Sets, Relations, and Functions

Wolfgang Schreiner
Research Institute for Symbolic Computation (RISC-Linz)
Johannes Kepler University, Linz, Austria

Wolfgang.Schreiner@risc.uni-linz.ac.at
http://www.risc.uni-linz.ac.at/people/schreine

The Datatype Set

Overview

- The Datatype Set
- Predicates as Sets
- Functions as Sets
- Sequences and Matrices

Motivation

- Most important mathematical domain.

  Default for interpretation of logic formulas.
- Universal domain.
  - Most other datatypes can be defined as sets.
  - Relations, functions, numbers, arrays, lists, trees, databases, ...
- Fundamental domain.
  - Not defined by other (more fundamental) domain.
  - Characterized by its properties (axioms).

Building material for other theories.

Basics

- Domain values are called sets.

  All objects are sets.

- Single binary predicate is element of $\in$.
  - All other predicates and functions defined by this predicate.
  - $x \in y$: x is element of y.

Behavior of $\in$ characterized by axioms.

Some Axioms

- Two sets are equal, iff they have the same elements:

$$\forall x, y : x = y \Leftrightarrow (\forall z : z \in x \Leftrightarrow z \in y).$$

- There exists a set that does not have any elements:

$$\exists x : \forall y : y \notin x.$$

  - We call this set empty set:

$$\emptyset := \textbf{such } x : \forall y : y \notin x.$$

  - Because of second axiom, $\emptyset$ is well defined:

$$\forall y : y \notin \emptyset.$$

  - Because of first axiom, $\emptyset$ is unique:

$$\forall z : (\forall y : y \notin z) \Rightarrow z = \emptyset.$$

Set Enumeration

Definition: term $\{T_0, T_1, \ldots, T_{n-1}\}$

- Terms T_i
- Set that contains exactly the values denoted by the T_i:

$$\forall x : x \in \{T_0, T_1, \ldots, T_{n-1}\} \Leftrightarrow (x = T_0 \vee x = T_1 \vee \ldots \vee x = T_{n-1}).$$

Special Case: $\{\} = \emptyset$

Example: $S := \{1, \emptyset, \{1, 2\}, a\}$

- $1 \in S$, $\emptyset \in S$, $\{1, 2\} \in S$, $a \in S$.
- $S = \{\{1, 2\}, a, \emptyset, 1\}$.
- $S = \{1, 1, 1, \emptyset, a, \{1, 2\}, a\}$.

Subset

Definition: subset

- x is subset of y iff every element of x is also a element of y:

$$x \subseteq y :\Leftrightarrow (\forall z \in x : z \in y).$$

Proposition: for all x, y, and z, we have

- Minimum: $\emptyset \subseteq x$,
- Reflexivity: $x \subseteq x$,
- Antisymmetry: $(x \subseteq y \wedge y \subseteq x) \Rightarrow x = y$,
- Transitivity: $(x \subseteq y \wedge y \subseteq z) \Rightarrow x \subseteq z$.

Ordering relationship among sets.

Subset and Equality

Proposition: For every x and y, we have

$$x = y \Leftrightarrow (x \subseteq y \wedge y \subseteq x).$$

Proof: Take arbitrary x and y. We prove $x = y \Leftrightarrow (x \subseteq y \wedge y \subseteq x)$.

- We prove $x = y \Rightarrow (x \subseteq y \wedge y \subseteq x)$. Assume $x = y$, i.e., by definition of '$=$',

  $$(1)\ \forall z : z \in x \Leftrightarrow z \in y.$$

  We have to prove $x \subseteq y \wedge y \subseteq x$.
  - We prove $x \subseteq y$, i.e., by definition of '$\subseteq$', $\forall z \in x : z \in y$. Take arbitrary z. We have to prove $z \in x \Rightarrow z \in y$. Assume (2) $z \in x$. We have to prove $z \in y$ which is a consequence of (1) and (2).
  - The proof of $y \subseteq x$ proceeds analogously.

Subset and Equality (Continued)

Proof (continued):

- We prove $(x \subseteq y \wedge y \subseteq x) \Rightarrow x = y$. Assume $x \subseteq y \wedge y \subseteq x$, i.e., by definition of '$\subseteq$'

  $$(1)\ \forall z \in x : z \in y;$$
  $$(2)\ \forall z \in y : z \in x.$$

  We prove $x = y$, i.e., by definition of '$=$', $\forall z : z \in x \Leftrightarrow z \in y$. Take arbitrary z. We have to prove $z \in x \Leftrightarrow z \in y$.
  - We prove $z \in x \Rightarrow z \in y$. Assume (3) $z \in x$. We have to prove $z \in y$ which is a consequence of (1) and (3).
  - We prove $z \in y \Rightarrow z \in x$. Assume (4) $z \in y$. We have to prove $z \in x$ which is a consequence of (2) and (4).

A well-structured argument based on definitions and given knowledge.

Set Quantifier

$$\{x \in S : A\}$$

- Variable x, term S, formula A.
- Term whose value is the set of all elements x in S with property A:

  $$\forall x : x \in \{x \in S : A\} \Leftrightarrow (x \in S \wedge A).$$

- Variable domain S dropped, if clear from context:

  $$\{x : A\}$$

Tool to construct subsets of given sets.

Example

- Let S be $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Then

  $$\{x \in S : x \leq 3 \vee x \text{ is even}\}$$

  is $\{1, 2, 3, 4, 6, 8, 10\}$.
- Let $\mathbb{N}$ be the set of all natural numbers. Then

  $$\{n \in \mathbb{N} : n > 1 \wedge \forall i : i | n \Rightarrow (i = 1 \vee i = n)\}$$

  is the set of all prime numbers.
- For every set S,

  $$\{x \in S : x \notin S\}$$

  is the empty set $\emptyset$.

Generalized Set Quantifier

$$\{T_x : x \in S \wedge A\}$$

- Term T_x with free variable x.

  Variable x has to be deduced from context.

- Term whose value is the set of all values of T_x where x is an element of S for which A holds:

$$\{T_x : x \in S \wedge A\} = \{y : (\exists x \in S : y = T_x \wedge A)\}$$

More convenient syntax for set construction.

Example

- The term

$$\{2 * x : 1 \leq x \leq 5\}$$

  is usually interpreted as

$$\{2 * x : x \in \mathbb{N} \wedge 1 \leq x \leq 5\}$$

  which denotes the set $\{2, 4, 6, 8, 10\}$.

- The term

$$\{x + y : 1 \leq y \leq 5\}$$

  typically denotes the set $\{x+1, x+2, x+3, x+4, x+5\}$ (assuming that only y is bound by the quantifier and that its domain is $\mathbb{N}$).

Generalized Set Quantifier

$$\{Z_{x,y} : x \in S \wedge y \in T \wedge A\}$$

- Term $Z_{x,y}$ with free variables x and y.

  Variables have to be deduced from context.

- Value of the term is the set

$$\{z : (\exists x \in S, y \in T : z = Z_{x,y}. \wedge A)\}$$

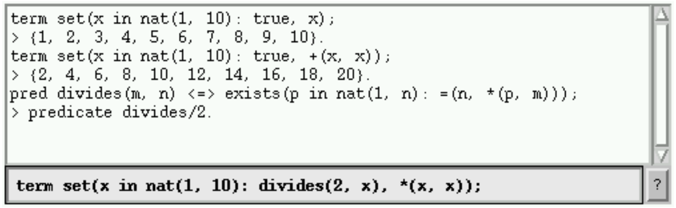
Set quantifier may bind arbitrary number of variables.

Example

$$\{x + y : 1 \leq x \leq 3 \wedge 0 \leq y \leq 2\}$$
$$=$$
$$\{x + y : x \in \mathbb{N} \wedge y \in \mathbb{N} \wedge 1 \leq x \leq 3 \wedge 0 \leq y \leq 2\}$$
$$=$$
$$\{s : (\exists x \in \mathbb{N}, y \in \mathbb{N} : s = x + y \wedge 1 \leq x \leq 3 \wedge 0 \leq y \leq 2)\}$$
$$=$$
$$\{1+0,\ 1+1,\ 1+2,\ 2+0,\ 2+1,\ 2+2,\ 3+0,\ 3+1,\ 3+2\}$$
$$=$$
$$\{1,\ 2,\ 3,\ 4,\ 5\}.$$

Logic Evaluator

$$\texttt{set}(x \text{ in } S\text{: } A, T) = \{T_x : x \in S \wedge A\}.$$



Operational Interpretation

```
public final class SetTerm implements Term
{
  private String x; private Term S; private Formula A; private Term T;

  public Value eval() throws EvalException
  {
    Set set = new Set();
    Iterator iterator = Model.iterator(S);
    while (iterator.hasNext()) {
      Context.begin(x, iterator.next());
      if (A.eval()) set.addElement(T.eval());
      Context.end(); }
    return set;
  }
}
```


Set Union

Definition: union

- Binary function: all elements in x or in y

$$x \cup y := \{z : z \in x \vee z \in y\}$$

- Unary function: all elements in some element of x

$$\bigcup x := \{z : (\exists y \in x : z \in y)\}$$

- Quantor: union of all term values

$$\bigcup_{x \in S \wedge A} T := \bigcup \{T : x \in S \wedge A\}.$$

Set Intersection

Definition: intersection

- Binary function: all elements in x and in y

$$x \cap y := \{z \in x : z \in y\}$$

- Unary function: all elements in every element of x

$$\bigcap x := \{z \in \bigcup x : (\forall y \in x : z \in y)\}$$

- Quantor: intersection of all term values

$$\bigcap_{x \in S \wedge A} T := \bigcap \{T : x \in S \wedge A\}$$

More Set Functions

- Definition: difference

$$x \backslash y := \{z \in x : z \notin y\}$$

The set of all elements in x but not in y.

- Definition: powerset

$$\mathbb{P}(x) := \{y : y \subseteq x\}$$

The set of all subsets of x.

Example

- Let $S := \{1, 2, 3, 4, 5\}$, $T := \{2, 5, 7\}$, $U := \{1, 3, 5, 7, 9\}$.

$$\begin{aligned}
&S \cap T = \{2, 5\};\\
&S \cup T = \{1, 2, 3, 4, 5, 7\};\\
&\bigcap\{S, T, U\} = \{5\};\\
&\bigcup\{S, T, U\} = \{1, 2, 3, 4, 5, 7, 9\};\\
&\mathbb{P}(T) = \{\emptyset, \{2\}, \{5\}, \{7\}, \{2, 5\}, \{2, 7\}, \{5, 7\}, \{2, 5, 7\}\}.
\end{aligned}$$

- Let $\mathbb{N}$ be the set of natural numbers and $\mathbb{N}_n := \{x \in \mathbb{N} : x < n\}$.

$$\begin{aligned}
&\textstyle\bigcup_{i \in \mathbb{N}} \mathbb{N}_i = \mathbb{N};\\
&\textstyle\bigcap_{i \in \mathbb{N}} \mathbb{N}_i = \{\};
\end{aligned}$$

Set Identities

For every A, B, and C, we have:

- Idempotency, Identity and Domination

$$\begin{aligned}
A \cup A = A,\ A \cup \emptyset = A,\\
A \cap A = A,\ A \cap \emptyset = \emptyset;
\end{aligned}$$

- Commutativity

$$\begin{aligned}
A \cup B = B \cup A,\\
A \cap B = B \cap A;
\end{aligned}$$

- Associativity

$$\begin{aligned}
A \cup (B \cup C) = (A \cup B) \cup C,\\
A \cap (B \cap C) = (A \cap B) \cap C;
\end{aligned}$$

Set Identities

For every A, B, and C, we have:

- Distributivity

$$\begin{aligned}
A \cap (B \cup C) = (A \cap B) \cup (A \cap C),\\
A \cup (B \cap C) = (A \cup B) \cap (A \cup C);
\end{aligned}$$

- Cancellation

$$\begin{aligned}
A \cup (A \cap B) = A,\\
A \cap (A \cup B) = A;
\end{aligned}$$

- De Morgan

$$\begin{aligned}
C \backslash (A \cup B) = C \backslash A \cap C \backslash B,\\
C \backslash (A \cap B) = C \backslash A \cup C \backslash B.
\end{aligned}$$

Proof

Proposition: $\forall A, B : A \cup B = B \cup A$

Proof: Take arbitrary A and B. By definition of $=$, we have to prove

$$(1)\ \forall x : x \in A \cup B \Leftrightarrow x \in B \cup A.$$

Take arbitrary x.

- We prove $x \in A \cup B \Rightarrow x \in B \cup A$. Assume

  $$(2)\ x \in A \cup B.$$

  We have to prove $x \in B \cup A$. By definition of $\cup$, we have to prove

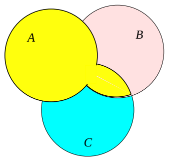
  $$(3)\ x \in B \vee x \in A.$$

  If $x \in B$, we are done. Thus assume (4) $x \notin B$. By (2) and definition of $\cup$, we have (5) $x \in A \vee x \in B$. From (4) and (5), we have $x \in A$ and thus (3).
- The proof of $x \in B \cup A \Rightarrow x \in A \cup B$ proceeds analogously.

Equality Reasoning

$$\forall A, B, C : (B \cup A) \cap (C \cup A) = (B \cap C) \cup A.$$

Venn diagram:



We prove the proposition as follows:

$$\begin{array}{rcl} (B \cup A) \cap (C \cup A) & = & \text{(commutativity)} \\ (A \cup B) \cap (A \cup C) & = & \text{(distributivity)} \\ A \cup (B \cap C) & = & \text{(commutativity)} \\ (B \cap C) \cup A. & & \end{array}$$

Tuples

Definition: n-tuple $\langle x_0, x_1, \ldots, x_{n-1} \rangle$

- tuple constructor $\langle\ \rangle$ (n-ary function)
- tuple selectors $._0, ._1, \ldots, ._{n-1}$ (n 1-ary functions)

$$\begin{array}{l} \langle x_0, x_1, \ldots, x_{n-1} \rangle_0 = x_0; \\ \langle x_0, x_1, \ldots, x_{n-1} \rangle_1 = x_1; \\ \ldots \\ \langle x_0, x_1, \ldots, x_{n-1} \rangle_{n-1} = x_{n-1}. \end{array}$$

Ordered sequence of elements (can be implemented as sets).

Example

- $T := \langle 1, 2 \rangle$.

  $$\begin{array}{l} T_0 = 1; \\ T_1 = 2. \end{array}$$

- $U := \langle 2, T, \{1\} \rangle$.

  $$\begin{array}{l} U_0 = 2; \\ U_1 = T; \\ U_2 = \{1\}. \end{array}$$

Tuple Equality

Proposition: Two n-tuples are equal iff their components are equal.

For every n and all $x_0, x_1, \ldots, x_{n-1}$ and all $y_0, y_1, \ldots, y_{n-1}$:

$$\langle x_0, x_1, \ldots, x_{n-1}\rangle = \langle y_0, y_1, \ldots, y_{n-1}\rangle \Leftrightarrow \\ (x_0 = y_0 \wedge x_1 = y_1 \wedge \ldots x_{n-1} = y_{n-1}).$$

Example:

- $\langle 1, 2\rangle \neq \langle 2, 1\rangle$;
- $\langle 1, 2\rangle \neq \langle 1, 2, 2\rangle$.

Cartesian Product

Definition: Cartesian Product of $S_0, \ldots, S_{n-1}$.

- Set of all n-tuples whose i-th component is in S_i:

$$S_0 \times \ldots \times S_{n-1} := \\ \{\langle x_0, \ldots, x_{n-1}\rangle : x_0 \in S_0 \wedge \ldots \wedge x_{n-1} \in S_{n-1}\}.$$

Example:

$$\{a, b\} \times \{0, 1, 2\} = \{\langle a, 0\rangle, \langle a, 1\rangle, \langle a, 2\rangle, \langle b, 0\rangle, \langle b, 1\rangle, \langle b, 2\rangle\}.$$

Predicates as Sets

Logic and Set Theory

- First-order predicate logic.
  - Variables may represent domain objects, not predicates or functions.
  - No quantifiers over predicates or functions.
  - Problem: "for all predicates p, ...", "there is a function f, such that ..."
- First-order predicate logic over domain of sets.
  - Domain objects are sets.
  - May encode predicates and functions as sets.
  - Interpret statements about sets as statements about predicates and functions.
  - Overcome limitations of first-order predicate logic.

The combination of first-order predicate logic and set theory is the working horse of mathematics.

Basic Idea

- A binary predicate p defines a set S of 2-tuples:
$$S := \{\langle x, y\rangle : p(x, y)\}.$$
- A set S of 2-tuples defines a binary predicate p:
$$p(x, y) :\Leftrightarrow \langle x, y\rangle \in S.$$

We may interpret a set of tuples as a predicate (relation).

Relations

Definition: relation

R is a relation between $S_0, \ldots, S_{n-1}$ $:\Leftrightarrow$
$R \subseteq S_0 \times \ldots \times S_{n-1}$.

R is an n-ary relation $:\Leftrightarrow$
$\exists S_0, \ldots, S_{n-1} : R$ is a relation between $S_0, \ldots, S_{n-1}$;

R is an n-ary relation on S $:\Leftrightarrow$
$R \subseteq S \times \ldots \times S$ (cartesian product of n instances of S).

R is a relation on S $:\Leftrightarrow$ R is a 2-ary (binary) relation on S.

Set of tuples of related values.

Relations

Definition: R holds on $x_0, \ldots, x_{n-1}$

$R(x_0, \ldots, x_{n-1}) :\Leftrightarrow \langle x_0, \ldots, x_{n-1}\rangle \in R.$

Example:
$$\forall x : \exists R : (\forall y : R(x, y) \Rightarrow x = y)$$
is interpreted as
$$\forall x : \exists R : (\forall y : \langle x, y\rangle \in R \Rightarrow x = y)$$

Blur distinction between relations as sets and predicates.

Example

- Let
$$R := \{\langle x, y\rangle : x \in \mathbb{N} \land y \in \mathbb{N} \land x \leq y\}$$

  R is a relation on $\mathbb{N}$ ($R \subseteq \mathbb{N} \times \mathbb{N}$); it is also a relation on $\mathbb{Z}$ ($R \subseteq \mathbb{Z} \times \mathbb{Z}$).
- Let
$$S := \{\langle x, x/2\rangle : x \in \mathbb{N}\}$$

  S is a relation between $\mathbb{N}$ and $\mathbb{Q}$ ($R \subseteq \mathbb{N} \times \mathbb{Q}$) and it is also a relation on $\mathbb{Q}$ ($R \subseteq \mathbb{Q} \times \mathbb{Q}$); it is not a relation on $\mathbb{N}$ ($R \not\subseteq \mathbb{N} \times \mathbb{N}$).

Representation of Relations

Predicate definition:

$$R(x,y) :\Leftrightarrow x \in \mathbb{N} \land y \in \mathbb{N} \land x < y < 5$$

Set quantifier:

$$R := \{\langle x,y\rangle : x \in \mathbb{N} \land y \in \mathbb{N} \land x < y < 5\}$$

Set enumeration:

$$R = \{\langle 0,1\rangle, \langle 0,2\rangle, \langle 0,3\rangle, \langle 0,4\rangle,$$
$$\langle 1,2\rangle, \langle 1,3\rangle, \langle 1,4\rangle, \langle 2,3\rangle, \langle 2,4\rangle, \langle 3,4\rangle\}.$$

Representation of Relations (Continued)

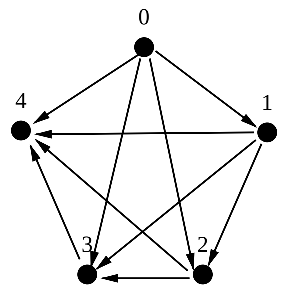
Truth table:

$x \backslash y$	0	1	2	3	4	...
0	false	true	true	true	true	
1	false	false	true	true	true	
2	false	false	false	true	true	
3	false	false	false	false	true	
4	false	false	false	false	false	
...						

Matrix entry for x/y is true iff x is in relation to y.

Representation of Relations (Continued)

Directed graph:



Graph has arrow from x to y iff x is in relation to y.

Domain and Range

Definition: domain and range of a binary relation R

- The domain of R is the set of first components of the tuples in R:

$$\text{domain}(R) := \{r_0 : r \in R\}.$$

- The range of R is the set of second components of the tuples in R:

$$\text{range}(R) := \{r_1 : r \in R\}.$$

Proposition:

$$\forall x : x \in \text{domain}(R) \Leftrightarrow \exists y : \langle x,y\rangle \in R,$$
$$\forall y : y \in \text{range}(R) \Leftrightarrow \exists x : \langle x,y\rangle \in R.$$

Example

Take the relation

$$R := \{\langle 0,0\rangle, \langle 0,1\rangle, \langle 0,2\rangle, \langle 1,2\rangle\}.$$

We have

$$\begin{aligned}\text{domain}(R) &= \{0,1\}\\ \text{range}(R) &= \{0,1,2\}.\end{aligned}$$

We have $\text{domain}(\emptyset) = \text{range}(\emptyset) = \emptyset$.

Inverse of a Relation

Definition: inverse of a binary relation R

$$R^{-1} := \{\langle b,a\rangle : a \in \text{domain}(R) \land b \in \text{range}(R) \land \langle a,b\rangle \in R\}.$$

Proposition:
The inverse of a relation from A to B is a relation from B to A.

$$\forall R, A, B : R \subseteq A \times B \Rightarrow R^{-1} \subseteq B \times A.$$

Example:

- $R := \{\langle 0,0\rangle, \langle 0,1\rangle, \langle 0,2\rangle, \langle 1,2\rangle\}$.
- $R^{-1} = \{\langle 0,0\rangle, \langle 1,0\rangle, \langle 2,0\rangle, \langle 2,1\rangle\}$.

Composition of Relations

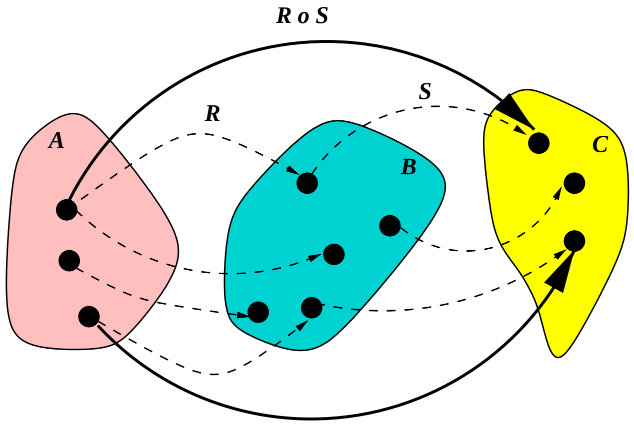
Definition: composition of two binary relations R and S.

$$\begin{aligned}R \circ S := \{\langle a,c\rangle : a \in \text{domain}(R) \land c \in \text{range}(S) \land\\ (\exists b : \langle a,b\rangle \in R \land \langle b,c\rangle \in S)\}.\end{aligned}$$

Proposition: The composition of a relation from A to B and of a relation from B to C is a relation from A to C.

$$\begin{aligned}&\forall R, S, A, B, C :\\ &\quad (R \subseteq A \times B \land S \subseteq B \times C) \Rightarrow R \circ S \subseteq A \times C.\end{aligned}$$

Visualization



Logic Evaluator

```
read relation;
> file 'set.txt' read.
> file 'relation.txt' read.
fun o(R: Relation, S: Relation) =
  let(A = domain(R), B = **(range(R), domain(S)), C = range(S):
    set(a in A, c in C:
      exists(b in B: and(in(tuple(a, b), R), in(tuple(b, c), S))),
      tuple(a, c)));
> function o/2.
fun R = set(x in nat(0, 5): true, tuple(x, *(2, x)));
> function R/0.
term R;
> {<0, 0>, <1, 2>, <2, 4>, <3, 6>, <4, 8>, <5, 10>}.
fun S = set(x in nat(0,2), y in nat(0,3): <=(x,*(2,y)), tuple(x,y));
> function S/0.
term S;
> {<0, 0>, <0, 1>, <0, 2>, <0, 3>, <1, 1>, <1, 2>, <1, 3>, <2, 1>,
<2, 2>, <2, 3>}.
term o(R, S);
> {<0, 0>, <0, 1>, <0, 2>, <0, 3>, <1, 1>, <1, 2>, <1, 3>}.
term o(S, R);
> {<0, 0>, <0, 2>, <0, 4>, <0, 6>, <1, 2>, <1, 4>, <1, 6>, <2, 2>,
<2, 4>, <2, 6>}.
```

```
term o(R, ^-1(R));
```


Composition Laws

For every A, B, C, D and every $R \subseteq A \times B$, $S \subseteq B \times C$, and $T \subseteq C \times D$, we have:

$$(R^{-1})^{-1} = R;$$
$$R \circ (S \circ T) = (R \circ S) \circ T;$$
$$(R \circ S)^{-1} = S^{-1} \circ R^{-1}.$$

Proof: see lecture notes (will be discussed later).

Functions as Sets

Basic Idea

- A unary function f defines a binary relation (the function graph):

$$S := \{\langle x, y\rangle : y = f(x)\}.$$

- A binary relation S defines a function, if the first component of every tuple in S is unique:

$$f(x) := \textbf{such } y : \langle x, y\rangle \in S.$$

We may interpret a special binary relation as a unary function.

Functions

Definition: function

$$\begin{aligned}&f \text{ is a function} :\Leftrightarrow\\ &\quad f \text{ is a binary relation} \wedge\\ &\quad \forall x, y_0, y_1 : (\langle x, y_0\rangle \in f \ \wedge \langle x, y_1\rangle \in f) \Rightarrow y_0 = y_1.\end{aligned}$$

Definition: partial function from A to B

$$\begin{aligned}&f : A \overset{\text{partial}}{\longrightarrow} B :\Leftrightarrow\\ &\quad f \text{ is a function} \wedge\\ &\quad f \subseteq A \times B.\end{aligned}$$

Definition: (total) function from A to B

$$\begin{aligned}&f : A \to B :\Leftrightarrow\\ &\quad f : A \overset{\text{partial}}{\longrightarrow} B \wedge\\ &\quad \forall x \in A : \exists y \in B : \langle x, y\rangle \in f.\end{aligned}$$

Functions

- Every unary function is a binary relation.

  The functions domain, range, inverse, and composition also apply to functions.
- n-ary functions can be modelled as unary functions.
  - Pack n arguments into a single n-tuple.
  - E.g., a binary function f is represented as (for some A, B, C)

$$f : A \times B \to C$$

Example

The following binary relations are functions:

- $\emptyset$;
- $\{\langle 0,0\rangle, \langle 1,0\rangle, \langle 2,1\rangle, \langle 3,2\rangle\}$;
- $\{\langle x, x^2\rangle : x \in \mathbb{N}\}$;
- $\{\langle x, x^2\rangle : x \in \mathbb{N} \wedge x \text{ is even}\} \cup \{\langle x, 0\rangle : x \in \mathbb{N} \wedge x \text{ is odd}\}$;
- $\{\langle x, y\rangle : x \in \mathbb{N} \wedge y \in \mathbb{N} \wedge x + y = 100\}$;
- $\{\langle\langle x, y\rangle, x + y\rangle : x \in \mathbb{N} \wedge y \in \mathbb{N}\}$.

Check unicity of first tuple component.

Example

The following binary relations are not functions:

- $\{\langle 0,0\rangle, \langle 0,1\rangle\}$;
- $\{\langle x, x^2\rangle : x \in \mathbb{N} \wedge x \text{ is prime}\} \cup \{\langle x, 0\rangle : x \in \mathbb{N} \wedge x \text{ is odd}\}$;
- $\{\langle x, y\rangle : x \in \mathbb{N} \wedge y \in \mathbb{N} \wedge x + y \leq 100\}$;

Every argument must be mapped to at most one result.

Function Applications

Definition: function application

$$\text{apply}(f,x) := \textbf{such } y : \langle x,y\rangle \in f.$$

Notation: we write $f(x)$ instead of $\text{apply}(f,x)$.
Notation: we write $f(x_0,\ldots,x_{n-1})$ to denote $f(\langle x_0,\ldots,x_{n-1}\rangle)$.

Proposition: a function maps every argument in its domain to a well-defined result.

$$\begin{aligned}&\forall f : f \text{ is function} \;\Rightarrow\\ &\quad \forall x \in \text{domain}(f) :\\ &\qquad f(x) \in \text{range}(f) \land \langle x, f(x)\rangle \in f.\end{aligned}$$

Example

A statement with quantified function variables

$$\forall y : \exists f : \forall x : f(x) = y$$

is interpreted as

$$\forall y : \exists f : \forall x : \text{apply}(f,x) = y$$

i.e., as

$$\forall y : \exists f : \forall x : \langle x,y\rangle \in f.$$

Blur distinction between functions as sets and functions in logic.

Defining Functions

Various common formats:

$$\begin{aligned}f &: A \to B\\ x &\mapsto T\end{aligned}$$

$$\begin{aligned}&f : A \to B\\ &f(x) := T\end{aligned}$$

$$\begin{aligned}&f : A \to B\\ &f := \lambda x.T\end{aligned}$$

$$f(x : A) : B = T$$

Defining Functions

All definition formats are interpreted as the formula

$$f : A \to B \;\land\; \forall x \in A : f(x) = T$$

or, equivalently, as

$$f = \{\langle x, T\rangle : x \in A\} \;\land\; (\forall x \in A : T \in B)$$

with a corresponding generalization to multiple arguments.

Example

The statement

$$\begin{array}{l} \mathrm{div} : \mathbb{N} \times \mathbb{N} \to \mathbb{Q} \\ \mathrm{div}(x, y) := x/y; \end{array}$$

is to be read as the formula

$$\begin{array}{l} \mathrm{div} = \{\langle\langle x, y\rangle, x/y\rangle : x \in \mathbb{N}, y \in \mathbb{N}\} \land \\ (\forall x \in \mathbb{N}, y \in \mathbb{N} : \mathrm{div}(x, y) \in \mathbb{Q}). \end{array}$$

Claim about function range has to be proved.

Function Inversion

Let

$$f := \{\langle 0, 0\rangle, \langle 1, 0\rangle, \langle 2, 1\rangle\}.$$

Then

$$f^{-1} = \{\langle 0, 0\rangle, \langle 0, 1\rangle, \langle 1, 2\rangle\}$$

is not a function, because it contains $\langle 0, 0\rangle$ and $\langle 0, 1\rangle$.

The inverse of a function is not necessarily a function.

Function Composition

Proposition: The composition of two functions is also a function,.

$$\begin{array}{l} \forall f, g, A, B, C : \\ \quad f : A \to B \land g : B \to C \ \Rightarrow \\ \quad\quad f \circ g : A \to C. \end{array}$$

Proof: see lecture notes (will be discussed later).

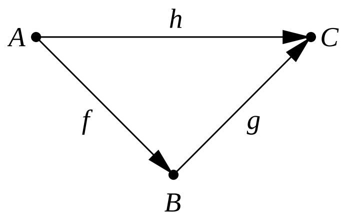
Proposition: direct characterization of function composition

$$\begin{array}{l} \forall f, g, A, B, C : \\ \quad f : A \to B \land g : B \to C \ \Rightarrow \\ \quad\quad \forall x \in A : (f \circ g)(x) = g(f(x)). \end{array}$$

Commutative Diagrams

Visualization of propositions about function compositions:

Proposition: $h = f \circ g$

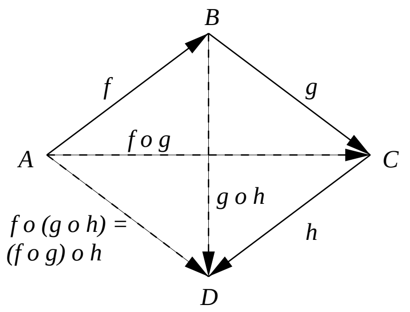


for $f : A \to B$ and $g : B \to C$ and $h : A \to C$.

Function Composition

Proposition: function composition is associative.

$$\forall A, B, C, D, f : A \to B, g : B \to C, h : C \to D :$$
$$f \circ (g \circ h) = (f \circ g) \circ h$$



Sequences and Matrices

Sequences

Definition: sequence of length n over S

s is a sequence of length n over S $:\Leftrightarrow s : \mathbb{N}_n \to S$.

Definition: length of a sequence

length(s) :=
such $n \in \mathbb{N}$: $\exists S : s$ is a sequence of length n over S.

Definition: finite sequence

s is a finite sequence over S $:\Leftrightarrow$
$\exists n \in N : s$ is a sequence of length n over S.

Definition: infinite sequence

s is an infinite sequence over S $:\Leftrightarrow s : \mathbb{N} \to S$.

Sequences

Indexed collections of elements:

- tables,
- arrays,
- vectors,
- lists.

i-th component $s(i)$ of a sequence s:

- s_i;
- $s[i]$;
- $s.i$.

Example

- $S := \{\langle 0,2\rangle, \langle 1,3\rangle, \langle 2,4\rangle, \langle 3,4\rangle, \langle 4,5\rangle\}$:
$$S = [2,3,4,4,5].$$
We have $\mathrm{length}(S) = 5$ and
$$S_0 = 2, S_1 = 3, S_2 = 4, S_3 = 4, S_4 = 5.$$
Furthermore $(\forall i \in \mathbb{N}_5 : 0 < S_i < 10)$.
- $T := \{\langle i, i^2\rangle : i \in \mathbb{N}\}$:
$$T = [0,1,4,9,16,25,\ldots].$$
We have $(\forall i \in \mathbb{N} : S_i = i^2)$.

Matrices

Definition: matrix with m rows and n columns over S

M is a $m \times n$-matrix over $S :\Leftrightarrow M : \mathbb{N}_m \times \mathbb{N}_n \to S$.

Matrix component $M(\langle i,j\rangle)$:

- $M(i,j)$;
- $M[i,j]$;
- $M_{i,j}$.

Example

$$\begin{aligned}&M := \\ &\quad \{\langle\langle 0,0\rangle,1\rangle, \langle\langle 0,1\rangle,2\rangle, \langle\langle 0,2\rangle,3\rangle, \\ &\quad \langle\langle 1,0\rangle,4\rangle, \langle\langle 1,1\rangle,5\rangle, \langle\langle 1,2\rangle,6\rangle, \\ &\quad \langle\langle 2,0\rangle,7\rangle, \langle\langle 2,1\rangle,8\rangle, \langle\langle 2,2\rangle,9\rangle\}\end{aligned}$$

Shorter:
$$M := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$
We have
$$\forall i \in \mathbb{N}_3, j \in \mathbb{N}_3 : M_{i,j} = 3i + j + 1.$$

Summary

- Sets
  - $\in$, $\emptyset$, set enumeration, $=$, $\subseteq$, set quantifier.
  - Union, intersection, difference, powerset.
  - Tuples, constructor $\langle\,\rangle$, selectors $._i$, $=$, Cartesian product.
- Relations.
  - Encoding and interpretation as sets.
  - Representation forms, truth tables, directed graphs.
  - Domain, range, inverse, composition.
- Functions.
  - Partial function, total function, function application.
  - Inversion, composition, commutative diagrams.
- Sequences and matrices as special functions.