

# CSE4213

## Formal Methods in Software Engineering

### Formal Definitions of Relations, Functions and Sequences

John Hurst

School of Computer Science & Software Engineering  
Monash University, Melbourne, Australia

20050317 / Lecture 6

# Outline

- 1 Relations
- 2 Functions
- 3 Sequences
- 4 Summary

# 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<sup>A</sup>T<sub>E</sub>X  $\$ S \rel T \$$

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

# Domains

ASCII  $\text{dom}(r)$

Publication  $\text{dom}(r)$

L<sup>A</sup>T<sub>E</sub>X  $\$ \backslash \text{dom}(r) \$$

formal  $\forall r \cdot r \in \mathbf{S} \leftrightarrow T \Rightarrow$

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

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

# Ranges

ASCII  $\text{ran}(r)$

Publication  $\text{ran}(r)$

L<sup>A</sup>T<sub>E</sub>X  $\$ \backslash \text{ran}(r) \$$

formal  $\forall r \cdot r \in \mathbf{S} \leftrightarrow T \Rightarrow$

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

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

# Forward Composition

ASCII  $p ; q$

Publication  $p ; q$

L<sup>A</sup>T<sub>E</sub>X  $\$ p \backslash \text{comp } q \$$

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  $p \circ q$

L<sup>A</sup>T<sub>E</sub>X `$ 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<sup>A</sup>T<sub>E</sub>X `$ \id(S) $`

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

comment transform a set into itself

# Domain Restriction

ASCII  $S \triangleleft r$

Publication  $S \triangleleft r$

$\text{\LaTeX}$   $\$ S \ \backslash\text{dres } r \$$

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  $S \ll r$

Publication  $S \triangleleft r$

L<sup>A</sup>T<sub>E</sub>X  $\$ S \setminus \text{ndres } r \$$

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

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

# Range Restriction

ASCII  $r \mid\triangleright T$

Publication  $r \triangleright T$

L<sup>A</sup>T<sub>E</sub>X  $\$ r \backslash rres T \$$

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  $r \setminus \gg T$

Publication  $r \triangleright T$

L<sup>A</sup>T<sub>E</sub>X  $\$ r \setminus \text{nrres } T \$$

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  $r\sim$

Publication  $r^{-1}$

L<sup>A</sup>T<sub>E</sub>X  $\$ \backslash inv r \$$

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<sup>A</sup>T<sub>E</sub>X  $\$ r[S] \$$

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  $r_1 <+ r_2$

Publication  $r_1 \triangleleft r_2$

L<sup>A</sup>T<sub>E</sub>X  $\$ r_1 \backslash\text{rovr } r_2 \$$

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<sup>A</sup>T<sub>E</sub>X**  $\$ r_1 \backslash\text{lovr } r_2 \$$

**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  $p \times q$

Publication  $p \otimes q$

L<sup>A</sup>T<sub>E</sub>X  $\$ p \otimes q \$$

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<sup>A</sup>T<sub>E</sub>X  $\$ p \parallel q \$$

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  $r^n$

L<sup>A</sup>T<sub>E</sub>X  $\$ r^n \$$

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

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

# Closure

ASCII `closure(r)`

Publication  $r^*$

L<sup>A</sup>T<sub>E</sub>X  $\$ r^* \$$

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** `prj1(S, T)`

**L<sup>A</sup>T<sub>E</sub>X** `$ \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** `prj2(S, T)`

**L<sup>A</sup>T<sub>E</sub>X** `$ \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

# Functions

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

# Partial Functions

ASCII  $S \dashrightarrow T$

Publication  $S \mapsto T$

L<sup>A</sup>T<sub>E</sub>X  $\$ S \backslash pfun T \$$

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

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<sup>A</sup>T<sub>E</sub>X  $\$ S \backslash\text{fun } T \$$

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  $S \rightarrow\!\!\rightarrow T$

Publication  $S \rightsquigarrow T$

L<sup>A</sup>T<sub>E</sub>X  $\$ S \backslash\text{pinj } T \$$

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

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

# Total Injections

ASCII  $S \rightarrow T$

Publication  $S \mapsto T$

L<sup>A</sup>T<sub>E</sub>X  $\$ S \ \backslash inj \ T \ \$$

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

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  $S \twoheadrightarrow T$

Publication  $S \twoheadrightarrow T$

L<sup>A</sup>T<sub>E</sub>X  $\$ S \backslash \text{surj } T \$$

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<sup>A</sup>T<sub>E</sub>X  $\$ S \surj T \$$

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

comment an *onto* mapping

# Bijections

ASCII  $S \rightarrow T$

Publication  $S \rightarrow T$

L<sup>A</sup>T<sub>E</sub>X  $\$ S \text{ \texttt{bij} } T \$$

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

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

# Lambda Abstraction

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

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

L<sup>A</sup>T<sub>E</sub>X  $\$ \backslash lambda z \cdot (P \mid E) \$$

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<sup>A</sup>T<sub>E</sub>X  $\$ f(E) \$$

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**)

# Empty Sequence

ASCII  $\langle \rangle$

Publication  $\langle \rangle$

$\text{\LaTeX}$   $\$ \backslash emptyseq \$$

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

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

# Finite Sequences

ASCII seq S

Publication seq(S)

L<sup>A</sup>T<sub>E</sub>X \$ \seq(S) \$

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

comment an ordered, numbered list of values

# Finite Non-Empty Sequences

ASCII `seq1(S)`

Publication `seq1(S)`

L<sup>A</sup>T<sub>E</sub>X `$ \seq_1(S) $`

formal `seq1(S) = seq(S) - {⟨⟩}`

comment

# Injective Sequences

ASCII `\iseq(S)`

Publication `iseq(S)`

L<sup>A</sup>T<sub>E</sub>X `$ \iseq(S) $`

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

comment all elements in the sequence are unique

# Permutations

ASCII `perm(S)`

Publication `perm(S)`

L<sup>A</sup>T<sub>E</sub>X `$ \perm(S) $`

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

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

# Sequence Concatenation

ASCII  $s \wedge t$

Publication  $s \wedge t$

L<sup>A</sup>T<sub>E</sub>X  $\$ s \backslash \text{cat } t \$$

formal  $s \wedge t =$

comment formal definition left as an exercise for the reader

# Prepend Element

ASCII  $E \rightarrow s$

Publication  $E \rightarrow s$

L<sup>A</sup>T<sub>E</sub>X  $\$ E \backslash\text{prepend } s \$$

formal  $E \rightarrow s = [E] \frown s$

comment

# Append Element

ASCII  $s \leftarrow E$

Publication  $\mathbf{s} \leftarrow E$

L<sup>A</sup>T<sub>E</sub>X  $\$s \backslash\text{append } E \$$

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

comment

# Singleton

ASCII  $[E]$

Publication  $[E]$

$\LaTeX$   $\$ [E] \$$

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

comment

# Sequence Construction

ASCII  $[E, F]$

Publication  $[E, F]$

L<sup>A</sup>T<sub>E</sub>X  $\$ [E, F] \$$

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

comment

# Size

ASCII `size(s)`

Publication `size(s)`

L<sup>A</sup>T<sub>E</sub>X `$ \size(s) $`

formal `size(s) = card(s)`

comment

## Reverse

ASCII `rev(s)`

Publication `rev(s)`

L<sup>A</sup>T<sub>E</sub>X `$ \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  $s / | \backslash n$

Publication  $s \uparrow n$

L<sup>A</sup>T<sub>E</sub>X  $\$ s \backslash take n \$$

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

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

## Drop

ASCII `s \ | / n`

Publication `s ↓ n`

L<sup>A</sup>T<sub>E</sub>X `$ s \drop n $`

formal `s ↓ n = (λ m · (m ∈ ℕ | m + n)); (1 .. n ≪ s)`

comment

# First Element

ASCII `first(s)`

Publication `first(s)`

L<sup>A</sup>T<sub>E</sub>X `$ \first(s) $`

formal `first(s) = s(1)`

comment defined only for a non-empty sequence

# Last Element

ASCII `last(s)`

Publication `last(s)`

L<sup>A</sup>T<sub>E</sub>X `$ \last(s) $`

formal `last(s) = s(size(s))`

comment defined only for a non-empty sequence

ASCII `tail(s)`

Publication `tail(s)`

L<sup>A</sup>T<sub>E</sub>X `$ \tail(s) $`

formal `tail(s) = s ↓ 1`

comment defined only for a non-empty sequence

ASCII `front(s)`

Publication `front(s)`

L<sup>A</sup>T<sub>E</sub>X `$ \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<sup>A</sup>T<sub>E</sub>X `$ \conc(ss) $`

formal `conc(⟨⟩) = ⟨⟩`

`conc(s ← E) = conc(s)  $\hat{\smile}$  E`

comment concatenation sequences of sequences

# Strings

ASCII " \ldots "

Publication "..."

L<sup>A</sup>T<sub>E</sub>X \$ ``\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!