Refinement Calculus

Using Z
Woodcock & Davies

Specification (schemas)

⇘ refinement

Design (schemas)

⇘ translation

Algorithm (abstract program)

⇘ refinement

Code (guarded commands)

⇘ translation

Code (programming language)

Specification statement

$$frame \ : \ [\, precondition, \ postcondition \,]$$

Example

$$available! : [\ true,\ available! = free(\theta BoxOffice)\ ]$$

Example

$$m : \left[\begin{array}{ccc} f(a) * f(b) \leq 0 & & -0.1 < f(m') < 0.1 \\ a \leq b & , & a \leq m' \leq b \end{array} \right]$$

Guarded commands

if · · · fi

do · · · od

begin · · · end

‘; ’

Abort

$$\mathsf{abort} \;=\; w : [\, false,\ true \,]$$

Choose

$$\mathsf{choose}\ w = w : [\, true, true \,]$$

Skip

$$\mathsf{skip} \;=\; [\, true,\ true \,]$$

Magic

$$\mathsf{magic} \;=\; w : [\, true, false \,]$$

Assignment

$$x, y : \left[\begin{array}{ll} x = X & x' = X - Y \\ y = Y \ , & y' = X \end{array} \right] ; \ \ x := y - x$$

Refinement

If a program P is correctly refined by another program Q, then we write $P \sqsubseteq Q$: pronounced 'P is refined by Q'.

$$P \sqsubseteq Q \;\wedge\; Q \sqsubseteq R \;\Rightarrow\; P \sqsubseteq R$$

Strengthen postcondition

If

$$pre \wedge post_2 \;\Rightarrow\; post_1$$

then

$$w : [\, pre,\ post_1 \,] \;\sqsubseteq\; w : [\, pre,\ post_2 \,]$$

Example

$$m : \left[\begin{array}{ccc} f(a) * f(b) \leq 0 & & -0.1 < f(m') < 0.1 \\ a \leq b & , & a \leq m' \leq b \end{array} \right]$$

$\sqsubseteq$ (strengthen postcondition)

$$m : \left[\begin{array}{ccc} f(a) * f(b) \leq 0 & & -0.01 < f(m') < 0.01 \\ a \leq b & , & a \leq m' \leq b \end{array} \right]$$

Weaken precondition

If

$$pre_1 \Rightarrow pre_2$$

then

$$w : [\, pre_1, post \,] \sqsubseteq w : [\, pre_2, post \,]$$

Example

$$m : \left[\begin{array}{ccc} f(a) * f(b) \leq 0 & & -0.01 < f(m') < 0.01 \\ a \leq b & , & a \leq m' \leq b \end{array} \right]$$

$\sqsubseteq$ (weaken precondition)

$$m : \left[\begin{array}{ccc} f(a) * f(b) \leq 0 & & -0.01 < f(m') < 0.01 \\ & , & a \leq m' \leq b \end{array} \right]$$

Feasibility check

$$pre \Rightarrow \exists\, x' : X;\ y' : Y \bullet post$$

Infeasible

$$m : \left[\begin{array}{ll} f(a) * f(b) \leq 0 & -0.01 < f(m') < 0.01 \\ , & a \leq m' \leq b \end{array} \right]$$

Introduce local block

If x does not appear in w, then

$$
\begin{aligned}
w : [\, pre,\ post \,] \ \sqsubseteq\ & \mathsf{begin} \\
& \quad \mathsf{var}\ x : T \mid inv \bullet \\
& \qquad w, x : [\, pre,\ post \,] \\
& \mathsf{end}
\end{aligned}
$$

Example

$$m : \left[\begin{array}{ccc} f(a) * f(b) \leq 0 & & -0.1 < f(m') < 0.1 \\ a \leq b & , & a \leq m' \leq b \end{array} \right]$$

$\sqsubseteq$ (introduce local block)

begin

var $x, y \bullet$

$$x, y, m : \left[\begin{array}{ccc} f(a) * f(b) \leq 0 & & -0.1 < f(m') < 0.1 \\ a \leq b & , & a \leq m' \leq b \end{array} \right]$$

end

Assignment introduction

If

$$pre \;\Rightarrow\; post[E/w', x/x']$$

then

$$w, x : [\, pre,\ post \,] \;\sqsubseteq\; w := E$$

Example

$$x : [\ true,\ x' = x + 1\ ] \sqsubseteq x := x + 1$$

Skip introduction

If

$$pre \Rightarrow post[w/w']$$

then

$$w : [\, pre, post \,] \sqsubseteq \mathsf{skip}$$

Example

$$x, y : [\, x = 5 \wedge y = x^3,\ x' = 5 \,]$$

$$\sqsubseteq \mathsf{skip}$$

Logical constants

$$x, y : \left[\begin{array}{ll} x = X & x' = X - Y \\ y = Y \ , & y' = X \end{array} \right] ; \ \ x := y - x$$

$$\mathsf{begin}\ \mathsf{con}\ X : T \bullet prog\ \mathsf{end}$$

Example

```
begin
    con X • x : [ x = X, x′ > X ]
end;
begin
    con X • x : [ x = X, x′ > X ]
end
```

Scope

```
begin
    con X • x : [ x = X, x′ > X ];  x : [ x = X, x′ > X ]
end
```

Angelic nondeterminism

$$\text{begin con } X \bullet x : [\, x = X,\ x' > X \,] \text{ end}$$

$$\text{begin con } X \bullet x : [\, \mathit{true},\ x' > x \,] \text{ end}$$

Introduce logical constant

If

$$pre_1 \Rightarrow (\exists\, C : T \bullet pre_2)$$

and C is a fresh name, then

$$w : [\, pre_1, post \,]$$

$$\sqsubseteq \begin{cases} \mathsf{begin} \\ \qquad \mathsf{con}\ C : T \bullet w : [\, pre_2, post \,] \\ \mathsf{end} \end{cases}$$

Eliminate logical constant

If C occurs nowhere in *prog*, then

$$\left.\begin{array}{l}\mathsf{begin}\\ \quad\mathsf{con}\ C : T \bullet prog\\ \mathsf{end}\end{array}\right\} \quad \sqsubseteq \quad prog$$

C-style constants

```
begin
    var pi : ℝ | pi = 22/7 •
        ⋮
end
```

Sequential composition

$$w, x : [\ pre,\ post\ ]$$

$$x : [\ pre,\ mid\ ]$$

$$w, x : [\ mid[X/x, x/x'],\ post[X/x]\ ]$$

Sequential composition introduction

$$w, x : [\, pre,\ post \,]$$

$$\sqsubseteq \quad \left\{ \begin{array}{l} \mathsf{begin} \\ \quad \mathsf{con}\ X \bullet \\ \qquad x : [\, pre,\ mid \,]\,; \\ \qquad w, x : [\, mid[X/x, x/x'],\ post[X/x] \,] \\ \mathsf{end} \end{array} \right.$$

Example

$x : [\ true,\ x' = x + 2\ ]$

$\sqsubseteq$ (sequential composition introduction)

$\mathsf{con}\ X \bullet$

$\quad x : [\ x = X,\ x' = x + 1\ ];$

$\quad x : [\ x = X + 1,\ x' = X + 2\ ]$

Swapping the hard way

$x, y : [\, true,\ x' = y \wedge y' = x \,]$

$\sqsubseteq$ (sequential composition introduction)

$\mathsf{con}\ X_1, Y_1 : \mathbb{Z} \bullet$

$x, y : [\, true,\ x' = x - y \wedge y' = x \,];$ [◁]

$$x, y : \left[\begin{array}{ll} x = X_1 - Y_1 & x' = Y_1 \\ y = X_1 & ,\ y' = X_1 \end{array}\right]$$ [†]

$\sqsubseteq$ (sequential composition introduction)

$\mathsf{con}\ X_2, Y_2 \bullet$

$$x, y : [\ \mathit{true},\ x' = x - y \wedge y' = y\ ]; \qquad [\lhd]$$

$$x, y : \left[\begin{array}{ccc} x = X_2 - Y_2 & & x' = X_2 - Y_2 \\ y = Y_2 & , & y' = X_2 \end{array}\right] \qquad [\ddagger]$$

$\sqsubseteq$ (assignment introduction)

$x := x - y$

$\ddagger$

$\sqsubseteq$ (assignment introduction)

$y := x + y$

$\dagger$

$\sqsubseteq$ (assignment introduction)

$x := y - x$

Flattened

$$x, y : [\, x = X \wedge y = Y,\ x = Y \wedge y = X \,]$$

$$\sqsubseteq$$

$$\begin{array}{l}
\textsf{begin con } X_1, Y_1 : \mathbb{Z} \bullet \\
\quad \textsf{begin con } X_2, Y_2 \bullet \\
\qquad x := x - y \ ; \\
\qquad y := x + y \\
\quad \textsf{end} \ ; \\
\quad x := y - x \\
\textsf{end}
\end{array}$$

$$\sqsubseteq$$

$$x := x - y \;;$$

$$y := x + y \;;$$

$$x := y - x$$

Simple sequential composition

If the predicates *mid* and *post* make no reference to before variables, then

$$w, x : [\, pre,\ post \,]$$

$$\sqsubseteq \quad x : [\, pre,\ mid \,] \ ; \ \ w, x : [\, mid[x/x'],\ post \,]$$

Leading assignment

$$w, x : [\ pre[E/x],\ post[E/x]\ ]$$

$$\sqsubseteq \quad x := E\ ;\quad w, x : [\ pre,\ post\ ]$$

Following assignment

$$w, x : [\ pre,\ post\ ]$$

$$\sqsubseteq \quad w : [\ pre,\ post[E/x']\ ]\ ;\quad x := E$$

Conditional statements

$$
\begin{array}{ll}
\mathsf{if} & G_1 \rightarrow com_1 \\
\Box & G_2 \rightarrow com_2 \\
 & \vdots \\
\Box & G_n \rightarrow com_n \\
\mathsf{fi} &
\end{array}
$$

Conditional introduction

If

$$pre \Rightarrow (G_1 \vee G_2 \vee \ldots \vee G_n)$$

then

$$w : [\, pre,\ post \,] \quad \sqsubseteq \quad \begin{cases} \text{if} \quad G_1 \rightarrow w : [\, G_1 \wedge pre,\ post \,] \\ \Box \quad G_2 \rightarrow w : [\, G_2 \wedge pre,\ post \,] \\ \vdots \\ \Box \quad G_n \rightarrow w : [\, G_n \wedge pre,\ post \,] \\ \text{fi} \end{cases}$$

Example

$$x, y : \left[\begin{array}{c} x \leq y \wedge x' = x \wedge y' = y \\ \vee \\ true \ , \ y \leq x \wedge x' = y \wedge y' = x \end{array} \right]$$

$\sqsubseteq$ (conditional introduction)

$$\text{if} \quad x \leq y \rightarrow x, y : \left[\begin{array}{c} x \leq y \wedge x' = x \wedge y' = y \\ \vee \\ x \leq y \ , \ y \leq x \wedge x' = y \wedge y' = x \end{array}\right] \qquad [\lhd]$$

$$\Box \quad y \leq x \rightarrow x, y : \left[\begin{array}{c} x \leq y \wedge x' = x \wedge y' = y \\ \vee \\ y \leq x \ , \ y \leq x \wedge x' = y \wedge y' = x) \end{array}\right] \qquad [\dagger]$$

fi

$\sqsubseteq$ (strengthen postcondition)

$x, y : [\, x \leq y,\ x \leq y \wedge x' = x \wedge y' = y \,]$

$\sqsubseteq$ (skip introduction)

skip

$\dagger$

$\sqsubseteq$ (strengthen postcondition)

$x, y : [\, y \leq x,\ y \leq x \wedge x' = y \wedge y' = x \,]$

$\sqsubseteq$ (assignment introduction)

$x, y := y, x$

$$
\begin{array}{ll}
\textsf{if} & x \leq y \rightarrow \textsf{skip} \\
\Box & y \leq x \rightarrow x, y := y, x \\
\textsf{fi} &
\end{array}
$$

Contract frame

$$w, x : [\ pre,\ post\ ] \sqsubseteq w : [\ pre,\ post[x/x']\ ]$$

Expand frame

$$w : [\, pre,\ post \,] = w, x : [\, pre,\ post \wedge x' = x \,]$$

Iteration

$$
\begin{array}{ll}
\mathsf{do}\ G_1 & \rightarrow\ com_1 \\
\Box\ \ G_2 & \rightarrow\ com_2 \\
\quad\vdots & \\
\Box\ \ G_n & \rightarrow\ com_n \\
\mathsf{od} &
\end{array}
$$

Loop introduction

$$w : \left[\begin{array}{c} \neg G[w'/w] \\ inv \ , \ inv[w'/w] \end{array} \right]$$

$$\sqsubseteq \quad \left\{ \begin{array}{l} \mathsf{do} \\ \qquad G \rightarrow \\ \qquad\qquad w : \left[\begin{array}{c} G \qquad inv[w'/w] \\ inv \ , \ 0 \leq V[w'/w] < V \end{array} \right] \\ \mathsf{od} \end{array} \right.$$

Point at the target

$i : [\ target \in \operatorname{ran} s,\ s(i') = target\ ]$

$$
\begin{array}{l}
\mathsf{begin} \\
\quad \mathsf{con} I : 1 \mathinner{..} \#s \bullet \\
\qquad i : [\, s(I) = target,\ s(i') = target \,] \\
\mathsf{end}
\end{array}
$$

$$i : [\, s(I) = target,\ s(I) = target \wedge i' \leq I \,];$$
$$i : [\, s(I) = target \wedge i \leq I,\ s(i') = target \,]$$

$$i : \left[\begin{array}{ccc} & & s(i') = target \\ s(I) = target & & s(I) = target \\ i \leq I & , & i' \leq I \end{array} \right]$$

$$
\begin{array}{l}
\mathtt{do} \\
\quad s(i) \neq target \rightarrow \\
\qquad i : \left[\begin{array}{ll} s(I) = target & s(I) = target \\ i \leq I & i' \leq I \\ s(i) \neq target \ , & 0 \leq I - i' < I - i \end{array} \right] \\
\mathtt{od}
\end{array}
$$

$i := 1;$ do

$s(i) \neq target \rightarrow$

$i := i + 1$

od

An integer array

$$ar : (1..n) \rightarrow \mathbb{N}$$

$$Init \;=\; ar : [\, true, \operatorname{ran} ar' = \{0\} \,]$$

$$Init \;=\; ar : [\, true, \forall\, j : 1..n \bullet ar'\, j = 0 \,]$$

$$zeroed(i, ar) \;=\; \forall\, j : 1..i \bullet ar\, j = 0$$

$$ar : [\ true,\ zeroed(n, ar')\ ]$$

$$\sqsubseteq$$

$$\mathsf{var}\ \ j \mid 1 \leq j \leq n + 1 \bullet$$

$$\quad j, ar : [\ true,\ zeroed(n, ar')\ ]$$

$\sqsubseteq$ (simple sequential composition)

$$j, ar : [\ true,\ zeroed(j' - 1, ar')\ ]\ ; \qquad [\lhd]$$

$$j, ar : [\ zeroed(j - 1, ar),\ zeroed(n, ar')\ ] \qquad [\dagger]$$

$\sqsubseteq$ (assignment introduction)

$j := 1$

$\dagger$

$\sqsubseteq$ (strengthen postcondition)

$j, ar : [\, zeroed(j-1, ar),\ zeroed(j'-1, ar') \wedge j' = n+1 \,]$

$\sqsubseteq$ (loop introduction)

$\mathsf{do}\ j \neq n + 1 \rightarrow$

$$j, ar : \left[\begin{array}{cc} j \neq n + 1 & 0 \leq n - j' + 1 < n - j + 1 \\ zeroed(j - 1, ar)\ , & zeroed(j' - 1, ar') \end{array} \right]$$

od

$\sqsubseteq$ (following assignment)

$$ar : \left[\begin{array}{c} j \neq n + 1 \\ zeroed(j - 1, ar) \ , \ zeroed(j, ar') \end{array} \right] ;$$

[◁]

$j := j + 1$

$\sqsubseteq$ (assignment introduction)

$$ar := ar \oplus \{j \mapsto 0\}$$

$$
\begin{array}{l}
\mathit{Init} \\
\sqsubseteq \\
\quad \mathsf{begin} \\
\quad\quad \mathsf{var}\ j \mid 1 \leq j \leq n+1 \bullet \\
\quad\quad\quad j := 1\,; \\
\quad\quad\quad \mathsf{do}\ j \neq n+1 \rightarrow \\
\quad\quad\quad\quad ar := update(ar, j, 0)\,; \\
\quad\quad\quad\quad j := j+1 \\
\quad\quad\quad \mathsf{od} \\
\quad \mathsf{end}
\end{array}
$$

```
PROCEDURE Init ;
  BEGIN
    FOR j := 1 TO n DO ar[j] := 0
  END
```

Base conversion

10011100

$$1 * 2^7 + 0 * 2^6 + 0 * 2^5 + 1 * 2^4 + 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 0 * 2^0$$

$$= 128 + 16 + 8 + 4$$

$$= 156$$

Horner's rule

$$a_1 + a_2 * \beta + a_3 * \beta^2 + \cdots + a_n * \beta^{n-1}$$
$$=$$
$$a_1 + \beta * (a_2 + \beta * (\cdots \beta * (a_{n-2} + \beta * (a_{n-1} + \beta * a_n)) \cdots))$$

Recurrence relation

$\sum_{i=1}^{n} a_i * \beta^{i-1} = H_{1,n}$

where

$H_{n,n} = a_n$
$H_{i,n} = a_i + \beta * H_{i+1,n}$ for $i < n$

$a_n a_{n-1} \ldots a_2 a_1$

$d : [\ true,\ d' = \sum_{i=1}^{n} a_i * \beta^{i-1}\ ]$

$d : [\ true,\ d' = H_{1,n}\ ]$

$d : [\ true,\ d' = H_{1,n}\ ]$

$\sqsubseteq$

$\quad \mathsf{var}\ j : 1 \mathinner{..} n \bullet$

$\quad d, j : [\ true,\ d' = H_{1,n}\ ]$

$\sqsubseteq$ (sequential composition introduction)

$\quad d, j : [\ true,\ d' = H_{j',n}\ ];$ $[\lhd]$

$\quad d, j : [\ d = H_{j,n},\ d' = H_{1,n}\ ]$ $[\dagger]$

$\sqsubseteq$ (assignment introduction)

$\quad d, j := a_n, n$

$\dagger$

$\sqsubseteq$ (strengthen postcondition)

$$d, j : [\ d = H_{j,n},\ d' = H_{1,n} \wedge j' = 1\ ]$$

$\sqsubseteq$ (strengthen postcondition)

$$d, j : [\ d = H_{j,n},\ d' = H_{j',n} \wedge j' = 1\ ]$$

$\sqsubseteq$ (loop introduction)

do

$j \neq 1 \rightarrow$

$$d, j : \left[\begin{array}{cc} j \neq 1 & 0 \leq j' < j \\ d = H_{j,n} \ , & d' = H_{j',n} \end{array} \right]$$

od

$$d, j : [\ \begin{pmatrix} j \neq 0 \\ d = H_{j+1,n} \end{pmatrix} [j - 1/j],\ \begin{pmatrix} 0 \leq j' \leq j \\ d' = H_{j',n} \end{pmatrix} [j - 1/j]\ ]$$

$\sqsubseteq$ (leading assignment)

$$j := j - 1\,;$$

$$d, j : \left[\begin{array}{cc} j \neq 0 & 0 \leq j' \leq j \\ d = H_{j+1,n}\ , & d' = H_{j',n} \end{array}\right]$$

$\sqsubseteq$ (contract frame)

$$d : \left[\begin{array}{cc} j \neq 0 & 0 \leq j \leq j \\ d = H_{j+1,n}\ , & d' = H_{j,n} \end{array}\right]$$

$\sqsubseteq$ (strengthen postcondition)

$$d : [\, j \neq 0 \wedge d = H_{j+1,n},\ d' = a_j + \beta * H_{j+1,n} \,]$$

$\sqsubseteq$ (strengthen postcondition)

$$d : [\, j \neq 0 \wedge d = H_{j+1,n},\ d' = a_j + \beta * d \,]$$

$\sqsubseteq$ (assignment introduction)

$$d := a_j + \beta * d$$

```
begin
    var j : 1 .. n •
        d, j := a_n, n;
        do j ≠ 1 →
            j := j − 1;
            d := a_j + x ∗ d
        od
end
```

```
PROCEDURE Translate ;
  BEGIN
    d := a[n] ;
    FOR j := n DOWNTO 1 DO
      d := a[j] + x * d
  END
```