Dagstuhl, July 8–10, 2002:

MPC, Dagstuhl July, 2002 — i

Eternity Variables
to Simulate Specifications

Wim H. Hesselink

Email: `wim@cs.rug.nl`

Aim of the talk — ii

How to prove that program K simulates specification L?

Programs are (executable) specifications

Four kinds of simulations:
functional, forward, backward, eternity

Theorem. Every simulation $F : K \rightarrowtriangle L$ that preserves quiescence, is provable by means of these special ones

A variation of theory of Abadi and Lamport (1991)

`www.cs.rug.nl/~wim/pub/whh275.pdf`

Overview — iii

1. Temporal Logic of Actions
2. Refinement mappings and simulations between specs
3. Forward and backward simulations
4. Eternity variables
5. Preservation of quiescence and Completeness

1. Temporal Logic of Actions — iv

A *specification* is a 4-tuple K :

$X = \mathit{states}(K)$: the state space
$Y = \mathit{init}(K) \subseteq X$: set of initial states
$N = \mathit{step}(K) \subseteq X^2$: next-state relation
$P = \mathit{prop}(K) \subseteq X^\omega$: (fairness) property

An *execution* is a list xs of states with $(xs_i, xs_{i+1}) \in N$ for all i

- *initial* iff $xs_0 \in Y$.
- a *behaviour* iff infinite and belongs to P

$\mathit{Beh}(K) \;=\; [\![Y]\!] \cap \Box[\![N]\!] \cap P$

To allow stuttering N is reflexive (and P is a "property").

Example — v

Specification $L0$

```
var k: Int := 0 ;
do  k = 0  ->  choose k in Int ;
[]  true   ->  k := k - 2 ;
od ;
prop: infinitely often k = 0.
```

$\mathit{states}(L0) = \mathtt{Int}$
$\mathit{init}(L0) = \{0\}$
$\mathit{prop}(L0) = \Box\Diamond[\![\mathtt{k} = 0]\!]$

$\mathit{step}(L0) \;=\; \{(k, k') \mid k = 0 \;\vee\; k' = k - 2 \;\vee\; k' = k\}$

Every state is reachable

The occurring states have `k` natural and even

2. Refinement Mappings — vi

When does spec K implement spec L?

K : the concrete program
L : the abstract program

A refinement mapping from K to L is a function $f : \mathit{states}(K) \to \mathit{states}(L)$ such that

$$x \in \mathit{init}(K) \;\Rightarrow\; f(x) \in \mathit{init}(L)$$
$$(x, x') \in \mathit{step}(K) \;\Rightarrow\; (f(x), f(x')) \in \mathit{step}(L)$$
$$xs \in \mathit{Beh}(K) \;\Rightarrow\; f^\omega(xs) \in \mathit{Beh}(L)$$

Example $K(m)$ for $m > 1$ — vii

var $\mathtt{j} : \mathit{Nat} := 0$;
do *true* $\to$ $\mathtt{j} := (\mathtt{j} + 1)$ **mod** m **od** ;
prop: `j` changes infinitely often.

$\mathit{states}(K(m)) = \mathbb{N}$
$\mathit{init}(K(m)) = \{0\}$
$\mathit{prop}(K(m)) = \Box\Diamond[\![\neq]\!]$

$(j, j') \in \mathit{step}(K(m)) \;\equiv\; j' \in \{j, (j+1) \;\mathbf{mod}\; m\}$

A refinement mapping f from $K(21)$ to $K(14)$?
Take $f : \mathbb{N} \to \mathbb{N}$ with $f(j) = \min(j, 13)$

The abstract behaviour stutters
whenever the concrete behaviour proceeds
from 13 to 20

---

vii.1. – Refinement mappings are not enough.
We sometimes need simulations

Simulations (new) — viii

Spec K *simulates* spec L
via relation $F \subseteq \mathit{states}(K) \times \mathit{states}(L)$
(notation $F : K \mathrel{-\!\!\triangleright} L$)
$\equiv$
for every $xs \in \mathit{Beh}(K)$
there is $ys \in \mathit{Beh}(L)$
with $(xs_n, ys_n) \in F$ for all n

Every refinement mapping
$f : \mathit{states}(K) \to \mathit{states}(L)$
induces a simulation $K \mathrel{-\!\!\triangleright} L$

If $F : K \mathrel{-\!\!\triangleright} L$ and $F \subseteq G$
then $G : K \mathrel{-\!\!\triangleright} L$

The smaller the simulation,
the more information it carries

Example with prescience — ix

K and L
both with state space $X = \{0, 1, 2, 3, 4\}$,
initial set $\{4\}$, and property $\Diamond [\![\{0, 1\}]\!]$.

$$\begin{array}{rcl} \mathit{step}(K) & = & 1_X \cup \{(4,2), (2,1), (2,0)\} \\ \mathit{step}(L) & = & 1_X \cup \{(4,3), (4,2), (3,1), (2,0)\} \end{array}$$

$$K : \begin{tikzcd} & & 0 \\ 4 \arrow[r] & 2 \arrow[ur] \arrow[dr] & \\ & & 1 \end{tikzcd} \qquad L : \begin{tikzcd} & 2 \arrow[r] & 0 \\ 4 \arrow[ur] \arrow[dr] & & \\ & 3 \arrow[r] & 1 \end{tikzcd}$$

Simulation $F = 1_X \cup \{(2, 3)\}$

Concrete state 2 splits
into abstract states 2 and 3

Visibility — x

Visible spec (K, v)
where v is a function on $\mathit{states}(K)$

The visible behaviours:
$\mathit{Obs}(K, v) = \{v^\omega(xs) \mid xs \in \mathit{Beh}(K)\}$

(K, v) *implements* (L, w) iff ...
$\mathit{Obs}(K, v) \subseteq \mathit{Obs}(L, w)$
(differs from Abadi-Lamport)

Theorem (new). (K, v) implements (L, w)
if and only if there is a simulation $F : K \mathrel{-\!\!\triangleright} L$
with $F \subseteq \{(x, y) \mid v(x) = w(y)\}$.

---

3. Forward Simulations — xi

$F \subseteq \mathit{states}(K) \times \mathit{states}(L)$
is a *forward simulation* iff

(F0) For every $x \in \mathit{init}(K)$,
there is $y \in \mathit{init}(L)$ with $(x, y) \in F$

(F1) For every $(x, y) \in F$
and every x' with $(x, x') \in \mathit{step}(K)$,
there is y' with $(y, y') \in \mathit{step}(L)$ and $(x', y') \in F$

(F2) Every infinite initial execution ys of L
with $(xs, ys) \in F^\omega$ for some $xs \in \mathit{Beh}(K)$
has $ys \in \mathit{prop}(L)$

Theorem.
Every forward simulation is a simulation

Example Different Periods — xii

Specs $K(m)$ and $K(2 \cdot m)$ as above

Relation F given by

$$(j, k) \in F \quad \equiv \quad k = j \quad \vee \quad k = j + m \ .$$

F is a forward simulation $K(m) \mathrel{-\!\!\triangleright} K(2 \cdot m)$

There is no refinement mapping
from $K(m)$ to $K(2 \cdot m)$.

Backward Simulations — xiii

$F \subseteq \mathit{states}(K) \times \mathit{states}(L)$
is a *backward simulation* (version Jonnson) iff

(B0) Every pair $(x, y) \in F$
with $x \in \mathit{init}(K)$ has $y \in \mathit{init}(L)$.

(B1) For every pair $(x', y') \in F$ and
every x with $(x, x') \in \mathit{step}(K)$,
there is y with $(x, y) \in F$ and $(y, y') \in \mathit{step}(L)$.

(B2) Every behaviour xs of K has
infinitely many n with
$(xs_n; F)$ nonempty and finite.

(B3) Every infinite initial execution ys of L
with $(xs, ys) \in F^\omega$ for some $xs \in \mathit{Beh}(K)$
has $ys \in \mathit{prop}(L)$.

Theorem.
Every backward simulation is a simulation

---

xiii.1. –

Every composition of simulations is a simulation.
A composition of forward/backward simulations need not be a forward/backward simulation.

The example with prescience is a backward simulation.

The finiteness condition in (B2) is inconvenient.
It is needed to apply König's Lemma.

4. Eternity Variables (new) — xiv

Let M be a type for an "eternity" variable $\mathtt{m}$

A relation $R \subseteq \mathit{states}(K) \times M$
is a *behaviour restriction* over K
$\equiv$
for every behaviour xs of K
there exists $m \in M$ with

$(\forall\, n :: (xs_n, m) \in R)$

Soundness of Eternity Extension — xv

Let R be a behaviour restriction over K.

Construct spec W :
$\mathit{states}(W) = R \subseteq \mathit{states}(K) \times M$
$\mathit{init}(W) = R \cap (\mathit{init}(K) \times M)$
$\mathit{prop}(W) = \{ws \mid \mathit{fst}^\omega(ws) \in \mathit{prop}(K)\}$

$$((x, m), (x', m')) \in \mathit{step}(W) \quad \equiv \quad (x, x') \in \mathit{step}(K) \quad \wedge \quad m' = m$$

Theorem. $F = \{(x, (x', m)) \mid x = x'\}$
gives a simulation $K \mathrel{-\!\!\triangleright} W$.

Proof. Let $xs \in \mathit{Beh}(K)$.
Choose $m \in M$ with $(\forall\, n :: (xs_n, m) \in R)$.
Define $ys_n = (xs_n, m) \;\in\; R = \mathit{states}(W)$.
Then $ys \in \mathit{Beh}(W)$ and all $(xs_n, ys_n) \in F$.

---

5. Towards Completeness — xvi

We want to write an arbitrary simulation F
as a composition of special ones:

forward simulations
(backward simulations)
eternity extensions

All these "preserve quiescence"

Therefore, F must "preserve quiescence"

Preservation of Quiescence — xvii

For $xs \in \mathit{Beh}(K)$
the set of quiescent indices is

$$Q_K(xs) \;=\; \{\, n \mid (xs \mid n) +\!\!+\, (xs_n^\omega) \in \mathit{Beh}(K)\}$$

$F : K \mathrel{-\!\!\triangleright} L$ preserves quiescence
$\equiv$
for every $xs \in \mathit{Beh}(K)$
there exists $ys \in \mathit{Beh}(L)$
with $(xs_n, ys_n) \in F$ for all n
and $Q_K(xs) \subseteq Q_L(ys)$.

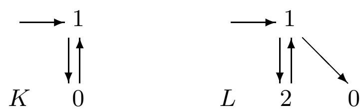
Quiescence Lost — xviii

K and L, with state spaces $X = \{0, 1, 2\}$
initial set $\{1\}$
property $\Diamond\Box \llbracket \{0\} \rrbracket$

$$\begin{array}{lcl} \mathit{step}(K) & = & 1_X \cup \{(1,0),(0,1)\}\ , \\ \mathit{step}(L) & = & 1_X \cup \{(1,0),(1,2),(2,1)\}\ . \end{array}$$

The quiescent indices are at the zero elements

Simulation $F = \{(0,0),(0,2),(1,1)\} : K \mathrel{-\!\!\triangleright} L$



Example: $xs = (1, 0, 0, 1, 0^\omega)$
corresponds to $ys = (1, 2, 2, 1, 0^\omega)$

Quiescence is lost where 0 becomes 2.

Semantic Completeness — xix

Theorem. Let $F : K \mathrel{-\!\!\triangleright} L$ preserve quiescence.
There is a forward simulation $H : K \mathrel{-\!\!\triangleright} K^\#$,
an eternity extension $E : K^\# \mathrel{-\!\!\triangleright} W$,
a refinement mapping $g : W \mathrel{-\!\!\triangleright} L$
with $(H; E; g) \subseteq F$.

Sketch of proof.
$K^\#$ is the "unfolding" of K
with $\mathit{states}(K^\#)$ the set of
stutterfree initial executions of K.

$R \subseteq \mathit{states}(K^\#) \times \mathit{Beh}(L)$ holds pairs (xs, ys)
with, for some $xt \in \mathit{Beh}(K)$,

$$xs \sqsubseteq xt \;\wedge\; (xt, ys) \in F^\omega \;\wedge\; Q_K(xt) \subseteq Q_L(ys)$$

This gives eternity extension $K^\# \mathrel{-\!\!\triangleright} W$.

Function $g : R \to \mathit{states}(L)$
maps (xs, ys) to ys_{n-1} where $n = \# xs$

Preservation of quiescence is needed
to make g a refinement mapping $W \mathrel{-\!\!\triangleright} L$

Comparison — xx

This result is simpler than
the Theorem of Abadi-Lamport (1991)

with backward simulation
instead of eternity extension

There the concrete specification had to be "machine-closed"

The abstract specification had to be of "finite invisible nondeterminism" and "internally continuous"

These conditions are not unreasonable but very technical

and therefore inconvenient

Comparison continued — xxi

Internal continuity is replaced by preservation of quiescence

Finite invisible nondeterminism is replaced by the condition that R be a behaviour restriction:

For every behaviour xs of K there exists $m \in M$ with

$$(\forall\, n :: (xs_n, m) \in R)$$

Usually solved by "approximating" m

---

6. Extended Example — xxii

Concrete specification *K0*

$$\begin{array}{l}
\textbf{var}\ \ \mathtt{j} :\ \mathit{Nat} := 0\ ;\\
\textbf{do}\ \ \mathit{true}\ \ \rightarrow\ \ \mathtt{j} := \mathtt{j} + 1\ ;\\
[\!]\ \ \mathtt{j} > 0\ \ \rightarrow\ \ \mathtt{j} := 0\ ;\\
\textbf{od}\ ;\\
\textbf{prop:}\ \mathtt{j}\ \text{decreases infinitely often.}
\end{array}$$

$\mathit{states}(K0) = \mathbb{N}$
$\mathit{init}(K0) = \{0\}$
$\mathit{prop}(K0) = \Box\Diamond[\![>]\!]$

$$\begin{array}{l}
(i, j) \in \mathit{step}(K0)\ \ \equiv\\
\quad j = i + 1\ \ \vee\ \ j = 0\ \ \vee\ \ j = i
\end{array}$$

Guessing the jumping points — xxiii

Abstract specification *K1*

$$\begin{array}{l}
\textbf{var}\ \ \mathtt{j} :\ \mathit{Nat} := 0\ ,\quad \mathtt{m} :\ \mathit{Nat} := 0\ ;\\
\textbf{do}\ \ \mathtt{j} < \mathtt{m}\ \ \rightarrow\ \ \mathtt{j} := \mathtt{j} + 1\ ;\\
[\!]\ \ \mathtt{j} = \mathtt{m}\ \ \rightarrow\ \ \mathtt{j} := 0\ ;\ \ \mathtt{m} := 0\ ;\\
[\!]\ \ \mathtt{j} = 0\ \ \rightarrow\ \ \mathtt{j} := 1\ ;\ \ \mathit{choose}\ \ \mathtt{m} \geq 1\ ;\\
\textbf{od}\ ;\\
\textbf{prop:}\ (\mathtt{j}, \mathtt{m})\ \text{changes infinitely often.}
\end{array}$$

$\mathit{states}(K1) = \mathbb{N} \times \mathbb{N}$
$\mathit{init}(K1) = \{(0, 0)\}$
$\mathit{prop}(K1) = \Box\Diamond[\![\neq]\!]$

$$\begin{array}{l}
((j, m)\ ,\ (j', m')) \in \mathit{step}(K1)\ \ \equiv\\
\quad (j < m\ \wedge\ j' = j + 1\ \wedge\ m' = m)\\
\quad \vee\ \ (j = m\ \wedge\ j' = m' = 0)\\
\quad \vee\ \ (j = 0\ \wedge\ j' = 1 \leq m')\\
\quad \vee\ \ (j' = j\ \wedge\ m' = m)\ .
\end{array}$$

How to let *K0* simulate *K1*?

Adding History Variables — xxiv

Extend *K0* with history variables `n` and `q` to obtain *K2*

$$\begin{array}{l}
\textbf{var}\ \ \mathtt{j} :\ \mathit{Nat} := 0\ ,\quad \mathtt{n} :\ \mathit{Nat} := 0\ ;\\
\quad \mathtt{q} :\ \textbf{array}\ \mathit{Nat}\ \textbf{of}\ \mathit{Nat} := ([\mathit{Nat}]\ 0)\ ;\\
\textbf{do}\ \ \mathit{true}\ \ \rightarrow\ \ \mathtt{j} := \mathtt{j} + 1\ ;\ \ \mathtt{q}[\mathtt{n}] := \mathtt{q}[\mathtt{n}] + 1\ ;\\
[\!]\ \ \mathtt{j} > 0\ \ \rightarrow\ \ \mathtt{j} := 0\ ;\ \ \mathtt{n} := \mathtt{n} + 1\ ;\\
\textbf{od}\ ;\\
\textbf{prop:}\ \mathtt{j}\ \text{decreases infinitely often.}
\end{array}$$

$F_{0,2} : K0 \rightarrow\!\!\triangleright\ K2$, the converse of the projection, is a forward simulation

---

Eternity Extension — xxv

Extend *K2* with eternity variable `m`

`m` is an infinite array
with the behaviour restriction (!)

$$R :\quad \mathtt{j} \leq \mathtt{m}[\mathtt{n}]\ \ \wedge\ \ (\forall\, i : 0 \leq i < \mathtt{n} : \mathtt{m}[i] = \mathtt{q}[i])\ .$$

This gives spec *K3* with

$$\begin{array}{l}
((j, n, q, m)\ ,\ (j', n', q', m')) \in \mathit{step}(K3)\ \ \equiv\\
\quad m = m'\ \wedge\ ((j, n, q)\ ,\ (j', n', q')) \in \mathit{step}(K2)
\end{array}$$

Define $f_{3,1} : \mathit{states}(K3) \rightarrow \mathit{states}(K1)$ by

$$f_{3,1}(j, n, q, m)\ \ =\ \ (j, (j = 0\ ?\ 0 : m[n]))$$

This is a refinement mapping $K3 \rightarrow\!\!\triangleright\ K1$
We thus have $K0 \rightarrow\!\!\triangleright\ K2 \rightarrow\!\!\triangleright\ K3 \rightarrow\!\!\triangleright\ K1$.
