Chapter 2

Logics

Course “Model checking”
Martin Steffen
Autumn 2021

Chapter 2

Learning Targets of Chapter “Logics”.

The chapter gives some basic information about “standard” logics, namely propositional logics and (classical) first-order logics (and maybe more).

Chapter 2

Outline of Chapter “Logics”.

Section

Introduction

Chapter 2 “Logics”
Course “Model checking”
Martin Steffen
Autumn 2021

Logics

What's logic?

General aspects of logics

- truth vs. provability
  - when does a formula *hold*, is *true*, is *satisfied*
  - valid
  - satisfiable
- syntax vs. semantics/models
- model theory vs. proof theory
- connection to computation, calculation, programs

Many different logics

- propositional and first-order logics (classical or otherwise)
- higher-order logics
- modal and temporal logics
- "program" logics
- special purpose logics, domain-specific constraints . . .
- fuzzy, probabilistic . . . .

Section

Propositional logic

Chapter 2 “Logics”
Course “Model checking”
Martin Steffen
Autumn 2021

Syntax (formulas, aka propositions)

$$
\begin{array}{lll}
\varphi ::= & & \text{propositions} \\
& p \mid \top \mid \bot & \text{atomic propositions} \\
\mid & \varphi \wedge \varphi \mid \neg\varphi \mid \varphi \rightarrow \varphi \mid \ldots & \text{compound propositions}
\end{array}
$$

Semantics: the meaning of propositions

- truth values
- σ
- different "notations"
  - $\sigma \models \varphi$
  - evaluate φ, given σ: $[\![\varphi]\!]^\sigma$

“Proof theory”

- decidable, so a “trivial problem” in that sense
- truth tables (brute force)
- one can try to do better, different derivation strategies (resolution, refutation, . . .)
- SAT is NP-complete

Semantics: the meaning of propositions

$$\sigma : P \to \mathbb{B} \ . \tag{1}$$

$$[\![\_]\!]^{-} : (P \to \mathbb{B}) \to \Phi \to \mathbb{B} \ . \tag{2}$$

$$[\![\varphi]\!]^{\sigma} = \top \ , \tag{3}$$

as

$$\sigma \models \varphi \tag{4}$$

Truth table (for $\wedge$)

		$\wedge$
$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$
$\top$	$\bot$	$\bot$
$\top$	$\top$	$\top$

.

Proof theory

model checking

$$\sigma \models^? \varphi$$

satisfiability

$$? \models \varphi$$

validity

$$\models^? \varphi$$

Section

Algebraic and first-order signatures

Chapter 2 “Logics”
Course “Model checking”
Martin Steffen
Autumn 2021

Logics so far: rather sterile

- propositional logic: cannot speak about "things" just regulates "truth" in a logical vacuum, but not "truth of something"
- predicates

Example (even as predicate)

- $even(4)$: true,
- $even(5)$, $even(4+1)$: false,
- $even(x)$: depends on value of *variable* x

Signature

- fixes the "syntactic playground"
- selection of
  - *functional* and
  - *relational*

  symbols, together with "arity" or sort-information

Sorts

- Sort
  - name of a domain (like `Nat`)
  - restricted form of type
- single-sorted vs. multi-sorted case
- single-sorted
  - one sort only
  - "degenerated"
  - *arity* = number of arguments (also for relations)

Terms

- given: signature Σ
- set of variables X (with typical elements $x, y', \ldots$)

$$
\begin{array}{rcll}
t & ::= & x & \text{variable} \\
  & | & f(t_1, \ldots, t_n) & f \text{ of arity } n
\end{array} \tag{5}
$$

- $T_\Sigma(X)$
- terms without variables (from $T_\Sigma(\emptyset)$ or short T_Σ): ground terms

Substutition

- Substitution = *replacement*, namely of variables by terms
- notation $t[s/x]$
- symbol θ

First-order signature (with relations)

- add relational symbols to Σ
- typical elements P, Q
- relation symbols with fixed arity n-ary predicates or relations)
- standard binary symbol: $\doteq$ (equality)

Section

First-order logic

Chapter 2 "Logics"
Course "Model checking"
Martin Steffen
Autumn 2021

Syntax: formulas of first-order logics

- given: first-order signature Σ

$$
\begin{array}{rcll}
\varphi & ::= & P(t, \ldots, t) \mid \top \mid \bot & \text{atomic formulas} \\
 & \mid & \varphi \wedge \varphi \mid \neg\varphi \mid \varphi \rightarrow \varphi \mid \ldots & \text{formulas} \\
 & \mid & \forall x.\varphi \mid \exists x.\varphi &
\end{array}
$$

First-order structures and models

- given Σ
- assume single-sorted case

first-order structure (or model) $M = (A, I)$

- A some domain/set
- interpretation I, respecting arity
  - $[\![f]\!]^I : A^n \to A$
  - $[\![P]\!]^I : A^n$

Giving meaning to variables

Variable assignment

given Σ and model

$$\sigma : X \to A$$

alternative names

- valuation
- state

(E)valuation of terms

- σ "straightforwardly extended/lifted to terms"
- how would one define that (or write it down, or implement)?

Free and bound occurrences of variables

- quantifiers *bind* variables
- *scope*
- other binding, scoping mechanisms
- variables can *occur* free or not (= *bound*) in a formula
- careful with substitution
- how could one define it?

Substitution

- basically:
  - generalize substitution from terms to formulas
  - careful about binders especially don't let substitution lead to variables being "captured" by binders

Example

$$\varphi = \exists x.x + 1 \doteq y \qquad \theta = [x/y]$$

Satisfaction

Definition ($\models$)

$M, \sigma \models \varphi$

- Σ fixed
- in model M and with variable assignment σ formula φ is true (holds)
- M and σ satisfy φ
- minority terminology: M, σ model of φ

Exercises

- substitutions and variable assignments: similar/different?
- there are infinitely many primes
- there is a person with at least 2 neighbors (or exactly)
- every even number can be written as the sum of 2 primes

Proof theory

- how to *infer*, derive, deduce formulas (from others), i.e., how to do a proof?
- mechanical process
- *proof* = deduction (sequence or tree of steps)
- theorem
  - syntactic: derivable formula
  - semantical a formula which holds (in a given model)
- (fo)-theory: set of formulas which are
  - derivable
  - true (in a given model)
- soundness and completeness

Deductions and proof systems

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

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

- $\varphi_1, \dots, \varphi_n$ are premises and ψ conclusion.

A simple form of derivation

Derivation of φ

Sequence of formulae, where each formula is

- an axiom or
- can be obtained by applying an inference rule to formulae earlier in the sequence.

- $\vdash \varphi$
- more general: set of formulas Γ

$$\Gamma \vdash \varphi$$

- proof = derivation
- theorem: derivable formula (= last formula in a proof)

Proof systems and proofs: remarks

- "definitions" from the previous slides: not very formal
- many different "representations" of how to draw conclusions exist, the one sketched on the previous slide
  - works with "sequences"
  - corresponds to the historically oldest "style" of proof systems ("Hilbert-style"), some would say outdated ...
  - 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

A proof system for prop. logic

$$\frac{}{\varphi \to (\psi \to \varphi)}\ \mathrm{Ax}_1$$

$$\frac{}{(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))}\ \mathrm{Ax}_2$$

$$\frac{}{((\varphi \to \bot) \to \bot) \to \varphi}\ \mathrm{DN}$$

$$\frac{\varphi \qquad \varphi \to \psi}{\psi}\ \mathrm{MP}$$

A small derivation

Example

$p \rightarrow p$ is a theorem of the proof system:

$$(p \rightarrow ((p \rightarrow p) \rightarrow p)) \rightarrow ((p \rightarrow (p \rightarrow p)) \rightarrow (p \rightarrow p)) \quad \text{Ax}_2 \quad (1)$$

$$p \rightarrow ((p \rightarrow p) \rightarrow p) \quad \text{Ax}_1 \quad (2)$$

$$(p \rightarrow (p \rightarrow p)) \rightarrow (p \rightarrow p) \quad \text{MP on (1) and (2)} \quad (3)$$

$$p \rightarrow (p \rightarrow p) \quad \text{Ax}_1 \quad (4)$$

$$p \rightarrow p \quad \text{MP on (3) and (4)} \quad (5)$$

Section

Modal logics

Chapter 2 “Logics”
Course “Model checking”
Martin Steffen
Autumn 2021

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".

IN5110 – Verification and specification of parallel systems

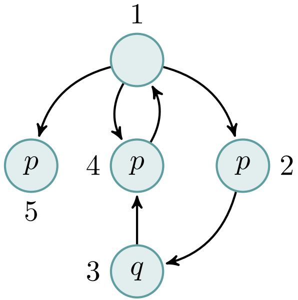
Kripke structures

Definition (Kripke frame and 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 a function of type $V : W \to (P \to \mathbb{B})$, called *valuation*.

isomorphically: $V : W \to 2^P$

Illustration



- $W = \{w_1, w_2, w_3, w_4, w_5\}$
- $R = \{(w_1, w_5), (w_1, w_4), (w_4, w_1), \dots\}$
- $V = [w_1 \mapsto \emptyset, w_2 \mapsto \{p\}, w_3 \mapsto \{q\}, \dots]$

"Box" and "Diamond"

- modal operators $\Box$ and $\Diamond$
- often pronounced "nessessarily" and "possibly"
- mental picture: depends on the kind of logic (temporal, epistemic, deontic ..) and, related to that, the form of the accessibility relation R
- formal definition: see next slide

Satisfaction

Definition (Satisfaction)

A modal formula φ is *true* (or it *holds*) in the world w of a model V, written $V, w \models \varphi$, if:

$$V, w \models p \quad \text{iff} \quad V(w)(p) = \top$$

$$V, w \models \neg\varphi \quad \text{iff} \quad V, w \not\models \varphi$$

$$V, w \models \varphi_1 \vee \varphi_2 \quad \text{iff} \quad V, w \models \varphi_1 \text{ or } V, w \models \varphi_2$$

$$V, w \models \Box\varphi \quad \text{iff} \quad V, w' \models \varphi, \text{ for all } w' \text{ such that } w\ R\ w'$$

$$V, w \models \Diamond\varphi \quad \text{iff} \quad V, w' \models \varphi, \text{ for some } w' \text{ such that } w\ R\ w'$$

Different kinds of relations

R a *binary relation* on W, i.e., $R \subseteq W \times W$

- reflexive
- transitive
- (right) Euclidian
- total
- order relation
- ...

Valid in a frame/for a set of frame

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

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

Example (Samples)

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

Some exercises

Prove the double implications from the previous slide!

Base line axiomatic system ("K")

$$\frac{\varphi \text{ is a propositional tautology}}{\varphi}\ \text{PL}$$

$$\frac{}{\Box(\varphi_1 \to \varphi_2) \to (\Box\varphi_1 \to \Box\varphi_2)}\ \text{K}$$

$$\frac{\varphi \to \psi \qquad \varphi}{\psi}\ \text{MP}$$

$$\frac{\varphi}{\Box\varphi}\ \textsc{Nec}$$

Sample axioms for different accessibility relations

$$\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi) \tag{K}$$

$$\Box\varphi \to \Diamond\varphi \tag{D}$$

$$\Box\varphi \to \varphi \tag{T}$$

$$\Box\varphi \to \Box\Box\varphi \tag{4}$$

$$\neg\Box\varphi \to \Box\neg\Box\varphi \tag{5}$$

$$\Box(\Box\varphi \to \psi) \to \Box(\Box\psi \to \varphi) \tag{3}$$

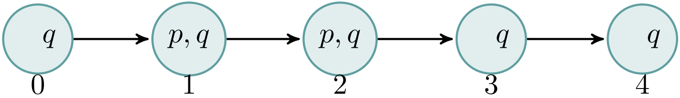
$$\Box(\Box(\varphi \to \Box\varphi) \to \varphi) \to (\Diamond\Box\varphi \to \varphi)) \tag{Dum}$$

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

Some exercises

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

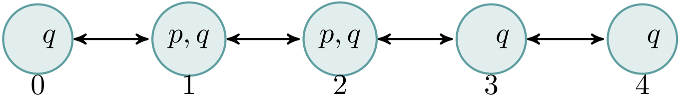


- $M, 0 \models \Diamond\Box p$
- $M, 0 \models \Diamond\Box p \rightarrow p$
- $M, 2 \models \Diamond(q \wedge \neg p) \wedge \Box(q \wedge \neg p)$
- $M, 0 \models q \wedge \Diamond(q \wedge \Diamond(q \wedge \Diamond(q \wedge \Diamond q)))$
- $M \models \Box q$

Exercises (2): bidirectional frames

Bidirectional frame

A frame (W, R) is bidirectional iff $R = R_F + R_P$ s.t. $\forall w, w'(wR_F w' \leftrightarrow w' R_P w)$.



Which of the following statements are correct in M and why?

1. $M, 0 \models \Diamond\Box p$
2. $M, 0 \models \Diamond\Box p \rightarrow p$
3. $M, 2 \models \Diamond(q \wedge \neg p) \wedge \Box(q \wedge \neg p)$
4. $M, 0 \models q \wedge \Diamond(q \wedge \Diamond(q \wedge \Diamond(q \wedge \Diamond q)))$
5. $M \models \Box q$
6. $M \models \Box q \rightarrow \Diamond\Diamond p$

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. $\Box\bot$
2. $\Diamond p \rightarrow \Box p$
3. $p \rightarrow \Box\Diamond p$
4. $\Diamond\Box p \rightarrow \Box\Diamond p$

Section

Dynamic logics

Chapter 2 “Logics”
Course “Model checking”
Martin Steffen
Autumn 2021

Dynamic logics

- different variants
- can be seen as special case of multi-modal logics
- also: generalization of Hoare-logics
- here: PDL on regular programs
- "P" stands for "propositional"

Multi-modal logic

Syntax

The formulas of a multi-modal logic are given by the following grammar.

$$\varphi \quad ::= \quad p \mid \bot \mid \varphi \rightarrow \varphi \mid \Diamond_0 \varphi \mid \Diamond_1 \varphi \mid \ldots \qquad (6)$$

where p is from a set of propositional atoms.

Multi-modal logic

Syntax

The formulas of a multi-modal logic are given by the following grammar.

$$\varphi \quad ::= \quad p \mid \bot \mid \varphi \rightarrow \varphi \mid \Diamond_0 \varphi \mid \Diamond_1 \varphi \mid \ldots \qquad (6)$$

where p is from a set of propositional atoms.

"Kripke frame" $(W, R_0, R_1, \ldots)$, where R_i are relations over W.

Multi-modal logic

Syntax

The formulas of a multi-modal logic are given by the following grammar.

$$\varphi \quad ::= \quad p \mid \bot \mid \varphi \rightarrow \varphi \mid \Diamond_0 \varphi \mid \Diamond_1 \varphi \mid \ldots \tag{6}$$

where p is from a set of propositional atoms.

"Kripke frame" $(W, R_0, R_1, \ldots)$, where R_i are relations over W.

Semantics

"natural" generalization of the "mono"-case

$$M, w \models \Diamond_a \varphi \text{ iff } \exists w' : w R_a w' \text{ and } M, w' \models \varphi \tag{7}$$

analogously for modality $\Diamond_b$ and relation R_b

Remarks

- *multi-modal*: obvious generalization of modal logic
- relations can overlap; i.e., their intersection need not be empty
- *infinitely* many relations and infinitely many modalities possible
- often: labels a, b, ... or similar from a label set or alphabet Σ

$$[a]\varphi \qquad \text{and} \qquad R_a$$

Notation: instead of R_a

$$w_1 \xrightarrow{a} w_2$$

Regular programs

DL

Dynamic logic is a multi-modal logic to talk about programs.

here: regular programs as abstract notation

Regular programs

- *atomic* programs, indivisible, single-step, basic programming constructs
- *sequential* composition
- *nondeterministic choice*
- *iteration*
- `skip` and `fail` programs (denoted $\mathbf{1}$ resp. $\mathbf{0}$)

Regular programs and tests

Definition (Regular programs)

The syntax of regular programs $\alpha, \beta \in \Pi$ is given according to the grammar:

$$\alpha ::= a, \ldots \in \Pi_0 \mid \mathbf{1} \mid \mathbf{0} \mid \alpha \cdot \alpha \mid \alpha + \alpha \mid \alpha^* \mid \varphi? \,. \quad (8)$$

Tests $\varphi?$

Tests can be seen as special atomic programs which may have *logical* structure, but their execution properly terminates in the same state iff the test succeeds (is true), otherwise fails if the test is deemed false in the current state.

Tests

- simple Boolean tests:

$$\varphi ::= \top \mid \bot \mid \varphi \rightarrow \varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi$$

- complex tests: $\varphi?$ where φ is a logical formula in *dynamic logic*

Propositional Dynamic Logic: Syntax

Definition (PDL syntax)

Given a set Π_0 of *atoms* (or atomic regular programs or atomic actions), with typical elements a, b, ... The formulas φ of *propositional dynamic logic* (PDL) over *regular programs* α are given as follows.

$$\begin{array}{lcl} \alpha & ::= & a, \ldots \in \Pi_0 \mid \mathbf{1} \mid \mathbf{0} \mid \alpha \cdot \alpha \mid \alpha + \alpha \mid \alpha^* \mid \varphi? \\ \varphi & ::= & p, q, \ldots \in P \mid \top \mid \bot \mid \varphi \to \varphi \mid [\alpha]\varphi \end{array} \tag{9}$$

where P is a set of atomic propositions.

1. programs, which we denote $\alpha ... \in \Pi$
2. formulas, which we denote $\varphi ... \in \Phi$

Propositional Dynamic Logic (PDL)

based on propositional logic, only

PDL: remarks

- programs α interpreted as a relation R_α (or $\xrightarrow{\alpha}$)

$\Rightarrow$ multi-modal logic.

- $[\alpha]\varphi$ defines many modalities, one modality for each program, each interpreted over the relation defined by the program α.
- ∞ many complex programs $\Rightarrow$ ∞ many relations/modalities
- relations of the basic programs: assumed given.
- operations on/composition of programs are interpreted as operations on relations.
- $[..]\varphi$ is the universal one, with $\langle..\rangle\varphi$ defined as usual.

Intuitive meaning/semantics of $[\alpha]\varphi$

"If program α is started in the current state, then, *if* it terminates, then in its final state, φ holds."

Exercises: “program constructs

Define the following programming constructs in PDL:

$$
\begin{aligned}
\mathbf{skip} &\triangleq \\
\mathbf{fail} &\triangleq \\
\mathbf{if}\ \varphi\ \mathbf{then}\ \alpha\ \mathbf{else}\ \beta &\triangleq \\
\mathbf{if}\ \varphi\ \mathbf{then}\ \alpha &\triangleq \\
\mathbf{case}\ \varphi_1\ \mathbf{then}\ \alpha_1;\ \ldots &\triangleq \\
\mathbf{case}\ \varphi_n\ \mathbf{then}\ \alpha_n & \\
\mathbf{while}\ \varphi\ \mathbf{do}\ \alpha &\triangleq \\
\mathbf{repeat}\ \alpha\ \mathbf{until}\ \varphi &\triangleq
\end{aligned}
$$

*(General **while** loop)*

$$\mathbf{while}\ \varphi_1\ \mathbf{then}\ \alpha_1 \mid \cdots \mid \varphi_n\ \mathbf{then}\ \alpha_n\ \mathbf{od} \triangleq$$

Exercises: “program constructs

Define the following programming constructs in PDL:

$$
\begin{array}{rcl}
\mathbf{skip} & \triangleq & \top? \\
\mathbf{fail} & \triangleq & \bot? \\
\mathbf{if}\ \varphi\ \mathbf{then}\ \alpha\ \mathbf{else}\ \beta & \triangleq & (\varphi? \cdot \alpha) + (\neg\varphi? \cdot \beta) \\
\mathbf{if}\ \varphi\ \mathbf{then}\ \alpha & \triangleq & (\varphi? \cdot \alpha) + (\neg\varphi? \cdot \mathbf{skip}) \\
\mathbf{case}\ \varphi_1\ \mathbf{then}\ \alpha_1;\ \ldots & \triangleq & (\varphi_1? \cdot \alpha_1) + \cdots + (\varphi_n? \cdot \alpha_n) \\
\mathbf{case}\ \varphi_n\ \mathbf{then}\ \alpha_n & & \\
\mathbf{while}\ \varphi\ \mathbf{do}\ \alpha & \triangleq & (\varphi? \cdot \alpha)^* \cdot \neg\varphi? \\
\mathbf{repeat}\ \alpha\ \mathbf{until}\ \varphi & \triangleq & \alpha \cdot (\neg\varphi? \cdot \alpha)^* \cdot \varphi? \\
 & & \\
\textit{(General } \textbf{\textit{while}} \textit{ loop)} & & \\
\mathbf{while}\ \varphi_1\ \mathbf{then}\ \alpha_1 \mid \cdots \mid \varphi_n\ \mathbf{then}\ \alpha_n\ \mathbf{od} & \triangleq & (\varphi_1? \cdot \alpha_1 + \cdots + \varphi_n? \cdot \alpha_n)^* \cdot \\
 & & \cdot(\neg\varphi_1 \wedge \cdots \wedge \varphi_n)?
\end{array}
$$

Making Kripke structures "multi-modal-prepared"

Definition (Labelled Kripke structures)

Assume a set of labels Σ. A *labelled Kripke structure* is a tuple (W, R, Σ) where

$$R = \bigcup_{l \in \Sigma} R_l \tag{10}$$

is the union of the relations indexed by the labels of Σ.

for us (at leat now): The labels of Σ can be thought as programs

- Σ: aka alphabet,
- alternative: $R \subseteq W \times \Sigma \times W$
- labels $l, l_1 \ldots$ but also $a, b, \ldots$ or others
- often: $\xrightarrow{a}$, like $w_1 \xrightarrow{a} w_2$ or $s_1 \xrightarrow{a} s_2$

Regular Kripke structures

- "labels" now have "structure"
- remember: regular program syntax
- interpretation of certain programs/labels fixed,
  - $\mathbf{0}$: failing program
  - $\alpha_1 \cdot \alpha_2$: sequential composition
  - ...
- thus, relations like $\mathbf{0}$, $R_{\alpha_1 \cdot \alpha_2}$, ... must obey side-conditions

Basically

leaving open the interpretation of the "atoms" a, we fix the interpretation/semantics of the constructs of regular programs

Regular Kripke structures

Definition (Regular Kripke structures (no tests))

A *regular Kripke structure* is a Kripke structure labelled as follows. For all atomic programs $a \in \Pi_0$, choose some relation R_a. For the remaining syntactic constructs (except tests), the corresponding relations are defined inductively as follows.

$$
\begin{array}{lcl}
R_{\mathbf{1}} & = & Id \\
R_{\mathbf{0}} & = & \emptyset \\
R_{\alpha_1 \cdot \alpha_2} & = & R_{\alpha_1} \circ R_{\alpha_2} \\
R_{\alpha_1 + \alpha_2} & = & R_{\alpha_1} \cup R_{\alpha_2} \\
R_{\alpha^*} & = & \bigcup_{n \geq 0} R_\alpha^n
\end{array}
$$

Interpreting PDL

Definition (Satisfaction relation)

A PDL formula φ is *true* in the world w of a regular Kripke model $M = (W, \rightarrow, V)$, written $M, w \models \varphi$, if:

$$
\begin{array}{lll}
M, w \models p & \text{iff} & p \in V(w) \text{ for all propositional atoms} \\
M, w \not\models \bot & & \\
M, w \models \top & & \\
M, w \models \varphi_1 \rightarrow \varphi_2 & \text{iff} & \text{whenever } M, w \models \varphi_1 \text{ then also } M, w \models \varphi_2 \\
M, w \models [\alpha]\varphi & \text{iff} & M, w' \models \varphi \text{ for all } w' \text{ such that } w \xrightarrow{\alpha} w' \\
M, w \models \langle\alpha\rangle\varphi & \text{iff} & M, w' \models \varphi \text{ for some } w' \text{ such that } w \xrightarrow{\alpha} w'
\end{array}
\tag{11}
$$

Semantics (cont'd)

- programs and formulas: mutually dependent via modalities and tests
- omitted so far in the regular Kripke structures: tests

$$\varphi?$$

- remember the intuitive meaning of tests

Test programs

Tests

interpreted as subsets of the identity relation.

$$R_{\varphi?} = \{(w, w) \mid w \models \varphi\} \subseteq Id \quad (12)$$

Some special cases:

- $R_{\top?} = Id$
- $R_{\bot?} = \emptyset$
- $R_{(\varphi_1 \wedge \varphi_2)?} = \{(w, w) \mid w \models \varphi_1 \text{ and } w \models \varphi_2\}$

Modalities

testing modal formulas $[\alpha]\varphi$ and $\langle\alpha\rangle\varphi$ is like looking into the future of the program and then deciding whether to take the action.

Axiomatic system of PDL

Take all tautologies of propositional logic (i.e., the axiom system of PL from some earlier lecture) and add
Axioms:

$$[\alpha](\varphi_1 \rightarrow \varphi_2) \rightarrow ([\alpha]\varphi_1 \rightarrow [\alpha]\varphi_2) \quad (1)$$

$$[\alpha](\varphi_1 \land \varphi_2) \leftrightarrow [\alpha]\varphi_1 \land [\alpha]\varphi_2 \quad (2)$$

$$[\alpha + \beta]\varphi \leftrightarrow [\alpha]\varphi \land [\beta]\varphi \quad (3)$$

$$[\alpha \cdot \beta]\varphi \leftrightarrow [\alpha][\beta]\varphi \quad (4)$$

$$[\varphi?]\psi \leftrightarrow \varphi \rightarrow \psi \quad (5)$$

$$\varphi \land [\alpha][\alpha^*]\varphi \leftrightarrow [\alpha^*]\varphi \quad (6)$$

$$\varphi \land [\alpha^*](\varphi \rightarrow [\alpha]\varphi) \rightarrow [\alpha^*]\varphi \quad \text{(IND)}$$

Rules: take the (MP) modus ponens and (G) generalization of modal logic.