A Sequent Calculus for Dynamic Logic

Composition rule

$$\frac{\Gamma \;\rightarrow\; [\alpha][\beta]F,\ \Delta}{\Gamma \;\rightarrow\; [\alpha;\beta]F,\ \Delta}$$

Assignment rule

$$\frac{\Gamma\{X \leftarrow X'\},\ X \doteq t\{X \leftarrow X'\} \;\rightarrow\; F,\ \Delta\{X \leftarrow X'\}}{\Gamma \;\rightarrow\; [X := t]F,\ \Delta}$$

X' a new variable

Example:

$$\frac{even(X'),\ X \doteq X' + 2 \;\rightarrow\; even(X)}{even(X) \;\rightarrow\; [X := X+2]\,even(X)}$$

A Sequent Calculus for Dynamic Logic

Conditional rule

$$\frac{\Gamma, B \;\rightarrow\; [\alpha]F,\; \Delta \qquad\qquad \Gamma, \neg B \;\rightarrow\; [\beta]F,\; \Delta}{\Gamma \;\rightarrow\; [\underline{\mathbf{if}}\; B\; \underline{\mathbf{then}}\; \alpha\; \underline{\mathbf{else}}\; \beta\; \underline{\mathbf{fi}}]F,\; \Delta}$$

Reasoning about Loops

To prove

$$[\underline{\textbf{while}}\ B\ \underline{\textbf{do}}\ body\ \underline{\textbf{od}}]\,F$$

find an (arbitrary) formula Inv such that

1. Inv is true before execution of the loop
2. $Inv \wedge B \rightarrow [body]Inv$ is true
3. $Inv \wedge \neg B \rightarrow F$ is true

Note

Inv is a loop invariant

Sequent Calculus for Dynamic Logic

Loop rule

$$\frac{\Gamma \rightarrow Inv, \Delta \qquad Inv, B \rightarrow [\alpha]\, Inv \qquad Inv, \neg B \rightarrow F}{\Gamma \rightarrow [\underline{\textbf{while}}\; B\; \underline{\textbf{do}}\; \alpha\; \underline{\textbf{od}}]\, F, \Delta}$$

Example

Invariant Inv: $I \leq X \wedge Y \doteq I * I$

$$I \doteq 0,\ Y \doteq 0 \rightarrow Inv \qquad Inv,\ B \rightarrow [\alpha]\, Inv \qquad Inv,\ \neg B \rightarrow Y \doteq X * X$$

$$\begin{array}{rcl} I \doteq 0,\ Y \doteq 0 & \rightarrow & [\underline{\textbf{while}}\ B\ \underline{\textbf{do}}\ \alpha\ \underline{\textbf{od}}]\, Y \doteq X * X \\ I \doteq 0 & \rightarrow & [Y := 0]\, [\underline{\textbf{while}}\ B\ \underline{\textbf{do}}\ \alpha\ \underline{\textbf{od}}]\, Y \doteq X * X \\ & \rightarrow & [I := 0]\, [Y := 0]\, [\underline{\textbf{while}}\ B\ \underline{\textbf{do}}\ \alpha\ \underline{\textbf{od}}]\, Y \doteq X * X \\ & \rightarrow & [I := 0;\ Y := 0;\ \underline{\textbf{while}}\ B\ \underline{\textbf{do}}\ \alpha\ \underline{\textbf{od}}]\, Y \doteq X * X \\ & \rightarrow & [\alpha_{square}]\, Y \doteq X * X \end{array}$$

B: $\quad I < X$

α: $\quad Y := Y + 2 * I + 1;\ I := I + 1$

Example

Left branch (pre-condition implies invariant)

$$I \doteq 0,\ Y \doteq 0 \ \rightarrow\ 0 \leq X \qquad\qquad I \doteq 0,\ Y \doteq 0 \ \rightarrow\ Y \doteq 0 * 0$$

$$I \doteq 0,\ Y \doteq 0 \quad \rightarrow \quad 0 \leq X \ \wedge\ Y \doteq 0 * 0$$

$$I \doteq 0,\ Y \doteq 0 \quad \rightarrow \quad I \leq X \ \wedge\ Y \doteq I * I$$

Example

Middle branch (invariant is indeed invariant)

$$I' < X,\ I \doteq I' + 1 \ \rightarrow\ I < X$$

$$Y' \doteq I' * I',\ Y \doteq Y' + 2 * I' + 1,\ I \doteq I' + 1 \ \rightarrow\ Y \doteq I * I$$

$$I' \leq X,\ Y' \doteq I' * I',\ I' < X,\ Y \doteq Y' + 2 * I' + 1,\ I \doteq I' + 1 \ \rightarrow\ Inv$$

$$I \leq X,\ Y' \doteq I * I,\ I < X,\ Y \doteq Y' + 2 * I + 1 \ \rightarrow\ [I := I + 1]\, Inv$$

$$I \leq X,\ Y \doteq I * I,\ I < X \ \rightarrow\ [Y := Y + 2 * I + 1]\, [I := I + 1]\, Inv$$

$$I \leq X,\ Y \doteq I * I,\ I < X \ \rightarrow\ [Y := Y + 2 * I + 1;\ I := I + 1]\, Inv$$

$$Inv,\ B \ \rightarrow\ [\alpha]\, Inv$$

Example

Right branch
(invariant and negated loop condition imply post-condition)

$$I \leq X,\ Y \doteq I * I,\ \neg(I < X),\ I \doteq X \quad \rightarrow \quad Y \doteq I * I$$

$$I \leq X,\ Y \doteq I * I,\ \neg(I < X),\ I \doteq X \quad \rightarrow \quad Y \doteq X * X$$

$$I \leq X,\ Y \doteq I * I,\ \neg(I < X) \quad \rightarrow \quad Y \doteq X * X$$

$$Inv \wedge \neg B \quad \rightarrow \quad Q$$

Example II: Multiplication

Invariant $\quad Inv\colon A * B + Z \doteq X * Y$

$$X \doteq A,\ Y \doteq B,\ Z \doteq 0 \quad \to \quad Inv$$

$$Inv,\ \neg B \doteq 0 \quad \to \quad [\alpha_{body}]\, Inv$$

$$Inv,\ B \doteq 0 \quad \to \quad Z \doteq X$$

$$X \doteq A,\ Y \doteq B,\ Z \doteq 0 \quad \to \quad [\alpha_{while}]\, Z \doteq X * Y$$

$$X \doteq A,\ Y \doteq B \quad \to \quad [Z := 0;\ \alpha_{while}]\, Z \doteq X * Y$$

$$X \doteq A,\ Y \doteq B \quad \to \quad [\alpha_{mult}]\, Z \doteq X * Y$$

Example II: Multiplication

Left branch (pre-condition implies invariant)

$$X \doteq A,\ Y \doteq B,\ Z \doteq 0 \quad \rightarrow \quad A * B + Z \doteq X * Y$$

Middle branch (invariant is indeed invariant)

$$A * B + Z \doteq X * Y,\ \neg B \doteq 0 \quad \rightarrow \quad [\alpha_{body}]\, A * B + Z \doteq X * Y$$

Right branch
(invariant and negated loop condition imply post-condition)

$$A * B + Z \doteq X * Y,\ B \doteq 0 \quad \rightarrow \quad Z \doteq X * Y$$

Induction Rule

Purpose

- Needed to prove first-order theorems on natural numbers (oracle not available in practice)
- Handling loops in $\langle\cdot\rangle$ modality

$$\frac{\begin{array}{c}\Gamma \ \rightarrow\ F\{N \leftarrow 0\},\, \Delta \\ \Gamma, F \ \rightarrow\ F\{N \leftarrow N+1\},\, \Delta \\ \Gamma, \forall N\, F \ \rightarrow\ \Delta\end{array}}{\Gamma \ \rightarrow\ \Delta}$$

N not occurring in Γ, Δ

N not occurring in any program in F

Induction Rule: Example

$$
\frac{
\begin{array}{rcl}
 & \rightarrow & even(2*0),\ even(2*3) \\
even(2*N) & \rightarrow & even(2*(N+1)),\ even(2*3) \\
\forall N\,(even(2*N)) & \rightarrow & even(2*3)
\end{array}
}{
\rightarrow \quad even(2*3)
}
$$

Loop Unwind Rule

Rule

$$\frac{\Gamma, \neg B \;\rightarrow\; F, \Delta \qquad\qquad \Gamma, B \;\rightarrow\; \langle\alpha\rangle\langle\underline{\textbf{while}}\ B\ \underline{\textbf{do}}\ \alpha\ \underline{\textbf{od}}\rangle F, \Delta}{\Gamma \;\rightarrow\; \langle\underline{\textbf{while}}\ B\ \underline{\textbf{do}}\ \alpha\ \underline{\textbf{od}}\rangle F, \Delta}$$

Note

Only useful

- in connection with induction rule, or
- if number of loop iterations has a (small) known upper bound

Loop Unwind / Induction: Example

Proof goal

$$\rightarrow \quad \langle \underline{\textbf{while}}\ I > 0\ \underline{\textbf{do}}\ I := I - 1\ \underline{\textbf{od}} \rangle\, I \doteq 0$$

Induction hypothesis

$$F(N) \quad \equiv \quad \forall I\, (I \leq N \;\rightarrow\; \langle \underline{\textbf{while}}\ I > 0\ \underline{\textbf{do}}\ I := I - 1\ \underline{\textbf{od}} \rangle\, I \doteq 0)$$

Admissibility of Substitutions Revisited

Problem

Previous definition of admissibility
is not sufficient if formulas contain programs

Example

$$
\begin{array}{rcll}
F & \equiv & J \doteq K \rightarrow [I := 0]\,(J \doteq K) & \text{valid} \\
F\{I \leftarrow J\} & \equiv & J \doteq K \rightarrow [J := 0]\,(J \doteq K) & \text{not valid} \\
F\{J \leftarrow I\} & \equiv & I \doteq K \rightarrow [I := 0]\,(I \doteq K) & \text{not valid} \\
F\{I \leftarrow 1\} & \equiv & J \doteq K \rightarrow [1 := 0]\,(J \doteq K) & \text{not a formula}
\end{array}
$$

Admissibility of Substitutions Revisited

Revised definition

A substitution $\{X \leftarrow t\}$ is admissible for a formula F iff

1. $t = X$, or
2. t is a variable not occurring in F, or
3. there is no variable Y in t such that a free occurrence of X in F is in the scope of
   (a) a quantification $\forall Y$ or $\exists Y$, or
   (b) a modality containing an assignment of the form $Y := s$