FIT3013 Lecture Notes

Formal Definitions of Relations, Functions and Sequences

John Hurst, 2008

Computer Science and Software Engineering
Monash University
(with acknowledgement to Schneider, the b-method, chapter 2)

20080730 / Lecture 6

Outline

Relations

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

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

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

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 <-> T`
Publication	$S \leftrightarrow T$
LaTeX	`$ S \rel T $`
formal	$S \leftrightarrow T = \mathbb{P}(S \times T)$

Domains

ASCII	`dom(r)`
Publication	$\mathrm{dom}(r)$
LaTeX	`$ \dom(r) $`
formal	$\forall r \cdot r \in S \leftrightarrow T \Rightarrow \mathrm{dom}(r) = \{x \mid x \in S \land (\exists y \cdot y \in T \land x \mapsto y \in r)\}$
comment	If $r \in S \leftrightarrow T$ then $\mathrm{dom}(r) \subseteq S$

Ranges

ASCII	`ran(r)`
Publication	$\mathrm{ran}(r)$
LaTeX	`$ \ran(r) $`
formal	$\forall r \cdot r \in S \leftrightarrow T \Rightarrow$ $\mathrm{ran}(r) = \{y \mid y \in T \land (\exists x \cdot x \in S \land x \mapsto y \in r)\}$
comment	If $r \in S \leftrightarrow T$ then $\mathrm{ran}(r) \subseteq T$

Forward Composition

ASCII	`p ; q`
Publication	$p \mathbin{;} q$
LaTeX	`$ 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	$p \circ q$
LaTeX	`$ 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	$\mathrm{id}(S)$
LaTeX	`$ \id(S) $`
formal	$\mathrm{id}(S) = \{x, y \mid x \in S \land y \in S \land x = y\}$
comment	transform a set into itself

Domain Restriction

ASCII	`S <| r`
Publication	$S \lhd r$
LaTeX	`$ S \dres r $`
formal	$S \lhd r = \{x, y \mid x \mapsto y \in r \wedge x \in S\}$
comment	subset a relation r so that its domain is a subset of the given set S

Domain Subtraction

ASCII	`S <<| r`
Publication	$S \ndres r$
LaTeX	`$ S \ndres r $`
formal	$S \ndres r = \{x, y \mid x \mapsto y \in r \land x \notin S\}$
comment	subset a relation *r* so that its domain is mutually exclusive with the given set *S*

Range Restriction

ASCII	`r |> T`
Publication	$r \rhd T$
LaTeX	`$ r \rres T $`
formal	$r \rhd T = \{x, y \mid x \mapsto y \in r \land y \in T\}$
comment	subset a relation *r* so that its range is a subset of the given set *T*

Range Subtraction

ASCII	`r |>> T`
Publication	$r \rhd T$
LaTeX	`$ r \nrres T $`
formal	$r \rhd T = \{x, y \mid x \mapsto y \in r \land y \notin T\}$
comment	subset a relation *r* so that its range is mutually exclusive with the given set *T*

Inverse

ASCII	`r~`
Publication	r^{-1}
LaTeX	`$ \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	$r[S]$
LaTeX	`$ r[S] $`
formal	$r[S] = \{y \mid \exists x \cdot x \in S \land x \mapsto y \in r\}$
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	$r_1 \Lsh r_2$
LaTeX	`$ r_1 \rovr r_2 $`
formal	$r_1 \Lsh r_2 = r_2 \cup (\mathrm{dom}(r_2) \lhd r_1)$
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 `r1 +> r2`

Publication $r_1 \Lsh r_2$

LaTeX `$ r_1 \lovr r_2 $`

formal $r_1 \Lsh r_2 = r_1 \cup (\mathrm{dom}(r_1) \lhd r_2)$

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	$p \otimes q$
LaTeX	`$ 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 || q`
Publication	$p \parallel q$
LaTeX	`$ 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	`iterate(r,n)`
Publication	r^n
LaTeX	`$ r^n $`
formal	$r \in S \leftrightarrow S \Rightarrow r^0 = \mathrm{id}(S) \wedge r^{n+1} = r;\ r^n$
comment	repeatedly apply a relation; both domain and range must be of the same type.

Closure

ASCII	`closure(r)`
Publication	r^*
LaTeX	`$ 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	$\text{prj1}(S, T)$
LaTeX	`$ \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	$\mathrm{prj2}(S, T)$
LaTeX	`$ \PRJy(S,T) $`
formal	$\mathrm{prj2}(S, T) = \{(x, y), z \mid x, y \in S \times T \wedge z = y\}$
comment	extract the right hand element of a maplet

Functions

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

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 $S \pfun T$

LaTeX `$ S \pfun T $`

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

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 --> T`
Publication	$S \rightarrow T$
LaTeX	`$ S \fun T $`
formal	$S \rightarrow T = \{f \mid f \in S \nrightarrow T \land \mathrm{dom}(f) = S\}$
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 >+> T`
Publication	$S \rightarrowtail\!\!\!\!\!\!+\ T$
$\LaTeX$	`$ S \pinj T $`
formal	$S \rightarrowtail\!\!\!\!\!\!+\ T = \{f \mid f \in S \nrightarrow T \wedge f^{-1} \in T \nrightarrow S\}$
comment	all elements in the domain map to unique elements in the range: a *one-to-one* mapping

Total Injections

ASCII	`S >-> T`
Publication	$S \rightarrowtail T$
LaTeX	`$ S \inj T $`
formal	$S \rightarrowtail T = S \nrightarrow T \cap S \rightarrow T$
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 +->> T`
Publication	$S \twoheadrightarrow T$
LaTeX	`$ S \surj T $`
formal	$S \twoheadrightarrow T = \{f \mid f \in S \nrightarrow T \wedge \mathrm{ran}(f) = T\}$
comment	an *onto* mapping

Total Surjections

ASCII	`S -->> T`
Publication	$S \twoheadrightarrow T$
LaTeX	`$ S \surj T $`
formal	$S \twoheadrightarrow T = S \mathrel{+\!\!\twoheadrightarrow} T \cap S \rightarrow T$
comment	an *onto* mapping

Bijections

ASCII	`S >->> T`
Publication	$S \rightarrowtail\!\!\!\twoheadrightarrow T$
LaTeX	`$ S \bij T $`
formal	$S \rightarrowtail\!\!\!\twoheadrightarrow T = S \rightarrowtail T \cap S \twoheadrightarrow T$
comment	a *one-to-one and onto* mapping

Lambda Abstraction

ASCII	`z \cdot (P | E)`
Publication	$\lambda z \cdot (P \mid E)$
LaTeX	`$ \lambda z \cdot (P | E) $`
formal	$\lambda z \cdot (P \mid E) = \{z, y \mid z \in \{z \mid P\} \land y = E\}$
comment	where $y \backslash P$ and $y \backslash E$. P must constrain the variables in z. "$y \backslash X$" 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	$f(E)$
LaTeX	`$ f(E) $`
formal	$E \mapsto y \in f \Rightarrow f(E) = y$
comment	

Sequences

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

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	`<>`
Publication	$\langle\rangle$
LaTeX	`$ \emptyseq $`
formal	$\langle\rangle = \{\}$
comment	same as the empty set: note the three different representations!

Finite Sequences

ASCII	`seq S`
Publication	$\text{seq}(S)$
LaTeX	`$ \seq(S) $`
formal	$\text{seq}(S) = \{f \mid f \in \mathbb{N}_1 \mapsto S \land \exists\, n \cdot n \in \mathbb{N} \land \text{dom}(f) = 1 \mathinner{..} n\}$
comment	an ordered, numbered list of values

Finite Non-Empty Sequences

ASCII	`seq1(S)`
Publication	$\mathsf{seq}_1(S)$
LaTeX	`$ \seq_1(S) $`
formal	$\mathsf{seq}_1(S) = \mathsf{seq}(S) - \{\langle\rangle\}$
comment	

Injective Sequences

ASCII	`\iseq(S)`
Publication	$\mathrm{iseq}(S)$
LaTeX	`$ \iseq(S) $`
formal	$\mathrm{iseq}(S) = \mathrm{seq}(S) \cap (\mathbb{N}_1 \rightarrowtail\!\!\!\!\!\!+\; S)$
comment	all elements in the sequence are unique

Permutations

ASCII	`perm(S)`
Publication	$\mathrm{perm}(S)$
LaTeX	`$ \perm(S) $`
formal	$\mathrm{perm}(S) = \mathrm{iseq}(S) \cap (\mathbb{N}_1 \twoheadrightarrow S)$
comment	one-to-one and onto (bijective) sequences

Sequence Concatenation

ASCII	`s ^ t`
Publication	$s \frown t$
LaTeX	`$ s \cat t $`
formal	$s \frown t =$
comment	formal definition left as an exercise for the reader

Prepend Element

ASCII	`E -> s`
Publication	$E \rightarrow s$
LaTeX	`$ E \prepend s $`
formal	$E \rightarrow s = [E] \frown s$
comment	

Append Element

ASCII	`s <- E`
Publication	$s \leftarrow E$
LaTeX	`$s \append E $`
formal	$s \leftarrow E = s \frown [E]$
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	$\mathrm{size}(s)$
LaTeX	`$ \size(s) $`
formal	$\mathrm{size}(s) = \mathrm{card}(s)$
comment	

Reverse

ASCII	`rev(s)`
Publication	$\mathrm{rev}(s)$
LaTeX	`$ \rev(s) $`
formal	$\forall i \cdot i \in \mathrm{dom}(s) \Rightarrow \mathrm{rev}(s)(i) = s(\mathrm{size}(s) + 1 - i)$
comment	

Take

ASCII	`s/|\n`
Publication	$s \uparrow n$
LaTeX	`$ s \take n $`
formal	$s \uparrow n = 1 \mathinner{..} n \lhd s$
comment	the prefix of length n from s (tail discarded)

Drop

ASCII	`s |/ n`
Publication	$s \downarrow n$
LaTeX	`$ s \drop n $`
formal	$s \downarrow n = (\lambda m \cdot (m \in \mathbb{N} \mid m + n));\ (1 \mathinner{..} n \lhd s)$
comment	

First Element

ASCII	`first(s)`
Publication	$\text{first}(s)$
LaTeX	`$ \first(s) $`
formal	$\text{first}(s) = s(1)$
comment	defined only for a non-empty sequence

Last Element

ASCII	`last(s)`
Publication	$\mathrm{last}(s)$
LaTeX	`$ \last(s) $`
formal	$\mathrm{last}(s) = s(\mathrm{size}(s))$
comment	defined only for a non-empty sequence

Tail

ASCII	`tail(s)`
Publication	$\mathrm{tail}(s)$
LaTeX	`$ \tail(s) $`
formal	$\mathrm{tail}(s) = s \downarrow 1$
comment	defined only for a non-empty sequence

Front

ASCII	`front(s)`
Publication	$\text{front}(s)$
LaTeX	`$ \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	$\mathrm{conc}(ss)$
LaTeX	`$ \conc(ss) $`
formal	$\mathrm{conc}(\langle\,\rangle) = \langle\,\rangle$ $\mathrm{conc}(s \leftarrow E) = \mathrm{conc}(s) \frown E$
comment	concatenation sequences of sequences

Strings

ASCII	`" \ldots "`
Publication	“. . .”
LaTeX	`$ ``\ldots'' $`
formal	left as an exercise for the reader
comment	sequences of characters, delimited by quotes

Summary

- Many specialized operators in B
- These are shorthand (*syntactic sugar*) for formal expressions
- Note that these are all operations that deal with relations!

Summary

- Many specialized operators in B
- These are shorthand (*syntactic sugar*) for formal expressions
- Note that these are all operations that deal with relations!

Summary

- Many specialized operators in B
- These are shorthand (*syntactic sugar*) for formal expressions
- Note that these are all operations that deal with relations!