advanced logic
2018 03 14
lecture 12

some context: KeY

specification in Java modeling language

is transformed into formula in dynamic logic

and compared with semantics in dynamic logic

more generally: dynamic logic is useful for sequential programming

see wiki for KeY model checker

see home page of KeY

more context

conference

phd position

overview

- propositional dynamic logic: semantics
- alternative semantics
- connection with Hoare logic
- bisimulation
- expressive power

overview

- propositional dynamic logic: semantics
- alternative semantics
- connection with Hoare logic
- bisimulation
- expressive power

Hoare logic

Hoare logic for program correctness

triples consisting of a precondition, a program, and a postcondition

the program is a 'while program'

MLOM 14.2

Hoare logic and PDL

PDL uses regular programs

while programs form a subset of the regular programs

$\{\phi\}\alpha\{\psi\}$ in Hoare logic is written in PDL as $\phi \to [\alpha]\psi$

MLOM 14.3

programs and formulas of PDL

we have programs built from atomic programs,
and possibily dependent on propositions

we have formulas built from atomic formulas,
and possibly dependent on programs

truth of ϕ depends on
truth of strictly smaller formulas and relation of strictly smaller programs

consider for example $\phi = \langle a \rangle p$

accessibility relation of α depends on
relation of strictly smaller programs and truth of strictly smaller formulas

consider for example $\alpha = p?; a$

recal: definition PDL frame

a Prog-frame $\mathcal{F} = (W, \{R_\alpha \mid \alpha \in \text{Prog}\})$ is a PDL-frame if

$R_{\alpha\beta} = R_\alpha; R_\beta$, and

$R_{\alpha\cup\beta} = R_\alpha \cup R_\beta$, and

$R_{\alpha^*} = (R_\alpha)^*$

so if we know all R_a then we know enough!

recall: definition PDL model

A: set of atomic program

Prog: set of regular programs over A

a model $\mathcal{M} = (W, \{R_\alpha \mid \alpha \in \text{Prog}\}, V)$ is a PDL-model if

$(W, \{R_\alpha \mid \alpha \in \text{Prog}\})$ is a PDL-frame, and

$R_{\phi?} = \{(w, w) \mid \mathcal{M}, w \vDash \phi\}$

we can find a PDL-model as extension of an A-model with in addition

$R_{\phi?} = \{(w, w) \mid \mathcal{M}, w \vDash \phi\}$

PDL extension: definition

we can get a PDL model as the extension of a model over labels A

Let $\mathcal{M} = (W, \{R_a \mid a \in A\}, V)$ be an A-model

Its PDL-extension is defined as $\hat{\mathcal{M}} = (W, \{\hat{R}_\alpha \mid \alpha \in \text{Prog}\}, V)$ with

$\hat{R}_a = R_a$

$\hat{R}_{\alpha;\beta} = \hat{R}_\alpha; \hat{R}_\beta$

$\hat{R}_{\alpha\cup\beta} = \hat{R}_\alpha \cup \hat{R}_\beta$

$\hat{R}_{\alpha^*} = (\hat{R}_\alpha)^*$

$\hat{R}_{\phi?} = \{(x, x) \mid \mathcal{M}, x \vDash \phi\}$

some formulas that are valid in all PDL-models

$\langle \alpha; \beta \rangle p \leftrightarrow \langle \alpha \rangle \langle \beta \rangle p$

$\langle \alpha \cup \beta \rangle p \leftrightarrow \langle \alpha \rangle p \vee \langle \beta \rangle p$

$\langle \alpha^* \rangle p \leftrightarrow p \vee \langle \alpha \rangle \langle \alpha^* \rangle p$

$\langle p? \rangle q \leftrightarrow p \wedge q$

$[\alpha^*]p \leftrightarrow p \wedge [\alpha^*](p \to [\alpha]p)$ (induction principle)

overview

- propositional dynamic logic: semantics
- alternative semantics
- connection with Hoare logic
- bisimulation
- expressive power

alternative semantics

the book takes a different approach to semantics of PDL

the definition of $\mathcal{M}, x \models \phi$ is as usual for ϕ a formula from prop1

for diamond formulas: $\mathcal{M}, w \models \langle\alpha\rangle\phi$ if and only if

there exists w' such that $\mathcal{M}, w, w' \models \alpha$ and $\mathcal{M}, w' \models \phi$

what is $\mathcal{M}, w, w' \models \alpha$?

$\mathcal{M}, w, w' \models a$ if $(w, w') \in R_a$ with a an atomic program

extend this to sequential composition, choice, iteration, and test

for example: $\mathcal{M}, w, w' \models ?\phi$ if $w = w'$ and $\mathcal{M}, w \models \phi$

MLOM 14.2

obvious question

we start with a model $\mathcal{M}$ consisting of

W, for every atomic program a a relation R_a, and V

two approaches to defining semantics:

by constructing the PDL-extension of $\mathcal{M}$ and considering $\hat{\mathcal{M}}, x \models \phi$

by considering $\mathcal{M}, x \models \phi$ and $\mathcal{M}, x, x' \models \alpha$

are the two semantics equivalent?

overview

- propositional dynamic logic: semantics
- alternative semantics
- connection with Hoare logic
- bisimulation
- expressive power

if then else

if p then a else b is encoded as $(p?; a) \cup (\neg p?; b)$

question: compute R_γ for $\gamma = (p?; a) \cup (\neg p?; b)$

while

while p do a is encoded as $(p?; a)^*; \neg p?$

question: compute R_γ for $\gamma = (p?; a)^*; \neg p?$

formula $\langle$while p do $a\rangle\, q$ is valid in model $\mathcal{M}$ in state x iff

there exist $n \geq 0$ and there exist $x_0, \ldots, x_n$ such that

$x = x_0$, and

$\mathcal{M}, x_0 \models p$ and $R_a x_0 x_1$

$\vdots$

$\mathcal{M}, x_{n-1} \models p$ and $R_a x_{n-1} x_n$

$\mathcal{M}, x_n \not\models p$ and $\mathcal{M}, x_n \models q$

back to the approach due to Hoare

we encode while-programs as regular programs

we encode $\{\phi\}P\{\psi\}$ as $\phi \rightarrow [Q]\psi$ with Q the translation of P

we show that all rules from Hoare Logic are derivable

so we can encode Hoare logic in propositional dynamic logic

vice versa:

the class of while-progams is the regular programing with union and test
star only used for encoding while and if

overview

- propositional dynamic logic: semantics
- alternative semantics
- connection with Hoare logic
- **bisimulation**
- expressive power

bisimulation for PDL-models

do we need to consider all infinitely many relations?

no, it is sufficient to consider all R_a with a atomic

because PDL-constructors are safe for bisimulation

as a consequence:

if E is a bisimulation between $\mathcal{M}, x$ and $\mathcal{M}', x'$

with $\mathcal{M}$ and $\mathcal{M}'$ A-models,

then E is a bisimulation between $\hat{\mathcal{M}}, x$ and $\hat{\mathcal{M}}', x'$

with $\hat{\mathcal{M}}$ and $\hat{\mathcal{M}}'$ their PDL-extensions

not all operators are safe for bisimulation

intersection is not safe for bisimulation

inverse is not safe for bisimulation

overview

- propositional dynamic logic: semantics
- alternative semantics
- connection with Hoare logic
- bisimulation
- expressive power

example

how can we express the following property:

p is alternatively true and false

along all execution paths of a starting in current state with p true

$p \wedge [a^*]((p \rightarrow [a]\neg p) \wedge (\neg p \rightarrow [a]p))$

or equivalently:

$[(aa)^*]p \wedge [a(aa)^*]\neg p$

without the condition on the start state:

$(p \rightarrow [a]\neg p) \wedge (\neg p \rightarrow [a]p)$

example

which property is expressed by the following formula:

$\langle \text{while } p \text{ do } \alpha \rangle \top$

while p do α terminates if and only if

it is possible by repeated execution of α to reach state with $\neg p$

the formula is equivalent to $\langle \alpha^* \rangle \neg p$