INF5140 – Specification and Verification of Parallel Systems

Spring 2017

Institutt for informatikk, Universitetet i Oslo

February 3, 2017

INF5140 – Specification and Verification of Parallel Systems

Logics, lecture 2

Spring 2017

February 3, 2017

Introduction

Logic is "the" specification language for us.[1]

There are many logics to choose from.
Today we see two of them:

- First-order logic (FOL) can be used to describe the state of a program.
- Modal logic can be used to describe the change of state of a program.

Other logics that we will see in other lectures:

- Temporal logics has features not available in FOL like possibility to describe sequences of states.
- Hoare logic is specially designed to reason about (imperative) programs.
- Dynamic logics: more expressive than Hoare logic, more abstract constructs and is more in the tradition of modal logic.

[1] Note: there is no such thing as "*the* logics". There are many . . .

First-order logic

Syntax

Language

The symbols of our first-order language are

- *variables* (a countable set of them $V = \{x, y, \dots\}$)
- *relation symbols* $\mathcal{P} = \{P, Q, \dots\}$ of varying arity (incl. $\doteq$ of arity 2)
- *function symbols* $\mathcal{F} = \{f, g, \dots\}$ of varying arity (if the arity of f is 0 then f is called a *constant* symbols)[a]
- the propositional *connectives* $\neg$, $\vee$, $\wedge$, $\rightarrow$ and $\leftrightarrow$
- the *quantifiers* $\forall$ and $\exists$

[a] Cf. also the notion of *signature* in the term-rewriting talk later (by L. Tveito, 2015)

Syntax: expressions

Expressions (terms)

- Variables are *atomic* expressions.
- If f is a function symbol of arity n, and $t_1, \ldots, t_n$ are terms, then the following is also an expression.

$$f(t_1, \ldots, t_n)$$

If $n = 0$, f is a *constant*.

Example

Using infix notation, the following are expressions:

x	$U \cup V$
$y - 1$	$U \cap V$
$(x + y) + z$	$U \setminus V$

Syntax: Atomic formulae

Atomic formulae

- $\top$ (top) and $\bot$ (bottom) are atomic formulae.
- If P is a relation symbol of arity n, and $t_1, \ldots, t_n$ are terms, then the following is an atomic formulae.

$$P(t_1, \ldots, t_n)$$

Example

Using infix notation, the following are atomic formulae.

$$\begin{array}{ll}
\top & x \in U \\
x < y + 1 & U \subseteq V \\
x \doteq x - 1 & U \cap V \doteq \emptyset
\end{array}$$

Boolean formulae

Boolean formulae

- All atomic formulae are boolean formulae.
- If φ and ψ are boolean formulae, so are the following.

$$\neg\varphi \qquad (\varphi \vee \psi) \qquad (\varphi \wedge \psi) \qquad (\varphi \rightarrow \psi) \qquad (\varphi \leftrightarrow \psi)$$

Example

Some examples of Boolean formulas are:

$$\begin{aligned}
&\neg\neg\top \\
&\neg(x < y + 1) \rightarrow \bot \\
&P \rightarrow (Q \rightarrow P)
\end{aligned}$$

FO formulae

First-order formulae

- All boolean formulae are first-order formulae.
- Let x be a variable. If φ is a first-order formulae, so are the following.

$$(\exists x)\varphi \qquad (\forall x)\varphi$$

- If φ and ψ are first-order formulae, so are the following.

$$\neg\varphi \qquad (\varphi \vee \psi) \qquad (\varphi \wedge \psi) \qquad (\varphi \rightarrow \psi) \qquad (\varphi \leftrightarrow \psi)$$

- $\mathcal{L}$ denotes the set of first-order formulae.

Example

$Q(y) \vee (\forall x)P(x)$

$(\forall x)(\forall y)(x < y \rightarrow (\exists z)(x < z \wedge z < y))$

First-order model

Definition

A model is a pair $M = (D, I)$, such that

- D is a non-empty set (the *domain*)
- I is mapping (the *interpretation*), such that
  - $f^I : D^n \rightarrow D$ for every function symbol f of arity n
  - $P^I \subseteq D^n$ for every relation symbol P of arity n

Observation

- We will assume an implicit model, whose domain will include the natural numbers and sets of natural numbers, and it will be obvious what function and relation symbols should be mapped to.
- E.g.: if $+$ is a function symbol $+^I$ is the addition function on the natural numbers, and $\doteq$ is mapped to a suitable $=$.
- Simplification here: no "sorts" or "types": only one sort ⇒ only one domain. Normally: *many-sorted*

Valuation / state

Given a model

Definition (Valuation)

A *valuation* s over a set of variables V is a mapping from V to D.

- other names: variable assignment
- here, in the context of using logics to speak about programs, where variables in the formula may refer to *program variables:* we will often call a valuation a state

Example

Let $V = \{x, y, z\}$, let x and z be variables of type natural number, and y a variable of type "set of natural numbers".

- $s(x) = 256$
- $s(y) = \{1, 2, 3\}$
- $s(z) = 512$

Valuation of an expression/term

Definition

To every FOL expression t we associate a value $s(t)$ from the domain D in a homomorphic way:

$$s(f(t_1, \dots, t_n)) = f^I(s(t_1), \dots, s(t_n))$$

Example

$$\begin{aligned} s((2 * x) + z) &= s([2 * x]^I) +^I s(z) \\ &= (s(2^I) *^I s(x)) +^I s(z) \\ &= (2 * s(x)) + s(z) \\ &= (2 * 256) + 512 \\ &= 1024 \end{aligned}$$

Free and bound variable occurrences

Definition

- A variable *occurrence* is free in a formula if it is not within the scope of a quantifier. A variable occurrence that is not free is bound.
- Let s_1 and $valn_2$ be states over V, and $x \in V$. s_2 is an x-variant of s_1 if

$$s_1(y) = s_2(y) \text{ for all } y \in V \setminus \{x\}.$$

Thus, x is the only variable the states disagree on.

Substitution

Definition (Substitution)

- Let φ be a first order formula, x a variable and t an expression.
- Then $\varphi[t/x]$ is φ, only with every free occurrence of the x replaced with t.

- Note, the same definition is also used in the lecture about term rewriting (used on terms, not on general FOL formula, but it's "the same".)
- Some other notation has been used like $\varphi_{x \leftarrow c}$. The one used here is the (most) standard one.
- A really exact definition would have to cater for situations like $(\forall x.x + y = 19)[x + 1/y]$.

Example

$$\varphi = (\forall x)P(x) \vee P(x)$$
$$\varphi[c/x] = (\forall x)P(x) \vee P(c)$$

Satisfaction

Definition (Satisfaction)

We define the notion that a state formula φ is true (false) relative to a model $M = (D, I)$ in a state s, written $M, s \models \varphi$ ($M, s \not\models \varphi$) as follows.

$$
\begin{array}{lll}
M, s \models \top & \text{and} & M, s \not\models \bot \\
M, s \models R(t_1, \ldots, t_n) & \text{iff} & (s(t_1), \ldots, s(t_n)) \in R^I \\
M, s \models \neg\varphi & \text{iff} & M, s \not\models \varphi \\
M, s \models \varphi \vee \psi & \text{iff} & M, s \models \varphi \text{ or } M, s \models \psi \\
M, s \models \varphi \wedge \psi & \text{iff} & M, s \models \varphi \text{ and } M, s \models \psi \\
M, s \models \varphi \rightarrow \psi & \text{iff} & M, s \not\models \varphi \text{ or } M, s \models \psi \\
M, s \models \varphi \leftrightarrow \psi & \text{iff} & M, s \models \varphi \rightarrow \psi \text{ and } M, s \models \psi \rightarrow \varphi \\
M, s \models (\forall x)\varphi & \text{iff} & M, t \models \varphi \text{ for every } t \text{ that is an } x\text{-variant of } s \\
M, s \models (\exists x)\varphi & \text{iff} & M, t \models \varphi \text{ for some } t \text{ that is an } x\text{-variant of } s
\end{array}
$$

“Truth” and validity

Definition

- We say that φ is true in the model M, written $M \models \varphi$, if

$$M, s \models \varphi \text{ for every state } s.$$

- We say that φ is valid, written $\models \varphi$, if

$$M \models \varphi \text{ for every model } M.$$

Observation

- We will abuse this notation, and write $\models \varphi$ if φ is true in our implicit model, and refer to this as state-validity.
- For instance: $\models x + y \doteq y + x$.
- In a model where $+^I$ is the subtraction function, this will obviously not hold.

Exercises

- Model the statement: “There are infinitely many primes”.

  $(\forall x)(\exists y)(x \leq y \wedge (\forall z)(z \text{ divides } y \rightarrow (z = 1 \vee z = y)))$
  where we define: z divides $y \triangleq (\exists w)(z \cdot w = y)$.
  Can define $\mathit{prime}(y) \triangleq (\forall z)(z \text{ divides } y \rightarrow (z = 1 \vee z = y))$

- “There is a person with at least two neighbors”

  $(\exists x, y, z)(y \neq z \wedge \mathit{Neigh}(x, y) \wedge \mathit{Neigh}(x, z))$
  where $\mathit{Neigh}(\cdot, \cdot)$ is a binary relation.

  Model now: “There is a person with exactly two neighbors”

  $(\exists x, y, z)(y \neq z \wedge \mathit{Neigh}(x, y) \wedge \mathit{Neigh}(x, z) \wedge$
  $((\forall w)\mathit{Neigh}(x, w) \rightarrow (w = y \vee w = z)))$.

- “Every even number can be written as a sum of two primes”

  $(\forall x)((\mathit{even}(x) \wedge x > 2) \rightarrow$
  $(\exists y, z)(\mathit{prime}(y) \wedge \mathit{prime}(z) \wedge y + z = x))$
  where the shorthand $\mathit{even}(x) \triangleq (\exists w)(2 \cdot w = x)$.

We assume the domain $-$ with standard $\cdot, +, >$.

Deductions and proof systems

Definition

A proof system for a given logic consists of

- axioms (or *axiom schemata*), which are formulae assumed to be true, and
- inference rules, of approx. the form

$$\frac{\varphi_1 \quad \cdots \quad \varphi_n}{\psi}$$

where $\varphi_1, \ldots, \varphi_n$ are premises and ψ the conclusion.

Derivations and proofs

Definition

- A derivation from a set of formulae S is a sequence of formulae, where each formula is either in S, an axiom or can be obtained by applying an inference rule to formulae earlier in the sequence.
- A proof is a derivation from the empty set.
- A theorem is the last formula in a proof.
- A proof system is
  - sound if every theorem is valid.
  - complete if evey valid formula is a theorem.
- We do not study soundness and completeness in this course.

Proof systems and proofs: remarks

- the "definitions" from the previous slides: not very formal
- in general: a proof system: a "mechanical" (= formal and constructive) way of conclusions from axioms (= "given" formulas), and other already proven formulas
- Many different "representations" of how to draw conclusions exists
- the one sketched on the previous slide
  - works with "sequences"
  - corresponds to the historically oldest "style" of proof systems ("Hilbert-style")
  - otherwise, in that naive form: impractical (but sound & complete).
  - nowadays, better ways and more suitable for computer support of representation exists (especially using trees). For instance natural deduction style system
- for the course, those variations don't matter.

a proof system for prop. logic

Observation

We can axiomatize a subset of *propositional logic* as follows.

$$\varphi \to (\psi \to \varphi) \tag{A1}$$

$$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi)) \tag{A2}$$

$$((\varphi \to \bot) \to \bot) \to \varphi \tag{DN}$$

$$\frac{\varphi \quad \varphi \to \psi}{\psi} \tag{MP}$$

Let us call this logic PPL.

Note: As said, it's only one of many different ways and styles to axiomatize logic (here prop. logic)

A proof system

Example

$p \to p$ is a theorem of PPL:

$$
\begin{align}
&\begin{array}{l}(p \to ((p \to p) \to p)) \to \\ ((p \to (p \to p)) \to (p \to p))\end{array} && \text{AX2} \tag{1}\\
&p \to ((p \to p) \to p) && \text{AX1} \tag{2}\\
&(p \to (p \to p)) \to (p \to p) && \text{MP on (1) and (2)} \tag{3}\\
&p \to (p \to p) && \text{AX1} \tag{4}\\
&p \to p && \text{MP on (3) and (4)} \tag{5}
\end{align}
$$

Observation

A proof can be represented as a tree of inferences where the leaves are axioms.

Modal logics

Introduction

- Modal logic: logic of "*necessity*" and "*possibility*", in that originally the intended meaning of the *modal* operators $\Box$ and $\Diamond$ was
  - $\Box\varphi$: φ is necessarily true.
  - $\Diamond\varphi$: φ is possibly true.
- Depending on what we intend to capture: we can interpret $\Box\varphi$ differently.

temporal	φ will always hold.
doxastic	I believe φ.
epistemic	I know φ.
intuitionistic	φ is provable.
deontic	It ought to be the case that φ.

- We will restrict here the modal operators to $\Box$ and $\Diamond$ (and mostly work with a temporal "mind-set".

Kripke structure

Definition (Kripke model)

- A Kripke frame is a structure (W, R) where
  - W is a non-empty set of *worlds*, and
  - $R \subseteq W \times W$ is called the *accessibility relation* between worlds.
- A Kripke model M is a structure (W, R, V) where
  - (W, R) is a frame, and
  - $V : W \to 2^{\Phi}$ labels each world with a set of propositional variables.

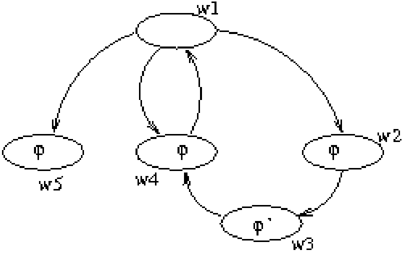
Remark: some also consider propositional variables as propositional constants, propositional "symbols", it's unimportant. Kripke models are sometimes called Kripke structures.

Example

Example

Let $M = (W, R, V)$ be the Kripke model such that

- $W = \{w_1, w_2, w_3, w_4, w_5\}$
- $R = \{(w_1, w_5), (w_1, w_4), (w_4, w_1), \ldots\}$
- $V = \langle w_1 : \emptyset, w_2 : \{\phi\}, w_3 : \{\phi'\}, \ldots\rangle$



Satisfaction

Definition

A modal formula φ is true in the world w of a model M, written $M, w \models \varphi$, if:

$$
\begin{array}{lll}
M, w \models p_i & \text{iff} & p_i \in V(w) \\
\\
M, w \models \neg\varphi & \text{iff} & M, w \not\models \varphi \\
M, w \models \varphi_1 \vee \varphi_2 & \text{iff} & M, w \models \varphi_1 \text{ or } M, w \models \varphi_2 \\
\\
M, w \models \Box\varphi & \text{iff} & M, w' \models \varphi \text{ for all } w' \text{ such that } wRw' \\
M, w \models \Diamond\varphi & \text{iff} & M, w' \models \varphi \text{ for some } w' \text{ such that } wRw'
\end{array}
$$

But what does box and diamond intuitively “mean”?

Observation

- The semantics only differs for $\Box$ and $\Diamond$.
- We don’t put any restriction on the accessibility relation R.
- The “mental picture” of what to think of $\Box$ and $\Diamond$ depends on the properties of R (and what we think R actually represent)

Different kinds of accessibility relations

Definition

A binary relation $R \subseteq W \times W$ is

- reflexive if every element in W is R-related to itself.

$$(\forall a) aRa$$

- transitive if

$$(\forall abc)(aRb \wedge bRc \rightarrow aRc)$$

- euclidean if

$$(\forall abc)(aRb \wedge aRc \rightarrow bRc)$$

- total if

$$(\forall a)(\exists b)(aRb)$$

Modal Logic

Semantics

If $(W, R, V), s \models \varphi$ for all s and V, we write

$$(W, R) \models \varphi$$

Example

- $(W, R) \models \Box\varphi \to \varphi$ iff R is reflexive.
- $(W, R) \models \Box\varphi \to \Diamond\varphi$ iff R is total.
- $(W, R) \models \Box\varphi \to \Box\Box\varphi$ iff R is transitive.
- $(W, R) \models \neg\Box\varphi \to \Box\neg\Box\varphi$ iff R is euclidean.

Observation

The axioms above are said to “hold on a frame”, which means, for *any* valuation and at *any* state.

Prove the double implications from the slide before!

1. The forward implications are based on the fact that we quantify over *all* valuations and all states. More precisely; assume an arbitrary frame (W, R) which does NOT have the property (e.g., reflexive). Find a valuation and a state where the axiom does not hold. You have now the contradiction . . .
2. For the backward implication take an arbitrary frame (W, R) which *has* the property (e.g., euclidian). Take an arbitrary valuation and an arbitrary state on this frame. Show that the axiom holds in this state under this valuation. Sometimes one may need to use an inductive argument or to work with properties derived from the main property on R (e.g., if R is euclidian then $(\forall w_1, w_2 \in W)(w_1 R w_2 \to w_2 R w_2))$

An axiomatic system

Every normal modal logic has the following inference rules.

$$\frac{\varphi \text{ is a tautology instance}}{\varphi} \quad \text{(PL)}$$

$$\frac{\varphi \quad \varphi \to \psi}{\psi} \quad \text{(MP)}$$

$$\frac{\varphi}{\Box\varphi} \quad \text{(G)}$$

We will only be concerned with normal modal logics.

Sample axioms for different accessibility relations

Formulae that can be used to axiomatize logics with different properties.

$$
\begin{aligned}
&\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi) && \text{(K)}\\
&\Box\varphi \to \Diamond\varphi && \text{(D)}\\
&\Box\varphi \to \varphi && \text{(T)}\\
&\Box\varphi \to \Box\Box\varphi && \text{(4)}\\
&\neg\Box\varphi \to \Box\neg\Box\varphi && \text{(5)}\\
&\Box(\Box\varphi \to \psi) \to \Box(\Box\psi \to \varphi) && \text{(3)}\\
&\Box(\Box(\varphi \to \Box\varphi) \to \varphi) \to (\Diamond\Box\varphi \to \varphi)) && \text{(Dum)}
\end{aligned}
$$

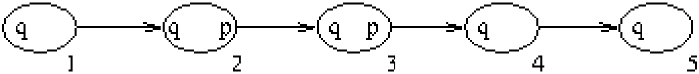
- Every normal logic has K as axiom schema.
- Observe that T implies D.

Different “flavors” of modal logic

Logic	Axioms	Interpretation	Properties of R
D	K D	deontic	total
T	K T		reflexive
K45	K 4 5	doxastic	transitive/euclidean
S4	K T 4		reflexive/transitive
S5	K T 5	epistemic	reflexive/euclidean reflexive/symmetric/transitive equivalence relation

Exercises

1. Consider the frame (W, R) with $W = \{1, 2, 3, 4, 5\}$ and $(i, i+1) \in R$

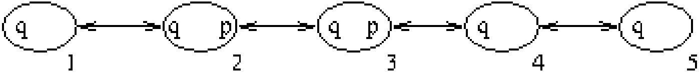


Choose the valuation $V(p) = \{2, 3\}$ and $V(q) = \{1, 2, 3, 4, 5\}$ to get the model $M = (W, R, V)$.
Which of the following statements are correct in M and why?

1.1 $M, 1 \models \Diamond\Box p$ Correct
1.2 $M, 1 \models \Diamond\Box p \to p$ Incorrect
1.3 $M, 3 \models \Diamond(q \wedge \neg p) \wedge \Box(q \wedge \neg p)$ Correct
1.4 $M, 1 \models q \wedge \Diamond(q \wedge \Diamond(q \wedge \Diamond(q \wedge \Diamond q)))$ Correct
1.5 $M \models \Box q$ Correct . . . but why?

Exercises 2 (bidirectional frames)

We call a frame (W, R) bidirectional iff $R = R_F \uplus R_P$ s.t. $\forall w, w'(wR_F w' \leftrightarrow w' R_P w)$.
i.e.: The R can be separated into two disjoind relations R_F and R_P, which one is the inverse of the other.



Consider the model $M = (W, R, V)$ from before.
Which of the following statements are correct in M and why?

0.1 $M, 1 \models \Diamond\Box p$ Incorrect

0.2 $M, 1 \models \Diamond\Box p \to p$ Correct

0.3 $M, 3 \models \Diamond(q \land \neg p) \land \Box(q \land \neg p)$ Incorrect

0.4 $M, 1 \models q \land \Diamond(q \land \Diamond(q \land \Diamond(q \land \Diamond q)))$ Correct

0.5 $M \models \Box q$ Correct . . . but is it the same explanation as before?

0.6 $M \models \Box q \to \Diamond\Diamond p$

Exercises 3 (validities)

Which of the following are valid in modal logic. For those that are not, argue why and find a class of frames on which they become valid.

1. $\Box\bot$
   Valid on frames where $R = \emptyset$.
2. $\Diamond p \to \Box p$
   Valid on frames where R is a partial function.
3. $p \to \Box\Diamond p$
   Valid on bidirectional frames.
4. $\Diamond\Box p \to \Box\Diamond p$
   Valid on Euclidian frames.

Further readings

- [Harel et al., 2000]
- [Blackburn et al., 2001]

Linear-Time Temporal Logic (LTL)

Remark

We have left out this semester Hoare logic.

Introduction

Temporal Logic?

- Temporal logic is the logic of "time"[a]
- It is a *modal* logic.
- There are different ways of modeling time.
  - linear time vs. branching time
  - time instances vs. time intervals
  - discrete time vs. continuous time
  - past and future vs. future only

[a] pay attention, it will be something kind of abstract, it's mostly not what's known as *real-time*, but there are variants of temporal logics which can handle real-time. They *won't* occur in this lecture .

FOL (repetition)

First Order Logic

- We have used FOL to express properties of states.
  - $\langle x : 21, y : 49 \rangle \Vdash x < y$
  - $\langle x : 21, y : 7 \rangle \nVdash x < y$
- A computation is a sequence of states.
- To express properties of computations, we need to extend FOL.
- This we can do using temporal logic.

LTL: speaking about “time”

In Linear Temporal Logic (LTL) (also called linear-time temporal logic) we can describe such properties as follows: assume time is a sequence[2] of discrete points i in time, then: if i is *now*,

- p holds in i and every following point (the future)
- p holds in i and every preceding point (the past)

We will only be concerned with the future.

$$\cdots \longrightarrow \bullet^{p}_{i-2} \longrightarrow \bullet^{p}_{i-1} \longrightarrow \bullet^{p}_{i} \longrightarrow \bullet^{p}_{i+1} \longrightarrow \bullet^{p}_{i+2} \longrightarrow \cdots$$

[2] a sequence is linear

LTL operators

We extend our first-order language[3] $\mathcal{L}$ to a temporal language $\mathcal{L}_T$ by adding the temporal operators $\Box$, $\Diamond$, $\bigcirc$, U, R and W.

Interpretation of the operators

$\Box\varphi$	φ will *always* (in every state) hold
$\Diamond\varphi$	φ will *eventually* (in some state) hold
$\bigcirc\varphi$	φ will hold at the *next* point in time
$\varphi U \psi$	ψ will eventually hold, and *until* that point φ will hold
$\varphi R \psi$	ψ holds until (incl.) the point (if any) where φ holds (*release*)
$\varphi W \psi$	φ will hold until ψ holds (*weak until* or *waiting for*)

[3] Note: it's equally ok to extend a propositional language the same way. The difference is between a first-order LTL or propositional LTL.

Syntax

We define LTL formulae as follows.

Definition

- $\mathcal{L} \subseteq \mathcal{L}_T$: first-order formulae are also LTL formulae.
- If φ is an LTL formula, so are the following.

$$\Box\varphi \quad \Diamond\varphi \quad \bigcirc\varphi \quad \neg\varphi$$

- If φ and ψ are LTL formulae, so are

$$\varphi U \psi \quad \varphi R \psi \quad (\varphi W \psi)$$
$$(\varphi \vee \psi) \quad (\varphi \wedge \psi) \quad (\varphi \to \psi) \quad (\varphi \leftrightarrow \psi)$$

- nothing else

Paths and computations

Definition

- A path is an infinite sequence

$$\sigma = s_0, s_1, s_2, \ldots$$

  of states.
- σ^k denotes the *path* $s_k, s_{k+1}, s_{k+2}, \ldots$
- σ_k denotes the *state* s_k.
- All computations are paths, but not vice versa.

Satisfaction (semantics)

Definition

We define the notion that an LTL formula φ is true (false) relative to a path σ, written $\sigma \models \varphi$ ($\sigma \not\models \varphi$) as follows.

$$\begin{array}{lll}
\sigma \models \varphi & \text{iff} & \sigma_0 \Vdash \varphi \text{ when } \varphi \in \mathcal{L} \\
\sigma \models \neg\varphi & \text{iff} & \sigma \not\models \varphi \\
\sigma \models \varphi \vee \psi & \text{iff} & \sigma \models \varphi \text{ or } \sigma \models \psi
\end{array}$$

$$\begin{array}{lll}
\sigma \models \Box\varphi & \text{iff} & \sigma^k \models \varphi \text{ for all } k \geq 0 \\
\sigma \models \Diamond\varphi & \text{iff} & \sigma^k \models \varphi \text{ for some } k \geq 0 \\
\sigma \models \bigcirc\varphi & \text{iff} & \sigma^1 \models \varphi
\end{array}$$

(cont.)

Satisfaction (semantics) (2)

Definition

(cont.)

$$\sigma \models \varphi U \psi \quad \text{iff} \quad \sigma^k \models \psi \text{ for some } k \geq 0, \text{ and}$$
$$\sigma^i \models \varphi \text{ for every } i \text{ such that } 0 \leq i < k$$

$$\sigma \models \varphi R \psi \quad \text{iff} \quad \text{for every } j \geq 0,$$
$$\text{if } \sigma^i \not\models \varphi \text{ for every } i < j \text{ then } \sigma^j \models \psi$$

$$\sigma \models \varphi W \psi \quad \text{iff} \quad \sigma \models \varphi U \psi \text{ or } \sigma \models \Box\varphi$$

Validity and semantic equivalence

Definition

- We say that φ is (temporally) valid, written $\models \varphi$, if
$$\sigma \models \varphi \text{ for all paths } \sigma.$$
- We say that φ and ψ are equivalent, written $\varphi \sim \psi$, if
$$\models \varphi \leftrightarrow \psi \text{ (i.e. } \sigma \models \varphi \text{ iff } \sigma \models \psi \text{, for all } \sigma).$$

Example

$\Box$ distributes over $\wedge$, while $\Diamond$ distributes over $\vee$.

$$\Box(\varphi \wedge \psi) \sim (\Box\varphi \wedge \Box\psi)$$
$$\Diamond(\varphi \vee \psi) \sim (\Diamond\varphi \vee \Diamond\psi)$$

Semantics

$\sigma \models \Box p$

$$\bullet_0^p \longrightarrow \bullet_1^p \longrightarrow \bullet_2^p \longrightarrow \bullet_3^p \longrightarrow \bullet_4^p \longrightarrow \dots$$

$\sigma \models \Diamond p$

$$\bullet_0 \longrightarrow \bullet_1 \longrightarrow \bullet_2 \longrightarrow \bullet_3^p \longrightarrow \bullet_4 \longrightarrow \dots$$

$\sigma \models \bigcirc p$

$$\bullet_0 \longrightarrow \bullet_1^p \longrightarrow \bullet_2 \longrightarrow \bullet_3 \longrightarrow \bullet_4 \longrightarrow \dots$$

$\sigma \models pUq$ (sequence of p's is finite)

$$\bullet_0^p \longrightarrow \bullet_1^p \longrightarrow \bullet_2^p \longrightarrow \bullet_3^q \longrightarrow \bullet_4 \longrightarrow \dots$$

$\sigma \models pRq$ (The sequence of qs may be infinite)

$$\bullet_0^q \longrightarrow \bullet_1^q \longrightarrow \bullet_2^q \longrightarrow \bullet_3^{p,q} \longrightarrow \bullet_4 \longrightarrow \dots$$

$\sigma \models pWq$. The sequence of ps may be infinite.
$(pWq \sim pUq \vee \Box p)$.

$$\bullet_0^p \longrightarrow \bullet_1^p \longrightarrow \bullet_2^p \longrightarrow \bullet_3^p \longrightarrow \bullet_4^p \longrightarrow \dots$$

The past

Observation

- [Manna and Pnueli, 1992] uses pairs (σ, j) of paths and positions instead of just the path σ because they have past-formulae: formulae without future operators (the ones we use) but possibly with past operators, like $\Box^{-1}$ and $\Diamond^{-1}$.

$$\begin{array}{lll} (\sigma, j) \models \Box^{-1}\varphi & \text{iff} & (\sigma, k) \models \varphi \text{ for all } k,\ 0 \leq k \leq j \\ (\sigma, j) \models \Diamond^{-1}\varphi & \text{iff} & (\sigma, k) \models \varphi \text{ for some } k,\ 0 \leq k \leq j \end{array}$$

- However, it can be shown that for any formula φ, there is a future-formula (formulae without past operators) ψ such that

$$(\sigma, 0) \models \varphi \quad \text{iff} \quad (\sigma, 0) \models \psi$$

The past: examples

Example

What is a future version of $\Box(p \rightarrow \Diamond^{-1}q)$?
$(\sigma, 0) \models \Box(p \rightarrow \Diamond^{-1}q)$

$$\bullet^{p\rightarrow\Diamond^{-1}q} \longrightarrow \bullet^{p\rightarrow\Diamond^{-1}q} \longrightarrow \bullet^{p\rightarrow\Diamond^{-1}q} \longrightarrow \bullet^{p\rightarrow\Diamond^{-1}q} \longrightarrow \bullet \longrightarrow$$

$(\sigma, 0) \models qR(p \rightarrow q)$

$$\bullet^{p\rightarrow q} \longrightarrow \bullet^{p\rightarrow q} \longrightarrow \bullet^{p\rightarrow q,q} \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \ldots$$

Examples

Example

$\varphi \rightarrow \Diamond\psi$: Inf φ holds initially, then ψ holds eventually.

$$\bullet^{\varphi} \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet^{\psi} \longrightarrow \bullet \longrightarrow \ldots$$

This formula will also hold in every path where φ does not hold initially.

$$\bullet^{\neg\varphi} \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \ldots$$

Example: Response

Example (Response)

$\Box(\varphi \rightarrow \Diamond\psi)$

Every φ-position coincides with or is followed by a ψ-position.

$$\bullet \longrightarrow \bullet^{\varphi} \longrightarrow \bullet \longrightarrow \bullet^{\psi} \longrightarrow \bullet \longrightarrow \bullet^{\varphi,\psi} \longrightarrow \ldots$$

This formula will also hold in every path where φ never holds.

$$\bullet^{\neg\varphi} \longrightarrow \bullet^{\neg\varphi} \longrightarrow \bullet^{\neg\varphi} \longrightarrow \bullet^{\neg\varphi} \longrightarrow \bullet^{\neg\varphi} \longrightarrow \ldots$$

Examples

Example

$\Box\Diamond\psi$

There are infinitely many ψ-positions.

$$\bullet^{\varphi} \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet^{\varphi} \longrightarrow \bullet \longrightarrow \bullet^{\varphi} \longrightarrow \bullet \longrightarrow \dots$$

This formula can be obtained from the previous one, $\Box(\varphi \to \Diamond\psi)$, by letting $\varphi = \top$: $\Box(\top \to \Diamond\psi)$.

Example: permanence

Example

$\Diamond\Box\varphi$
Eventually φ will hold permanently.

$$\bullet \longrightarrow \bullet^{\varphi} \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet^{\varphi} \longrightarrow \bullet^{\varphi} \longrightarrow \bullet^{\varphi} \longrightarrow \dots$$

Equivalently: there are finitely many $\neg\varphi$-positions.

LTL example

Example

$(\neg\varphi) W \psi$

The first φ-position must coincide or be preceded by a ψ-position.

$$\bullet^{\neg\varphi} \longrightarrow \bullet^{\neg\varphi} \longrightarrow \bullet^{\neg\varphi} \longrightarrow \bullet^{\psi} \longrightarrow \bullet^{\varphi} \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \dots$$

φ may never hold

$$\bullet^{\neg\varphi} \longrightarrow \bullet^{\neg\varphi} \longrightarrow \bullet^{\neg\varphi} \longrightarrow \bullet^{\neg\varphi} \longrightarrow \bullet^{\neg\varphi} \longrightarrow \bullet^{\neg\varphi} \longrightarrow \bullet^{\neg\varphi} \longrightarrow .$$

LTL Example

Example

$\Box(\varphi \to \psi W \chi)$
Every φ-position initiates a sequence of ψ-positions, and if terminated, by a χ-position.

$$\bullet \longrightarrow \bullet^{\varphi,\psi} \longrightarrow \bullet^{\psi} \longrightarrow \bullet^{\psi} \longrightarrow \bullet^{\chi} \longrightarrow \bullet \longrightarrow \bullet^{\varphi,\psi} \longrightarrow \dots$$

The sequence of ψ-positions need not terminate.

$$\bullet \longrightarrow \bullet^{\varphi,\psi} \longrightarrow \bullet^{\psi} \longrightarrow \bullet^{\psi} \longrightarrow \bullet^{\psi} \longrightarrow \bullet^{\psi} \longrightarrow \bullet^{\psi} \longrightarrow \dots$$

Nested waiting-for

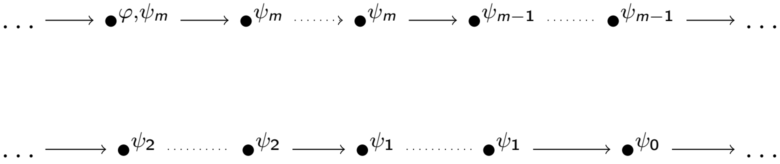
A nested waiting-for formula is of the form

$$\Box(\varphi \to (\psi_m W(\psi_{m-1} W \cdots (\psi_1 W \psi_0) \cdots))),$$

where $\varphi, \psi_0, \ldots, \psi_m \in \mathcal{L}$. For the sake of convenience, we write

$$\Box(\varphi \to \psi_m \, W \, \psi_{m-1} \, W \, \cdots \, W \, \psi_1 \, W \, \psi_0).$$

Every φ-position initiates a succession of intervals, beginning with a ψ_m-interval, ending with a ψ_1-interval and possibly terminated by a ψ_0-position. Each interval may be empty or extend to infinity.



Capturing informally understood temporal specifications formally

It can be difficult to correctly formalize informally stated requirements in temporal logic.

Example

How does one formalize the informal requirement "φ implies ψ"?

- $\varphi \rightarrow \psi$? $\varphi \rightarrow \psi$ holds in the initial state.
- $\Box(\varphi \rightarrow \psi)$? $\varphi \rightarrow \psi$ holds in every state.
- $\varphi \rightarrow \Diamond\psi$? φ holds in the initial state, ψ will hold in some state.
- $\Box(\varphi \rightarrow \Diamond\psi)$? We saw this earlier.
- None of these is necessarily what we intended

Duals

Definition (Duals)

For binary boolean connectives[a] $\circ$ and $\bullet$, we say that $\bullet$ is the dual of $\circ$ if

$$\neg(\varphi \circ \psi) \sim (\neg\varphi \bullet \neg\psi).$$

Similarly for unary connectives: $\bullet$ is the dual of $\circ$ if $\neg \circ \varphi \sim \bullet\neg\varphi$.

[a] Those are not concrete connectives or operators, they are meant as "placeholders"

Duality is symmetric:

- If $\bullet$ is the dual of $\circ$ then
- $\circ$ is the dual of $\bullet$, thus
- we may refer to two connectives as dual (of each other).

Dual connectives

Which connectives are duals?

- $\wedge$ and $\vee$ are duals:

$$\neg(\varphi \wedge \psi) \sim (\neg\varphi \vee \neg\psi).$$

- $\neg$ is its own dual:

$$\neg\neg\varphi \sim \neg\neg\varphi.$$

- What is the dual of $\rightarrow$? It's $\nleftarrow$:

$$\begin{aligned} \neg(\varphi \nleftarrow \psi) &\sim \varphi \leftarrow \psi \\ &\sim \psi \rightarrow \varphi \\ &\sim \neg\varphi \rightarrow \neg\psi \end{aligned}$$

Complete sets of connectives

- A set of connectives is complete (for boolean formulae) if every other connective can be defined in terms of them.
- Our set of connectives is complete (e.g., $\nleftarrow$ can be defined), but also subsets of it, so we don't actually need all the connectives.

Example

$\{\vee, \neg\}$ is complete.

- $\wedge$ is the dual of $\vee$.
- $\varphi \rightarrow \psi$ is equivalent to $\neg\varphi \vee \psi$.
- $\varphi \leftrightarrow \psi$ is equivalent to $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.
- $\top$ is equivalent to $p \vee \neg p$
- $\bot$ is equivalent to $p \wedge \neg p$

Duals in LTL

We can extend the notions of duality and completeness to temporal formulae.

Duals of temporal operators

- What is the dual of $\Box$? And of $\Diamond$?
- $\Box$ and $\Diamond$ are duals.

$$\neg\Box\varphi \sim \Diamond\neg\varphi$$
$$\neg\Diamond\varphi \sim \Box\neg\varphi$$

- Any other?
- U and R are duals.

$$\neg(\varphi U \psi) \sim (\neg\varphi)R(\neg\psi)$$
$$\neg(\varphi R \psi) \sim (\neg\varphi)U(\neg\psi)$$

Complete set of LTL operators

We don't need all our temporal operators either.

Proposition

$\{\vee, \neg, U, \bigcirc\}$ *is complete for LTL.*

Proof:

- $\Diamond\varphi \sim \top U \varphi$
- $\Box\varphi \sim \bot R \varphi$
- $\varphi R \psi \sim \neg(\neg\varphi U \neg\psi)$
- $\varphi W \psi \sim \Box\varphi \vee (\varphi U \psi)$

□

Classification of properties

We can classify properties expressible in LTL.

Classification

safety	$\Box\varphi$
liveness	$\Diamond\varphi$
obligation	$\Box\varphi \vee \Diamond\psi$
recurrence	$\Box\Diamond\varphi$
persistence	$\Diamond\Box\varphi$
reactivity	$\Box\Diamond\varphi \vee \Diamond\Box\psi$

Safety

- important basic class of properties
- relation to testing and run-time verification
- "nothing bad ever happens"

Definition (Safety)

- A safety formula is of the form

  $$\Box\varphi$$

  for some first-order formula φ.
- A conditional safety formula is of the form

  $$\varphi \to \Box\psi$$

  for (first-order) formulae φ and ψ.
- Safety formulae express *invariance* of some state property φ: that φ holds in every state of the computation.

Safety property example

Example

- *Mutual exclusion* is a safety property. Let C_i denote that process P_i is executing in the critical section. Then

  $$\Box\neg(C_1 \wedge C_2)$$

  expresses that it should always be the case that not both P_1 and P_2 are executing in the critical section.
- Observe that the negation of a safety formula is a liveness formula; the negation of the formula above is the liveness formula

  $$\Diamond(C_1 \wedge C_2)$$

  which expresses that eventually it *is* the case that both P_1 and P_2 are executing in the critical section.

Liveness properties

Definition (Liveness)

- A liveness formula is of the form

  $$\Diamond\varphi$$

  for some first-order formula φ.
- A conditional liveness formula is of the form

  $$\varphi \to \Diamond\psi$$

  for first-order formulae φ and ψ.
- Liveness formulae *guarantee* that some event φ eventually happens: that φ holds in at least one state of the computation.

Connection to Hoare logic

Observation

- Partial correctness is a safety property. Let P be a program and ψ the post condition.

$$\Box(\mathit{terminated}(P) \to \psi)$$

- In the case of full partial correctness, where there is a precondition φ, we get a *conditional safety* formula,

$$\varphi \to \Box(\mathit{terminated}(P) \to \psi),$$

which we can express as $\{\ \varphi\ \}\ P\ \{\ \psi\ \}$ in Hoare Logic.

Total correctness and liveness

Observation

- Total correctness is a liveness property. Let P be a program and ψ the post condition.

$$\Diamond(\mathit{terminated}(P) \wedge \psi)$$

- In the case of full total correctness, where there is a precondition φ, we get a *conditional liveness* formula,

$$\varphi \rightarrow \Diamond(\mathit{terminated}(P) \wedge \psi).$$

Duality of partial and total correctness

Observation

Partial and total correctness are dual.

Let

$$PC(\psi) \triangleq \Box(\mathit{terminated} \rightarrow \psi)$$
$$TC(\psi) \triangleq \Diamond(\mathit{terminated} \land \psi)$$

Then

$$\neg PC(\psi) \sim PC(\neg\psi)$$
$$\neg TC(\psi) \sim TC(\neg\psi)$$

Obligation

Definition (Obligation)

- A simple obligation formula is of the form

  $$\Box\varphi \vee \Diamond\psi$$

  for first-order formula φ and ψ.
- An equivalent form is

  $$\Diamond\chi \to \Diamond\psi$$

  which states that some state satisfies χ only if some state satisfies ψ.

Obligation (2)

Proposition

Every safety and liveness formula is also an obligation formula.

Proof: This is because of the following equivalences.

$$\Box\varphi \sim \Box\varphi \vee \Diamond\bot$$
$$\Diamond\varphi \sim \Box\bot \vee \Diamond\varphi$$

and the facts that $\models \neg\Box\bot$ and $\models \neg\Diamond\bot$. □

Recurrence

Definition (Recurrence)

- A recurrence formula is of the form

$$\Box\Diamond\varphi$$

  for some first-order formula φ.
- It states that infinitely many positions in the computation satisfies φ.

Observation

A response formula, of the form $\Box(\varphi \rightarrow \Diamond\psi)$, is equivalent to a recurrence formula, of the form $\Box\Diamond\chi$, if we allow χ to be a past-formula.

$$\Box(\varphi \rightarrow \Diamond\psi) \sim \Box\Diamond(\neg\varphi)W^{-1}\psi$$

Recurrence

Proposition

Weak fairness[a] *can be specified as the following recurrence formula.*

$$\Box\Diamond(\mathit{enabled}(\tau) \rightarrow \mathit{taken}(\tau))$$

[a] weak and strong fairness will be "recurrent" (sorry for the pun) themes. For instance they will show up again in the TLA presentation.

Observation

An equivalent form is

$$\Box(\Box\mathit{enabled}(\tau) \rightarrow \Diamond\mathit{taken}(\tau)),$$

which looks more like the first-order formula we saw last time.

Persistence

Definition (Persistence)

- A persistence formula is of the form

$$\Diamond\Box\varphi$$

  for some first-order formula φ.
- It states that all but finitely many positions satisfy φ[a]
- Persistence formulae are used to describe the eventual stabilization of some state property.

[a] In other words: only finitely ("but") many position satisfy $\neg\varphi$. So at some point onwards, it's always φ.

Recurrence and Persistence

Observation

Recurrence and persistence are duals.

$$\neg(\Box\Diamond\varphi) \sim (\Diamond\Box\neg\varphi)$$
$$\neg(\Diamond\Box\varphi) \sim (\Box\Diamond\neg\varphi)$$

Reactivity

Definition (Reactivity)

- A simple reactivity formula is of the form

$$\Box\Diamond\varphi \vee \Diamond\Box\psi$$

  for first-order formula φ and ψ.
- A very general class of formulae are conjunctions of reactivity formulae.
- An equivalent form is

$$\Box\Diamond\chi \rightarrow \Box\Diamond\psi,$$

  which states that if the computation contains infinitely many χ-positions, it must also contain infinitely many ψ-positions.

Reactivity

Proposition

Strong fairness can be specified as the following reactivity formula.

$$\Box\Diamond \mathit{enabled}(\tau) \rightarrow \Box\Diamond \mathit{taken}(\tau)$$

GCD Example

Below is a computation σ of our recurring GCD program.

- a and b are fixed: $\sigma \models \Box(a \doteq 21 \wedge b \doteq 49)$.
- $at(l)$ denotes the formulae $(\pi \doteq \{l\})$.
- *terminated* denotes the formula $at(l_8)$.

P-computation

States are of the form $\langle \pi, x, y, g \rangle$.

$$\begin{array}{rllll}
\sigma : & \langle l_1, 21, 49, 0 \rangle \rightarrow & \langle l_2^b, 21, 49, 0 \rangle \rightarrow & \langle l_6, 21, 49, 0 \rangle \rightarrow \\
& \langle l_1, 21, 28, 0 \rangle \rightarrow & \langle l_2^b, 21, 28, 0 \rangle \rightarrow & \langle l_6, 21, 28, 0 \rangle \rightarrow \\
& \langle l_1, 21, 7, 0 \rangle \rightarrow & \langle l_2^a, 21, 7, 0 \rangle \rightarrow & \langle l_4, 21, 7, 0 \rangle \rightarrow \\
& \langle l_1, 14, 7, 0 \rangle \rightarrow & \langle l_2^a, 14, 7, 0 \rangle \rightarrow & \langle l_4, 14, 7, 0 \rangle \rightarrow \\
& \langle l_1, 7, 7, 0 \rangle \rightarrow & \langle l_7, 7, 7, 0 \rangle \rightarrow & \langle l_8, 7, 7, 7 \rangle \rightarrow \cdots
\end{array}$$

GCD Example

Does the following properties hold for σ? And why?

1. $\Box \mathit{terminated}$ (safety)
2. $at(l_1) \to \mathit{terminated}$
3. $at(l_8) \to \mathit{terminated}$
4. $at(l_7) \to \Diamond \mathit{terminated}$ (conditional liveness)
5. $\Diamond at(l_7) \to \Diamond \mathit{terminated}$ (obligation)
6. $\Box(\mathrm{gcd}(x, y) \doteq \mathrm{gcd}(a, b))$ (safety)
7. $\Diamond \mathit{terminated}$ (liveness)
8. $\Diamond\Box(y \doteq \mathrm{gcd}(a, b))$ (persistence)
9. $\Box\Diamond \mathit{terminated}$ (recurrence)

Exercises

Exercises

1. Show that the following formulae are (not) LTL-valid.
   - 1.1 $\Box\varphi \leftrightarrow \Box\Box\varphi$
   - 1.2 $\Diamond\varphi \leftrightarrow \Diamond\Diamond\varphi$
   - 1.3 $\neg\Box\varphi \rightarrow \Box\neg\Box\varphi$
   - 1.4 $\Box(\Box\varphi \rightarrow \psi) \rightarrow \Box(\Box\psi \rightarrow \varphi)$
   - 1.5 $\Box(\Box\varphi \rightarrow \psi) \vee \Box(\Box\psi \rightarrow \varphi)$
   - 1.6 $\Box\Diamond\Box\varphi \rightarrow \Diamond\Box\varphi$
   - 1.7 $\Box\Diamond\varphi \leftrightarrow \Box\Diamond\Box\Diamond\varphi$
2. A *modality* is a sequence of $\neg$, $\Box$ and $\Diamond$, including the empty sequence ϵ. Two modalities σ and τ are *equivalent* if $\sigma\varphi \leftrightarrow \tau\varphi$ is valid.
   - 2.1 Which are the non-equivalent modalities in LTL, and
   - 2.2 what are their relationship (ie. implication-wise)?

References I

[Blackburn et al., 2001] Blackburn, P., de Rijke, M., and Venema, Y. (2001).
Modal Logic.
Cambridge University Press.

[Harel et al., 2000] Harel, D., Kozen, D., and Tiuryn, J. (2000).
Dynamic Logic.
Foundations of Computing. MIT Press.

[Manna and Pnueli, 1992] Manna, Z. and Pnueli, A. (1992).
The temporal logic of reactive and concurrent systems—Specification.
Springer Verlag, New York.