Outline

Contents

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 <-> T`

Publication $S \leftrightarrow T$

LaTeX `$ S \rel T $`

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

Domains

ASCII `dom(r)`

Publication $\mathsf{dom}(r)$

LaTeX `$ \dom(r) $`

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

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

Ranges

ASCII `ran(r)`

Publication $\mathsf{ran}(r)$

LaTeX `$ \ran(r) $`

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

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

Forward Composition

ASCII `p ; q`

Publication $p\ ;\ q$

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 $\mathsf{id}(S)$

LaTeX `$ \id(S) $`

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

comment transform a set into itself

Domain Restriction

ASCII `S <| r`

Publication $S \lhd r$

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 $r \ntriangleright T$

LaTeX `$ r \nrres T $`

formal $r \ntriangleright T = \{x, y \mid x \mapsto y \in r \land y \notin T\}$

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

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 $r_1 \lhd\!\!+ r_2$

LaTeX `$ r_1 \rovr r_2 $`

formal $r_1 \lhd\!\!+ r_2 = r_2 \cup (\mathsf{dom}(r_2) \unlhd r_1)$

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 $r_1 +\!\!\rhd r_2$

LaTeX `$ r_1 \lovr r_2 $`

formal $r_1 +\!\!\rhd r_2 = r_1 \cup (\mathsf{dom}(r_1) \unlhd r_2)$

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

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 $\mathsf{prj1}(S, T)$

LaTeX `$ \PRJx(S,T) $`

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

comment extract the left hand element of a maplet

Projection

ASCII `prj2(S,T)`

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

LaTeX `$ \PRJy(S,T) $`

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

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 $S \nrightarrow T$

LaTeX `$ S \pfun T $`

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

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 $S \rightarrow T = \{f \mid f \in S \nrightarrow T \land \mathsf{dom}(f) = S\}$

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 $S \pinj T$

LaTeX `$ S \pinj T $`

formal $S \pinj T = \{f \mid f \in S \pfun T \land f^{-1} \in T \pfun S\}$

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 $S \rightarrowtail T = S \pinj T \cap S \rightarrow T$

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 $S \twoheadrightarrow T = \{f \mid f \in S \pfun T \land \mathsf{ran}(f) = T\}$

comment an *onto* mapping

Total Surjections

ASCII `S -->> T`

Publication $S \twoheadrightarrow T$

LaTeX `$ S \surj T $`

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

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

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

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 `<>`

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 $\mathsf{seq}(S)$

LaTeX `$ \seq(S) $`

formal $\mathsf{seq}(S) = \{f \mid f \in \mathbb{N}_1 \nrightarrow S \land \exists\, n \cdot n \in \mathbb{N} \land \mathsf{dom}(f) = 1 \mathinner{..} n\}$

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 $\mathsf{iseq}(S)$

LaTeX `$ \iseq(S) $`

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

comment all elements in the sequence are unique

Permutations

ASCII `perm(S)`

Publication $\mathsf{perm}(S)$

LaTeX `$ \perm(S) $`

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

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 $\mathsf{size}(s)$

LaTeX `$ \size(s) $`

formal $\mathsf{size}(s) = \mathsf{card}(s)$

comment

Reverse

ASCII `rev(s)`

Publication $\mathsf{rev}(s)$

LaTeX `$ \rev(s) $`

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

comment

Take

ASCII `s/|\n`

Publication $s \upharpoonright n$

LaTeX `$ s \take n $`

formal $s \upharpoonright n = 1 \mathrel{..} n \lhd s$

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

Drop

ASCII `s \|/ n`

Publication $s \downharpoonright n$

LaTeX `$ s \drop n $`

formal $s \downharpoonright n = (\lambda\, m \cdot (m \in \mathbb{N} \mid m + n));\ (1 \mathrel{..} n \ntriangleleft s)$

comment

First Element

ASCII `first(s)`

Publication $\mathsf{first}(s)$

LaTeX `$ \first(s) $`

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

comment defined only for a non-empty sequence

Last Element

ASCII `last(s)`

Publication $\mathsf{last}(s)$

LaTeX `$ \last(s) $`

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

comment defined only for a non-empty sequence

Tail

ASCII `tail(s)`

Publication $\mathsf{tail}(s)$

LaTeX `$ \tail(s) $`

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

comment defined only for a non-empty sequence

Front

ASCII `front(s)`

Publication $\mathsf{front}(s)$

LaTeX `$ \front(s) $`

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

comment defined only for a non-empty sequence

Generalized Concatenation

ASCII `conc(ss)`

Publication $\mathsf{conc}(ss)$

LaTeX `$ \conc(ss) $`

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

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