Vienna Development Method for Specification Language (VDM-SL)

Formal Methods in Software Engineering

VDM-SL

- Formal Method
- IMB Laboratory Vienna
- System description (functionality performed on data)
- Data => State
- Various Components
  - VDM++ for object-oriented and concurrent systems
  - Specification Language

Data Types

Data Type	DT	Description	Example Values
bool	B	Boolean datatype	false, true
nat	N	natural numbers (including zero)	0, 1, 2, 3, 4, 5 ...
nat1	N1	natural numbers (excluding zero)	1, 2, 3, 4, 5, ...
int	Z	integers	..., −3, −2, −1, 0, 1, 2, 3, ...
rat		rational numbers	a/b, where a and b are integers, b is not 0
real	R	real numbers	...
char	char	characters	A, B, C, ...
token		structureless tokens	...
<A>		the quote type containing the value <A>	...

Examples

- Types
  - UserId = nat
  - CGPA = real
- Invarients
  - UserId = nat
  - inv uid == uid <= 9999
  - CGPA = real
  - CGPA<=4

Basic Type Constructors

Construct	Description
T1 | T2 | ... | Tn	Union of types T1,...,Tn
T1*T2*...*Tn	Cartesian product of types T1,...,Tn
T :: f1:T1 ... fn:Tn	Composite (Record) type

Examples

- SignalColour = <Red> | <Yellow> | <Green>
- Subjects= <Formal Methods> | <Algorithms> | <Subject A>
- Composit (Field:Type)
  - T :: f1:A1
         f2:A2
         ...
         fn:An
  - Date :: day:nat1
            month:nat1
            year:nat
    inv mk_Date(d,m,y) == d <=31 and m<=12

Composit (Field:Type)

- T :: f1:A1
  f2:A2
  ...
  fn:An
- Date :: day:nat1
  month:nat1
  year:nat
  inv mk_Date(d,m,y) == d <=31 and m<=12
- mk_Date(22,2,2023)
  - day=22
  - month=2
  - year=2023
- mk_Date(22,2,2023).month =>2

Case Study: Incubator

- Temperature (-10,+10)
- Increase()
- Decrease()
- View/display()

System State

- state <SystemName> of
- <StateName> : <Type> end

- Example
  - **state** IncubatorSystem **of**
  - temp : int/Z
  - **end**

Operations

- the operation header
- ext wr: the external clause
- pre: the precondition
- post: the postcondition
- Example
  - increment()
  - ext wr temp: Z
  - pre temp<10
  - post temp=temp+1

Example

- decrement()
- ext wr temp: Z
- pre temp>-10
- post temp=temp-1

Example 2: Quiz Evaluation

- As a home task

Constants

- Constant_Name Type : Value
- MIN Z:-10
- MAX Z:10
- Pre Condition
- Temp<MIN

Functions: Explicit

- Signature
  - Function_name Inputs_types → Output_type
- Definition
  - Function_name(inputs) Δ output
- Add: $R\ X\ R\ \rightarrow\ R$
- Add(x,y) Δ x + y

Functions: implicit

- Pre conditions
- Post conditions
- Add(x:R,y:R) z:R
- Pre True
- Post $z = x + y$

Example

- Abs(x:Z) r:N
- Pre True
- Post $x < 0 \wedge r = -x \ \vee x \geq 0 \wedge r = x$

- $x < 0 \wedge r = -x$
- OR
- $x \geq 0 \wedge r = x$

Example: Explicit

- Abs: $Z \rightarrow N$
- Abs(x) Δ
  - if x<0
  - Then –x
  - Else x

Example: Explicit

- factorial: N → N
- factorial(n) Δ
  - if n=0
  - then 1
  - else n*factorial(n-1)

State Invariant

- Global Constraint
- Inv: $State \rightarrow Boolean$
- inv mk-IncubatorMonitor(t) Δ $\mathrm{MIN} \leq t \leq \mathrm{MAX}$

Incubator Controller

- requestedTemp : Integer
- actualTemp : Integer

- setInitialTemp(Integer)
- requestChange(Integer) : Signal
- increment() : Signal
- decrement() : Signal
- getRequestedTemp() : Integer
- getActualTemp() : Integer

Signal

- <<enumeration>>
- Signal
- INCREASE
- DECREASE
- DO_NOTHING

- types
  - Signal = <INCREASE>|<DECREASE>|<DO_NOTHING>

State

- **state** IncubatorController **of**
- requestedTemp : [Z]
- actualTemp : [Z]

Constructor

- inv mk-IncubatorController (r, a) Δ
  - $MIN \leq r \leq MAX \vee r = nil \wedge$
  - $MIN \leq a \leq MAX \vee a = nil$
- init mk-IncubatorController (r, a) Δ $\mathrm{r} = \mathrm{nil} \wedge a = nil$

Is Range

- inRange(val: Z) result: B
- pre TRUE
- post $result \Leftrightarrow MIN \leq val \leq MAX$

- inv mk-IncubatorController(r, a) Δ
- (inRange(r)v r=nil) ∧
- (inRange(a) v a=nil)

- setInitialTemp(tempIn : Z)
- ext wr actualTemp : [Z]
- pre inRange(tempIn) v actualTemp = nil
- post actualTemp = tempIn

Change

- requestChange(tempIn : Z) signalOut : Signal
- Ext
  - wr requestedTemp : [Z]
  - rd actualTemp : [Z]
- pre inRange(tempIn) ∧ actualTemp ≠ nil
- Post
  - requestedTemp tempIn ∧
  - (tempIn > actualTemp ∧ signalOut = <INCREASE>
  - ∨
  - tempIn actualTemp ∧ signalOut = <DECREASE>
  - ∨
  - tempIn actualTemp ∧ signalOut = <DO_NOTHING>)

increment

- increment () signalOut : Signal
- Ext
  - rd requestedTemp : [Z]
  - wr actualTemp : [Z]
- Pre
  - actualTemp < requestedTemp ∧
  - actualTemp ≠ nil ∧ requestedTemp ≠ nil
- Post
  - actualTemp = actualTemp + 1 ∧
  - (actualTemp < requestedTemp ∧
  - signalOut = <INCREASE> ∨
  - actualTemp = requestedTemp ∧
  - signalOut = <DO_NOTHING>)

Comments

- -- comments

Complete System

- Book FORMAL SOFTWARE DEVELOPMENT Chapter 3
  - Section 3.17
  - Page 41 to 43