Dynamische Logik

A Simple Programming Language

Logical basis

Typed first-order predicate logic
(Types, variables, terms, formulas, ...)

Assumption for examples

The signature contains a type Nat and appropriate symbols:

- function symbols $0, s, +, *$
  (terms $s(0), s(s(0)), \ldots$ written as $1, 2, \ldots$)
- predicate symbols $\doteq$, $<$, $\leq$, $>$, $\geq$

<u>NOTE:</u> This is a “convenient assumption” not a definition

A Simple Programming Language

Programs

- **Assignments:** $X := t$ — X: variable, t: term
- **Test:** <u>**if**</u> B <u>**then**</u> α <u>**else**</u> β <u>**fi**</u> — B: quant.-free formula, α, β: programs
- **Loop:** <u>**while**</u> B <u>**do**</u> α <u>**od**</u> — B: quantifier-free formula, α: program
- **Composition:** $\alpha;\beta$ — α, β programs

WHILE is computationally complete

WHILE: Examples

Compute the square of X and store it in Y

$$Y := X * X$$

If X is positive then add one else subtract one

$$\underline{\textbf{if}}\ X > 0\ \underline{\textbf{then}}\ X := X + 1\ \underline{\textbf{else}}\ X := X - 1\ \underline{\textbf{fi}}$$

WHILE: Example – Square of a Number

Compute the square of X (the complicated way)

Making use of: $n^2 = 1 + 3 + 5 + \cdots + (2n-1)$

$$\left.\begin{array}{l} I := 0; \\ Y := 0; \\ \underline{\textbf{while}}\ I < X\ \underline{\textbf{do}} \\ \quad Y := Y + 2 * I + 1; \\ \quad I := I + 1 \\ \underline{\textbf{od}} \end{array}\right\}\ \alpha_{\text{square}}$$

WHILE: Example – Multiplication

Russian multiplication

$$\left.\begin{array}{l}
Z := 0; \\
\underline{\textbf{while}}\ \neg(B \doteq 0)\ \underline{\textbf{do}} \\
\quad \underline{\textbf{if}}\ ((B/2)*2 \doteq B)\ \underline{\textbf{then}} \\
\qquad A := 2*A; \\
\qquad B := B/2 \\
\quad \underline{\textbf{else}} \\
\qquad Z := Z + A; \\
\qquad A := 2*A; \\
\qquad B := B/2 \\
\quad \underline{\textbf{fi}} \\
\underline{\textbf{od}}
\end{array}\right\} \alpha_{mult}$$

WHILE: Operational Semantics

Given

A (fixed) first-order structure $\mathcal{A}$ interpreting
the function and predicate symbols in the signature

State

$$s \quad = \quad (\mathcal{A}, \beta) \qquad\qquad \text{where}$$

β a variable assignment (i.e. function interpreting the variables)

WHILE: Operational Semantics

State update

$$s[X/e] \quad = \quad (\mathcal{A}, \beta[X/e])$$

with

$$\beta[X/e](Y) \quad = \quad \begin{cases} e & \text{if } Y = X \\ \beta(Y) & \text{otherwise} \end{cases}$$

WHILE: Operational Semantics

Define the relation $s[\![\alpha]\!]s'$ as follows

- $s[\![X := t]\!]s'$ iff $s' = s[X/s(t)]$
- $s[\![\underline{\textbf{if}}\ B\ \underline{\textbf{then}}\ \alpha\ \underline{\textbf{else}}\ \beta\ \underline{\textbf{fi}}]\!]s'$ iff
  $s \models B$ and $s[\![\alpha]\!]s'$ or $s \models \neg B$ and $s[\![\beta]\!]s'$
- $s[\![\underline{\textbf{while}}\ B\ \underline{\textbf{do}}\ \alpha\ \underline{\textbf{od}}]\!]s'$ iff there are states $s = s_0, \ldots, s_t = s'$ s.t.
  $s_i \models B$ for $0 \leq i \leq t-1$ and $s_t \models \neg B$ and
  $s_0[\![\alpha]\!]s_1,\ s_1[\![\alpha]\!]s_2,\ \ldots,\ s_{t-1}[\![\alpha]\!]s_t$
- $s[\![\alpha;\beta]\!]s'$ iff there is a state s'' such that
  $s[\![\alpha]\!]s''$ and $s''[\![\beta]\!]s'$

$[\![\alpha]\!]$ is a partial function

A Different Approach to WHILE

Programs

- $X := t$ (atomic program)
- $\alpha;\beta$ (sequential composition)
- $\alpha \cup \beta$ (non-deterministic choice)
- α^* (non-deterministic iteration, n times for some $n \geq 0$)
- $F?$ (test)
  remains in initial state if F is true,
  does not terminate if F is false

A Different Approach to WHILE

Restriction to deterministic programs

Non-deterministic program constructors may only be used in

$$\underline{\textbf{if}}\ B\ \underline{\textbf{then}}\ \alpha\ \underline{\textbf{else}}\ \beta\ \underline{\textbf{fi}} \quad \equiv \quad (B?;\alpha) \cup ((\neg B)?;\beta)$$

$$\underline{\textbf{while}}\ B\ \underline{\textbf{do}}\ \alpha\ \underline{\textbf{od}} \quad \equiv \quad (B?;\alpha)^*;(\neg B)?$$

Expressing Program Properties

Logic for expressing properties

Full first-order logic (usually with arithmetic)

Partial correctness assertion (Hoare formula)

$$\{P\}\,\alpha\,\{Q\}$$

Meaning:

If α is started in a state satisfying P and terminates, then its final state satisfies Q

Formally:

$\{P\}\,\alpha\,\{Q\}$ is valid iff

for all states s, s', if $s \models P$ and $s[\![\alpha]\!]s'$, then $s' \models Q$

Expressing Program Properties

Examples

$\{\text{true}\}\ X := X + 1\ \{X > 1\}$

$\{even(X)\}\ X := X + 2\ \{even(X)\}$
where $even(X) \equiv \exists Z\,(X \doteq 2 * Z)$

$\{\text{true}\}\ \alpha_{square}\ \{Y = X * X\}$

An Annotated Program

$Z := 0;$
assert $X \doteq A \wedge Y \doteq B;$
while $\neg(B \doteq 0)$ **do**
    assert $A * B + Z \doteq X * Y;$
    if $((B/2) * 2 \doteq B)$ **then**
        $A := 2 * A;$
        $B := B/2$
    else
        $Z := Z + A;$
        $A := 2 * A;$
        $B := B/2$
    fi
od
assert $B \doteq 0$
assert $Z \doteq X * Y$

Note
X, Y are "external" variables

Dynamic Logic

The idea of dynamic logic

- Annotated programs use formulas within programs
- Dynamic Logic uses programs within formulas
- Instead of "assert F" after program segment α, write: $[\alpha]F$

A multi-modal logic

- the states are the possible worlds
- two modalities $[\alpha]$ and $\langle\alpha\rangle$ for each program α
- state s' is α-reachable from state s iff $s[\![\alpha]\!]s'$

Dynamic Logic: Semantics

Semantics

- $[\alpha]F$ true in a state s iff
  F is true in all states that are α-reachable from s
  (partial correctness)
- $\langle\alpha\rangle F$ true in a state s iff
  F is true in some state that is α-reachable from s
  (total correctness)
- A formula is valid iff it is valid in all states

Dynamic Logic: Examples

Example formulas (validity depends on α,β)

$(\langle\alpha\rangle X \doteq Y) \leftrightarrow (\langle\beta\rangle X \doteq Y)$

$\exists X \langle\alpha\rangle$ true

Valid formulas

$[X := 1] X \doteq 1$

$[$**<u>while</u>** true **<u>do</u>** $X := X$ **<u>od</u>**$]$ false

$\langle\alpha^*\rangle F \rightarrow (F \vee \langle\alpha^*\rangle(\neg F \wedge \langle\alpha\rangle F))$

Multiplication example

$\forall A, B, X, Y, Z (X \doteq A \wedge Y \doteq B \rightarrow [\alpha_{mult}] Z \doteq X * Y)$

Dynamic Logic: More Examples

Hoare formulas

$\{P\}\,\alpha\,\{Q\}$ the same as $P \longrightarrow [\alpha]\,Q$

Duality of the modal operators

$[\alpha]\,P \;\leftrightarrow\; \neg\langle\alpha\rangle\neg P$

Some DL-Tautologies

<u>Assumption:</u> X does not occur in π

$$
\begin{array}{rcll}
(\exists X\ \langle\pi\rangle F) & \leftrightarrow & (\langle\pi\rangle \exists X\, F) & \\
(\forall X\ [\pi] F) & \leftrightarrow & ([\pi] \forall X\, F) & \\
 & & & \\
(\exists X\ [\pi] F) & \rightarrow & ([\pi] \exists X\, F) & \\
([\pi] \exists X\ F) & \rightarrow & (\exists X\, [\pi] F) & \text{provided } \pi \text{ is deterministic} \\
 & & & \\
(\langle\pi\rangle \forall X\ F) & \rightarrow & (\forall X\ \langle\pi\rangle F) & \\
(\forall X\ \langle\pi\rangle F) & \rightarrow & (\langle\pi\rangle \forall X\ F) & \text{provided } \pi \text{ is deterministic} \\
 & & & \\
(\langle\pi\rangle (F \wedge G)) & \rightarrow & ((\langle\pi\rangle F) \wedge \langle\pi\rangle G) & \\
((\langle\pi\rangle F) \wedge \langle\pi\rangle G) & \rightarrow & (\langle\pi\rangle (F \wedge G)) & \text{provided } \pi \text{ is deterministic}
\end{array}
$$

A Sequent Calculus for Dynamic Logic

Sequent

$$\Gamma \rightarrow \Delta$$

Meaning

$\bigwedge \Gamma$ logically implies $\bigvee \Delta$
(for all variable assignments, i.e.,
free variables in the sequent are implicitly universally quantified)

Sequent Rules

Form of sequent rules

$$\frac{\Gamma_1 \rightarrow \Delta_1}{\Gamma_2 \rightarrow \Delta_2} \qquad \text{or} \qquad \frac{\Gamma_1 \rightarrow \Delta_1 \quad \Gamma_1' \rightarrow \Delta_1'}{\Gamma_2 \rightarrow \Delta_2}$$

(rules can also have more than two premisses)

Meaning

The conclusion is true in a state
whenever all premisses are true in that state

<u>In particular:</u>
The conclusion is valid whenever all premisses are valid

Sequent Calculus for First-order Logic

Axioms

$$\frac{}{F, \Gamma \; \Rightarrow \; F, \Delta} \qquad \frac{}{\text{false}, \Gamma \; \Rightarrow \; \Delta} \qquad \frac{}{\Gamma \; \Rightarrow \; \text{true}, \Delta}$$

Negation

$$\frac{\Gamma \; \Rightarrow \; F, \Delta}{\Gamma, \neg F \; \Rightarrow \; \Delta} \qquad \frac{\Gamma, F \; \Rightarrow \; \Delta}{\Gamma \; \Rightarrow \; \neg F, \Delta}$$

Implication

$$\frac{\Gamma \; \Rightarrow \; F, \Delta \qquad \Gamma, G \; \Rightarrow \; \Delta}{\Gamma, F \to G \; \Rightarrow \; \Delta} \qquad \frac{\Gamma, F \; \Rightarrow \; G, \Delta}{\Gamma \; \Rightarrow \; F \to G, \Delta}$$

Sequent Calculus for First-order Logic

Conjunction

$$\frac{\Gamma, F, G \rightarrow \Delta}{\Gamma, F \wedge G \rightarrow \Delta} \qquad \frac{\Gamma \rightarrow F, \Delta \qquad \Gamma \rightarrow G, \Delta}{\Gamma \rightarrow F \wedge G, \Delta}$$

Disjunction

$$\frac{\Gamma, F \rightarrow \Delta \qquad \Gamma, G \rightarrow \Delta}{\Gamma, F \vee G \rightarrow \Delta} \qquad \frac{\Gamma \rightarrow F, G, \Delta}{\Gamma \rightarrow F \vee G, \Delta}$$

Sequent Calculus for First-order Logic

Universal quantification

$$\frac{\Gamma,\ \forall X\,F,\ F\{X \leftarrow t\} \ \rightarrow\ \Delta}{\Gamma,\ \forall X\,F \ \rightarrow\ \Delta}$$

t an arbitrary term,

$\{X \leftarrow t\}$ admissible for F

$$\frac{\Gamma \ \rightarrow\ F\{X \leftarrow Z\},\ \Delta}{\Gamma \ \rightarrow\ \forall X\,F,\ \Delta}$$

Z a new variable

Existential quantification

$$\frac{\Gamma \ \rightarrow\ \exists X\,F,\ F\{X \leftarrow t\},\ \Delta}{\Gamma \ \rightarrow\ \exists X\,F,\ \Delta}$$

t an arbitrary term,

$\{X \leftarrow t\}$ admissible for F

$$\frac{\Gamma,\ F\{X \leftarrow Z\} \ \rightarrow\ \Delta}{\Gamma,\ \exists X\,F \ \rightarrow\ \Delta}$$

Z a new variable

Example Proof

$$
\begin{array}{llcl}
 & & \text{axiom} & \\
6 & p(V),\ p(U) & \rightarrow & p(U),\ \forall Y\, p(Y) \\
 & & \Big| & \text{impl-right} \\
5 & p(V) & \rightarrow & p(U),\ p(U) \rightarrow \forall Y\, p(Y) \\
 & & \Big| & \text{ex-right} \\
4 & p(V) & \rightarrow & p(U),\ \exists X (p(X) \rightarrow \forall Y\, p(Y)) \\
 & & \Big| & \text{all-right} \\
3 & p(V) & \rightarrow & \forall Y\, p(Y),\ \exists X\, (p(X) \rightarrow \forall Y\, p(Y)) \\
 & & \Big| & \text{impl-right} \\
2 & & \rightarrow & p(V) \rightarrow \forall Y\, p(Y),\ \exists X\, (p(X) \rightarrow \forall Y\, p(Y)) \\
 & & \Big| & \text{ex-right} \\
1 & & \rightarrow & \exists X\, (p(X) \rightarrow \forall Y\, p(Y))
\end{array}
$$

Admissibility of Substitutions

Motivation

We want to have that

$$\begin{aligned} \forall X\, F &\rightarrow F\sigma \\ F\sigma &\rightarrow \exists X\, F \end{aligned}$$

is valid for all formulas F and substitutions σ

Admissibility of Substitutions

Definition

A substitution

$$\{X \leftarrow t\}$$

is admissible for a formula F iff
there is no variable Y such that

- Y occurs in t
- there is a quantification $\forall Y$ or $\exists Y$ in F
- there is a free occurrence of X in the scope of that quantification

Sequent Calculus for Dynamic Logic

Cut rule

$$\frac{\Gamma \rightarrow F, \Delta \qquad F, \Gamma \rightarrow \Delta}{\Gamma \rightarrow \Delta}$$

Equality rules

$$\frac{}{\Gamma \rightarrow t \doteq t, \Delta}$$

$$\frac{s \doteq t, \Gamma\{s \leftarrow t\} \rightarrow \Delta\{s \leftarrow t\}}{s \doteq t, \Gamma \rightarrow \Delta} \qquad \frac{t \doteq s, \Gamma\{s \leftarrow t\} \rightarrow \Delta\{s \leftarrow t\}}{t \doteq s, \Gamma \rightarrow \Delta}$$

Sequent Calculus for Dynamic Logic

Oracle for first-order logic

$$\frac{}{\Gamma \Rightarrow \Delta}$$

if no programs occur in Γ, Δ and $\mathcal{A} \models \bigwedge \Gamma \to \bigvee \Delta$

Only of theoretical use! Not computable!