

## 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* ]

Jsing Z

19-4

## Example

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

## Example

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

Ising Z

19-6

## Guarded commands

```
if . . . fi  
do . . . od  
begin . . . end  
,
```

**Abort**

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

Jsing Z

19-8

**Choose**

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

**Skip**

`skip = [ true, true ]`

Ising Z

19-10

**Magic**

`magic = w:[ true, false ]`

## Assignment

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

Ising Z

19-12

## 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

$$w : [ \text{pre}, \text{post}_2 ] \sqsubseteq w : [ \text{pre}, \text{post}_1 ]$$

then

$$w : [ \text{pre}, \text{post}_1 ] \sqsubseteq w : [ \text{pre}, \text{post}_2 ]$$

Ising Z

19-14

### Example

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

$\sqsubseteq$  (strengthen postcondition)

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

## Weaken precondition

If

$$pre_1 \Rightarrow pre_2$$

then

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

Ising Z

19-16

## Example

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

$\sqsubseteq$  (weaken precondition)

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

## Feasibility check

$$\textit{pre} \Rightarrow \exists x' : X; y' : Y \bullet \textit{post}$$

Ising Z

19-18

## Infeasible

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

## Introduce local block

If  $x$  does not appear in  $w$ , then

```
w : [ pre, post ]  ⊢ begin
    var x : T | inv •
    w, x : [ pre, post ]
end
```

Jsing Z

19-20

## Example

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

⊐ (introduce local block)

```
begin
    var x, y •
    x, y, m : \left[ \begin{array}{l} f(a) * f(b) \leq 0 \quad -0.1 < f(m') < 0.1 \\ a \leq b \quad , \quad a \leq m' \leq b \end{array} \right]
end
```

## Assignment introduction

If

$$\text{pre} \Rightarrow \text{post}[E/w', x/x']$$

then

$$w, x : [ \text{pre}, \text{post} ] \sqsubseteq w := E$$

Jsing Z

19-22

## Example

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

## Skip introduction

If

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

then

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

Jsing Z

19-24

## Example

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

$\sqsubseteq \text{skip}$

## Logical constants

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

begin con  $X : T \bullet prog$  end

Ising Z

19-26

## Example

```
begin
  con  $X \bullet x : [x = X, x' > X]$ 
end;
begin
  con  $X \bullet x : [x = X, x' > X]$ 
end
```

## Scope

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

Ising Z

19-28

## Angelic nondeterminism

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

```
begin con X • x : [ true, x' > x ] end
```

## Introduce logical constant

If

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

and C is a fresh name, then

$$w : [ pre_1, post ]$$

$$\equiv \begin{cases} \text{begin} \\ \quad \text{con } C : T \bullet w : [ pre_2, post ] \\ \text{end} \end{cases}$$

Ising Z

19-30

## Eliminate logical constant

If C occurs nowhere in *prog*, then

$$\begin{aligned} \text{begin} \\ \quad \text{con } C : T \bullet prog \\ \text{end} \end{aligned} \quad \equiv \quad prog$$

## C-style constants

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

Ising Z

19-32

## Sequential composition

```
w, x : [ pre, post ]
x : [ pre, mid ]
w, x : [ mid[X/x, x/x'], post[X/x] ]
```

## Sequential composition introduction

$w, x : [ \text{pre}, \text{post} ]$

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

Ising Z

19-34

## Example

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

$\sqsubseteq$  (sequential composition introduction)

$\text{con } X \bullet$

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

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

## Swapping the hard way

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

$\sqsubseteq$  (sequential composition introduction)

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

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

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

Ising Z

19-36

$\sqsubseteq$  (sequential composition introduction)

con  $X_2, Y_2 \bullet$

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

$x, y : \left[ \begin{array}{ll} x = X_2 - Y_2 & x' = X_2 - Y_2 \\ y = Y_2 & , y' = X_2 \end{array} \right] ;$  [ $\#$ ]

$\sqsubseteq$  (assignment introduction)

$x := x - y$

$\dagger$  $\sqsubseteq$  (assignment introduction) $y := x + y$  $\ddagger$  $\sqsubseteq$  (assignment introduction) $x := y - x$ 

19-38

Ising Z

## Flattened

 $x, y : [x = X \wedge y = Y, x = Y \wedge y = X]$  $\sqsubseteq$ begin con  $X_1, Y_1 : \mathbb{Z}$  •begin con  $X_2, Y_2 : \mathbb{Z}$  • $x := x - y ;$  $y := x + y$ 

end ;

 $x := y - x$ 

end

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

Ising Z

19-40

## Simple sequential composition

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

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

## Leading assignment

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

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

Ising Z

19-42

## Following assignment

$w, x : [ \text{pre}, \text{post} ]$

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

## Conditional statements

```

if   G1 → com1
    G2 → com2
    :
fi
Gn → comn

```

Sing Z

19-44

## Conditional introduction

If

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

then

$$w : [ \text{pre}, \text{post} ] \quad \equiv \quad \left\{ \begin{array}{l} \text{if } G_1 \rightarrow w : [ G_1 \wedge \text{pre}, \text{post} ] \\ \quad \square \quad G_2 \rightarrow w : [ G_2 \wedge \text{pre}, \text{post} ] \\ \quad \vdots \\ \quad \square \quad G_n \rightarrow w : [ G_n \wedge \text{pre}, \text{post} ] \\ \text{fi} \end{array} \right.$$

## Example

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

Ising Z

19-46

$\sqsubseteq$  (conditional introduction)

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

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

fi

$\sqsubseteq$  (strengthen postcondition)

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

$\sqsubseteq$  (skip introduction)

skip

Ising Z

19-48

$\dagger$

$\sqsubseteq$  (strengthen postcondition)

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

$\sqsubseteq$  (assignment introduction)

$x, y := y, x$

```
if  $x \leq y \rightarrow \text{skip}$ 
 $\square y \leq x \rightarrow x, y := y, x$ 
fi
```

Ising Z

19-50

## Contract frame

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

## Expand frame

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

Ising Z

19-52

## Iteration

```
do G1 → com1
  □ G2 → com2
  ⋮
  □ Gn → comn
od
```

## Loop introduction

$$\begin{aligned} w : & \left[ \neg G[w'/w] \right. \\ & \left. \text{inv , } \text{inv}[w'/w] \right] \\ \sqsubseteq & \left\{ \begin{array}{l} \text{do} \\ \quad G \rightarrow \\ \quad w : \left[ \begin{array}{ll} G & \text{inv}[w'/w] \\ \text{inv , } & 0 \leq V[w'/w] < V \end{array} \right] \\ \text{od} \end{array} \right\} \end{aligned}$$

Ising Z

19-54

## Point at the target

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

```
begin
  conI : 1 .. #s •
    i : [ s(I) = target, s(i') = target ]
end
```

Ising Z

19-56

```
i : [ s(I) = target, s(I) = target ∧ i' ≤ I ];
i : [ s(I) = target ∧ i ≤ I, s(i') = target ]
```

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

```
do  
  s(i) ≠ target →  
    s(I) = target      s(I) = target  
    i: [                ]  
      i ≤ I            i' ≤ I  
      s(i) ≠ target , 0 ≤ I - i' < I - i  
    od
```

Ising Z

19-58

```
i := 1; do  
  s(i) ≠ target →  
    i := i + 1  
  od
```

## An integer array

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

Ising Z

19-60

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

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

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

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

$\sqsubseteq$

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

$j, ar : [ \text{true}, \text{zeroed}(n, ar') ]$

Ising Z

19-62

$\sqsubseteq$  (simple sequential composition)

$j, ar : [ \text{true}, \text{zeroed}(j' - 1, ar') ] ;$

$j, ar : [ \text{zeroed}(j - 1, ar), \text{zeroed}(n, ar') ]$

[ $\lhd$ ]

[ $\vdash$ ]

$\sqsubseteq$  (assignment introduction)

$j := 1$

Ising Z

19-64

$^+$

$\sqsubseteq$  (strengthen postcondition)

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

$\sqsubseteq$  (loop introduction)

do  $j \neq n + 1 \rightarrow$

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

od

Ising Z

19-66

$\sqsubseteq$  (following assignment)

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

[ $\lhd$ ]

$j := j + 1$

$\sqsubseteq$  (assignment introduction)

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

*Init*  
 $\sqsubseteq$   
begin

var  $j \mid 1 \leq j \leq n + 1$  •  
 $j := 1;$   
do  $j \neq n + 1 \rightarrow$   
     $ar := update(ar, j, 0);$   
     $j := j + 1$   
od  
end

Ising Z

19-68

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

## Base conversion

10011100

$$\begin{aligned}1 * 2^7 + 0 * 2^6 + 0 * 2^5 + 1 * 2^4 + \\1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 0 * 2^0 \\= 128 + 16 + 8 + 4 \\= 156\end{aligned}$$

Ising Z

19-70

## Horner's rule

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

## 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  $i < n$

Ising Z

19-72

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

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

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

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

$\sqsubseteq$

$\text{var } j : 1 \dots n \bullet$

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

$\sqsubseteq$  (sequential composition introduction)

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

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

$\sqsubseteq$  (assignment introduction)

$d, j := a_n, n$

Ising Z

19-74

$\dagger$

$\sqsubseteq$  (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

$j \neq 1 \rightarrow$

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

od

Ising Z

19-76

$d,j : \left[ \begin{array}{l} j \neq 0 \\ d = H_{j+1,n} \end{array} \right] \sqcup [j - 1/j], \left( \begin{array}{l} 0 \leq j' \leq j \\ d' = H_{j',n} \end{array} \right) \sqcup [j - 1/j]$

$\sqsubseteq$  (leading assignment)

$j := j - 1;$

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

$\sqsubseteq$  (contract frame)

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

Ising Z

19-78

$\sqsubseteq$  (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 := an,n;
  do j ≠ 1 →
    j := j - 1;
    d := aj + x * d
  od
end
```

Jsing Z

19-80

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