Channels

Example: Leader election

We have n processes on a directed ring, each having a unique $\mathrm{id} \in \{1, \dots, n\}$.

```
send(id)
loop forever
   receive(x)
   if (x = id) then STOP fi
   if (x > id) then send(x)
```


Channels

Definition: Channels

- Declaration:
  - c : channel [k] of bool — size k
  - c : channel [∞] of int — unbounded
  - c : channel [0] of colors — Rendez-vous
- Primitives:
  - empty(c)
  - $c!e$ — add the value of expression e to channel c
  - $c?x$ — read a value from c and assign it to variable x
- Domain: Let D_m be the domain for a single message.
  - $D_c = D_m^k$ — size k
  - $D_c = D_m^*$ — unbounded
  - $D_c = \{\varepsilon\}$ — Rendez-vous
- Politics: FIFO, LIFO, BAG, ...

Channels

Semantics: (lossy) FIFO

Send
$$\frac{s_i \xrightarrow{c!e} s'_i \wedge \nu'(c) = \nu(e) \cdot \nu(c)}{(\bar{s}, \nu) \xrightarrow{c!e} (\bar{s}', \nu')}$$

Receive
$$\frac{s_i \xrightarrow{c?x} s'_i \wedge \nu(c) = \nu'(c) \cdot \nu'(x)}{(\bar{s}, \nu) \xrightarrow{c?e} (\bar{s}', \nu')}$$

Lossy send
$$\frac{s_i \xrightarrow{c!e} s'_i}{(\bar{s}, \nu) \xrightarrow{c!e} (\bar{s}', \nu)}$$

Implicit assumption: all variables that do not occur in the premise are not modified.

Exercises:

1. Implement a FIFO channel using rendez-vous with an intermediary process.
2. Give the semantics of a LIFO channel.
3. Model the alternating bit protocol (ABP) using a lossy FIFO channel.
   Fairness assumption: For each channel, if infinitely many messages are sent, then infinitely many messages are delivered.

High-level descriptions

Summary

- Sequential program = transition system with variables
- Concurrent program with shared variables
- Concurrent program with Rendez-vous
- Concurrent program with FIFO communication
- Petri net
- ...

Models: expressivity versus decidability

Definition: (Un)decidability

- Automata with 2 integer variables = Turing powerful
  Restriction to variables taking values in finite sets
- Asynchronous communication: unbounded fifo channels = Turing powerful
  Restriction to bounded channels

Definition: Some infinite state models are decidable

- Petri nets. Several unbounded integer variables but no zero-test.
- Pushdown automata. Model for recursive procedure calls.
- Timed automata.
- ...

Outline

Static and dynamic properties

Definition: Static properties

Example: Mutual exclusion

Safety properties are often static.

They can be reduced to reachability.

Definition: Dynamic properties

Example: Every request should be eventually granted.

$$\bigwedge_i \forall t, (\mathrm{Call}_i(t) \longrightarrow \exists t' \geq t, (\mathrm{atLevel}_i(t') \wedge \mathrm{openDoor}_i(t')))$$

The elevator should not cross a level for which a call is pending without stopping.

$$\bigwedge_i \forall t \forall t', (\mathrm{Call}_i(t) \wedge t \leq t' \wedge \mathrm{atLevel}_i(t')) \longrightarrow \exists t \leq t'' \leq t', (\mathrm{atLevel}_i(t'') \wedge \mathrm{openDoor}_i(t'')))$$

First Order specifications

First order logic

- These specifications can be written in $\mathrm{FO}(<)$.
- $\mathrm{FO}(<)$ has a good expressive power.
  ... but $\mathrm{FO}(<)$-formulae are not easy to write and to understand.
- $\mathrm{FO}(<)$ is decidable.
  ... but satisfiability and model checking are non elementary.

Definition: Temporal logics

- no variables: time is implicit.
- quantifications and variables are replaced by modalities.
- Usual specifications are easy to write and read.
- Good complexity for satisfiability and model checking problems.

Linear versus Branching

Let $M = (S, T, I, \mathrm{AP}, \ell)$ be a Kripke structure.

Definition: Linear specifications

Example: The printer manager is fair.
On each run, whenever some process requests the printer, it eventually gets it.

Execution sequences (runs): $\sigma = s_0 \to s_1 \to s_2 \to \cdots$ with $s_i \to s_{i+1} \in T$

Two Kripke structures having the same execution sequences satisfy the same linear specifications.

Actually, linear specifications only depend on the label of the execution sequence

$$\ell(\sigma) = \ell(s_0) \to \ell(s_1) \to \ell(s_2) \to \cdots$$

Models are words in Σ^ω with $\Sigma = 2^{\mathrm{AP}}$.

Definition: Branching specifications

Example: Each process has the possibility to print first.

Such properties depend on the execution tree.

Execution tree = unfolding of the transition system

References

Bibliography

[6] S. Demri and P. Gastin.
Specification and Verification using Temporal Logics.
In Modern applications of automata theory, IISc Research Monographs 2. World Scientific, To appear.
http://www.lsv.ens-cachan.fr/~gastin/mes-publis.php

A large list of references is given in this paper.

Bibliography

[7] V. Diekert and P. Gastin.
First-order definable languages.
In *Logic and Automata: History and Perspectives*, vol. 2, *Texts in Logic and Games*, pp. 261–306. Amsterdam University Press, (2008).
http://www.lsv.ens-cachan.fr/~gastin/mes-publis.php

A large overview of formalisms expressively equivalent to First-Order.

Some original References

[8] J. Kamp.
Tense Logic and the Theory of Linear Order.
PhD thesis, UCLA, USA, (1968).

[10] P. Gastin and D. Oddoux.
Fast LTL to Büchi automata translation.
In *CAV'01*, vol. 2102, *Lecture Notes in Computer Science*, pp. 53–65. Springer, (2001).
http://www.lsv.ens-cachan.fr/~gastin/mes-publis.php

[9] P. Wolper.
The tableau method for temporal logic: An overview,
Logique et Analyse. **110–111**, 119–136, (1985).

[11] A. Sistla and E. Clarke.
The complexity of propositional linear temporal logic.
Journal of the Association for Computing Machinery. **32** (3), 733–749, (1985).

Some original References

[12] O. Lichtenstein and A. Pnueli.
Checking that finite state concurrent programs satisfy their linear specification.
In *ACM Symposium PoPL'85*, 97–107.

[13] D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi.
On the temporal analysis of fairness.
In *7th Annual ACM Symposium PoPL'80*, 163–173. ACM Press.

[14] D. Gabbay.
The declarative past and imperative future: Executable temporal logics for interactive systems.
In *Temporal Logics in Specifications, April 87.* LNCS 398, 409–448, 1989.

Outline

Introduction

Models

Specifications

4 Linear Time Specifications
- Definitions
- Main results
- Büchi automata
- From LTL to BA
- Hardness results

Branching Time Specifications

Linear Temporal Logic (Pnueli 1977)

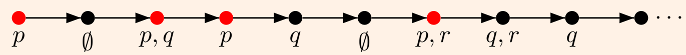
Definition: Syntax: $\mathrm{LTL}(\mathrm{AP}, \mathsf{X}, \mathsf{U})$

$$\varphi ::= \bot \mid p \ (p \in \mathrm{AP}) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi \,\mathsf{U}\, \varphi$$

Definition: Semantics: $w = a_0 a_1 a_2 \cdots \in \Sigma^\omega$ with $\Sigma = 2^{\mathrm{AP}}$ and $i \in \mathbb{N}$

$w, i \models p$	if	$p \in a_i$
$w, i \models \neg\varphi$	if	$w, i \not\models \varphi$
$w, i \models \varphi \vee \psi$	if	$w, i \models \varphi$ or $w, i \models \psi$
$w, i \models \mathsf{X}\,\varphi$	if	$w, i+1 \models \varphi$
$w, i \models \varphi \,\mathsf{U}\, \psi$	if	$\exists k.\ i \leq k$ and $w, k \models \psi$ and $\forall j.\ (i \leq j < k) \to w, j \models \varphi$

Example:



Linear Temporal Logic (Pnueli 1977)

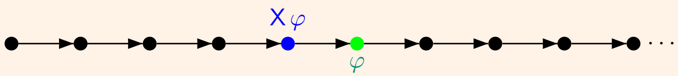
Definition: Syntax: $\mathrm{LTL}(\mathrm{AP}, \mathsf{X}, \mathsf{U})$

$$\varphi ::= \bot \mid p \ (p \in \mathrm{AP}) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi \,\mathsf{U}\, \varphi$$

Definition: Semantics: $w = a_0 a_1 a_2 \cdots \in \Sigma^\omega$ with $\Sigma = 2^{\mathrm{AP}}$ and $i \in \mathbb{N}$

$w, i \models p$	if	$p \in a_i$
$w, i \models \neg\varphi$	if	$w, i \not\models \varphi$
$w, i \models \varphi \vee \psi$	if	$w, i \models \varphi$ or $w, i \models \psi$
$w, i \models \mathsf{X}\,\varphi$	if	$w, i+1 \models \varphi$
$w, i \models \varphi \,\mathsf{U}\, \psi$	if	$\exists k.\ i \leq k$ and $w, k \models \psi$ and $\forall j.\ (i \leq j < k) \to w, j \models \varphi$

Example:



Linear Temporal Logic (Pnueli 1977)

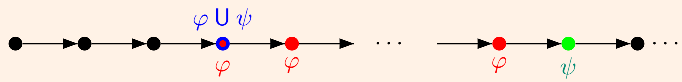
Definition: Syntax: $\mathrm{LTL}(\mathrm{AP}, \mathsf{X}, \mathsf{U})$

$$\varphi ::= \bot \mid p \ (p \in \mathrm{AP}) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi \,\mathsf{U}\, \varphi$$

Definition: Semantics: $w = a_0 a_1 a_2 \cdots \in \Sigma^\omega$ with $\Sigma = 2^{\mathrm{AP}}$ and $i \in \mathbb{N}$

$w, i \models p$	if	$p \in a_i$
$w, i \models \neg\varphi$	if	$w, i \not\models \varphi$
$w, i \models \varphi \vee \psi$	if	$w, i \models \varphi$ or $w, i \models \psi$
$w, i \models \mathsf{X}\,\varphi$	if	$w, i+1 \models \varphi$
$w, i \models \varphi \,\mathsf{U}\, \psi$	if	$\exists k.\ i \leq k$ and $w, k \models \psi$ and $\forall j.\ (i \leq j < k) \to w, j \models \varphi$

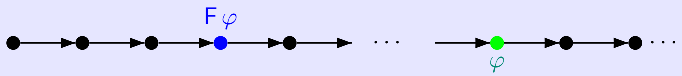
Example:



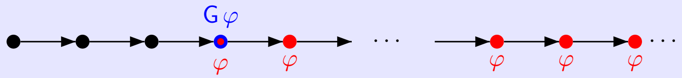
Linear Temporal Logic (Pnueli 1977)

Definition: Macros

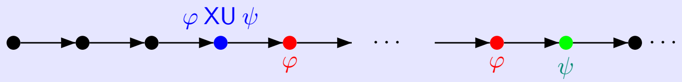
- Eventually: $\mathsf{F}\,\varphi = \top \,\mathsf{U}\, \varphi$



- Always: $\mathsf{G}\,\varphi = \neg\,\mathsf{F}\,\neg\varphi$



- Weak until: $\varphi\,\mathsf{W}\,\psi = \mathsf{G}\,\varphi \vee \varphi\,\mathsf{U}\,\psi$
- $\neg(\varphi\,\mathsf{U}\,\psi) = (\mathsf{G}\,\neg\psi) \vee (\neg\psi\,\mathsf{U}\,(\neg\varphi \wedge \neg\psi)) = \neg\psi\,\mathsf{W}\,(\neg\varphi \wedge \neg\psi)$
- Release: $\varphi\,\mathsf{R}\,\psi = \psi\,\mathsf{W}\,(\varphi \wedge \psi) = \neg(\neg\varphi\,\mathsf{U}\,\neg\psi)$
- Next until: $\varphi\,\mathsf{XU}\,\psi = \mathsf{X}(\varphi\,\mathsf{U}\,\psi)$



- $\mathsf{X}\,\psi = \bot\,\mathsf{XU}\,\psi$ and $\varphi\,\mathsf{U}\,\psi = \psi \vee (\varphi \wedge \varphi\,\mathsf{XU}\,\psi)$.

Linear Temporal Logic (Pnueli 1977)

Definition: Specifications:

- Safety: $\mathsf{G}\,\mathrm{good}$
- MutEx: $\neg\,\mathsf{F}(\mathrm{crit}_1 \wedge \mathrm{crit}_2)$
- Liveness: $\mathsf{G}\,\mathsf{F}\,\mathrm{active}$
- Response: $\mathsf{G}(\mathrm{request} \to \mathsf{F}\,\mathrm{grant})$
- Response': $\mathsf{G}(\mathrm{request} \to \mathsf{X}(\neg\mathrm{request}\,\mathsf{U}\,\mathrm{grant}))$
- Release: $\mathrm{reset}\,\mathsf{R}\,\mathrm{alarm}$
- Strong fairness: $\mathsf{G}\,\mathsf{F}\,\mathrm{request} \to \mathsf{G}\,\mathsf{F}\,\mathrm{grant}$
- Weak fairness: $\mathsf{F}\,\mathsf{G}\,\mathrm{request} \to \mathsf{G}\,\mathsf{F}\,\mathrm{grant}$

Linear Temporal Logic (Pnueli 1977)

Examples:

Every elevator request should be eventually satisfied.

$$\bigwedge_i \mathsf{G}(\mathrm{Call}_i \to \mathsf{F}(\mathrm{atLevel}_i \wedge \mathrm{openDoor}_i))$$

The elevator should not cross a level for which a call is pending without stopping.

$$\bigwedge_i \mathsf{G}(\mathrm{Call}_i \to \neg\mathrm{atLevel}_i\,\mathsf{W}\,(\mathrm{atLevel}_i \wedge \mathrm{openDoor}_i))$$

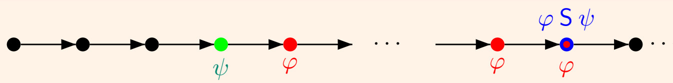
Past LTL

Definition: Semantics: $w = a_0 a_1 a_2 \cdots \in \Sigma^\omega$ with $\Sigma = 2^{\mathrm{AP}}$ and $i \in \mathbb{N}$

$w, i \models \mathsf{Y}\,\varphi$ if $i > 0$ and $w, i-1 \models \varphi$

$w, i \models \varphi\,\mathsf{S}\,\psi$ if $\exists k.\ k \leq i$ and $w, k \models \psi$ and $\forall j.\ (k < j \leq i) \to w, y \models \varphi$

Example:



Example: LTL versus PLTL

$\mathsf{G}(\mathrm{grant} \to \mathsf{Y}(\neg\mathrm{grant}\,\mathsf{S}\,\mathrm{request}))$

Theorem (Laroussinie & Markey & Schnoebelen 2002)

PLTL may be exponentially more succinct than LTL.

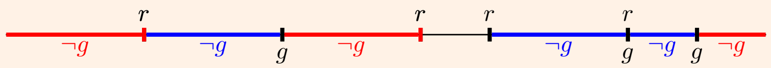
Past LTL

Definition: Semantics: $w = a_0 a_1 a_2 \cdots \in \Sigma^\omega$ with $\Sigma = 2^{\mathrm{AP}}$ and $i \in \mathbb{N}$

$w, i \models \mathsf{Y}\, \varphi$ if $i > 0$ and $w, i-1 \models \varphi$

$w, i \models \varphi\, \mathsf{S}\, \psi$ if $\exists k.\ k \leq i$ and $w, k \models \psi$ and $\forall j.\ (k < j \leq i) \to w, y \models \varphi$

Example:



Example: LTL versus PLTL

$\mathsf{G}(\text{grant} \to \mathsf{Y}(\neg\text{grant}\, \mathsf{S}\, \text{request}))$

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

Theorem (Laroussinie & Markey & Schnoebelen 2002)

PLTL may be exponentially more succinct than LTL.

Expressivity

Theorem [8, Kamp 68]

$$\mathrm{LTL}(\mathsf{Y}, \mathsf{S}, \mathsf{X}, \mathsf{U}) = \mathrm{FO}_\Sigma(\leq)$$

Separation Theorem [13, Gabbay, Pnueli, Shelah & Stavi 80]

For all $\varphi \in \mathrm{LTL}(\mathsf{Y}, \mathsf{S}, \mathsf{X}, \mathsf{U})$ there exist $\overleftarrow{\varphi_i} \in \mathrm{LTL}(\mathsf{Y}, \mathsf{S})$ and $\overrightarrow{\varphi_i} \in \mathrm{LTL}(\mathsf{X}, \mathsf{U})$ such that for all $w \in \Sigma^\omega$ and $k \geq 0$,

$$w, k \models \varphi \iff w, k \models \bigvee_i \overleftarrow{\varphi_i} \wedge \overrightarrow{\varphi_i}$$

Corollary: $\mathrm{LTL}(\mathsf{Y}, \mathsf{S}, \mathsf{X}, \mathsf{U}) = \mathrm{LTL}(\mathsf{X}, \mathsf{U})$

For all $\varphi \in \mathrm{LTL}(\mathsf{Y}, \mathsf{S}, \mathsf{X}, \mathsf{U})$ there exist $\overrightarrow{\varphi} \in \mathrm{LTL}(\mathsf{X}, \mathsf{U})$ such that for all $w \in \Sigma^\omega$,

$$w, 0 \models \varphi \iff w, 0 \models \overrightarrow{\varphi}$$

Elegant algebraic proof of $\mathrm{LTL}(\mathsf{X}, \mathsf{U}) = \mathrm{FO}_\Sigma(\leq)$ due to Wilke 98.