



Specification Formalisms: Temporal Logic and Automata on Infinite Words

Literature: Peled ch. 5

Mads Dam

2005 Mads Dam IMIT, KTH

1

2G1516 Formal Methods

Temporal Logic

Logic of transition system executions

Propositional/first-order logic = state assertions

Temporal assertions = assertions on system executions

- Invariably (along this execution) $x \leq y + z$
- Sometime (along this execution) an acknowledgement packet is sent
- If T is infinitely often enabled (along this execution) then T is eventually executed
- Last packet received along channel a (along this execution) had the shape (b,c,d)
- No matter which execution is followed from now (this state), a reply will eventually (along that execution) be sent
- No matter what choice B made in the past, it would necessarily come to pass that ψ

2005 Mads Dam IMIT, KTH

2

2G1516 Formal Methods

Runs/Executions/Paths

Fix transition system $T = (Q, R, Q_0)$

Computation path (aka run, execution sequence):

Infinite sequence

$$\xi = q_0 q_1 q_2 \dots q_i \dots$$

such that for all $i \geq 0$, $q_i R q_{i+1}$

Notation:

- $\xi(k) = q_k$ (= k'th state of ξ)
- $\xi^k = q_k q_{k+1} \dots$ (= k'th suffix of ξ , i.e. the path)

2005 Mads Dam IMIT, KTH

3

2G1516 Formal Methods

LTL – Linear Time Temporal Logic

Logic of future time path properties

η : Primitive state assertions

Syntax:

$$\phi ::= \eta \mid \neg\phi \mid \phi \wedge \phi \mid \langle \rangle\phi \mid \Box\phi \mid \phi \cup \phi \mid O\phi$$

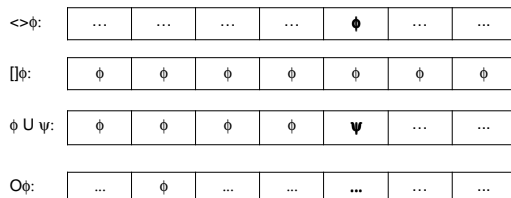
- η : η holds now/at the current time instant
- $\langle \rangle\phi$: At some future time instant ϕ is true
- $\Box\phi$: For all future time instants ϕ is true
- $\phi \cup \psi$: ϕ is true until ψ becomes true
- $O\phi$: ϕ is true at the next time instant

2005 Mads Dam IMIT, KTH

4

2G1516 Formal Methods

Pictorially



2005 Mads Dam IMIT, KTH

5

2G1516 Formal Methods

Semantics

Satisfaction relation $\xi \models \phi$

Assume interpretation function $\rho: \eta \mapsto Q' \subseteq Q$

$\rho(\eta)$: Set of states for which η holds

$\xi \models \eta$ iff $\xi(0) \in \rho(\eta)$

$\xi \models \neg\phi$ iff not $\xi \models \phi$

$\xi \models \phi \wedge \psi$ iff $\xi \models \phi$ and $\xi \models \psi$

$\xi \models \langle \rangle\phi$ iff exists $k \in \mathbb{N}$. $\xi^k \models \phi$

$\xi \models \Box\phi$ iff for all $k \in \mathbb{N}$. $\xi^k \models \phi$

$\xi \models \phi \cup \psi$ iff exists $k \in \mathbb{N}$. $\xi^k \models \psi$ and for all $i: 0 \leq i < k$. $\xi^i \models \phi$

$\xi \models O\phi$ iff $\xi^1 \models \phi$

For transition system $T = (Q, R, Q_0)$:

$T \models \phi$ iff for all runs ξ of T with $\xi(0) \in Q_0$, $\xi \models \phi$

2005 Mads Dam IMIT, KTH

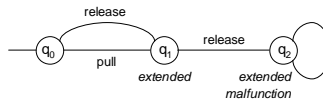
6

2G1516 Formal Methods

Some LTL Formulas

- $\phi \vee \psi = \neg(\neg\phi \wedge \neg\psi)$
- $\phi \rightarrow \psi = \neg\phi \vee \psi$ (\rightarrow is seriously overloaded!)
- $\langle \rangle \phi = \text{true} \cup \phi$
- $\Box \phi = \neg \langle \rangle \neg \phi$
- $\phi \vee \psi = \Box \psi \vee (\psi \cup (\phi \wedge \psi))$
 - (aka "release" in Peled)
- $\langle \rangle \Box \phi$
 - ϕ holds from some point forever
- $\Box \langle \rangle \phi$
 - ϕ holds infinitely often
- $\Box \langle \rangle \phi \rightarrow \Box \langle \rangle \psi$
 - if ϕ holds infinitely often then so does ψ

Spring Example



Primitive state assertions: *extended*, *malfunction*

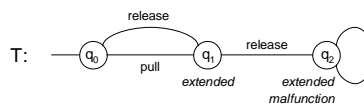
- Sample paths:
- $q_0 q_1 q_0 q_1 q_2 q_2 \dots$
 - $q_0 q_1 q_2 q_2 q_2 \dots$
 - $q_0 q_1 q_0 q_1 q_0 q_1 \dots$

Satisfaction by Single Path



- | | |
|--|---|
| $\xi \models \text{extended?}$ | $\xi \models \Box \langle \rangle \text{extended?}$ |
| $\xi \models \text{Oextended?}$ | $\xi \models \text{extended} \cup \text{malfunction?}$ |
| $\xi \models \text{OOextended?}$ | $\xi \models (\neg \text{extended}) \cup \text{extended?}$ |
| $\xi \models \langle \rangle \text{extended?}$ | $\xi \models (\langle \rangle \text{extended}) \cup \text{malfunction?}$ |
| $\xi \models \Box \text{extended?}$ | $\xi \models (\langle \rangle \neg \text{extended}) \cup \text{malfunction?}$ |
| $\xi \models \langle \rangle \Box \text{extended?}$ | $\xi \models \Box (\neg \text{extended} \rightarrow \text{Oextended})$ |
| $\xi \models \langle \rangle \Box \text{malfunction?}$ | |

Satisfaction by Transition System



- | | |
|--|---|
| $T \models \text{extended?}$ | $T \models \Box \langle \rangle \text{extended?}$ |
| $T \models \text{Oextended?}$ | $T \models \text{extended} \cup \text{malfunction?}$ |
| $T \models \text{OOextended?}$ | $T \models (\neg \text{extended}) \cup \text{extended?}$ |
| $T \models \langle \rangle \text{extended?}$ | $T \models (\langle \rangle \text{extended}) \cup \text{malfunction?}$ |
| $T \models \Box \text{extended?}$ | $T \models (\langle \rangle \neg \text{extended}) \cup \text{malfunction?}$ |
| $T \models \langle \rangle \Box \text{extended?}$ | $T \models \Box (\neg \text{extended} \rightarrow \text{Oextended})$ |
| $T \models \langle \rangle \Box \text{malfunction?}$ | |

Example: Mutex

Assume there are 2 processes, P_i and P_r

State assertions:

- tryCS_i : Process i is trying to enter critical section
 E.g. $\text{tryCS}_i: pc_i = l_4$
- inCS_i : Process i is inside its critical section
 E.g. $\text{inCS}_i: pc_i = l_5 \vee pc_i = l_6$

Mutual exclusion:

$$\Box (\neg (\text{inCS}_i \wedge \text{inCS}_j))$$

Responsiveness:

$$\Box (\text{tryCS}_i \rightarrow \langle \rangle \text{inCS}_i)$$

Process keeps trying until access is granted:

$$\Box (\text{tryCS}_i \rightarrow ((\text{tryCS}_i \cup \text{inCS}_i) \vee \Box \text{tryCS}_i))$$

Example: Fairness

States: Pairs (q, α)

α label of last transition taken, so

$$\frac{q \xrightarrow{\alpha} q'}{(q, \beta) \rightarrow^{\alpha} (q', \alpha)}$$

Σ : Finite set of labels partitioned into subsets P

P : "(finite) set of labels of some process"

State assertions:

- en_P : Some transition labelled $\alpha \in P$ is enabled
 i.e. $(q, \beta) \in P(\text{en}_P)$ iff $\exists q'. q \xrightarrow{\alpha} q'$
- exec_P : Label of last executed transition is in P
 i.e. $(q, \alpha) \in P(\text{exec}_P)$ iff $\alpha \in P$

Note: $\text{en}_P \leftrightarrow \bigvee_{\alpha \in P} \text{en}_{\{\alpha\}}$ and $\text{exec}_P \leftrightarrow \bigvee_{\alpha \in P} \text{exec}_{\{\alpha\}}$

Fairness Conditions

Weak transition fairness:

$$\bigwedge_{\alpha \in \Sigma} \neg \langle \rangle [(\text{en}_{\{\alpha\}} \wedge \neg \text{exec}_{\{\alpha\}})]$$

Or equivalently

$$\bigwedge_{\alpha \in \Sigma} (\langle \rangle [(\text{en}_{\{\alpha\}} \rightarrow \langle \rangle \text{exec}_{\{\alpha\}})])$$

Strong transition fairness:

$$\bigwedge_{\alpha \in \Sigma} (\langle \rangle [(\text{en}_{\{\alpha\}} \rightarrow \langle \rangle \text{exec}_{\{\alpha\}})])$$

Weak process fairness:

$$\bigwedge_p \neg \langle \rangle [(\text{en}_p \wedge \neg \text{exec}_p)]$$

Strong process fairness:

$$\bigwedge_p (\langle \rangle [(\text{en}_p \rightarrow \langle \rangle \text{exec}_p)])$$

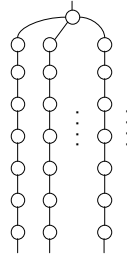
2005 Mads Dam IMIT, KTH

13

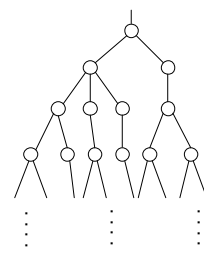
2G1516 Formal Methods

Branching Time Logic

Sets of paths?



Or computation tree?



2005 Mads Dam IMIT, KTH

14

2G1516 Formal Methods

Computation Tree Logic - CTL

Syntax:

$$\phi ::= \eta \mid \neg \phi \mid \phi \wedge \psi \mid \text{AF} \phi \mid \text{AG} \phi \mid \text{A}(\phi \text{ U } \psi) \mid \text{AX} \phi$$

Formulas hold of states, not paths

A: Path quantifier, along all paths from this state

F: $\langle \rangle$, G: $[]$, X: O

So:

- AF ϕ : Along all paths, at some future time instant ϕ is true
- AG ϕ : Along all paths, for all future time instants ϕ is true
- A(ϕ U ψ): Along all paths, ϕ is true until ψ becomes true
- AX ϕ : ϕ is true for all next states

Note: CTL is closed under negation so also express dual modalities EF, EG, EU, EX (E is existential path quantifier)

2005 Mads Dam IMIT, KTH

15

2G1516 Formal Methods

CTL, Semantics

Interpretation function $\rho: \eta \mapsto Q' \subseteq Q$ the same

$q \models \eta$ iff $q \in \rho(\eta)$

$q \models \neg \phi$ iff not $q \models \phi$

$q \models \phi \wedge \psi$ iff $q \models \phi$ and $q \models \psi$

$q \models \text{AF} \phi$ iff for all ξ such that $\xi(0)=q$ exists $k \in \mathbb{N}$ such that $\xi(k) \models \phi$

$q \models \text{AG} \phi$ iff for all ξ such that $\xi(0)=q$, for all $k \in \mathbb{N}$, $\xi(k) \models \phi$

$q \models \text{A}(\phi \text{ U } \psi)$ iff for all ξ such that $\xi(0)=q$, exists $k \in \mathbb{N}$. $\xi(k) \models \psi$ and for all $i: 0 \leq i < k$. $\xi(i) \models \phi$

$q \models \text{AX} \phi$ iff for all q' such that $q \rightarrow q'$, $q' \models \phi$
(iff for all q' such that $q \rightarrow q'$, $q' \models \phi$)

For transition system $T = (Q, R, Q_0)$:

$T \models \phi$ iff for all $q_0 \in Q_0$, $q_0 \models \phi$

2005 Mads Dam IMIT, KTH

16

2G1516 Formal Methods

CTL – LTL: Brief Comparison

LTL in branching time framework:

- $\phi \mapsto \text{A} \phi$ (ϕ to hold for all paths)

CTL $\not\subseteq$ LTL: EF ϕ not expressible in LTL

LTL $\not\subseteq$ CTL: $\langle \rangle [] \eta$ not expressible in LTL

CTL*: Extension of CTL with free alternation A, F, G, U, X

Advantages and disadvantages:

- LTL often "more natural"
- Satisfiability: LTL: PSPACE complete, CTL: DEXPTIME complete
- Model checking: LTL: PSPACE complete, CTL: In P

2005 Mads Dam IMIT, KTH

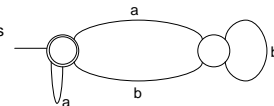
17

2G1516 Formal Methods

Automata Over Finite Words

Finite state automaton $A = (Q, \Sigma, \Delta, I, F)$:

- Q: Finite set of states
- Σ : Finite alphabet
- $\Delta \subseteq Q \times \Sigma \times Q$: Transition relation
Write $q \xrightarrow{a} q'$ for $\Delta(q, a, q')$ as before
- $I \subseteq Q$: Start states
- $F \subseteq Q$: Accepting states



Word $a_1 a_2 \dots a_n$ is accepted, if there is sequence

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} q_n$$

such that $q_0 \in I$ and $q_n \in F$

2005 Mads Dam IMIT, KTH

18

2G1516 Formal Methods

Automata Over Infinite Words

Intuition: Letters $a \in \Sigma$ might represent states, or state properties
 A computation path is an infinite word over object states

Infinite word w :

- Function $w: \mathbb{N} \rightarrow \Sigma$
- Equivalently: Infinite sequence $w = a_0 a_1 a_2 \dots a_n \dots$

Buchi automaton: Finite state automaton, but on infinite words

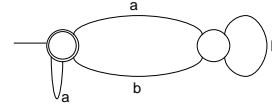
Word w is accepted if accepting state visited infinitely often

2005 Mads Dam IMIT, KTH

19

2G1516 Formal Methods

Example



Which infinite words are accepted?

- ababab ... (= ab^ω) ?
- aaaaaa... (= a^ω) ?
- bbbbb... (= b^ω) ?
- aaabbbb... (= $aaab^\omega$) ?
- abababababba... ?

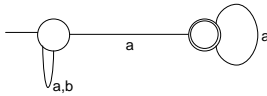
2005 Mads Dam IMIT, KTH

20

2G1516 Formal Methods

Nondeterminism

- What is the language accepted by this automaton?
- What is the corresponding LTL property if $b = \text{inCS}$ and $a = \neg b$?



2005 Mads Dam IMIT, KTH

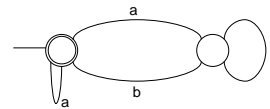
21

2G1516 Formal Methods

Another Example

Letters represent propositions

Example: $[\] \langle \rangle \text{inCS}$, $a = \text{inCS}$, $b = \neg \text{inCS}$



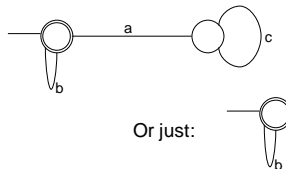
2005 Mads Dam IMIT, KTH

22

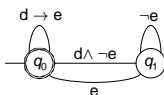
2G1516 Formal Methods

Yet More Examples

- $a = \text{inCS}_1 \wedge \text{inCS}_2$
- $b = \neg a$
- $c = \text{true}$
- Property: $[\] \neg a$



- Property: $[\](d \rightarrow \langle \rangle e)$
- Idea:
 - q_0 : Have seen $\neg d \vee e$
 - q_1 : Saw d , now wait for e



2005 Mads Dam IMIT, KTH

23

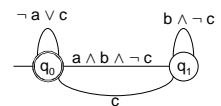
2G1516 Formal Methods

Even More...

Property: $[\](a \rightarrow (bUc))$

Idea:

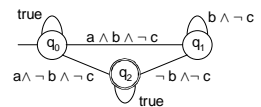
- q_0 : Body of $[\]$ immediately ok
- q_1 : Awaiting c



Property: $\neg[\](a \rightarrow (bUc)) = \langle \rangle(a \wedge \neg(bUc))$

Idea:

- $\neg(bUc)$: b becomes false some time without c having become true first
- q_0 : Waiting ...
- q_1 : Have seen a with b and $\neg c$
- q_2 : Committing ...



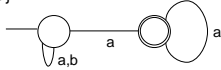
2005 Mads Dam IMIT, KTH

24

2G1516 Formal Methods

Deterministic Buchi Automata

Consider $\phi = \langle \rangle [a$ where $\Sigma = \{a, b\}$



Suppose A recognizes ϕ

A deterministic

A reaches accepting state on some input a^{n1}

And on $a^{n1}ba^{n2}$

And on $a^{n1}ba^{n2}ba^{n3}$

And on $a^{n1}ba^{n2}ba^{n3}b \dots b \dots b \dots$

So: Nondeterministic Buchi automata strictly more expressive than deterministic ones

And: Deterministic B. A. not closed under complement

2005 Mads Dam IMIT, KTH

25

2G1516 Formal Methods

Alternative Formalisms

- Next lecture: LTL \mapsto Buchi automata
- Buchi automata strictly richer than LTL
- B. A. recognisable languages remarkably stable
 - Monadic second order logic of successor

$$\exists X(0 \in X \wedge \forall y \forall z(\text{succ}(y, z) \rightarrow (y \in X \leftrightarrow \neg z \in X)) \wedge \forall y(y \in X \rightarrow a(y)))$$

(all even symbols are a's)

- LTL with propositional quantification

$$\exists X((X \wedge \Box(X \leftrightarrow O \neg X) \wedge \Box(x \rightarrow a))$$

- ω -regular expressions

$$a((a \cup b)a)^\omega$$

- Linear-time μ -calculus

$$\forall X.a \wedge OOX$$

2005 Mads Dam IMIT, KTH

26

2G1516 Formal Methods