Verification – Lecture 6
Precedence Properties

Bernd Finkbeiner – Sven Schewe
Rayna Dimitrova – Lars Kuhtz – Anne Proetzsch

Wintersemester 2007/2008

Linear Invariants

Review

A linear invariant is of the form

$$\underbrace{\sum_{i=1}^{r} a_i \cdot y_i}_{\text{body}} + \underbrace{\sum_{\ell \in \mathcal{L}} b_\ell \cdot at\_\ell}_{\text{compensation expression}} = K$$

where

$a_i,\ b_\ell,\ K$ – integer constants.

$\mathcal{L}$ – set of all locations in P

$y_1, \ldots, y_r$ – all linear variables in P

Linear Variables

Review

Definition: integer variable y is linear in P if

$$y' = y + c \qquad \text{for every } \rho_\tau$$

where c is some integer constant

Example: semaphore variables are linear

$$\underbrace{y' = y + 1}_{\textbf{release}} \qquad \underbrace{y' = y - 1}_{\textbf{request}} \qquad \underbrace{y' = y}_{\text{otherwise}}$$

Increments

Review

- $\Delta(y, \tau) = c$ if $\rho_\tau \to y' = y + c$

  therefore $\rho_\tau \to y' = y + \Delta(y, \tau)$

- $$\Delta(at\_\ell, \tau) = \begin{cases} 1 & \text{if } \ell = \ell_j \\ -1 & \text{if } \ell = \ell_i \\ 0 & \text{otherwise} \end{cases}$$

  if $\rho_\tau \to move(\ell_i, \ell_j)$

  therefore $\rho_\tau \to at'\_\ell = at\_\ell + \Delta(at\_\ell, \tau)$

Automatic Invariant Construction

Review

Construct

$$\varphi: \quad \sum_{i=1}^{r} a_i \cdot y_i \; + \; \sum_{\ell \in \mathcal{L}} b_\ell \cdot at\_\ell \; = \; K$$

Our procedure guarantees that the generated assertions are P-invariants!

Equations

Review

We obtain the values of the coefficients from a set of equations as follows:

(I) The invariant has to hold at the first state of every computation

Θ implies $y_i = y_i^0 \; (i = 1 \ldots r)$
and $\pi = \{\ell_0^1, \ldots \ell_0^m\}$

and so we get

$$\sum_{i=1}^{r} a_i \cdot y_i^0 \; + \; (b_{\ell_0^1} + \cdots + b_{\ell_0^m}) \; = \; K$$

Equations (cont'd)

Review

(**T**) the assertion has to be preserved by all transitions (we want it to be inductive):

$$\underbrace{\left(\sum_{i=1}^{r} a_i \cdot y_i \;+\; \sum_{\ell \in \mathcal{L}} b_\ell \cdot at\_\ell = K\right)}_{\varphi} \wedge \rho_\mathcal{T}$$

$$\rightarrow \underbrace{\left(\sum_{i=1}^{r} a_i \cdot y'_i \;+\; \sum_{\ell \in \mathcal{L}} b_\ell \cdot at'\_\ell = K\right)}_{\varphi'}$$

Equations (cont'd)

Review

$$\underbrace{\left(\sum_{i=1}^{r} a_i \cdot y_i \;+\; \sum_{\ell \in \mathcal{L}} b_\ell \cdot at\_\ell = K\right)}_{\varphi} \wedge \rho_\mathcal{T}$$

$$\rightarrow \underbrace{\left(\sum_{i=1}^{r} a_i \cdot y'_i \;+\; \sum_{\ell \in \mathcal{L}} b_\ell \cdot at'\_\ell = K\right)}_{\varphi'}$$

or $\rho_\mathcal{T} \rightarrow \sum_{i=1}^{r} a_i \cdot (y'_i - y_i) \;+\; \sum_{\ell \in \mathcal{L}} b_\ell \cdot (at'\_\ell - at\_\ell) \;=\; 0$

resulting in the set of equations

$$\boxed{\sum_{i=1}^{r} a_i \cdot \Delta(y_i, \tau) \;+\; \sum_{\ell \in \mathcal{L}} b_\ell \cdot \Delta(at\_\ell, \tau) \;=\; 0}$$

for every transition $\tau \in \mathcal{T}$

Linear Invariants for Cyclic Programs

Review

Program $\ell_0^1\colon\ S_1 \ \|\ \ldots \|\ \ell_0^j\colon\ S_j\ \|\ \ldots\ \|\ \ell_0^m\colon\ S_m$

where S_j is of the form

$$\ell_0^j\colon\ \textbf{loop forever do}\ \underbrace{\ell_1^j, \ell_2^j, \ldots, \ell_k^j}_{\text{cycle } C}$$

Define

$$\Delta(y, C) \;=\; \Delta(y, \tau_1) \;+\; \cdots \;+\; \Delta(y, \tau_n)$$

Invariant Construction

Review

$$\underbrace{\sum_{i=1}^{r} a_i \cdot y_i}_{\text{body}} \;+\; \underbrace{\sum_{\ell \in \mathcal{L}} b_\ell \cdot at\_\ell}_{\text{compensation expression}} \;=\; K$$

3 Phases:

1. Compute a_i's
2. Compute b_ℓ's
3. Compute K

Phase 1: Bodies

Review

For cycle $\underbrace{\ell_1, \ell_2, \ldots, \ell_k}_{C}$

$$\sum_{i=1}^{r} a_i \cdot \Delta(y_i, \tau_{\ell_1}) - b_{\ell_1} + b_{\ell_2} = 0$$

$$\sum_{i=1}^{r} a_i \cdot \Delta(y_i, \tau_{\ell_2}) - b_{\ell_2} + b_{\ell_3} = 0$$

$$\vdots$$

$$\sum_{i=1}^{r} a_i \cdot \Delta(y_i, \tau_{\ell_k}) + b_{\ell_1} - b_{\ell_k} = 0$$

$$\sum_{i=1}^{r} a_i \cdot \Delta(y_i, C) = 0$$

Phase 2: Compensation Expressions

Review

$b_{\ell_0} = 0$

For $\tau: \ell_j \rightarrow \ell_k$ where $j < k$

$$\sum_{i=1}^{r} a_i \cdot \Delta(y_i, \tau) - b_{\ell_j} + b_{\ell_k} = 0$$

Assume that for all $j < k$, b_{ℓ_j} is known.
Compute b_{ℓ_k} from

$$b_{\ell_k} = b_{\ell_j} - \sum_{i=1}^{r} a_i \cdot \Delta(y_i, \tau)$$

(independently for each cycle)

Phase 3: Right Constants

Review

$$K = \sum_{i=1}^{r} a_i \cdot y_i^0$$

Note: This set of equations has the same solutions as the equations (T) + (I) except for solutions of the form

$$at\_\ell_1 + \ \ldots \ at\_\ell_k = 1$$

which are produced by (T) + (I), but not by this set.

Example: PRODUCER-CONSUMER

Review

$$\begin{array}{l} \textbf{local } r,\ ne,\ nf: \textbf{integer where } r=1,\ ne=N,\ nf=0 \\ \qquad b \qquad\quad : \textbf{list of integer where } b=\Lambda \end{array}$$

$$\left[\begin{array}{l} \textbf{local } x\text{: } \textbf{integer} \\ \ell_0\text{: } \textbf{loop forever do} \\ \qquad \left[\begin{array}{l} \ell_1\text{: } \textbf{produce } x \\ \ell_2\text{: } \textbf{request } ne \\ \ell_3\text{: } \textbf{request } r \\ \ell_4\text{: } b := b \bullet x \\ \ell_5\text{: } \textbf{release } r \\ \ell_6\text{: } \textbf{release } nf \end{array}\right] \end{array}\right] \quad || \quad \left[\begin{array}{l} \textbf{local } y\text{: } \textbf{integer} \\ m_0\text{: } \textbf{loop forever do} \\ \qquad \left[\begin{array}{l} m_1\text{: } \textbf{request } nf \\ m_2\text{: } \textbf{request } r \\ m_3\text{: } (y,\ b) := (hd(b),\ tl(b)) \\ m_4\text{: } \textbf{release } r \\ m_5\text{: } \textbf{release } ne \\ m_6\text{: } \textbf{consume } y \end{array}\right] \end{array}\right]$$

Increments along each cycle:

	Prod	Cons
r	0	0
ne	-1	1
nf	1	-1
$\|b\|$	1	-1

Example: PRODUCER-CONSUMER

Review

local r, ne, nf: **integer where** $r = 1$, $ne = N$, $nf = 0$
b : **list of integer where** $b = \Lambda$

```
[local x: integer                 ]      [local y: integer                       ]
[ℓ0: loop forever do              ]      [m0: loop forever do                    ]
[    [ℓ1: produce x  ]            ]      [    [m1: request nf           ]        ]
[    [ℓ2: request ne ]            ]  ||  [    [m2: request r            ]        ]
[    [ℓ3: request r  ]            ]      [    [m3: (y, b) := (hd(b), tl(b))]     ]
[    [ℓ4: b := b • x ]            ]      [    [m4: release r            ]        ]
[    [ℓ5: release r  ]            ]      [    [m5: release ne           ]        ]
[    [ℓ6: release nf ]            ]      [    [m6: consume y            ]        ]
```

We look for linear invariants with the body

$$a_r \cdot r \;+\; a_e \cdot ne \;+\; a_f \cdot nf \;+\; a_b \cdot |b|$$

Example (cont'd)

Review

For each cycle: $\sum_{i=1}^{r} a_i \cdot \Delta(y_i, C) = 0$

Therefore
Prod: $-a_e + a_f + a_b = 0$
Cons: $a_e - a_f - a_b = 0$

Solutions			Bodies
1.	$a_r = 1$,	$a_e = a_f = a_b = 0$	B_1: r
2.	$a_e = a_f = 1$,	$a_r = a_b = 0$	B_2: $ne + nf$
3.	$a_e = a_b = 1$,	$a_r = a_f = 0$	B_3: $ne + |b|$

Example (cont'd)

Review

```
local r, ne, nf : integer where r = 1, ne = N, nf = 0
      b         : list of integer where b = Λ

[ local x: integer                ]      [ local y: integer                        ]
[ ℓ0: loop forever do             ]      [ m0: loop forever do                     ]
[     [ ℓ1: produce x   ]         ]  ||  [     [ m1: request nf               ]    ]
[     [ ℓ2: request ne  ]         ]      [     [ m2: request r                ]    ]
[     [ ℓ3: request r   ]         ]      [     [ m3: (y, b) := (hd(b), tl(b)) ]    ]
[     [ ℓ4: b := b • x  ]         ]      [     [ m4: release r                ]    ]
[     [ ℓ5: release r   ]         ]      [     [ m5: release ne               ]    ]
[     [ ℓ6: release nf  ]         ]      [     [ m6: consume y                ]    ]
```

compensation expressions

coefficients of $b_{\ell_1}, \ldots, b_{m_6}$
(corresponding to bodies B_1, B_2, B_3)

– Prod –				– Cons –			
	B_1	B_2	B_3		B_1	B_2	B_3
b_{ℓ_1}	0	0	0	b_{m_1}	0	0	0
b_{ℓ_2}	0	0	0	b_{m_2}	0	1	0
b_{ℓ_3}	0	1	1	b_{m_3}	1	1	0
b_{ℓ_4}	1	1	1	b_{m_4}	1	1	1
b_{ℓ_5}	1	1	0	b_{m_2}	0	1	1
b_{ℓ_6}	0	1	0	b_{m_6}	0	0	0

Bodies

B_1: r

B_2: $ne + nf$

B_3: $ne + |b|$

Example (cont'd)

Review

Right constants

Initial values $r = 1,\ ne = N,\ nf = 0,\ |b| = 0$

$$K_1 = 1 \cdot \underbrace{1}_{r} = 1$$

$$K_2 = 1 \cdot \underbrace{N}_{ne} + 1 \cdot \underbrace{0}_{nf} = N$$

$$K_3 = 1 \cdot \underbrace{N}_{ne} + 1 \cdot \underbrace{0}_{|b|} = N$$

The resulting invariants

φ_1: $r + at\_\ell_{4,5} + at\_m_{3,4} = 1$

φ_2: $ne + nf + at\_\ell_{3..6} + at\_m_{2..5} = N$

φ_3: $ne + |b| + at\_\ell_{3,4} + at\_m_{4,5} = N$

Are the Generated Invariants Useful?

```
local r, ne, nf: integer where r = 1, ne = N, nf = 0
      b        : list of integer where b = Λ

⎡local x: integer           ⎤      ⎡local y: integer                    ⎤
⎢ℓ0: loop forever do        ⎥      ⎢m0: loop forever do                 ⎥
⎢    ⎡ℓ1: produce x  ⎤      ⎥      ⎢    ⎡m1: request nf                ⎤⎥
⎢    ⎢ℓ2: request ne ⎥      ⎥  ||  ⎢    ⎢m2: request r                 ⎥⎥
⎢    ⎢ℓ3: request r  ⎥      ⎥      ⎢    ⎢m3: (y, b) := (hd(b), tl(b))  ⎥⎥
⎢    ⎢ℓ4: b := b • x ⎥      ⎥      ⎢    ⎢m4: release r                 ⎥⎥
⎢    ⎢ℓ5: release r  ⎥      ⎥      ⎢    ⎢m5: release ne                ⎥⎥
⎣    ⎣ℓ6: release nf ⎦      ⎦      ⎣    ⎣m6: consume y                 ⎦⎦
```

Specification:

$$\Box \underbrace{\neg(at\_\ell_4 \ \wedge \ at\_m_3)}_{\psi_1} \qquad \Box \underbrace{at\_m_3 \ \rightarrow \ |b| > 0}_{\psi_3}$$

$$\Box \underbrace{at\_\ell_4 \ \rightarrow \ |b| < N}_{\psi_2}$$

Example (cont'd)

```
local r, ne, nf: integer where r = 1, ne = N, nf = 0
      b        : list of integer where b = Λ

⎡local x: integer           ⎤      ⎡local y: integer                    ⎤
⎢ℓ0: loop forever do        ⎥      ⎢m0: loop forever do                 ⎥
⎢    ⎡ℓ1: produce x  ⎤      ⎥      ⎢    ⎡m1: request nf                ⎤⎥
⎢    ⎢ℓ2: request ne ⎥      ⎥  ||  ⎢    ⎢m2: request r                 ⎥⎥
⎢    ⎢ℓ3: request r  ⎥      ⎥      ⎢    ⎢m3: (y, b) := (hd(b), tl(b))  ⎥⎥
⎢    ⎢ℓ4: b := b • x ⎥      ⎥      ⎢    ⎢m4: release r                 ⎥⎥
⎢    ⎢ℓ5: release r  ⎥      ⎥      ⎢    ⎢m5: release ne                ⎥⎥
⎣    ⎣ℓ6: release nf ⎦      ⎦      ⎣    ⎣m6: consume y                 ⎦⎦
```

Additional Bottom-Up invariants:

$$\underbrace{r \geq 0}_{\chi_0} \ \wedge \ \underbrace{ne \geq 0}_{\chi_1} \ \wedge \ \underbrace{nf \geq 0}_{\chi_2} \ \wedge \ \underbrace{|b| \geq 0}_{\chi_3}$$

Example (cont'd)

$$\psi_1 : \quad \underbrace{r + at\_\ell_{4,5} + at\_m_{3,4} = 1}_{\varphi_1} \wedge \underbrace{r \geq 0}_{\chi_0} \rightarrow \underbrace{\neg(at\_\ell_4 \wedge at\_m_3)}_{\psi_1}$$

$$\psi_2 : \quad \underbrace{ne + |b| + at\_\ell_{3,4} + at\_m_{4,5} = N}_{\varphi_3} \wedge \underbrace{ne \geq 0}_{\chi_1} \rightarrow \underbrace{at\_\ell_4 \rightarrow |b| < N}_{\psi_2}$$

Since $at\_\ell_4 \rightarrow at\_\ell_{3,4} = 1$

and $ne \geq 0$, $at\_\ell_{3,4} = 1$, $at\_m_{4,5} \geq 0$ implies $|b| < N$

Example (cont'd)

$$\psi_3 : \quad \underbrace{ne + nf + at\_\ell_{3..6} + at\_m_{2..5} = N}_{\varphi_2} \wedge \underbrace{ne + |b| + at\_\ell_{3,4} + at\_m_{4,5} = N}_{\varphi_3} \wedge \underbrace{nf \geq 0}_{\chi_2} \rightarrow \underbrace{at\_m_3 \rightarrow |b| > 0}_{\psi_3}$$

Since φ_2, φ_3 yields

$nf - |b| + at\_\ell_{3..6} - at\_\ell_{3,4} + 1 = 0$

Thus

$$|b| = \underbrace{nf}_{\geq 0} + \underbrace{(at\_\ell_{3..6} - at\_\ell_{3,4})}_{\geq 0} + 1 > 0$$

Precedence Properties

Precedence Properties

are of the form

$$p \Rightarrow q_m \, \mathcal{W} \, (q_{m-1} \cdots (q_1 \, \mathcal{W} \, q_0) \ldots)$$

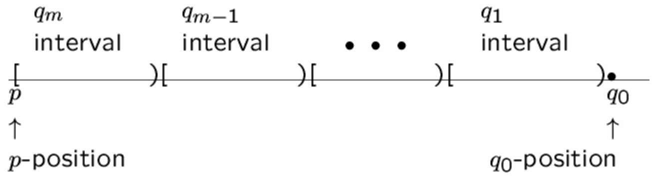
also written

$$\boxed{p \Rightarrow q_m \, \mathcal{W} \, q_{m-1} \cdots q_1 \, \mathcal{W} \, q_0}$$

for assertions $p, q_0, q_1, \ldots, q_m$.

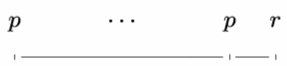
Each interval may be empty, may extend to infinity.

Models that satisfy these formulas



Simple Precedence

$$\boxed{p \;\Rightarrow\; p \,\mathcal{W}\, r}$$



can be reduced to first-order VCs by verification rule WAIT-B:

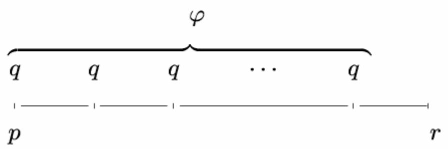
Rule WAIT-B (basic waiting-for)

For assertions p, r,

$$\frac{P \;\Vdash\; \{p\}\;\mathcal{T}\;\{p \,\vee\, r\}}{P \;\vDash\; p \;\Rightarrow\; p \,\mathcal{W}\, r}$$

General Waiting-For

$$\boxed{p \;\Rightarrow\; q \,\mathcal{W}\, r}$$



Rule WAIT (general waiting-for)

For assertions p, q, r, φ

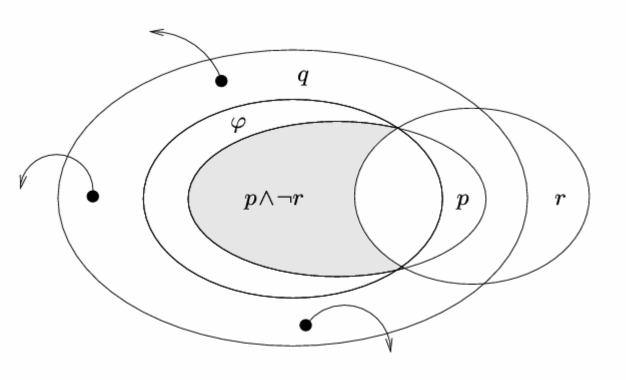
W1. $p \;\rightarrow\; \varphi \vee r$

W2. $\varphi \;\rightarrow\; q$

W3. $\{\varphi\}\mathcal{T}\{\varphi \vee r\}$

$$p \;\Rightarrow\; q \,\mathcal{W}\, r$$

Strengthening & Weakening



$\varphi \to q$ "φ strengthens q"

$p \to \varphi \lor r$, i.e., $p \land \neg r \to \varphi$ "φ weakens $p \land \neg r$"

Example

local y_1, y_2: boolean where $y_1 = \text{F}, y_2 = \text{F}$
s : integer where $s = 1$

P_1 ::
ℓ_0: loop forever do
- ℓ_1: noncritical
- ℓ_2: $(y_1, s) := (\text{T}, 1)$
- ℓ_3: await $(\neg y_2) \lor (s = 2)$
- ℓ_4: critical
- ℓ_5: $y_1 := \text{F}$

||

P_2 ::
m_0: loop forever do
- m_1: noncritical
- m_2: $(y_2, s) := (\text{T}, 2)$
- m_3: await $(\neg y_1) \lor (s = 1)$
- m_4: critical
- m_5: $y_2 := \text{F}$

We proved mutual exclusion

ψ_1: $\neg(at\_\ell_4 \land at\_m_4)$

Using invariants

φ_0: $s = 1 \lor s = 2$

φ_1: $y_1 \leftrightarrow at\_\ell_{3..5}$

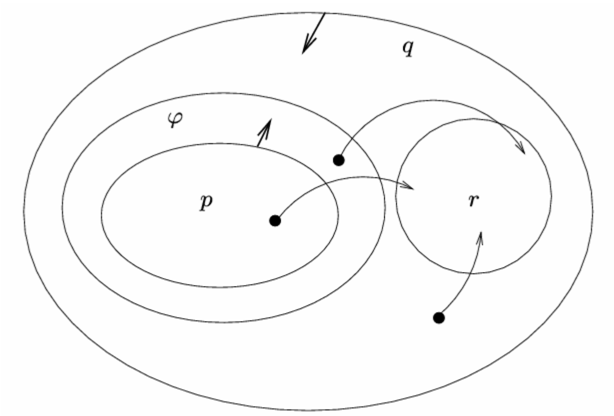
φ_2: $y_2 \leftrightarrow at\_m_{3..5}$

φ_3: $at\_\ell_3 \land at\_m_4 \to y_2 \land s = 1$

φ_4: $at\_\ell_4 \land at\_m_3 \to y_1 \land s = 2$

$$\psi_2: \underbrace{at\_\ell_3 \land at\_m_{0..2}}_{p} \Rightarrow \underbrace{\neg at\_m_4}_{q} \ \mathcal{W} \ \underbrace{at\_\ell_4}_{r}$$

Derivation of Intermediate Assertions



escape transition: transition that establishes r

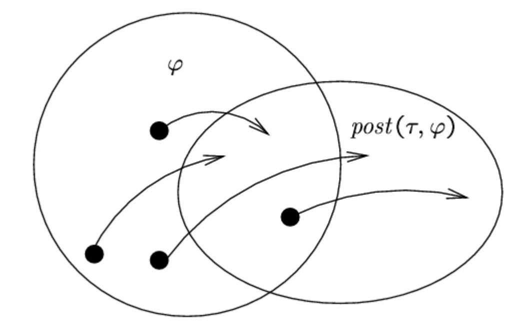
Forward Propagation

characterizes all states that can be reached from a $(p \wedge \neg r)$-state without taking an escape transition.

Based on postcondition:

$post(\tau, \varphi)$: $\exists V^0 . \varphi(V^0) \wedge \rho_\tau(V^0, V)$

$post(\tau, \varphi)$ characterizes all states that are τ-successors of a φ-state.



Forward Propagation (cont'd)

1. $\Phi_0 = p \land \neg r$

2. Repeat

   $\Phi_{k+1} = \Phi_k \lor post(\tau, \Phi_k)$
   for any non-escape transition

   Until

   $post(\tau, \Phi_t) \rightarrow \Phi_t$
   for all non-escape transitions

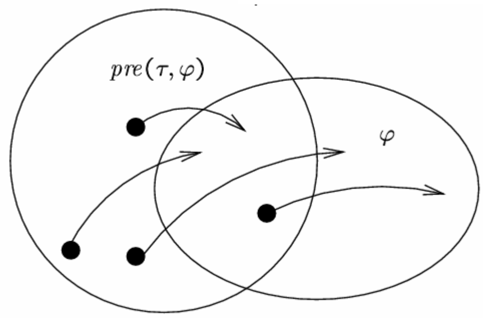
Backward Propagation

characterizes all states that can reach a q-state without taking an escape transition

Based on precondition:

$pre(\tau, \varphi)$: $\forall V' . \rho_\tau(V, V') \rightarrow \varphi(V')$

$pre(\tau, \varphi)$ characterizes all states of which all τ-successors satisfy φ.



Backward Propagation (cont'd)

1. $\Gamma_0 = q$

2. Repeat

   $\Gamma_{k+1} = \Gamma_k \wedge pre(\tau, \Gamma_k)$

   for any non-escape transition

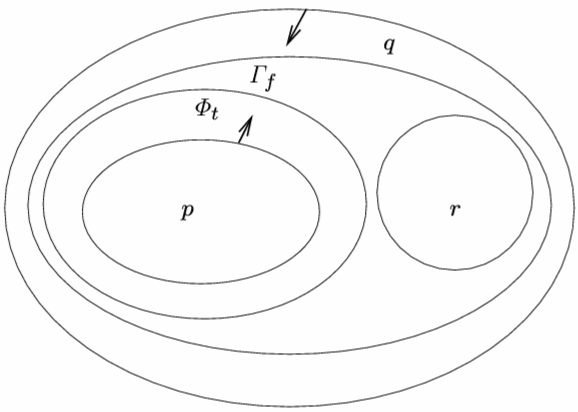
   Until

   $\Gamma_f \rightarrow pre(\tau, \Gamma_f)$

   for all non-escape transitions

If this terminates (it may not), Γ_f is a good assertion to be used in rule WAIT.
W1–W3 are satisfied if $p \Rightarrow q \mathcal{W} r$ holds.

Forward vs. Backward

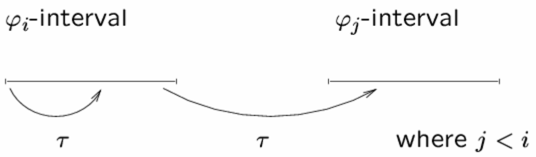


Nested Waiting-For Formulas

Rule NWAIT-B (basic nested waiting-for)

For assertions $\varphi_0, \varphi_1, \ldots, \varphi_m$

$$\frac{\{\varphi_i\}\ \mathcal{T}\ \left\{\bigvee_{j\le i} \varphi_j\right\} \quad \text{for} \quad i = 1, \ldots, m}{\left(\bigvee_{j=0}^{m} \varphi_j\right) \rightarrow \varphi_m\ \mathcal{W}\ \varphi_{m-1}\ \mathcal{W}\ \cdots\ \varphi_1\ \mathcal{W}\ \varphi_0}$$



General Rule

Rule NWAIT (nested waiting-for)

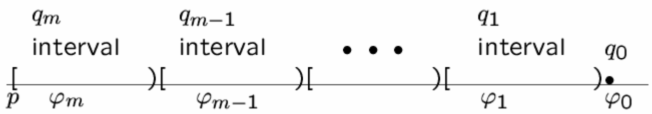
For assertions p, $q_0, q_1, \ldots, q_m$ and $\varphi_0, \varphi_1, \ldots, \varphi_m$

N1. $p \rightarrow \bigvee_{j=0}^{m} \varphi_j$

N2. $\varphi_i \rightarrow q_i$ for $i = 0, 1, \ldots, m$

N3. $\{\varphi_i\}\mathcal{T}\left\{\bigvee_{j\le i} \varphi_j\right\}$ for $i = 1, \ldots, m$

$$p \Rightarrow q_m\ \mathcal{W}\ q_{m-1}\ \cdots\ q_1\ \mathcal{W}\ q_0$$



Example

local y_1, y_2: boolean where $y_1 = \text{F}, y_2 = \text{F}$
s : integer where $s = 1$

$P_1 ::$
ℓ_0 : loop forever do
- ℓ_1 : noncritical
- ℓ_2 : $(y_1, s) := (\text{T}, 1)$
- ℓ_3 : await $(\neg y_2) \vee (s = 2)$
- ℓ_4 : critical
- ℓ_5 : $y_1 := \text{F}$

||

$P_2 ::$
m_0 : loop forever do
- m_1 : noncritical
- m_2 : $(y_2, s) := (\text{T}, 2)$
- m_3 : await $(\neg y_1) \vee (s = 1)$
- m_4 : critical
- m_5 : $y_2 := \text{F}$

$$\underbrace{at\_\ell_3}_{p} \Rightarrow \underbrace{\neg at\_m_4}_{q_3} \; \mathcal{W} \; \underbrace{at\_m_4}_{q_2} \; \mathcal{W} \; \underbrace{\neg at\_m_4}_{q_1} \; \mathcal{W} \; \underbrace{at\_\ell_4}_{q_0}$$

p: at_ℓ_3

q_3: $\neg at\_m_4 \wedge at\_\ell_3 \wedge at\_m_3 \wedge s = 1$
"P_2 has priority over P_1"

q_2: $at\_m_4 \wedge at\_\ell_3$

q_1: $\neg at\_m_4 \wedge at\_\ell_3 \wedge (at\_m_3 \rightarrow s = 2)$
"P_1 has priority over P_2"

q_0: at_ℓ_4

Completeness

If the formula

$$p \Rightarrow q_m \; \mathcal{W} \; (q_{m-1} \cdots (q_1 \; \mathcal{W} \; q_0) \ldots)$$

is P-valid, then there exist assertions $\varphi_0, \varphi_1, \ldots, \varphi_m$, such that the premises of rule NWAIT are provable from state-validities.