

CSE4213

Formal Methods in Software Engineering

Formal Definitions of Relations, Functions and Sequences

John Hurst

20050317 / Lecture 6

Contents

1	Relations	1
2	Functions	6
3	Sequences	9

1 Relations

Relations

- Relations are the building blocks of data representation in B
- Relations are mappings from an argument set to a result set

Formal Definitions

- Have seen an informal introduction to the role of relations, and their subsets, functions
- Here develop formal definitions of relations and their usage
- Excellent example of set definition by comprehension

Relations

ASCII $S \leftrightarrow T$

Publication $S \leftrightarrow T$

L^AT_EX $\$ S \ \rel \ T \$$

formal $S \leftrightarrow T = \mathbb{P}(S \times T)$

Domains

ASCII `dom(r)`

Publication `dom(r)`

L^AT_EX `\dom(r)`

formal $\forall r \cdot r \in S \leftrightarrow T \Rightarrow \text{dom}(r) = \{x \mid x \in S \wedge (\exists y \cdot y \in T \wedge x \mapsto y \in r)\}$

comment If $r \in S \leftrightarrow T$ then $\text{dom}(r) \subseteq S$

Ranges

ASCII `ran(r)`

Publication `ran(r)`

L^AT_EX `\ran(r)`

formal $\forall r \cdot r \in S \leftrightarrow T \Rightarrow \text{ran}(r) = \{y \mid y \in T \wedge (\exists x \cdot x \in S \wedge x \mapsto y \in r)\}$

comment If $r \in S \leftrightarrow T$ then $\text{ran}(r) \subseteq T$

Forward Composition

ASCII `p ; q`

Publication `p ; q`

L^AT_EX `p \comp q`

formal $\forall p, q \cdot p \in S \leftrightarrow T \wedge q \in T \leftrightarrow U \Rightarrow p ; q = \{x, y \mid (\exists z \cdot x \mapsto z \in p \wedge z \mapsto y \in q)\}$

comment relations between S and T , and T and U , can be used to build relations between S and U through T

Backward Composition

ASCII `p circ q`

Publication `p o q`

L^AT_EX `p \circ q`

formal $p \circ q = q ; p$

comment reverse composition; useful where the nature of p and q suggests applying q first

Identity

ASCII id(S)

Publication id(S)

L^AT_EX $\$ \backslash id(S) \$$

formal $id(S) = \{x, y \mid x \in S \wedge y \in S \wedge x = y\}$

comment transform a set into itself

Domain Restriction

ASCII S <| r

Publication $S \triangleleft r$

L^AT_EX $\$ S \backslash dres r \$$

formal $S \triangleleft r = \{x, y \mid x \mapsto y \in r \wedge x \in S\}$

comment subset a relation r so that its domain is a subset of the given set S

Domain Subtraction

ASCII S <<| r

Publication $S \triangleleft\!\!\triangleleft r$

L^AT_EX $\$ S \backslash ndres r \$$

formal $S \triangleleft\!\!\triangleleft r = \{x, y \mid x \mapsto y \in r \wedge x \notin S\}$

comment subset a relation r so that its domain is mutually exclusive with the given set S

Range Restriction

ASCII r |> T

Publication $r \triangleright T$

L^AT_EX $\$ r \backslash rres T \$$

formal $r \triangleright T = \{x, y \mid x \mapsto y \in r \wedge y \in T\}$

comment subset a relation r so that its range is a subset of the given set T

Range Subtraction

ASCII `r |>> T`

Publication $r \triangleright T$

L^AT_EX `$ r \nrres T $`

formal $r \triangleright T = \{x, y \mid x \mapsto y \in r \wedge y \notin T\}$

comment subset a relation r so that its range is mutually exclusive with the given set T

Inverse

ASCII `r~`

Publication r^{-1}

L^AT_EX `$ \inv r $`

formal $r^{-1} = \{y, x \mid x \mapsto y \in r\}$

comment invert a relation r so that its range and domain are swapped

Relational Image

ASCII `r[S]`

Publication $r[S]$

L^AT_EX `$ r[S] $`

formal $r[S] = \{y \mid \exists x \cdot x \in S \wedge x \mapsto y \in r\}$

comment relational image applies the relation to each element of a set (in the domain), to build a new set (in the range)

Right Overriding

ASCII `r1 <+ r2`

Publication $r_1 \triangleleft r_2$

L^AT_EX `$ r_1 \rovr r_2 $`

formal $r_1 \triangleleft r_2 = r_2 \cup (\text{dom}(r_2) \triangleleft r_1)$

comment Build a new relation by removing all relations in the domain of r_2 from r_1 , and replacing them with the relations in r_2 . Often r_2 consists of just a single maplet.

Left Overriding

ASCII $r_1 +> r_2$

Publication $r_1 \triangleright r_2$

L^AT_EX $\$ r_1 \ \lovr r_2 \$$

formal $r_1 \triangleright r_2 = r_1 \cup (\text{dom}(r_1) \triangleleft r_2)$

comment Build a new relation by removing all relations in the domain of r_1 from r_2 , and replacing them with the relations in r_1 . Often r_1 consists of just a single maplet.

Direct Product

ASCII $p >< q$

Publication $p \otimes q$

L^AT_EX $\$ p \ \otimes q \$$

formal $p \otimes q = \{x, (y, z) \mid x \mapsto y \in p \wedge x \mapsto z \in q\}$

comment a relation returning pairs of values in the ranges of p and q

Parallel Product

ASCII $p \parallel q$

Publication $p \parallel q$

L^AT_EX $\$ p \ \parallel q \$$

formal $p \parallel q = \{(x, y), (m, n) \mid x \mapsto m \in p \wedge y \mapsto n \in q\}$

comment a relation from domain pairs in p, q to range pairs in p, q

Iteration

ASCII $\text{iterate}(r, n)$

Publication r^n

L^AT_EX $\$ r^{\wedge}n \$$

formal $r \in S \leftrightarrow S \Rightarrow r^0 = \text{id}(S) \wedge r^{n+1} = r; r^n$

comment repeatedly apply a relation; both domain and range must be of the same type.

Closure

ASCII `closure(r)`

Publication r^*

L^AT_EX $\$ r^* \$$

formal $r^* = \bigcup n \cdot (n \in \mathbb{N} \mid r^n)$

comment repeatedly apply a relation, saving all the generated sets until no new elements are added

Projection

ASCII `prj1(S, T)`

Publication $\text{prj1}(S, T)$

L^AT_EX $\$ \backslash\text{PRJx}(S, T) \$$

formal $\text{prj1}(S, T) = \{(x, y), z \mid x, y \in S \times T \wedge z = x\}$

comment extract the left hand element of a maplet

Projection

ASCII `prj2(S, T)`

Publication $\text{prj2}(S, T)$

L^AT_EX $\$ \backslash\text{PRJy}(S, T) \$$

formal $\text{prj2}(S, T) = \{(x, y), z \mid x, y \in S \times T \wedge z = y\}$

comment extract the right hand element of a maplet

2 Functions

Functions

- Functions are special cases of relations
- Each element in the domain must have at most one maplet into the range

Partial Functions

ASCII `S +-> T`

Publication $S \leftrightarrow T$

L^AT_EX $\$ S \backslash\text{pfun } T \$$

formal $S \leftrightarrow T = \{r \mid r \in S \leftrightarrow T \wedge r^{-1}; r \subseteq \text{id}(T)\}$

comment a subset of relations where every element of the domain has at most one element in the range:
a *many-to-one* mapping

Total Functions

ASCII $S \twoheadrightarrow T$

Publication $S \rightarrow T$

L^AT_EX $\$ S \setminus\text{fun } T \$$

formal $S \rightarrow T = \{f \mid f \in S \twoheadrightarrow T \wedge \text{dom}(f) = S\}$

comment a subset of partial functions where the domain is equal to the function argument set: a *many-to-one* mapping

Partial Injections

ASCII $S \rhd \! \! \! \rightarrow T$

Publication $S \rightsquigarrow T$

L^AT_EX $\$ S \setminus\text{pinj } T \$$

formal $S \rightsquigarrow T = \{f \mid f \in S \twoheadrightarrow T \wedge f^{-1} \in T \twoheadrightarrow S\}$

comment all elements in the domain map to unique elements in the range: a *one-to-one* mapping

Total Injections

ASCII $S \rhd \! \! \! \rightarrow T$

Publication $S \rightarrowtail T$

L^AT_EX $\$ S \setminus\text{inj } T \$$

formal $S \rightarrowtail T = S \rightsquigarrow T \cap S \rightarrow T$

comment all elements in the function argument set map to unique elements across the entire result set: a *one-to-one* mapping

Partial Surjections

ASCII $S \dashrightarrow T$

Publication $S \twoheadrightarrow T$

L^AT_EX $\$ S \setminus\text{surj } T \$$

formal $S \twoheadrightarrow T = \{f \mid f \in S \twoheadrightarrow T \wedge \text{ran}(f) = T\}$

comment an *onto* mapping

Total Surjections

ASCII $S \twoheadrightarrow T$

Publication $S \twoheadrightarrow T$

L^AT_EX $\$ S \surj T \$$

formal $S \twoheadrightarrow T = S \twoheadrightarrow T \cap S \rightarrow T$

comment an *onto* mapping

Bijections

ASCII $S \xrightarrow{\sim} T$

Publication $S \xrightarrow{\sim} T$

L^AT_EX $\$ S \bij T \$$

formal $S \xrightarrow{\sim} T = S \xrightarrow{\sim} T \cap S \rightarrow T$

comment a *one-to-one and onto* mapping

Lambda Abstraction

ASCII $z \cdot (P \mid E)$

Publication $\lambda z \cdot (P \mid E)$

L^AT_EX $\$ \lambda z \cdot (P \mid E) \$$

formal $\lambda z \cdot (P \mid E) = \{z, y \mid z \in \{z \mid P\} \wedge y = E\}$

comment where $y \setminus P$ and $y \setminus E$. P must constrain the variables in z . “ $y \setminus X$ ” means “ y is not free in X ”, and means that a) the selection of elements z cannot depend upon y , and b) nor can y depend upon itself.

Function Application

ASCII $f(E)$

Publication $f(E)$

L^AT_EX $\$ f(E) \$$

formal $E \mapsto y \in f \Rightarrow f(E) = y$

comment

3 Sequences

Sequences

- Sequences are functions from natural numbers to arbitrary values
- The domain must be an interval of the form $1 .. n$ (*finite* and *coherent*)

Empty Sequence

ASCII $\langle \rangle$

Publication $\langle \rangle$

L^AT_EX $\$ \backslash emptyseq \$$

formal $\langle \rangle = \{\}$

comment same as the empty set: note the three different representations!

Finite Sequences

ASCII $seq\ S$

Publication $seq(S)$

L^AT_EX $\$ \backslash seq(S) \$$

formal $seq(S) = \{f \mid f \in \mathbb{N}_1 \twoheadrightarrow S \wedge \exists n \cdot n \in \mathbb{N} \wedge \text{dom}(f) = 1 .. n\}$

comment an ordered, numbered list of values

Finite Non-Empty Sequences

ASCII $seq_1(S)$

Publication $seq_1(S)$

L^AT_EX $\$ \backslash seq_1(S) \$$

formal $seq_1(S) = seq(S) - \{\langle \rangle\}$

comment

Injective Sequences

ASCII $\backslash iseq(S)$

Publication $iseq(S)$

L^AT_EX $\$ \backslash iseq(S) \$$

formal $iseq(S) = seq(S) \cap (\mathbb{N}_1 \twoheadrightarrow S)$

comment all elements in the sequence are unique

Permutations

ASCII perm(S)

Publication perm(S)

L^AT_EX $\$ \backslash\text{perm}(S) \$$

formal perm(S) = iseq(S) \cap ($\mathbb{N}_1 \rightarrow S$)

comment one-to-one and onto (bijective) sequences

Sequence Concatenation

ASCII $s \hat{\ } t$

Publication $s \hat{\ } t$

L^AT_EX $\$ s \backslash\text{cat } t \$$

formal $s \hat{\ } t =$

comment formal definition left as an exercise for the reader

Prepend Element

ASCII $E \rightarrow s$

Publication $E \rightarrow s$

L^AT_EX $\$ E \backslash\text{prepend } s \$$

formal $E \rightarrow s = [E] \hat{\ } s$

comment

Append Element

ASCII $s \leftarrow E$

Publication $s \leftarrow E$

L^AT_EX $\$ s \backslash\text{append } E \$$

formal $s \leftarrow E = s \hat{\ } [E]$

comment

Singleton

ASCII [E]

Publication [E]

LaTeX \$ [E] \$

formal [E] = $\{1 \mapsto E\}$

comment

Sequence Construction

ASCII [E, F]

Publication [E, F]

LaTeX \$ [E, F] \$

formal [E, F] = $[E] \leftarrow F$

comment

Size

ASCII size(s)

Publication size(s)

LaTeX \$ \text{size}(s) \$

formal size(s) = card(s)

comment

Reverse

ASCII rev(s)

Publication rev(s)

LaTeX \$ \text{rev}(s) \$

formal $\forall i \cdot i \in \text{dom}(s) \Rightarrow \text{rev}(s)(i) = s(\text{size}(s) + 1 - i)$

comment

Take

ASCII `s/|\n`

Publication $s \uparrow n$

L^AT_EX `$ s \take n $`

formal $s \uparrow n = 1..n \triangleleft s$

comment the prefix of length n from s (tail discarded)

Drop

ASCII `s \ | / n`

Publication $s \downarrow n$

L^AT_EX `$ s \drop n $`

formal $s \downarrow n = (\lambda m \cdot (m \in \mathbb{N} \mid m + n)); (1..n \triangleleft s)$

comment

First Element

ASCII `first(s)`

Publication $\text{first}(s)$

L^AT_EX `$ \first(s) $`

formal $\text{first}(s) = s(1)$

comment defined only for a non-empty sequence

Last Element

ASCII `last(s)`

Publication $\text{last}(s)$

L^AT_EX `$ \last(s) $`

formal $\text{last}(s) = s(\text{size}(s))$

comment defined only for a non-empty sequence

Tail

ASCII `tail(s)`

Publication `tail(s)`

L^AT_EX `$ \tail(s) $`

formal $\text{tail}(s) = s \downarrow 1$

comment defined only for a non-empty sequence

Front

ASCII `front(s)`

Publication `front(s)`

L^AT_EX `$ \front(s) $`

formal $\text{front}(s) = s \uparrow (\text{size}(s) - 1)$

comment defined only for a non-empty sequence

Generalized Concatenation

ASCII `conc(ss)`

Publication `conc(ss)`

L^AT_EX `$ \conc(ss) $`

formal $\text{conc}(\langle \rangle) = \langle \rangle$ $\text{conc}(s \leftarrow E) = \text{conc}(s) \frown E$

comment concatenation sequences of sequences

Strings

ASCII `" \ldots "`

Publication `"..."`

L^AT_EX `$ ``\ldots'' $`

formal left as an exercise for the reader

comment sequences of characters, delimited by quotes