6SENG005W Formal Methods

Summary of the

B-Method's

Abstract Machine Notation (AMN)

B-Method's Abstract Machine Notation (AMN)

The B-Method's notation (mathematical symbols, pseudo-code and B machines) is called *Abstract Machine Notation* (AMN).

The two B tools used on the module **Atelier B** and **ProB** require ASCII versions of B's AMN.

The following tables present AMN in two versions: a "pretty printed" symbol version & an ASCII version that is machine readable by the two B tools.

1. *Numbers*
2. *Sets*
3. *Logic*
4. *Ordered Pairs & Relations*
5. *Functions*
6. *Sequences*
7. *B Machine "clauses"*

AMN: Number Types & Operators

B Symbol	ASCII	Description
$\mathbb{N}$	`NAT`	Set of natural numbers from 0
$\mathbb{N}_1$	`NAT1`	Set of natural numbers from 1
$\mathbb{Z}$	`INTEGER`	Set of integers
$\mathrm{pred}(x)$	`pred(x)`	predecessor of x
$\mathrm{succ}(x)$	`succ(x)`	successor of x
$x + y$	`x + y`	x plus y
$x - y$	`x - y`	x minus y
$x \times y$	`x * y`	x multiply y
$x \div y$	`x / y`	x divided by y
$x \bmod y$	`x mod y`	remainder after x divided by y
$x \mathbin{\widehat{}} y$	`x ** y`	x to the power y, x^y
$\min(A)$	`min( A )`	minimum number in set A
$\max(A)$	`max( A )`	maximum number in set A
$x \mathinner{..} y$	`x .. y`	range of numbers from x to y inclusive

AMN: Number Relations

B Symbol	ASCII	Description
$x = y$	`x = y`	x equal to y
$x \neq y$	`x /= y`	x not equal to y
$x < y$	`x < y`	x less than y
$x \leq y$	`x <= y`	x less than or equal to y
$x > y$	`x > y`	x greater than y
$x \geq y$	`x >= y`	x greater than or equal to y

AMN: Set Definitions

B Symbol	ASCII	Description
$x \in A$	`x : A`	x is an element of set A
$x \notin A$	`x /: A`	x is not an element of set A
$\varnothing, \{\,\}$	`{}`	Empty set
$\{\,1\,\}$	`{ 1 }`	Singleton set (1 element)
$\{\,1, 2, 3\,\}$	`{ 1, 2, 3 }`	Set of elements: 1, 2, 3
$x\,..\,y$	`x .. y`	Range of integers from x to y inclusive
$\mathbb{P}(A)$	`POW(A)`	Power set of A
$\mathrm{card}(A)$	`card(A)`	Cardinality, number of elements in set A

AMN: Set Operators & Relations

B Symbol	ASCII	Description
$A \cup B$	`A \/ B`	Union of A and B
$A \cap B$	`A /\ B`	Intersection of A and B
$A - B$	`A - B`	Set subtraction of A and B
$\bigcup AA$	`union( AA )`	Generalised union of a set of sets AA
$\bigcap AA$	`inter( AA )`	Generalised intersection of a set of sets AA
$A \subseteq B$	`A <: B`	A is a subset of or equal to B
$A \not\subseteq B$	`A /<: B`	A is not a subset of or equal to B
$A \subset B$	`A <<: B`	A is a strict subset of B
$A \not\subset B$	`A /<<: B`	A is not a strict subset of B
$\{x \mid x \in TS \wedge C\}$	`{ x | x : TS & C }`	Set comprehension

AMN: Logic

B Symbol	ASCII	Description
$\neg P$	`not P`	Logical negation (not) of P
$P \wedge Q$	`P & Q`	Logical and of P, Q
$P \vee Q$	`P or Q`	Logical or of P, Q
$P \Rightarrow Q$	`P => Q`	Logical implication of P, Q
$P \Leftrightarrow Q$	`P <=> Q`	Logical equivalence of P, Q
$\forall\, xx \cdot (P \Rightarrow Q)$	`!(xx).(P => Q)`	Universal quantification of xx over $(P \Rightarrow Q)$
$\exists\, xx \cdot (P \wedge Q)$	`#(xx).(P & Q)`	Existential quantification of xx over $(P \wedge Q)$
$TRUE$	`TRUE`	Truth value $TRUE$.
$FALSE$	`FALSE`	Truth value $FALSE$
$BOOL$	`BOOL`	Set of boolean values $\{ TRUE, FALSE \}$
$bool(P)$	`bool(P)`	Convert predicate P into $BOOL$ value

AMN: Ordered Pairs & Relations

B Symbol	ASCII	Description
$X \times Y$	`X * Y`	Cartesian product of X and Y
(x, y),	`x |-> y`	Ordered pair
$x \mapsto y$	`x |-> y`	Ordered pair, (maplet)
$\mathrm{prj}_1(S,T)(x,y)$	`prj1(S,T)(x, y)`	Ordered pair projection function
$\mathrm{prj}_2(S,T)(x,y)$	`prj2(S,T)(x, y)`	Ordered pair projection function
$\mathbb{P}(X \times Y)$	`POW(X * Y)`	Set of relations between X and Y
$X \leftrightarrow Y$	`X <-> Y`	Set of relations between X and Y
$\mathrm{dom}(R)$	`dom(R)`	Domain of relation R
$\mathrm{ran}(R)$	`ran(R)`	Range of relation R

AMN: Relations Operators

B Symbol	ASCII	Description
$A \lhd R$	`A <| R`	Domain restriction of R to the set A
$A \lhd\!\!\!\!- R$	`A <<| R`	Domain subtraction of R by the set A
$R \rhd B$	`R |> B`	Range restriction of R to the set B
$R \rhd\!\!\!\!- B$	`R |>> B`	Range anti-restriction of R by the set B
$R[B]$	`R[B]`	Relational Image of the set B of relation R
$R_1 <\!\!+ R_2$	`R1 <+ R2`	R_1 overridden by relation R_2
$R ; Q$	`R ; Q`	Forward Relational composition
$\mathrm{id}(X)$	`id(X)`	Identity relation
R^{-1}	`R~`	Inverse relation
R^n	`iterate(R,n)`	Iterated Composition of R
R^+	`closure1(R)`	Transitive closure of R
R^*	`closure(R)`	Reflexive-transitive closure of R

AMN: Functions

B Symbol	ASCII	Description
$X \pfun Y$	`X +-> Y`	Partial function from X to Y
$X \fun Y$	`X --> Y`	Total function from X to Y
$X \pinj Y$	`X >+> Y`	Partial injection from X to Y
$X \inj Y$	`X >-> Y`	Total injection from X to Y
$X \psur Y$	`X +->> Y`	Partial surjection from X to Y
$X \surj Y$	`X -->> Y`	Total surjection from X to Y
$X \bij Y$	`X >->> Y`	(Total) Bijection from X to Y
$f \ovl g$	`f <+ g`	Function f overridden by function g

AMN: Sequences

B Symbol	ASCII	Description
$[\,]$	`[]`	Empty Sequence
$[\,e1\,]$	`[ e1 ]`	Singleton Sequence
$[\,e1,\ e2\,]$	`[ e1, e2 ]`	Constructed (enumerated) Sequence
$\mathrm{seq}(X)$	`seq( X )`	Set of Sequences over set X
$\mathrm{seq}_1(X)$	`seq1( X )`	Set of non-empty Sequences over set X
$\mathrm{iseq}(X)$	`iseq( X )`	Set of injective Sequences over set X
$\mathrm{iseq}_1(X)$	`iseq1( X )`	Set of non-empty injective Sequences over set X
$\mathrm{size}(s)$	`size( s )`	Size (length) of Sequence s

AMN: Sequences Operators

B Symbol	ASCII	Description
$s \frown t$	`s^t`	Concatenation of Sequences s & t
$e \rightarrow s$	`e -> s`	Append element e to front of sequences s
$s \leftarrow e$	`s <- e`	Append element e to tail of sequences s
$rev(s)$	`rev( s )`	Reverse of sequence s
$first(s)$	`first( s )`	First element of sequence s
$last(s)$	`last( s )`	Last element of sequence s
$front(s)$	`front( s )`	Front of sequence s, excluding last element
$tail(s)$	`tail( s )`	Tail of sequence s, excluding first element
$conc(SS)$	`conc(SS)`	Concatenation of sequence of sequences SS
$s \uparrow n$	`s /| n`	Take first n elements of sequence s
$s \downarrow n$	`s |/ n`	Drop first n elements of sequence s

B AMN Operation Statements

Operation Statements

```
xx := E

xx1, xx2, .., xxn := E1, E2, .., En

skip

BEGIN S END

PRE PC THEN S END

IF ( B ) THEN S END
IF ( B ) THEN S ELSE T END
IF ( B1 ) THEN S ELSIF ( B2 ) THEN T ELSE U END

CASE E OF EITHER
         val1 THEN S1
  OR     val2 THEN S2
  OR     val3 THEN S3
  ...
  ELSE Sn
END

S || T
```

B Machine Clauses I

Name

```
MACHINE Name( mparam1, ..., mparamN )

  CONSTRAINTS          CP1 & CP2 & ..... & CPn

  SETS                 SS1 ; SS2 ; ... ; SSn
  CONSTANTS            C1, C2, ... , Cn
  PROPERTIES           PR1 & PR2 & ... & PRn

  VARIABLES            var1, var2, ..., varn
  INVARIANT            INV1 & INV2 & ... & INVn
  INITIALISATION       var1, ..., varn  :=  val1, ..., valn

  OPERATIONS

  output1, .., outputM <-- operation_1( param1, .., paramN )
     =  PRE   PreCondition
        THEN
               Substitution
        END

    ...
END
```

B Machine Clauses II

Clause	Purpose
MACHINE	declaration of the abstract machine's name and the list of eventual parameters of the machine
INCLUDES	when an abstract machine includes another abstract machine it integrates the data of the included machine
PROMOTES	when an abstract machine includes another abstract machine it can use the PROMOTES clause to selectively add any of its operations to its interface, by promoting them.
EXTENDS	when an abstract machine extends another abstract machine it integrates the data of the included machine & makes all of its operations part of its interface.
SEES	when a component sees an abstract machine, it refers to this machine and can consult its data and use the operations which do not modify these data.
USES	is a generalisation of the SEES relationship & the using machine has the same access to the used machine. Plus the using machine can refer to the used machine's state variables in its invariant.

B Machine Clauses III

Clause	Purpose
CONSTRAINTS	declaration of the properties of the abstract machine's parameters
SETS	declaration of deferred and enumerated sets
CONSTANTS	declaration of constants
PROPERTIES	declaration of the sets's & constants's types & properties
DEFINITIONS	declaration of textual definitions which will be expanded in the component before any further analysis
VARIABLES	declaration of variables
INVARIANT	declaration of invariant properties of the variables
INITIALISATION	initialisation of variables
OPERATIONS	declaration of the operations in the form of a header and body