 5 Global temporal logics
- 6 Local temporal logics
- 7 MSO-definable local TL
- 8 Local μ -calculus

First Order Logic

Syntax: $FO_\Sigma(\preceq)$ ($\Sigma = 2^{\text{AP}}$)

$\varphi ::= \perp \mid P_q(x) \mid \neg\varphi \mid \varphi \vee \psi \mid x \preceq y \mid \exists x \varphi$ ($q \in \text{AP}, x, y \in \text{Var}$)

Semantics

- A formula is evaluated on labeled DAGs $t = [V, E, \lambda]$.
- $\sigma : \text{Var} \rightarrow V$ is a interpretation of (free) variables.
- \preceq interpreted as a relation between $< = E^+$ and $\leq = < \setminus <^2$.

$$\begin{aligned} t, \sigma \models P_q(x) & \text{ if } q \in \lambda(\sigma(x)) \\ t, \sigma \models \neg\varphi & \text{ if } t, \sigma \not\models \varphi \\ t, \sigma \models \varphi \vee \psi & \text{ if } t, \sigma \models \varphi \text{ or } t, \sigma \models \psi \\ t, \sigma \models x \preceq y & \text{ if } \sigma(x) \preceq \sigma(y) \\ t, \sigma \models \exists x \varphi & \text{ if } \exists v \in V : t, \{\sigma \cup [x \mapsto v]\} \models \varphi \end{aligned}$$

Macros

$\forall x \varphi : \neg \exists x \neg \varphi$ $\varphi \wedge \psi : \neg(\neg\varphi \vee \neg\psi)$ $\varphi \Rightarrow \psi : \psi \vee \neg\varphi$ $\varphi \Leftrightarrow \psi \dots$

A sentence $\varphi \in FO(\preceq)$ defines the language $\mathcal{L}(\varphi) = \{t \mid t \models \varphi\} \subseteq \mathbb{R}(\Sigma, D)$.

First Order Logic — Examples

On words (as Σ -labeled total orders)

- $\lambda(x) = a$ where $a \in \Sigma = 2^{\text{AP}} : \bigwedge_{q \in a} P_q(x) \wedge \bigwedge_{q \notin a} \neg P_q(x)$
- Conversely, $P_q(x) : \bigvee_{a \in \Sigma} \lambda(x) = a$
- $\varphi = \forall x (\lambda(x) = a \vee \lambda(x) = b) \wedge \forall y \forall z (y < z) \Rightarrow [\lambda(y) \neq \lambda(z)]$

$bababababa \dots \models \varphi$
 $abaabaabaaba \dots \not\models \varphi$

- $<$ is expressible in $FO_{\Sigma}(\leq)$:

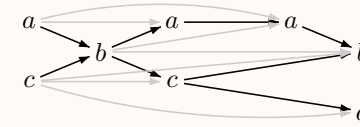
$$x < y : (x < y) \wedge (\neg \exists z, x < z < y)$$

- Conversely, $< = <^+$ is **not** expressible in $FO_{\Sigma}(<)$.
- $\min(x) : \neg \exists z, z < x$. Also valid on DAGs, as well as:
- $(\min = A) : \bigwedge_{a \in A} \exists x [\lambda(x) = a \wedge \min(x)] \wedge \bigwedge_{a \in \Sigma \setminus A} \neg \exists x [\lambda(x) = a \wedge \min(x)]$.
- $\mathcal{L}(\varphi \wedge \min = \{a\} \wedge \max = \{b\}) = (ab)^+$.
- The language $(aa)^+$ **cannot** be expressed!

First Order Logic — Examples

On traces

$D = a - b - c - d$ E_t in grey $<$ in black



- $x E_t y : x < y \wedge \bigvee_{(a,b) \in D} (\lambda(x) = a) \wedge (\lambda(y) = b)$
- $x < y : \bigvee_{k=0}^{|\Sigma|} \exists x = x_0, \dots, x_k = y, \bigwedge [x_i E_t x_{i+1}]$
- $x \parallel y : \neg(x < y) \wedge \neg(y < x)$.
- $\text{Path}_k(x_1, \dots, x_k) : \bigwedge_{1 \leq i < j \leq k} \neg(x_i \parallel x_j) \wedge \forall z \bigvee_{i,j} (x_i < z < x_j) \Rightarrow \bigvee_i z = x_i$
- $\text{MaxAntichain}(x_1, \dots, x_k) : \bigwedge_{1 \leq i < j \leq k} (x_i \parallel x_j) \wedge \forall y \bigvee_{1 \leq i \leq k} \neg(y \parallel x_i)$

On MSCs (FIFO)

$\text{msg}(x, y) : x < y \wedge \bigvee_{a,i,j} (\lambda(x) = a) \wedge (\lambda(y) = a) \wedge \neg \exists z (x < z < y) \wedge (\lambda(z) = \lambda(y))$

Every request should be granted: $\forall x \lambda(x) = !r \Rightarrow \exists y (x < y) \wedge \lambda(y) = ?g$

Not FO-expressible with natural MSC relation $<_{\text{Msg}} \cup \cup <_p$.

Specifications in FO

Specifications:

- Local
 - Safety $\neg \exists x \lambda(x) \in \text{Bad}_i$
 - Liveness $\exists x \lambda(x) = \text{act}_i \wedge \forall y [(\lambda(y) = \text{act}_i) \Rightarrow \exists z (z > y) \wedge (\lambda(z) = \text{act}_i)]$
 - Response $\forall x \lambda(x) = !\text{req}_i \Rightarrow \exists y (y > x) \wedge \lambda(y) = ?\text{grant}_i$
 - Response' $\text{Response} \wedge \forall x \forall y [(x < y) \wedge (\lambda(x) = \lambda(y) = !\text{req}_i)] \Rightarrow \exists z (x < z < y) \wedge \lambda(z) = ?\text{grant}_i$
 - Release (words) $\forall x [\text{alarm} \in \lambda(x) \Rightarrow \text{off} \in \lambda(x) \vee \forall y (x < y \Rightarrow \text{alarm} \in \lambda(y))]$
- Global
 - MutEx $\neg \exists x_1 \dots \exists x_k \text{MaxAntichain}(x_1, \dots, x_k) \wedge (\lambda(x_i))_{1 \leq i \leq k} \in \text{Bad}$
 - Response $\forall x_1 \dots \forall x_k [\text{MaxAntichain}(x_1, \dots, x_k) \wedge (\lambda(x_i))_{1 \leq i \leq k} = \text{Req}] \Rightarrow \exists y_1 \dots \exists y_\ell, \text{MaxAntichain}(y_1, \dots, y_\ell) \wedge (x_i)_{1 \leq i \leq k} < (y_j)_{1 \leq j \leq \ell} \wedge (\lambda(y_i))_{1 \leq i \leq k} = \text{Grant}$

First Order Logic — SAT

Fact: A word $w \in \Sigma^*$ is the linearization of a **unique** trace $[w]$.

Theorem

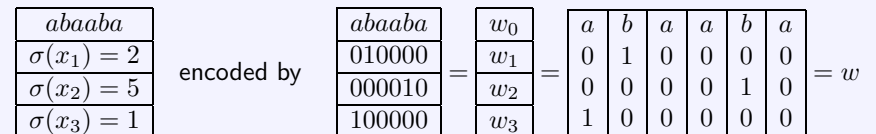
From any $FO(<)$ formula φ over traces, one can effectively build a finite word automaton \mathcal{A}_φ such that $\mathcal{L}(\mathcal{A}_\varphi) = \{w \in \Sigma^* \mid [w] \models \varphi\}$.

Proof sketch

Induction on φ . Problem: free variables. **Büchi: encode valuation σ in the word.**

For $\ell \in \mathbb{N}$, let $\Sigma_\ell = \Sigma \times \{0, 1\}^\ell$. A letter $a \in \Sigma_\ell$ is written $a = (a_0, a_1, \dots, a_\ell)$.

We have $\Sigma_\ell^* = \Sigma^* \times (\{0, 1\}^*)^\ell = \Sigma^* \times \mathcal{P}(\mathbb{N})^\ell$.



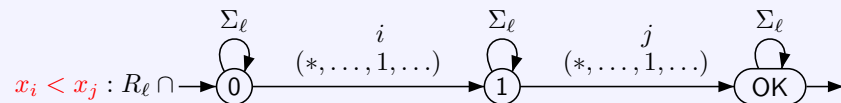
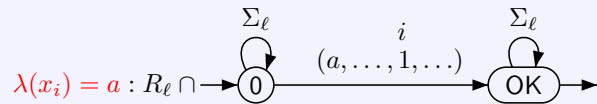
Each w_i is in $0^*10^* \rightsquigarrow w \in R_\ell$ regular language.

First Order Logic — SAT (2)

Proof sketch: translating formulae for words

From $\varphi \in \text{FO}_{\Sigma}(\leq)$ with free variables $\text{Var} = \{x_1, \dots, x_\ell\}$, build \mathcal{A}_φ over Σ_ℓ^* st.

$$\mathcal{L}(\mathcal{A}_\varphi) = \{w \in R_\ell \mid w_0, \sigma \models \varphi\}$$



$$\mathcal{A}_{\varphi \vee \psi} = \mathcal{A}_\varphi \cup \mathcal{A}_\psi$$

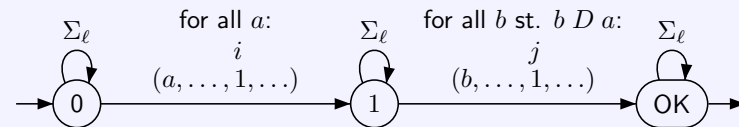
$$\mathcal{A}_{\neg \varphi} = \text{Complement}(\mathcal{A}_\varphi) \quad (\rightarrow \text{determinization})$$

$$\mathcal{A}_{\exists x_j \varphi} = \text{Project away } j+1\text{-th component from } \mathcal{A}_\varphi \quad (\rightsquigarrow \text{non-det. automaton})$$

First Order Logic — SAT (3)

Proof sketch: reduction from traces to words

- Let \leq_t be the trace ordering and \leq the ordering on linearizations.
- $x <_t y$: $\bigvee_{n=0}^{|\Sigma|} \exists x_1 \dots \exists x_n, x = x_1 < x_2 < \dots < x_n = y \wedge [\lambda(x_1) D \lambda(x_2) D \dots D \lambda(x_n)]$
- It remains to translate the formula (on words) $\lambda(x_i) D \lambda(x_j)$:



First Order Logic — Results

Decidability (Büchi 1960) - complexity (Stockmeyer 1974)

The satisfiability problem for $\text{FO}(<)$ is decidable but non elementary.

Expressiveness (McNaughton & Papert 1971, Schützenberger 1965)

The word languages expressible in $\text{FO}(<)$ are those

- described by a star-free (SF) expression: using union, complement, product.
- whose syntactic monoid is group-free (or aperiodic, AP) (\Rightarrow decidable).

Summary

	Words ($<$)	Traces ($<$)	MSC (E_m)/HMSCs
SAT / MC	Non elementary		Undecidable/Non elementary
Expressiveness	FO = AP = SF		
	Good but may be improved (MSO)		
Readability & Ease of expression	Bad if one has to comply the syntax OK if "English sentences" allowed		

Outline

- 1 Distributed behaviors and specifications
- 2 First order logic
- 3 **Monadic second order logic**
- 4 Linear temporal logic for sequences
- 5 Global temporal logics
- 6 Local temporal logics
- 7 MSO-definable local TL
- 8 Local μ -calculus

Monadic Second Order Logic

Syntax: $\text{MSO}(\preceq) \quad (\Sigma = 2^{\text{AP}})$

$\varphi ::= \perp \mid P_q(x) \mid \neg\varphi \mid \varphi \vee \psi \mid x \preceq y \mid \exists x\varphi \mid x \in X \mid \exists X\varphi$

Semantics

- A formula is evaluated on labeled DAGs $t = [V, E, \lambda]$.
- \prec interpreted as a relation between $\leq = E^+$ and $\leq = \prec \setminus \prec^2$.
- First order variables $x, y \in \text{Var}_1$. Second order variables $X, Y \in \text{Var}_2$.
- σ is a interpretation of (free) variables st. $\sigma(x) \in V$ and $\sigma(X) \in 2^V$.

$t, \sigma \models P_q(x)$ if $q \in \lambda(\sigma(x))$
 $t, \sigma \models x \preceq y$ if $\sigma(x) \preceq \sigma(y)$
 $t, \sigma \models \neg\varphi$ if $t, \sigma \not\models \varphi$
 $t, \sigma \models \varphi \vee \psi$ if $t, \sigma \models \varphi$ or $t, \sigma \models \psi$
 $t, \sigma \models \exists x\varphi$ if $t, \sigma \cup [x \mapsto v] \models \varphi$ for some $v \in V$
 $t, \sigma \models x \in X$ if $\sigma(x) \in \sigma(X)$
 $t, \sigma \models \exists X\varphi$ if $t, \sigma \cup [X \mapsto U] \models \varphi$ for some $U \subseteq V$

Monadic Second Order Logic — Examples

MSO strictly more expressive than FO

- $\leq = \prec^+$ is expressible in $\text{MSO}(\prec)$ (we use the finite past property).

$$x \leq y : \exists X [(x \in X) \wedge (y \in X) \wedge \forall z \in X (z = x) \vee \exists u (u \in X \wedge u \prec z)]$$

- Let $\varphi(x)$ be an $\text{MSO}(\prec)$ formula with one free variable x . One can express “there is a path whose vertices satisfy $\varphi(x)$ ” in $\text{MSO}(\prec)$.

$$\exists X \forall x \forall y [(x, y \in X) \implies \neg(x \parallel y) \wedge \varphi(x) \wedge \forall z [(x \prec z \prec y) \implies z \in X]]$$

- The language $(aa)^+$ can be expressed by the word formula: “always a ” \wedge

$$\varphi_{\text{Even}} = \exists X \exists Y [\forall z ((z \in X) \iff \neg(z \in Y)) \wedge \forall z \forall t (z \prec t) \implies (z \in X \iff t \in Y) \wedge \exists x \exists y (x \in X \wedge y \in Y \wedge \neg \exists z [(z \prec x) \vee (z \succ y)])]$$

Monadic Second Order Logic — Results

Decidability (Büchi 1960) & complexity

The satisfiability problem for $\text{MSO}(\prec)$ is decidable but non elementary.

Expressiveness (Words: Büchi 1960. Traces: Ebinger & Muscholl 1993)

The languages expressible in $\text{MSO}(\prec)$ are the recognizable languages.

Summary

	Words	Traces	MSC/HMSCs
SAT & MC	Non elementary		Undecidable/Non elementary
Expressiveness	recognizable		recognizable for finitely generated
			Very good
Readability & Ease of expression			Same as FO

Outline

- 1 Distributed behaviors and specifications
- 2 First order logic
- 3 Monadic second order logic
- 4 Linear temporal logic for sequences
- 5 Global temporal logics
- 6 Local temporal logics
- 7 MSO-definable local TL
- 8 Local μ -calculus

Linear Temporal Logic (Pnueli 1977)

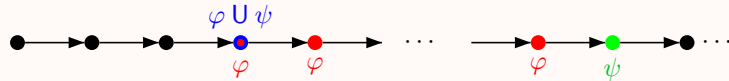
Syntax: LTL(AP, X, U)

$$\varphi ::= \perp \mid p \ (p \in AP) \mid \neg\varphi \mid \varphi \vee \psi \mid X\varphi \mid \varphi U \psi$$

Semantics: $t = [\mathbb{N}, \leq, \lambda]$ with $\lambda : \mathbb{N} \rightarrow \Sigma = 2^{AP}$ and $x \in \mathbb{N}$

- $t, x \models p$ if $p \in \lambda(x)$
- $t, x \models \neg\varphi$ if $t, x \not\models \varphi$
- $t, x \models \varphi \vee \psi$ if $t, x \models \varphi$ or $t, x \models \psi$
- $t, x \models X\varphi$ if $\exists y. x < y \ \& \ t, y \models \varphi$
- $t, x \models \varphi U \psi$ if $\exists z. x \leq z \ \& \ t, z \models \psi \ \& \ \forall y. (x \leq y < z) \Rightarrow t, y \models \varphi$

Example



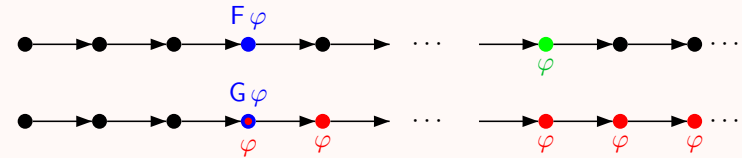
A formula $\varphi \in LTL(AP, X, U)$ defines the trace language $\mathcal{L}(\varphi) = \{t \mid t, 0 \models \varphi\}$.

Linear Temporal Logic (Pnueli 1977)

Macros:

- $\circ F\varphi = \top U \varphi$
- $\circ G\varphi = \neg F \neg\varphi$

Example



Linear Temporal Logic (Pnueli 1977)

Specifications:

- \circ Safety: G good
- \circ MutEx: $\neg F(\text{crit}_1 \wedge \text{crit}_2)$
- \circ Liveness: $G F$ active
- \circ Response: $G(\text{request} \Rightarrow F \text{ grant})$
- \circ Response': $G(\text{request} \Rightarrow X(\neg \text{request} U \text{ grant}))$
- \circ Release: $\text{reset } R \text{ alarm} = (\text{alarm } U (\text{alarm} \wedge \text{reset})) \vee (G \text{ alarm})$

Release:

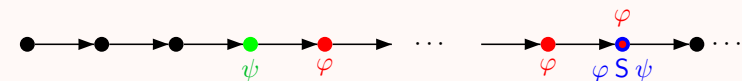
- $\circ \varphi R \psi = (\psi U (\varphi \wedge \psi)) \vee (G \psi) = \neg(\neg\varphi U \neg\psi)$

Past LTL

Semantics: $t = [\mathbb{N}, \leq, \lambda]$ with $\lambda : \mathbb{N} \rightarrow \Sigma = 2^{AP}$ and $x \in \mathbb{N}$

- $t, x \models Y\varphi$ if $\exists y. y < x \ \& \ t, y \models \varphi$
- $t, x \models \varphi S \psi$ if $\exists z. z \leq x \ \& \ t, z \models \psi \ \& \ \forall y. (z < y \leq x) \Rightarrow t, y \models \varphi$

Example



LTL versus PLTL

$G(\text{grant} \Rightarrow Y(\neg \text{grant} S \text{ request}))$

$$= (\text{request } R \neg \text{grant}) \wedge G(\text{grant} \Rightarrow (\text{request} \vee X(\text{request } R \neg \text{grant})))$$

Theorem (Laroussinie & Schnoebelen 2002)

PLTL may be exponentially more succinct than LTL.

Past LTL

Semantics: $t = [\mathbb{N}, \leq, \lambda]$ with $\lambda : \mathbb{N} \rightarrow \Sigma = 2^{AP}$ and $x \in \mathbb{N}$

$t, x \models Y \varphi$ if $\exists y. y < x \ \& \ t, y \models \varphi$

$t, x \models \varphi S \psi$ if $\exists z. z \leq x \ \& \ t, z \models \psi \ \& \ \forall y. (z < y \leq x) \Rightarrow t, y \models \varphi$

Example



LTL versus PLTL

$G(\text{grant} \Rightarrow Y(\neg \text{grant} S \text{request}))$

$= (\text{request} R \neg \text{grant}) \wedge G(\text{grant} \Rightarrow (\text{request} \vee X(\text{request} R \neg \text{grant})))$

Theorem (Laroussinie & Schnoebelen 2002)

PLTL may be exponentially more succinct than LTL.

Results

Expressivity

Theorem (Kamp 68(!))

$PLTL = FO$.

Theorem (Gabbay, Pnueli, Shelah & Stavi 80)

$PLTL = LTL = FO$.

Elegant algebraic proof of $LTL = FO$ due to Wilke 98.

Complexity

Theorem (Sistla & Clarke 85, Lichtenstein et. al 85)

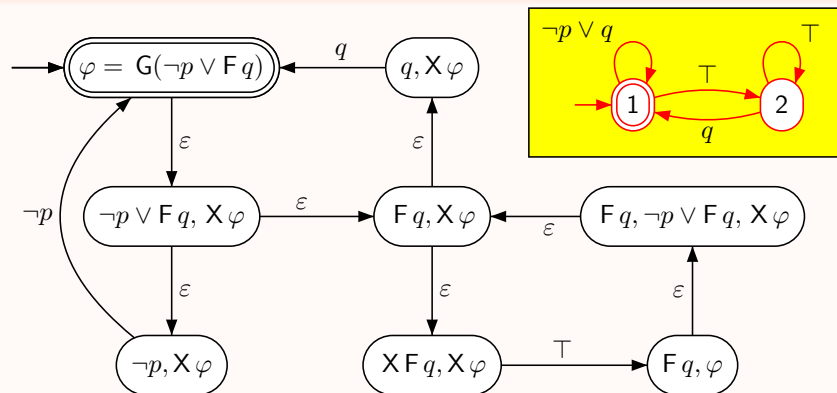
Satisfiability and Model Checking for LTL are PSPACE-complete.

Proof: Tableau construction.

From an LTL (or PLTL) formula, one can build a Büchi automaton \mathcal{A}_φ accepting precisely the models of φ . \square

Tableau construction

Example: $\varphi = G(p \Rightarrow Fq)$



State = set of obligations.

Saturation = reduce obligations to literals and next-formulas.

Transition = check literals and move forward.

Suitable accepting states.

Linear Temporal Logic — Results

Summary

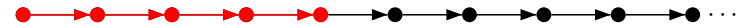
	Words
SAT & MC	PSPACE
Expressiveness	LTL = FO Good
Readability & Ease of expression	Good with the natural syntax

Outline

- 1 Distributed behaviors and specifications
- 2 First order logic
- 3 Monadic second order logic
- 4 Linear temporal logic for sequences
- 5 **Global temporal logics**
- 6 Local temporal logics
- 7 MSO-definable local TL
- 8 Local μ -calculus

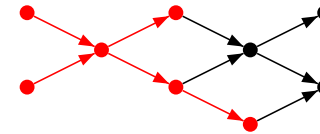
Global Temporal Logics

On words, an LTL formula is evaluated at **positions** of the word.



A position corresponds to a **partial computation**.

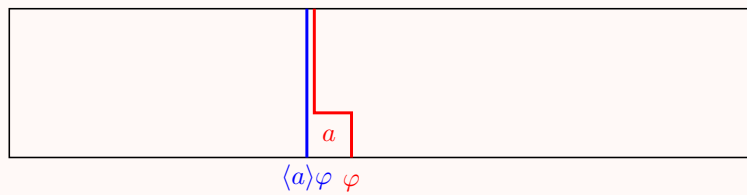
A partial computation of a distributed system is a **prefix** defined by a downward closed subset of events.



On partial orders, a **global** temporal logic formula is evaluated at **prefixes**.

Next Modality for Global TL

Example



Semantics of next and previous

$$\begin{aligned}
 t, r \models \langle a \rangle \varphi & \text{ if } ra \leq t \ \& \ t, ra \models \varphi \\
 t, r \models \langle a \rangle^{-1} \varphi & \text{ if } r = r'a \ \& \ t, r' \models \varphi
 \end{aligned}$$

Macros

$$a = \langle a \rangle \top \qquad a^{-1} = \langle a \rangle^{-1} \top$$

Existential next

Existential next

$$EX \varphi = \bigvee_{a \in \Sigma} \langle a \rangle \varphi$$

Using EX, formulas may be exponentially smaller.

Remark: $\langle a \rangle \varphi$ can be expressed using EX and a .

$$\begin{aligned}
 a^k &= \bigwedge_{0 \leq \ell < k} \neg EX^\ell \neg a \\
 \langle a \rangle \varphi &= \bigvee_{1 \leq k \leq n} (a^k \wedge EX(\neg a^k \wedge \varphi)) \vee (a^{n+1} \wedge \varphi)
 \end{aligned}$$

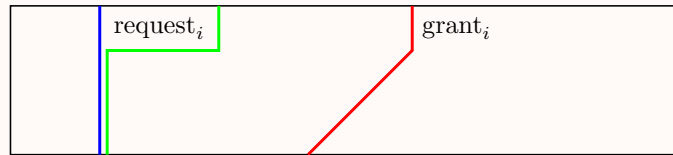
for some n depending on φ .

Specifications in Global TL

Specifications:

- Safety: $G \text{ good}$
- MutEx: $\neg F(\text{crit}_1 \wedge \text{crit}_2)$
- Liveness: $G F \text{ active}_i$
- Response: $G(\text{request}_i \Rightarrow F \text{ grant}_i)$
- Response': $G(\text{request}_i \Rightarrow \text{EX}(\neg \text{request}_i \cup \text{grant}_i))$

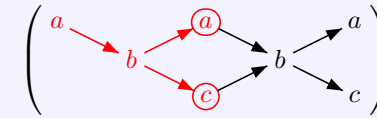
Response'



OK since request_i and grant_i cannot be concurrent.

$\text{globTL}(\langle a \rangle, U) \subseteq \text{FO}_{\Sigma}(\leq)$

A finite prefix $s \leq t$ is characterized by its **maximal events**



The number of maximal events is bounded by $|\Sigma|$.
Hence, we can define a prefix by a tuple \bar{x} of first order variables.

With each globTL-formula φ we associate a $\text{FO}_{\Sigma}(\leq)$ -formula $\tilde{\varphi}(\bar{x})$ such that
 $t, s \models \varphi$ iff $t, \sigma \models \tilde{\varphi}(\bar{x})$
if $\sigma(\bar{x})$ corresponds to s .

Example

$$\begin{aligned} \langle a \rangle \tilde{\varphi}(\bar{x}) &= \exists \bar{y}, ((\bar{y} = \bar{x} \cdot a) \wedge \varphi(\bar{y})) \\ \varphi \cup \psi(\bar{x}) &= \exists \bar{y}, (\bar{x} \leq \bar{y} \wedge \psi(\bar{y}) \wedge \forall \bar{z}, (\bar{x} \leq \bar{z} < \bar{y} \Rightarrow \varphi(\bar{z}))) \end{aligned}$$

Universal Until for Global TL

Expressive completeness

- $\text{FO}_{\Sigma}(\leq) = \text{globTL}(\langle a \rangle, U, \langle a \rangle^{-1}, S)$: **finite traces, future and past modalities**
Ebinger (PhD'94).
- $\text{FO}_{\Sigma}(\leq) = \text{globTL}(\langle a \rangle, U, a^{-1})$: **infinite traces, future modalities and past constants**
Thiagarajan, Walukiewicz (LICS'97).
- $\text{FO}_{\Sigma}(\leq) = \text{globTL}(\langle a \rangle, U)$: **infinite traces, future modalities only**
Diekert, Gastin (ICALP'00, JCSS'02).

Complexity

- **The satisfiability problem is non elementary** Walukiewicz (ICALP'98).
- **Modular decision procedure** Gastin, Meyer, Petit (MFCS'98).

Global Temporal Logic — Results

Summary

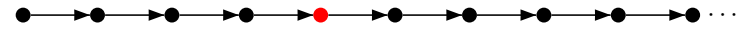
	Traces
SAT & MC	non elementary
Expressiveness	globTL = FO Good
Readability & Ease of expression	Good with the natural syntax

Outline

- 1 Distributed behaviors and specifications
- 2 First order logic
- 3 Monadic second order logic
- 4 Linear temporal logic for sequences
- 5 Global temporal logics
- 6 Local temporal logics
- 7 MSO-definable local TL
- 8 Local μ -calculus

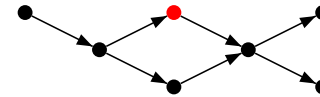
Local Temporal Logics

On words, an LTL formula is evaluated at **positions** of the word.



A position corresponds to an **action** in a sequential run.

An action of a distributed computation is an **event** in the trace.



On traces, a **local** temporal logic formula is evaluated at **events**.

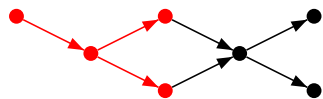
Local versus Global

We can identify an event with its lower set.

Hence all local configurations are global configurations.



But some global configurations are not local.



Local TL

Syntax: $\text{locTL}(\Sigma, \text{EX}, \text{U})$

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

Semantics: $t = [V, \leq, \lambda]$ with $\lambda : V \rightarrow \Sigma$ and $x \in V$

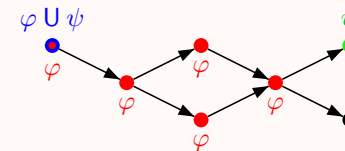
$$t, x \models a \quad \text{if} \quad \lambda(x) = a$$

$$t, x \models \text{EX}\varphi \quad \text{if} \quad \exists y. x < y \ \& \ t, y \models \varphi$$

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

Example

There may be several next events



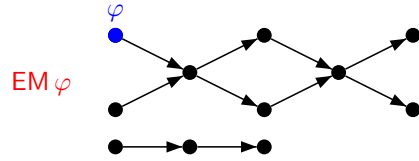
Macros:

$$F\varphi = \text{T}\text{U}\varphi$$

$$G\varphi = \neg F\neg\varphi$$

Initial Formulae

When does a trace satisfy a formula?



No canonical initial event. Several minimal events. Disjoint components.

First solution: **Initial formulae**

Boolean combinations of $EM \varphi$, where φ is a local formula

$$t \models EM \varphi \text{ if } \exists x. (x \in \min(t) \ \& \ t, x \models \varphi)$$

Initial Formulae

Local temporal logics based on EM and future modalities are not expressively complete (Walukiewicz).

Example with $(\Sigma, D) = a - b - c - d$

$$t_1 = \left(\begin{array}{cccc} a \rightarrow b \rightarrow c \rightarrow b & \cdots & & \\ & \uparrow & & \\ & d \rightarrow c & & \end{array} \right)$$

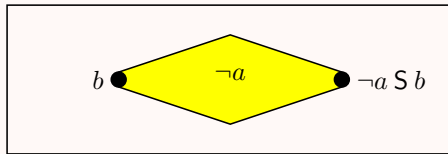
$$t_2 = \left(\begin{array}{cccc} a \rightarrow b & & & \\ & \downarrow & & \\ d \rightarrow c \rightarrow b \rightarrow c & \cdots & & \end{array} \right)$$

For all future formulae φ :

$$t_1 \models EM \varphi \text{ iff } t_2 \models EM \varphi$$

Past constants

$\neg a S b$: used to compare the latest a and the latest b



Theorem (Gastin, Kumar & Mukund MFCS'03)

$$\text{locTL}_{(\Sigma, D)}(\text{EM}, \text{EX}, \text{U}, \neg a S b) = \text{FO}_{(\Sigma, D)}^3(\leq).$$

Proof

- \subseteq Easy. $\text{locTL}_{(\Sigma, D)}(\text{EM}, \text{EX}, \text{U}, \neg a S b) \subseteq \text{FO}_{(\Sigma, D)}^3(\leq)$ by the very definition
- \supseteq Difficult.

Corollary

$$\text{FO}_{\Sigma}(\leq) = \text{FO}_{\Sigma}^3(\leq).$$

Local TL with past constants ...

Remark: $\neg d S d$ means there is a d in the past.

The two examples of Walukiewicz can be distinguished

$$(\Sigma, D) = a - b - c - d$$

$$t_1 = \left(\begin{array}{cccc} a \rightarrow b \rightarrow c \rightarrow b & \cdots & & \\ & \uparrow & & \\ & d \rightarrow c & & \end{array} \right) \models \text{EM}(a \wedge \text{EX}(\neg d S d))$$

$$t_2 = \left(\begin{array}{cccc} a \rightarrow b & & & \\ & \downarrow & & \\ d \rightarrow c \rightarrow b \rightarrow c & \cdots & & \end{array} \right) \not\models \text{EM}(a \wedge \text{EX}(\neg d S d))$$

Future only local TL

Theorem (Diekert & Gastin, LPAR'01)

► Skip proof

$\text{locTL}_{(\Sigma, D)}(\text{EM}, \text{EX}, \text{U}) = \text{FO}_{(\Sigma, D)}(\leq)$ if and only if (Σ, D) is a cograph.

(Σ, D) is a **cograph** iff

- it does not contain a P_4 : $a - b - c - d$.
- it is built from singletons using **disjoint unions and complementations**.
- it is built from singletons using **disjoint unions and direct products**.
- Traces are **series-parallel** posets.
- Systems obtained using **serial** and **parallel** compositions.

Expressivity for traces

Theorem (Diekert & Gastin, LPAR'01)

$\text{locTL}_{(\Sigma, D)}(\text{EM}, \text{EX}, \text{U}) = \text{FO}_{(\Sigma, D)}(\leq)$ if and only if (Σ, D) is a cograph.

Proof

only if See previous example.

if $\text{locTL}_{(\Sigma, D)}(\text{EM}, \text{EX}, \text{U}) \subseteq \text{FO}_{(\Sigma, D)}^3(\leq)$ by the very definition
Conversely, we use an induction on Σ .

- $\Sigma = \{a\}$: A first order language is either a finite set or the union of a finite set and a set of the form $a^n a^*$ for some $n \geq 0$.

$$\begin{aligned} a^3 a^* &= \mathcal{L}(\text{EM EX EX T}) \\ \{a^3\} &= a^3 a^* \setminus a^4 a^* \end{aligned}$$

- The case $\Sigma = A \uplus B$ with $A \times B \subseteq I$ is easy.
- The case $\Sigma = A \uplus B$ with $A \times B \subseteq D$ is more difficult.

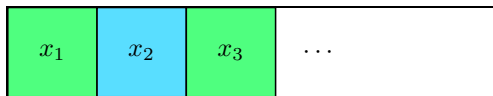
► Skip proof

Proof Sketch

Assume $\Sigma = A \uplus B$ with $A \times B \subseteq D$.

Let $\Delta = \mathbb{M} \cup ((\text{alphinf} \not\subseteq A) \cap (\text{alphinf} \not\subseteq B))$.

Each trace $x \in \Delta$ has a unique (finite or infinite) factorization $x = x_1 x_2 x_3 \dots$ in non-empty finite traces alternating \mathbb{M}_A^+ and \mathbb{M}_B^+ .



Let $L \in \text{FO}_{\Sigma}(\leq) = \text{AP}$ and let $h : \mathbb{M} \rightarrow S$ aperiodic recognizing L .

Let $T = h(\mathbb{M}^+)$ and let $e : T^* \rightarrow S$ be the evaluation morphism.

Let $\sigma : \Delta \rightarrow T^\infty$ defined by $\sigma(x) = h(x_1)h(x_2)\dots$.

Lemma

$L \cap \Delta = \sigma^{-1}(K)$ where $K = [\sigma(L \cap \Delta)]_{\approx_e}$ is aperiodic.

Using Kamp's theorem, $L \cap \Delta = \sigma^{-1}(\mathcal{L}(f))$ for some $f \in \text{LTL}_T(XU)$.

We can avoid to use Kamp's theorem using a generalization of a proof technique introduced by Wilke for words (STACS'99).

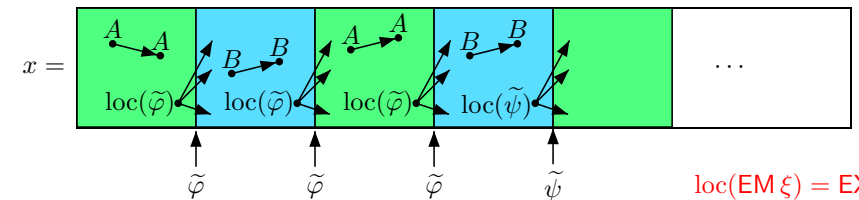
Proof Sketch: $\sigma^{-1}(\mathcal{L}(\varphi XU \psi))$

Lemma

$\forall \varphi \in \text{LTL}_T(XU)$, $\exists \tilde{\varphi} \in \text{locTL}_{\Sigma}(\text{EX}, \text{U})$ such that $\mathcal{L}(\tilde{\varphi}) \cap \Delta = \sigma^{-1}(\mathcal{L}(\varphi))$.

Let $x \in \Delta$ such that $\sigma(x) \models \varphi XU \psi$.

$$\sigma(x) = \begin{array}{ccccccccc} h(x_1) & h(x_2) & h(x_3) & h(x_4) & h(x_5) & \dots \\ \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \\ \varphi & \varphi & \varphi & \psi & & \end{array}$$



$\text{loc}(\text{EM } \xi) = \text{EX } \xi$

$\widetilde{\varphi XU \psi} =$

$$\text{EM} \left(((A \wedge \text{EX } A) \vee (B \wedge \text{EX } B) \vee \text{loc}(\tilde{\varphi})) \text{U} (((A \wedge \text{EX } B) \vee (B \wedge \text{EX } A)) \wedge \text{loc}(\tilde{\psi})) \right)$$

Proof Sketch: $\sigma^{-1}(\mathcal{L}(s)), s \in T$

$$\sigma(x) \models s \text{ iff } x \in \sigma^{-1}(s \cdot T^\infty)$$

$$\sigma^{-1}(s \cdot T^\infty) = \Delta \cap \left((h^{-1}(s) \cap \mathbb{M}_A^+) (\min \subseteq B) \cup (h^{-1}(s) \cap \mathbb{M}_B^+) (\min \subseteq A) \right)$$

By induction on Σ , $h^{-1}(s) \cap \mathbb{M}_A^+$ is expressible in $\text{locTL}_A(\text{EX}, \text{U})$.

Lemma

$\forall \alpha \in \text{locTL}_A, \exists \tilde{\alpha} \in \text{locTL}_\Sigma$ such that $\mathcal{L}(\tilde{\alpha}) = (\mathcal{L}(\alpha) \cap \mathbb{M}_A^+) (\min \subseteq B)$.



$$x_1 \models \alpha \text{ iff } x \models \tilde{\alpha}$$

$$\widetilde{\text{EX}} \varphi = \text{EX}(A \wedge \tilde{\varphi}) \quad \& \quad \widetilde{\varphi \text{ U } \psi} = (A \wedge \tilde{\varphi}) \text{ U } (A \wedge \tilde{\psi}).$$

Proof Sketch: Last Step

We have seen that L aperiodic implies $L \cap \Delta$ expressible in $\text{locTL}_\Sigma(\text{EX}, \text{U})$.

$$\mathbb{R} = \Delta \cup (\text{alphinf} \subseteq A) \cup (\text{alphinf} \subseteq B)$$

$$L \cap (\text{alphinf} \subseteq A) = \bigcup_{\text{finite}} (L_1 \cap (\max \subseteq B)) \cdot (L_2 \cap (\text{alph} \subseteq A))$$

$L_1 \cap (\max \subseteq B) = (L_1 \cap \Delta) \cap (\max \subseteq B)$ is expressible in $\text{locTL}_\Sigma(\text{EX}, \text{U})$.

Lemma

$\forall \alpha \in \text{locTL}_\Sigma(\text{EX}, \text{U}), \exists \tilde{\alpha} \in \text{locTL}_\Sigma(\text{EX}, \text{U})$ such that

$$\mathcal{L}(\tilde{\alpha}) = (\mathcal{L}(\alpha) \cap (\max \subseteq B)) (\text{alph} \subseteq A)$$

By induction on Σ , $L_2 \cap (\text{alph} \subseteq A)$ is expressible in $\text{locTL}_A(\text{EX}, \text{U})$.

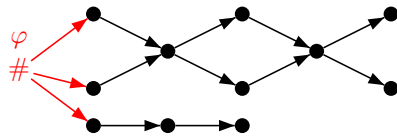
Lemma

$\forall \alpha \in \text{locTL}_A(\text{EX}, \text{U}), \exists \tilde{\alpha} \in \text{locTL}_\Sigma(\text{EX}, \text{U})$ such that

$$\mathcal{L}(\tilde{\alpha}) = (\max \subseteq B) (\mathcal{L}(\alpha) \cap (\text{alph} \subseteq A))$$

Initial Formulae

When does a trace satisfy a formula?



No canonical initial event. Several minimal events. Disjoint components.

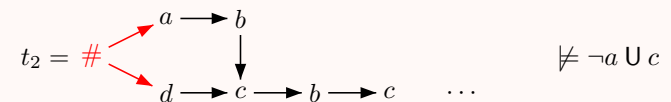
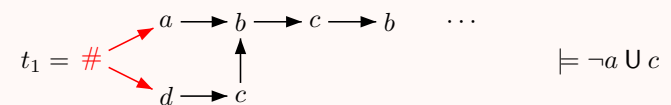
Second solution: **rooted partial orders**

Let φ be a local formula

$$t \models \varphi \text{ if } \#t, \# \models \varphi$$

Rooted partial orders

Example with $(\Sigma, D) = a - b - c - d$



Future only local TL on rooted traces

Theorem (Diekert & Gastin, LATIN'04)

→ Skip proof

- $\text{locTL}_{\#}(a, \text{EX}, \text{U}) = \text{FO}_{\Sigma}(\leq)$.
- $\text{locTL}_{\#}(a, (\text{X}_a \leq \text{X}_b), \text{X}_C, \text{U}_C) = \text{FO}_{\Sigma}(\leq)$,
where $C \in \mathcal{C}$ covering of Σ by dependence cliques.
- $\text{locTL}_{\#}(a, (\text{X}_a \leq \text{X}_b), \text{X}_a, \text{U}_a) = \text{FO}_{\Sigma}(\leq)$, $\mathcal{C} = \{\{a\} \mid a \in \Sigma\}$.

Semantics: $t = [V, \leq, \lambda]$ with $\lambda : V \rightarrow \Sigma$ and $x \in V$

For $C \subseteq \Sigma$ **dependence clique**, we let $x_C = \min\{y \mid x < y \ \& \ \lambda(y) \in C\}$ if it exists.

- $t, x \models \text{X}_C \varphi$ if x_C exists and $t, x_C \models \varphi$
- $t, x \models (\text{X}_a \leq \text{X}_b)$ if both x_a and x_b exist, and $x_a \leq x_b$

$\varphi \text{U}_C \psi = (C \Rightarrow \varphi) \text{U} (C \wedge \psi)$

- $t, x \models \varphi \text{U}_C \psi$ if $\exists z. x \leq z \ \& \ t, z \models \psi \ \& \ \lambda(z) \in C \ \& \ \forall y. (x \leq y < z \ \& \ \lambda(y) \in C) \Rightarrow t, y \models \varphi$

Proof sketch

$\text{locTL}_{\#}(a, (\text{X}_a \leq \text{X}_b), \text{X}_C, \text{U}_C) \subseteq \text{FO}_{\Sigma}^3(\leq)$ by the very definition.

- $t, x \models \text{X}_C \varphi$ if $\exists z. x < z \ \& \ t, z \models \varphi \ \& \ \lambda(z) \in C \ \& \ \forall y. x < y < z \Rightarrow \lambda(y) \notin C$
- $t, x \models \varphi \text{U}_C \psi$ if $\exists z. x \leq z \ \& \ t, z \models \psi \ \& \ \lambda(z) \in C \ \& \ \forall y. (x \leq y < z \ \& \ \lambda(y) \in C) \Rightarrow t, y \models \varphi$

Conversely, we use an induction on Σ .

$\Sigma = \{a\}$: A first order language is either a finite set or the union of a finite set and a set of the form $a^n a^*$ for some $n \geq 0$.

$$\begin{aligned} a^3 a^* &= \mathcal{L}(\text{EX EX EX T}) \\ \{a^3\} &= a^3 a^* \setminus a^4 a^* \end{aligned}$$

$|\Sigma| > 1$: This is the difficult case.

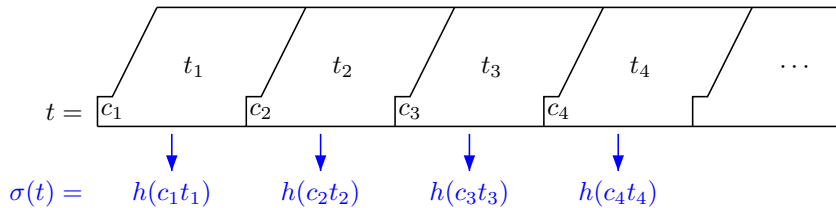
→ Skip proof

Factorization

Each trace $t \in (\min \in C)$ has a unique (finite or infinite) factorization

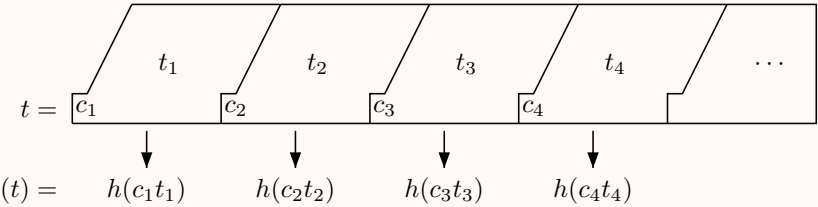
$$t = (c_1 t_1)(c_2 t_2)(c_3 t_3)(c_4 t_4) \cdots$$

with $c_i \in C$, $\text{alph}(t_i) \cap C = \emptyset$ and $\min(c_i t_i) = c_i$.



Let $h : \mathbb{M} \rightarrow S$ be a morphism to a finite aperiodic monoid S and let $T = h(\mathbb{M})$.

Define $\sigma : (\min \in C) \rightarrow T^\infty$.



Theorem (Ebinger, Muscholl) A trace language is **first-order** iff it is **aperiodic**.

Lemma Let $L \subseteq \mathbb{R}$ be recognized by h . Let $K = \sigma(L \cap (\min \in C))$. Then,

- $K \subseteq T^\infty$ is an aperiodic word language. $K = \mathcal{L}(f)$ for some $f \in \text{LTL}$.
- $\forall t \in (\min \in C), \quad t \in L \text{ iff } \sigma(t) \in K \text{ iff } \sigma(t) \models f$

Kamp's Theorem For words, $\text{FO}_{\Sigma}(\leq) = \text{AP} = \text{LTL}$.

Reduction Lemma $\forall L \subseteq \mathbb{R}$ expressible in $\text{FO}_{\Sigma}(\leq)$, $\exists f \in \text{LTL}$ such that

- $\forall t \in (\min \in C), \quad t \in L \text{ iff } \sigma(t) \models f$

Lifting Lemma Let $f \in \text{LTL}$, there exists $\tilde{f} \in \text{locTL}$ such that

- $\forall t \in (\min \in C), \quad \sigma(t) \models f \text{ iff } t \models \tilde{f}$

→ Skip proof

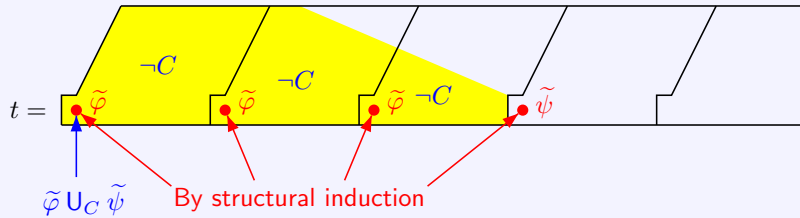
Lifting Lemma Let $f \in \text{LTL}$, there exists $\tilde{f} \in \text{locTL}$ such that

- $\forall t \in (\min \in C), \quad \sigma(t) \models f \text{ iff } t \models \tilde{f}$

$$\widetilde{\varphi \text{ U } \psi} = \tilde{\varphi} \text{ U}_C \tilde{\psi} = (\neg C \vee \tilde{\varphi}) \text{ U } (C \wedge \tilde{\psi})$$

Assume $\sigma(t) \models \varphi \text{ U } \psi$

$$\sigma(t) = \begin{array}{cccc} h(c_1 t_1) & h(c_2 t_2) & h(c_3 t_3) & h(c_4 t_4) & \dots \\ \varphi & \varphi & \varphi & \psi & \dots \end{array}$$



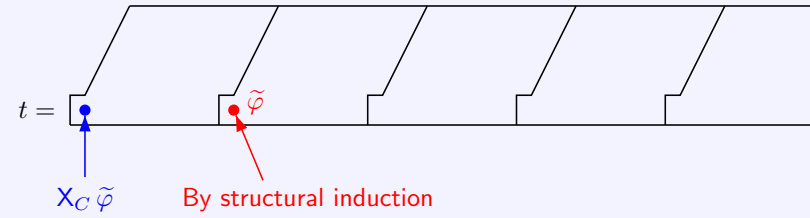
Lifting Lemma Let $f \in \text{LTL}$, there exists $\tilde{f} \in \text{locTL}$ such that

- $\forall t \in (\min \in C), \quad \sigma(t) \models f \text{ iff } t \models \tilde{f}$

$$\widetilde{X\varphi} = X_C \tilde{\varphi}$$

Assume $\sigma(t) \models X\varphi$

$$\sigma(t) = \begin{array}{cccc} h(c_1 t_1) & h(c_2 t_2) & h(c_3 t_3) & h(c_4 t_4) & \dots \\ \varphi & \varphi & \varphi & \varphi & \dots \end{array}$$



Lifting Lemma Let $f \in \text{LTL}$, there exists $\tilde{f} \in \text{locTL}$ such that

- $\forall t \in (\min \in C), \quad \sigma(t) \models f \text{ iff } t \models \tilde{f}$

$$\tilde{u} \text{ for } u \in S$$

Let $t = (c_1 t_1)(c_2 t_2)(c_3 t_3) \dots$ as usual and let $A = \Sigma \setminus C$. Then,

$$\begin{aligned} \sigma(t) \models u & \text{ iff } h(c_1 t_1) = u \\ & \text{ iff } h(t_1) = v \text{ for some } v \in S \text{ with } h(c_1)v = u \\ & \text{ iff } \#t_1 \models \varphi \text{ for some } \varphi \in \text{locTL}_A \\ & \text{ iff } c_1 t_1 \models \varphi \text{ for some } \varphi \in \text{locTL}_A \\ & \text{ iff } t \models \tilde{\varphi}^A \text{ where } \tilde{\varphi}^A \in \text{locTL}_\Sigma \end{aligned}$$

$t_1 \in h^{-1}(v) \cap \mathbb{R}_A$ which is an aperiodic trace language over $A \subsetneq \Sigma$.

By induction on $|\Sigma|$, we find $\varphi \in \text{locTL}_A$ such that

$$\mathcal{L}(\varphi) = \bigcup_{v|h(c_1)v=u} h^{-1}(v) \cap \mathbb{R}_A$$

Lifting Lemma Let $f \in \text{LTL}$, there exists $\tilde{f} \in \text{locTL}$ such that

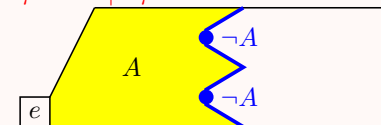
- $\forall t \in (\min \in C), \quad \sigma(t) \models f \text{ iff } t \models \tilde{f}$

$$\tilde{u} \text{ for } u \in S$$

Let $t = (c_1 t_1)(c_2 t_2)(c_3 t_3) \dots$ as usual and let $A = \Sigma \setminus C$. Then,

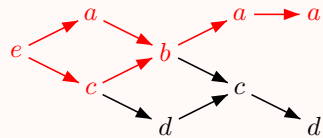
$$\begin{aligned} \sigma(t) \models u & \text{ iff } h(c_1 t_1) = u \\ & \text{ iff } h(t_1) = v \text{ for some } v \in S \text{ with } h(c_1)v = u \\ & \text{ iff } \#t_1 \models \varphi \text{ for some } \varphi \in \text{locTL}_A \\ & \text{ iff } c_1 t_1 \models \varphi \text{ for some } \varphi \in \text{locTL}_A \\ & \text{ iff } t \models \tilde{\varphi}^A \text{ where } \tilde{\varphi}^A \in \text{locTL}_\Sigma \end{aligned}$$

Second lifting lemma For all $\varphi \in \text{locTL}$ and $A \subseteq \Sigma$, there exists $\tilde{\varphi}^A \in \text{locTL}$ such that for all $es \in \mathbb{R}$ with $\min(es) = e$, if s_A is the largest prefix of s containing letters from A only, then $es_A \models \varphi$ iff $es \models \tilde{\varphi}^A$.



Lemma $\forall \varphi, \forall A, \exists \overline{\varphi}^A$ such that $es_A \models \varphi$ iff $es \models \overline{\varphi}^A$
 where s_A is the largest prefix of s using actions from A only.

$$\overline{X_b \top}^A = X_b \top \wedge \bigwedge_{c \notin A} \neg(X_c \leq X_b)$$

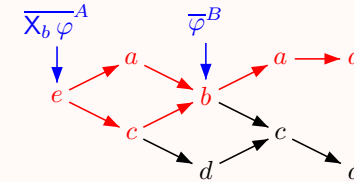


- If $A = \{a\}$ then $es_A \not\models X_b \top$
- If $A = \{a, b, c\}$ then $es_A \models X_b \top$

Lemma $\forall \varphi, \forall A, \exists \overline{\varphi}^A$ such that $es_A \models \varphi$ iff $es \models \overline{\varphi}^A$
 where s_A is the largest prefix of s using actions from A only.

First guess $\overline{X_b \varphi}^A = \overline{X_b \top}^A \wedge X_b \overline{\varphi}^A$ **Solution** $\overline{X_b \varphi}^A = \overline{X_b \top}^A \wedge$

$$\bigvee_B (\text{Switch}(A, B) \wedge X_b \overline{\varphi}^B)$$



- $c \notin B$ since $es \models (X_{dc} \leq X_{bc})$
- $a \in B$ since $es \not\models (X_{da} \leq X_{ba})$

The formula $\text{Switch}(A, B)$ uses constants $(X_{ac} \leq X_{bc})$.

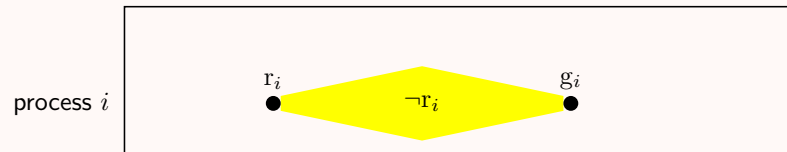
Lemma $(X_{ac} \leq X_{bc})$ can be expressed in $\text{locTL}(X_C, U_C, (X_a \leq X_b))$.

Specifications in local TL

Specifications: Local properties are easy to express

- Local safety: $G(\text{process}_i \Rightarrow \text{good}_i)$
- Liveness: $GF \text{active}_i$
- Response: $G(\text{request}_i \Rightarrow F \text{grant}_i)$
- Response': $G(\text{request}_i \Rightarrow X_i(\neg \text{request}_i \ U_i \text{grant}_i))$
- Response'': $G(\text{request}_i \Rightarrow (\neg \text{request}_i \ SU \text{grant}_i))$

Response'



OK since request_i and grant_i cannot be concurrent.

Specifications in local TL

Global properties are hard to express in $\text{locTL}_{\#}(a, EX, U)$.

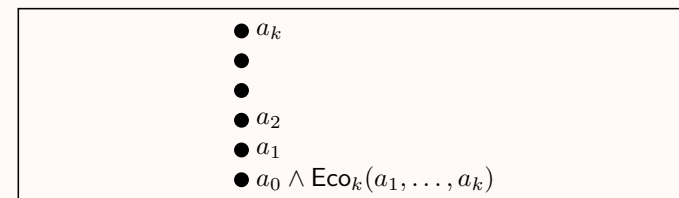
Solution: Introduce new modalities

- without changing the expressive power
- without changing the complexity

Specifications: global properties with new modalities

- MutEx: $\neg F(\text{crit}_1 \wedge \text{Eco crit}_2)$
- Safety: $\neg F \bigvee_{a \in \text{Bad}} a_0 \wedge \text{Eco}_k(a_1, \dots, a_k)$

Safety



Complexity of local TL

- TrPTL: process based Logic Thiagarajan (LICS'94).
Satisfiability is decidable in EXPTIME. Gossip automata
- TLC: Existential until Alur, Peled, Penczek (LICS'95).
Satisfiability is PSPACE-complete. Tableau construction
- locTL(EX, U): Universal until, pure future Diekert, Gastin (LPAR'01).
Satisfiability is PSPACE-complete. Alternating automata
- locTL(EX, EY, U, S): past modalities Gastin, Mukund (ICALP'02).
Satisfiability is PSPACE-complete. 2-way alternating automata

Local Temporal Logic — Results

Summary

	Traces
SAT & MC	PSPACE
Expressiveness	locTL = FO Good
Readability & Ease of expression	May be hard with the basic modalities

Outline

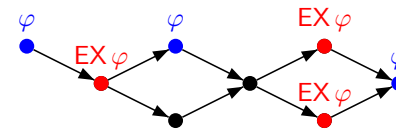
- 1 Distributed behaviors and specifications
- 2 First order logic
- 3 Monadic second order logic
- 4 Linear temporal logic for sequences
- 5 Global temporal logics
- 6 Local temporal logics
- 7 MSO-definable local TL
- 8 Local μ -calculus

MSO modalities

Fix some partial order $t = [V, \leq, \lambda]$ and some local formula φ .

For $v \in V$, the semantics defines when $t, v \models \varphi$.

Then, we let $\varphi^t = \{v \in V \mid t, v \models \varphi\}$ or equivalently $t, v \models \varphi$ iff $v \in \varphi^t$.



Now, $(EX \varphi)^t = \{u \in V \mid \exists v, u < v \ \& \ v \in \varphi^t\}$.

Define $\llbracket EX \rrbracket(X_1, x) = \exists y, x < y \ \& \ y \in X_1$.

Then, $u \in (EX \varphi)^t = \{u \in V \mid t \models_{MSO} \llbracket EX \rrbracket(\varphi^t, u)\}$.

EX is a unary modality with semantics

$$\llbracket EX \rrbracket(X_1, x) = \exists y, x < y \ \& \ y \in X_1$$

which is an MSO formula with one set variable and one free position variable.

MSO-definable temporal logic

Definition

An **MSO-modality** M is defined by its arity ℓ and its semantics

$$\llbracket M \rrbracket (X_1, \dots, X_\ell, x) \in \text{MSO}_\Sigma(\leq)$$

with ℓ free set variables and one free position variable.

An MSO-definable local logic $\text{TL}(M_1, \dots, M_k)$ is defined by a finite number of MSO-modalities M_i with their arity ℓ_i and semantics $\llbracket M_i \rrbracket$.

Syntax: $\varphi ::= M_1(\varphi_1, \dots, \varphi_{\ell_1}) \mid \dots \mid M_k(\varphi_1, \dots, \varphi_{\ell_k})$

Semantics: $M(\varphi_1, \dots, \varphi_\ell)^t = \{v \in t \mid t \models_{\text{MSO}} \llbracket M \rrbracket (\varphi_1^t, \dots, \varphi_\ell^t, v)\}$

Complexity of local TL

Gastin & Kuske, CONCUR'03

Theorem

The satisfiability problem and the model checking problem for **each MSO-definable logics over words** is decidable in PSPACE.

Theorem

The satisfiability problem and the model checking problem for **each MSO-definable local logics over traces** is decidable in PSPACE.

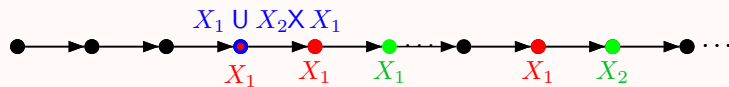
Corollary

All previously known complexity results.

Before the proof, we give some examples of MSO-definable TL.

Example 1: LTL over words

$$\text{LTL} = \text{TL}(X, U, a, \neg, \vee)$$



$\llbracket X \rrbracket (X_1, x)$	$= \exists y, (x < y) \wedge (y \in X_1)$
$\llbracket U \rrbracket (X_1, X_2, x)$	$= \exists y, (x \leq y) \wedge (y \in X_2) \wedge (\forall z, x \leq z < y \Rightarrow z \in X_1)$
$\llbracket a \rrbracket (x)$	$= \lambda(x) = a$
$\llbracket \neg \rrbracket (X_1, x)$	$= \neg(x \in X_1)$
$\llbracket \vee \rrbracket (X_1, X_2, x)$	$= (x \in X_1) \vee (x \in X_2)$

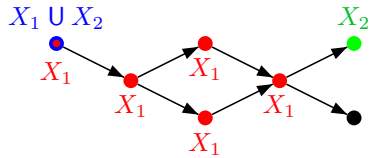
and more ...

Even(φ): even number of positions satisfying φ and above the current one.

$$\llbracket \text{Even} \rrbracket (X_1, x) = \exists Y, (|Y| \text{ is even}) \wedge \forall y, (y \in Y \Leftrightarrow (y \in X_1 \wedge x \leq y))$$

Example 2: locTL over traces

$$\text{locTL} = \text{TL}(a, \neg, \vee, \text{EX}, \text{U})$$



Semantics

$$\llbracket \text{EX} \rrbracket (X_1, x) = \exists y, (x \prec y) \wedge (y \in X_1)$$

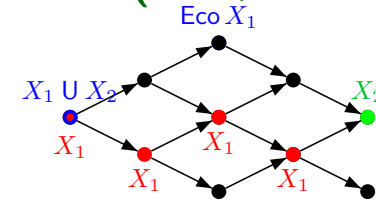
$$\llbracket \text{U} \rrbracket (X_1, X_2, x) = \exists y, (x \leq y) \wedge (y \in X_2) \wedge (\forall z, x \leq z < y \Rightarrow z \in X_1)$$

We can also define the initial modality

$$\llbracket \text{EM} \rrbracket (X_1, x) = \exists y, (y \in X_1) \wedge (\forall z, z \leq y \Rightarrow z = y)$$

or past modalities (EY and S), ...

Example 3: TLC (Alur, Peled, Penczek 95)



$$\llbracket \text{Eco} \rrbracket (X_1, x) = \exists z (\neg(x \leq z) \wedge \neg(z \leq x) \wedge (z \in X_1))$$

$$\llbracket \text{EU} \rrbracket (X_1, X_2, x) = \exists z ((x \leq z) \wedge (z \in X_2) \wedge \exists Y (Y \setminus \{z\} \subseteq X_1 \wedge Y \text{ is a maximal totally ordered set contained in } \uparrow x \cap \downarrow z))$$

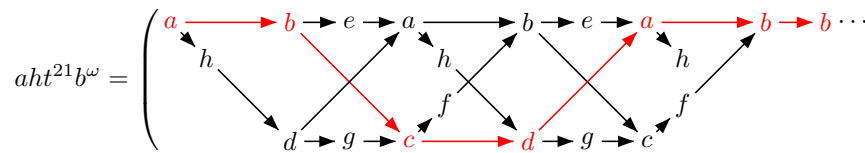
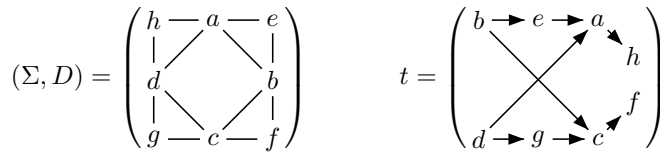
$$\text{TLC} = \text{TL}(a, \neg, \vee, \text{EX}, \text{EY}, \text{Eco}, \text{EU}, \text{ES}, \text{EG})$$

Theorem (Diekert & Gastin, LPAR'01)

► Skip proof

- The existential until is not FO-definable in general.
- We don't know whether $\text{FO}_{\Sigma}(\leq) \subseteq \text{TLC}$ in general.
- $\text{TLC} = \text{FO}_{\Sigma}(\leq)$ if and only if (Σ, D) is a cograph.

EU is not First Order



Let $\varphi = (a \vee b \vee c \vee d) \text{EU} G b$

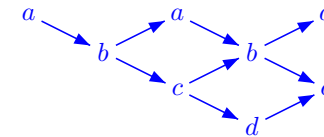
$$aht^2b^\omega \models \varphi \quad \text{but} \quad aht^1b^\omega \not\models \varphi$$

$$aht^*b^\omega \cap \mathcal{L}(\text{EM } \varphi) = ah(t^2)^*b^\omega \notin \text{FO}_{\Sigma}(\leq)$$

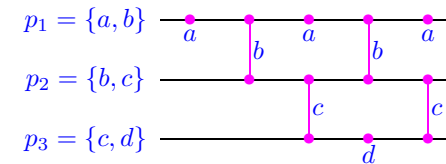
Example 4: Process based TL

$$(\Sigma, D) = a - b - c - d$$

The trace



can be redrawn as



TrPTL: a process based logic (Thiagarajan)

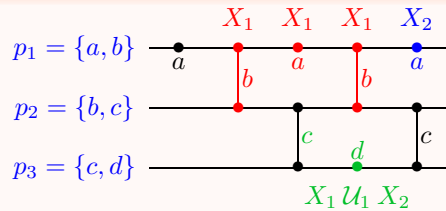
$$\text{TrPTL} = \text{locTL}(a, \neg, \vee, \mathcal{O}_i, \mathcal{U}_i)$$

Semantics

$$\llbracket \mathcal{O}_i \rrbracket (X_1, x) = \exists y \in P_i, \neg(y \leq x) \wedge y \in X_1 \wedge \forall z \in P_i, \neg(z \leq x) \Rightarrow y \leq z$$

$$\llbracket \mathcal{U}_i \rrbracket (X_1, X_2, x) = \exists x' \in P_i, (x' \leq x) \wedge (\forall y \in P_i, y \leq x \Rightarrow y \leq x') \wedge \exists y \in P_i, (x' \leq y) \wedge y \in X_2 \wedge \forall z \in P_i, (x' \leq z < y) \Rightarrow z \in X_1$$

Example



From the maximal i -event in $\downarrow x$, the sequence along process i satisfies $\varphi \cup \psi$.

Process based logics

Theorem (Diekert & Gastin, LATIN'04)

$$\text{locTL}(a, (X_a \leq X_b), \neg, \vee, X_i, U_i) = \text{FO}_{\Sigma}(\leq)$$

Here, **contrary to TrPTL**, the **next** and **until** modalities are **pure future**.

Expressivity is open for $\text{TrPTL} = \text{locTL}(a, \neg, \vee, \mathcal{O}_i, \mathcal{U}_i)$.

Theorem (Gastin & Kuske, CONCUR'03)

The satisfiability problem for **each MSO-definable** logics over **words** is decidable in **PSPACE**.

Proof sketch

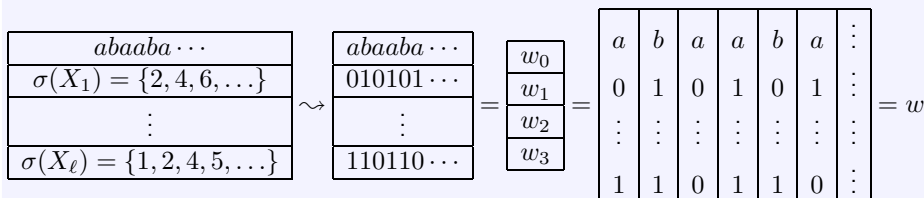
Skip proof

$w = a_0 a_1 \dots \in \{0, 1\}^\omega$ is identified with $\text{supp}(w) = \{p \mid a_p = 1\} \subseteq \mathbb{N}$.

For $\ell \in \mathbb{N}$, let $\Sigma_\ell = \Sigma \times \{0, 1\}^\ell$.

A letter $a \in \Sigma_\ell$ is written $a = (a_0, a_1, \dots, a_\ell)$.

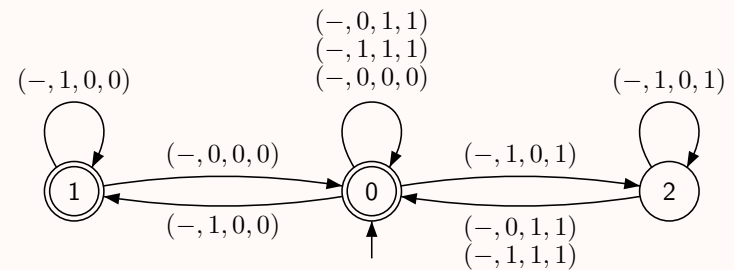
We have $\Sigma_\ell^\omega = \Sigma^\omega \times (\{0, 1\}^\omega)^\ell = \Sigma^\omega \times \mathcal{P}(\mathbb{N})^\ell$.



Theorem (Büchi)

Let M be an ℓ -ary modality. There exists a Büchi-automaton \mathcal{A}_M over $\Sigma_{\ell+1}$ such that $(w, w_1, \dots, w_{\ell+1}) \in \mathcal{L}(\mathcal{A}_M)$ iff $\text{supp}(w_{\ell+1}) = \{p \mid w \models \llbracket M \rrbracket(\text{supp}(w_1), \dots, \text{supp}(w_\ell), p)\}$.

Example $\mathcal{A}_U (X_3 = X_1 \cup X_2)$



Proof. Apply Büchi's theorem to the MSO formula

$$\llbracket \overline{M} \rrbracket (X_1, \dots, X_{\ell+1}) = \forall x (x \in X_{\ell+1} \Leftrightarrow \llbracket M \rrbracket (X_1, \dots, X_\ell, x)).$$

Decision procedure

Proposition

Let $\varphi \in \text{TL}(M_1, \dots, M_k)$ and $\varphi_1, \dots, \varphi_n$ be the subformulas of φ . We can construct an automaton \mathcal{A} over Σ_n such that

$$\mathcal{L}(\mathcal{A}) = \{(w, w_1, \dots, w_n) \in \Sigma_n^\omega \mid \text{supp}(w_j) = \varphi_j^w \text{ for all } 1 \leq j \leq n\}$$

Proof

Let $1 \leq j \leq n$. If $\varphi_j = M(\varphi_{k_1}, \dots, \varphi_{k_\ell})$, let \mathcal{A}_j be a copy of the automaton \mathcal{A}_M reading input lines $(0, k_1, \dots, k_\ell, j)$.

Assume by induction that $\text{supp}(w_{k_j}) = \varphi_{k_j}^w$ for $1 \leq j \leq \ell$. Then, TFAE

- (w, w_1, \dots, w_n) is accepted by \mathcal{A}_j ,
- $\text{supp}(w_j) = \{p \mid w \models \llbracket M \rrbracket(\text{supp}(w_{k_1}), \dots, \text{supp}(w_{k_\ell}), p)\}$,
- $\text{supp}(w_j) = \{p \mid w, p \models M(\varphi_{k_1}, \dots, \varphi_{k_\ell})\}$,
- $\text{supp}(w_j) = \varphi_j^w$.

Thus, the required automaton \mathcal{A} is obtained as the intersection of the automata \mathcal{A}_j for $1 \leq j \leq n$.

Decision procedure

Corollary

- $w, p \models \varphi_i$ if there exists $\bar{w} \in \mathcal{L}(\mathcal{A})$ such that $\bar{w}_i(p) = 1$.
- To check whether φ is satisfiable, we essentially need to check \mathcal{A} for emptiness.

The automaton \mathcal{A} needs not be constructed.

Instead, we guess an accepting path of \mathcal{A} , storing only the current global state $q = (q_1, \dots, q_n)$.

In order to compute the next state, we guess a letter in Σ_n and we compute the next state $q' = (q'_1, \dots, q'_n)$ according to the transition rules of the automata $\mathcal{A}_{M_1}, \dots, \mathcal{A}_{M_k}$.

Since the automata $\mathcal{A}_{M_1}, \dots, \mathcal{A}_{M_k}$ do not depend on the input formula $\varphi \in \text{TL}(M_1, \dots, M_k)$, checking whether φ is satisfiable can be done in PSPACE.

Reduction of traces to words

Let $t = [V, \leq, \lambda] \in \mathbb{R}(\Sigma, D)$ and $w = [V, \preceq, \lambda]$ be any linearization of t .

We want a formula $\eta(x, y)$ such that for all $x, y \in V$,

$$w \models \eta(x, y) \text{ iff } x \leq y \text{ in } t$$

The FO-formula $\eta(x, y)$ is defined by

$$\exists x_0 \dots \exists x_k, (x = x_0 \preceq x_1 \dots \preceq x_k = y) \wedge \bigwedge_{1 \leq i \leq k} (\lambda(x_{i-1}), \lambda(x_i)) \in D$$

Reduction of traces to words

Let $\text{TL}(M_1, \dots, M_k)$ be a local logic defined by the modality names M_i with arity ℓ_i and semantics $\llbracket M_i \rrbracket$.

Consider the local logic $\text{TL}'(M_1, \dots, M_k)$ defined by the modality names M_i with arity ℓ_i and semantics $\llbracket M_i \rrbracket'$ obtained by replacing each occurrence of $x \leq y$ in the MSO formula $\llbracket M_i \rrbracket$ by $\eta(x, y)$.

Note that TL and TL' have the same formulas.

Let $t = [V, \leq, \lambda] \in \mathbb{R}(\Sigma, D)$ and $w = [V, \preceq, \lambda]$ be any linearization of t .

We have for all $x \in V$, $t, x \models_{\text{TL}} \varphi$ iff $w, x \models_{\text{TL}'} \varphi$.

Therefore, φ is TL-satisfiable over traces iff φ is TL' satisfiable over words.

Hence, we can apply the result on words.

MSO-definable local TL

Summary

- Generic definition of local logics for traces.
- Allows to define local TL with very good readability and ease of expression (just introduce modalities for the desired properties).
- Generic decision procedure in PSPACE.
- The automata constructed are usually as good as the automata obtained with direct constructions.
- Applies also to the model checking problem.

	Traces
SAT & MC	PSPACE
Expressiveness	between FO and MSO Very good
Readability & Ease of expression	Good

Outline

- 1 Distributed behaviors and specifications
- 2 First order logic
- 3 Monadic second order logic
- 4 Linear temporal logic for sequences
- 5 Global temporal logics
- 6 Local temporal logics
- 7 MSO-definable local TL
- 8 Local μ -calculus

μ -calculus

Example: $\alpha = \mu Y.(\psi \vee (\varphi \wedge EX Y))$

Semantics: Let $t = [V, \rightarrow, \lambda]$. Consider

$$f: 2^V \rightarrow 2^V$$

$$S \mapsto \llbracket \psi \rrbracket^t \cup (\llbracket \varphi \rrbracket^t \cap \text{pred}(S))$$

where $\llbracket \psi \rrbracket^t = \{x \in V \mid t, x \models \psi\}$ and $\text{pred}(S) = \{x \in V \mid \exists y \in S, x \rightarrow y\}$.

Then f is monotone and admits a least fixed point

$$\llbracket \alpha \rrbracket^t = \bigcap \{S \in 2^V \mid f(S) \subseteq S\} = \bigcup_{n \geq 0} f^n(\emptyset)$$

We have

- $f^1(\emptyset) = \llbracket \psi \rrbracket^t = \{x \in V \mid t, x \models \psi\}$
- $f^n(\emptyset) = \{x \in V \mid \exists x = x_0 \rightarrow x_1 \rightarrow \dots \rightarrow x_k$ with $k \leq n, t, x_k \models \psi$ & $t, x_i \models \varphi$ for all $0 \leq i < k\}$

Therefore, $\alpha = \varphi \text{ EU } \psi$.

μ -calculus

Syntax: $\mu L(a, EX)$

$$\varphi ::= a \mid \neg \varphi \mid \varphi \vee \varphi \mid EX \varphi \mid Y \mid \mu Y. \varphi$$

where the construction $\mu Y. \varphi$ is only valid when Y occurs positively in φ .

Semantics: Let $t = [V, \rightarrow, \lambda]$ and Var be the set of variables

The semantics of φ is a map $\llbracket \varphi \rrbracket^t: (2^V)^{\text{Var}} \rightarrow 2^V$

$$\sigma \mapsto \llbracket \varphi \rrbracket^t(\sigma)$$

$$\begin{aligned} \llbracket a \rrbracket^t &= \lambda^{-1}(a) \\ \llbracket \neg \varphi \rrbracket^t(\sigma) &= V \setminus \llbracket \varphi \rrbracket^t(\sigma) \\ \llbracket \varphi \vee \psi \rrbracket^t(\sigma) &= \llbracket \varphi \rrbracket^t(\sigma) \cup \llbracket \psi \rrbracket^t(\sigma) \\ \llbracket EX \varphi \rrbracket^t(\sigma) &= \text{pred}(\llbracket \varphi \rrbracket^t(\sigma)) \\ \llbracket Y \rrbracket^t(\sigma) &= \sigma(Y) \\ \llbracket \mu Y. \varphi \rrbracket^t(\sigma) &= \text{least fixed point of } S \mapsto \llbracket \varphi \rrbracket^t(\sigma[Y \mapsto S]) \end{aligned}$$

If φ is closed then $\llbracket \varphi \rrbracket^t$ is a constant and we write $t, x \models \varphi$ for $x \in \llbracket \varphi \rrbracket^t$.

μ -calculus

Macros

- $\varphi \text{ EU } \psi = \mu Y.(\psi \vee (\varphi \wedge \text{EX} Y))$
- $\text{F } \varphi = \top \text{ EU } \varphi$
- $\text{G } \varphi = \neg \text{F } \neg \varphi$

Specifications: some local properties are easy to express

- Local safety: $\text{G}(\text{process}_i \Rightarrow \text{good}_i)$
- Liveness: G F active_i
- Response: $\text{G}(\text{request}_i \Rightarrow \text{F grant}_i)$

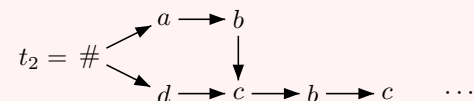
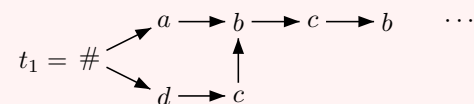
What about global properties such as MutEx?

What about Response' which uses an universal until?

Expressivity of μ -calculus

$$\text{FO}_{\Sigma}(\leq) \not\subseteq \mu\text{L}(a, \text{EX}) \not\subseteq \text{FO}_{\Sigma}(\leq)$$

- EU is expressible in $\mu\text{L}(a, \text{EX})$ but not in $\text{FO}_{\Sigma}(\leq)$.
- The traces t_1 and t_2 are bisimilar and thus undistinguishable in $\mu\text{L}(\text{pure future})$.

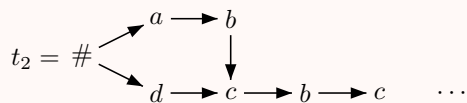
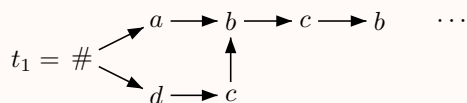


μ -calculus

Theorem (Walukiewicz 02)

- $\mu\text{L}(a, (X_a \leq X_b), \text{EX}) = \text{MSO}_{\Sigma}(\leq)$
- $\mu\text{L}(a, \text{Eco } a, \text{EX}) = \text{MSO}_{\Sigma}(\leq)$

$$t_1 \not\models (X_b \leq X_c) \qquad t_2 \models (X_b \leq X_c)$$



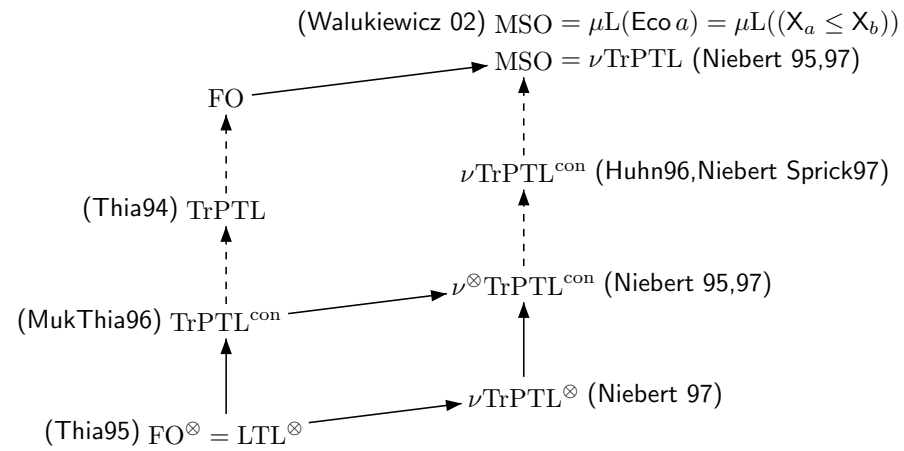
μ -calculus

Specifications: some local properties are easy to express

- MutEx: $\neg \text{F}(c_1 \wedge \text{Eco } c_2)$
- MutEx: $\neg \text{F}(\text{EX F } c_1 \wedge \text{EX F } c_2 \wedge \neg(X_{c_1} \leq X_{c_2}) \wedge \neg(X_{c_2} \leq X_{c_1}))$
- Response': $\text{G}(r_i \Rightarrow (\text{EX F } g_i \wedge \neg(X_{r_i} \leq X_{g_i})))$
OK since r_i and g_i are dependent and must be ordered.

But I don't know how to express the universal until $\varphi \text{ U } \psi$.

μ -calculus and trace based logics



μ -calculus

Summary

	Traces
SAT & MC	PSPACE
Expressiveness	MSO Very good
Readability & Ease of expression	Hard (at least for me) OK with suitable macros