Plan

CÔTE D'AZUR
master BBC
Jean-Paul Comet
R. Thomas
CTL
Software Engineering
General Schema
Hoare Logic
Hoare Logic
Examples

Standard Hoare logic : *swap(x,y)*

$\{(x = x_0) \wedge (y = y_0)\}$

aux := x ; $\leftarrow \{(x = x_0) \wedge (y = y_0)\}$

x := y ; $\leftarrow \{(aux = x_0) \wedge (y = y_0)\}$

y := aux $\leftarrow \{(aux = x_0) \wedge (x = y_0)\}$

$\{(y = x_0) \wedge (x = y_0)\}$

$$\frac{}{\{Q[v \leftarrow expr]\}\ \ v := expr\ \ \{Q\}}\ :=$$

$$\frac{\{P\}p_1\{Q'\} \qquad \{Q'\}p_2\{Q\}}{\{P\}p_1;\, p_2\{Q\}}\ ;$$

$$\frac{\dfrac{\overline{\{Q_3\}a_1\{Q_2\}}\ := \qquad \overline{\{Q_2\}a_2\{Q_1\}}\ :=}{\{P\}a_1;\, a_2\{Q_1\}}\ ; \qquad \overline{\{Q_1\}a_3\{Q\}}\ :=}{\{P\}a_1;\, a_2;\, a_3\{Q\}}\ ;$$

CÔTE D'AZUR
master BBC
Jean-Paul Comet
R. Thomas
CTL
Software Engineering
General Schema
Hoare Logic
Hoare Logic
Examples

Standard Hoare logic : *abs(x)*

$\{(x = x_0)\}$ $\leftarrow$ $\left\{ \left(\begin{array}{l} (x<0) \wedge \\ (-x \geq 0) \wedge \\ ((-x)^2 = x_0^2) \end{array} \right) \vee \left(\begin{array}{l} (x \geq 0) \wedge \\ (x \geq 0) \wedge \\ (x^2 = x_0^2) \end{array} \right) \right\}$

if (x¡0) :

r := −x

else :

r := x

$\{(r \geq 0) \wedge (r^2 = x_0^2)\}$

$$\frac{\{Q_1\}p_1\{Q\} \qquad \{Q_2\}p_2\{Q\}}{\{(e \wedge Q_1) \vee (\neg e \wedge Q_2)\}\ \texttt{if}\ e\ \texttt{then}\ p_1\ \texttt{else}\ p_2\ \{Q\}}\ \textbf{if}$$

Also :

While loop : $\dfrac{\{e \wedge I\}p\{I\} \qquad (\neg e \wedge I) \Longrightarrow Q}{\{I\}while\ e\ with\ I\ do\ p\{Q\}}$

Empty program : $\dfrac{P \Longrightarrow Q}{\{P\}\varepsilon\{Q\}}$ use sparingly : loses *weakest* precondition !

CÔTE D'AZUR
master BBC
Jean-Paul Comet
R. Thomas
CTL
Software Engineering
General Schema
Hoare Logic
Hoare Logic
Examples

Standard Hoare logic : Euclidian division

$\{(p = p_0) \wedge (q = q_0)\}$

r := 0 ;

while $p \geq q$ with $\{q{=}q_0 \wedge p{\geq}0 \wedge p_0{=}p + r.q_0\}$:

p := p − q

r := r + 1

$\{0 \leq p_0 - r.q_0 < q_0\}$

$$\frac{}{\{Q[v \leftarrow expr]\}\ \ v := expr\ \ \{Q\}}\ :=$$

$$\frac{\{P\}p_1\{Q'\} \qquad \{Q'\}p_2\{Q\}}{\{P\}p_1;\, p_2\{Q\}}\ ;$$

$$\frac{\{e \wedge I\}p\{I\} \qquad \neg e \wedge I \Rightarrow Q}{\{I\}\ \text{while}\ e\ \text{with}\ I\ \text{do}\ p\ \{Q\}}\ loop$$

Assertion language (Pre/Post)

Terms : v gene | $n \in \mathbb{N}$ | $K_{v,\{\dots\}}$ parameter symbols | $+$ | $-$

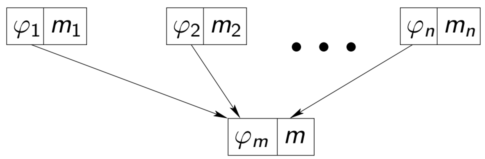
atoms : $t \geqslant t'$ | $t < t'$ | $t = t'$ | ...

Connectives : $\neg \mid \wedge \mid \vee \mid \Longrightarrow$

Example :

$$(a \leqslant 3 \ \wedge \ d+1 < K_{d,\{m,c\}}) \ \vee \ (K_{d,\{c\}} < K_{d,\{m,c\}} \ \wedge \ c \geqslant 3)$$

From multiplexes to assertions : flattening

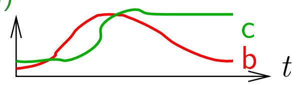


$$\overline{\varphi_m} \ \equiv \ \varphi_m[m_i \leftarrow \varphi_i] \text{ for all } i \text{ and recursively}$$

Trace specifications

- $x+ \mid x- \mid x := n \mid assert(\varphi)$
- $p_1; p_2; \cdots; p_n$
- *if* φ *then* p_1 *else* p_2
- *while* φ *with* ψ *do* p
- $\forall(p_1, p_2, \cdots, p_n)$
- $\exists(p_1, p_2, \cdots, p_n)$

Examples :



- $b+; c+; b-$
- $\exists(b+, b-, c+, c-, \varepsilon)$
- *while* $(b < 2)$ *with* $(c > 0)$
  do $\exists(b+, b-, \forall((c-; a-), c+))$ *od*;
  $b-$

Assertions that formalize Thomas'framework

ω is the set of ressources of v :

$$\Phi_v^\omega \ \equiv \ \Big(\bigwedge_{m \in \omega} \overline{\varphi_m}\Big) \wedge \Big(\bigwedge_{m \in G^{-1}(v)\setminus\omega} \neg\overline{\varphi_m}\Big)$$

v can increase :

$$\Phi_v^+ \ \equiv \ \bigwedge_{\omega \subset G^{-1}(v)} (\Phi_v^\omega \Longrightarrow K_{v,\omega} > v)$$

v can decrease :

$$\Phi_v^- \ \equiv \ \bigwedge_{\omega \subset G^{-1}(v)} (\Phi_v^\omega \Longrightarrow K_{v,\omega} < v)$$

Genetic, a la Hoare, inference rules

Incrementation rule : $\dfrac{}{\{\ \Phi_v^+ \ \wedge \ Q[v \leftarrow v+1]\ \}\ v+\ \{Q\}}$

Decrementation rule : $\dfrac{}{\{\ \Phi_v^- \ \wedge \ Q[v \leftarrow v-1]\ \}\ v-\ \{Q\}}$

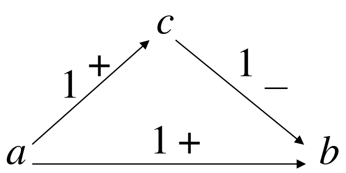
Assertion rule : $\dfrac{}{\{\ \varphi \wedge Q\ \}\ assert(\varphi)\ \{Q\}}$

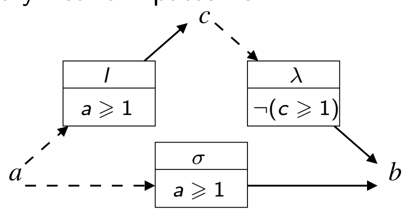
Universal quantifier rule : $\dfrac{\{P_1\}p_1\{Q\} \qquad \{P_2\}p_2\{Q\}}{\{P_1 \wedge P_2\}\ \forall(p_1,p_2)\ \{Q\}}$

Existential quantifier rule : $\dfrac{\{P_1\}p_1\{Q\} \qquad \{P_2\}p_2\{Q\}}{\{P_1 \vee P_2\}\ \exists(p_1,p_2)\ \{Q\}}$

Example : Feedforward “loop”

Uri Alon most frequent regulatory network patterns





Behaviour of b after switching a from off to on ?

Simple off→on→off behaviour of b with the help of c :

$$\{(a=1 \ \wedge \ b=0 \ \wedge \ c=0)\} \ b+ \ ; \ c+ \ ; \ b- \ \ \{b=0\}$$

possible if and only if :

$K_{b,\{\sigma,\lambda\}}=1 \ \wedge \ K_{c,\{l\}}=1 \ \wedge \ K_{b,\{\sigma\}}=0$

Feedforward example (continued)

off→on→off behaviour of b without the help of c :

$$\{(a=1 \ \wedge \ b=0 \ \wedge \ c=0)\} \ b+ \ ; \ b- \ \ \{b=0\}$$

$$\left\{\begin{array}{l} b=0 \\ ((c \geqslant 1)\wedge(a<1)) \Longrightarrow ((K_b=1)\wedge(K_b=0)) \\ ((c \geqslant 1)\wedge(a \geqslant 1)) \Longrightarrow ((K_{b,\sigma}=1)\wedge(K_{b,\sigma}=0)) \\ ((c<1)\wedge(a<1)) \Longrightarrow ((K_{b,\lambda}=1)\wedge(K_{b,\lambda}=0)) \\ ((c<1)\wedge(a \geqslant 1)) \Longrightarrow ((K_{b,\sigma\lambda}=1)\wedge(K_{b,\sigma\lambda}=0)) \end{array}\right\} \ \textbf{not}$$

satisfiable !

Feedforward example (continued)

Although $b+; c+; b-$ is possible, if c becomes “on” before b, then b will never be able to get “on”

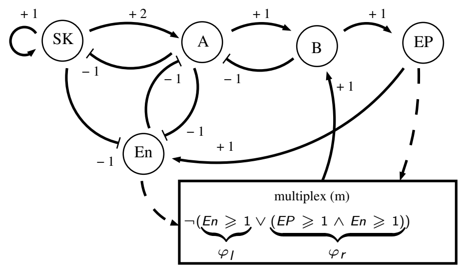
Proof by refutation :

$$\left\{\begin{array}{l} a=1 \ \wedge \ b=0 \ \wedge \ c=1 \ \wedge \\ K_{b,\sigma\lambda}=1 \ \wedge \ K_{c,l}=1 \ \wedge \ K_{b,\sigma}=0 \end{array}\right\}$$
$$\textit{while } b<1 \textit{ with } I \textit{ do } \exists(b+, b-, c+, c-)$$
$$\{ \ b=1 \ \}$$

the triple is inconsistent, whatever the loop invariant I !

Cell cycle in mammals

- A 22 gene model reduced to 5 variables using multiplexes



SK = Cyclin E/Cdk2, Cyclin H/Cdk7
A = Cyclin A/Cdk1
B = Cyclin B/Cdk1
En = APC^{G1}, CKI (p21, p27), Wee1
EP = APC^{M}, Phosphatases

- 48 states, 26 parameters, 339 738 624 possible valuations, 12 trace specifications and a few temporal properties

Cell cycle in mammals (continued)

- 13 parameters have been entirely identified (50%) and only 8192 valuations remain possible according to the generated constraints (0.002%)
- Additional reachability constraints (e.g. endoreplication and quiescent phase) have been necessary, on an extended *hybrid* extension of the Thomas' framework, to identify (almost) all parameters
- This initial Hoare logic identification step was crucial : it gave us the sign of the derivatives in all the (reachable) states

Correctness, Completeness and Decidability

- If there is a proof tree for $\{P\}p\{Q\}$ then for each initial state satisfying P, there are traces in the regulatory network that realize the trace specification p, and for all of them, if terminating, they satisfy Q at the end.
- If for each initial state satisfying P there are traces that realize p in the regulatory network and if they all satisfy Q at the end, then there exists a proof tree for $\{P\}p\{Q\}$.
- There is a simple algorithm to compute, for each Q, the minimal loop invariant I such that $\{I\}$*while e with I do* $p\{Q\}$.
  (However well chosen slightly non minimal invariants can considerably simplify the proof tree. . .)

Take Home Messages

Make explicit the hypotheses that motivate the biologist

A far as possible formalize them to get a computer aided approach
Behavioural *properties* are as much important as *models*
Mathematical models are not reality : let's use this freedom !
(several views of a same biological object)
Modelling is significant only with respect to the considered experimental *reachability* and *observability* (for refutability)

Formal proofs can suggest wet experiments
"Kleenex" models help understanding main behaviours
Specialized qualitative approaches can make complex models simple
The more detailed models are not the more comprehensible ones