Formal Method in Software Engineering (SE-313)

Course Teacher
Assistant Professor
Engr. Mustafa Latif

Introduction to Z Formal Specification

Z Formal Specification

Structure

Tuples and records

- Sets collect whole groups together, but we often need to associate particular individuals.
- Tuples associate particular elements of any type, in a fixed order.
- Tuples are instances of Cartesian product types, sometimes called cross product types.
- Example Date has data structure with three components the day, month, and year.
  - DAY = = 1 .. 31; MONTH = = 1 .. 12; YEAR == Z
  - DATE = = DAY x MONTH x YEAR
  -

$$\begin{array}{|l} landing, opening : DATE \\ \hline landing = (20, 7, 1969) \\ opening = (9, 11, 1989) \end{array}$$

3

Z Formal Specification

Structure

Tuples and records

- Example Consider a database of people who work at a company. The database records several items of information about each person: their name, an identification number, and the department where they work. Each item belongs to a different type:

$[NAME]$

$ID == \mathbb{N}$

$DEPARTMENT ::= administration \mid manufacturing \mid research$

4

Z Formal Specification

Structure

Tuples and records

- We can define a tuple that contains all three items:
  - EMPLOYEE == ID x NAME x DEPARTMENT
- Now we can define variables of this type:

$$\begin{array}{|l} Frank, Aki : EMPLOYEE \\ \hline Frank = (0019, frank, administration) \\ Aki = (7408, aki, research) \end{array}$$

Z Formal Specification

Structure

Relations, tables, and databases

- We usually work with whole sets of tuples.
- A set of tuples is called a relation.
- Relations can model tables and databases.

ID	Name	Department
0019	Frank	Administration
0308	Philip	Research
6302	Frank	Manufacturing
7408	Aki	Research
0517	Doug	Research
0038	Philip	Administration
⋮	⋮	⋮

$$\begin{array}{|l} Employee : \mathbb{P}\, EMPLOYEE \\ \hline Employee = \{ \\ \quad \vdots \\ \quad (0019, frank, administration), \\ \quad (0308, philip, research), \\ \quad (6302, frank, manufacturing), \\ \quad (7408, aki, research), \\ \quad (0517, doug, research), \\ \quad (0038, philip, administration), \\ \quad \vdots \end{array}$$

Z Formal Specification

Structure

Pairs and binary relations

- A particularly common kind of tuple is the pair: It has just two components.
- We can use a pair to associate a name with a telephone extension number, as in (aki, 4117). $aki \mapsto 4117$.
- Z also provides the first and second operators for extracting each component from a pair:

$$first(aki, 4117) = aki$$

$$second(aki, 4117) = 4117$$

- Operators like these that extract components from structures are called projection operators.

Z Formal Specification

Structure

Pairs and binary relations

- A binary relation is a set of pairs. Z provides an alternate syntax for declaring binary relations:
  - $\mathbb{P}$ (NAME x PHONE) can also be written as NAME $\leftrightarrow$ PHONE.

Z Formal Specification

Structure

Partial listing of the company telephone directory

Name	Phone
Aki	4117
Philip	4107
Doug	4107
Doug	4136
Philip	0113
Frank	0110
Frank	6190
⋮	⋮

notate it in Z:

- PHONE = = 0.. 9999

$$
\begin{array}{|l}
phone : NAME \leftrightarrow PHONE \\
\hline
phone = \{ \\
\quad \vdots \\
\quad aki \mapsto 4117, \\
\quad philip \mapsto 4107, \\
\quad doug \mapsto 4107, \\
\quad doug \mapsto 4136, \\
\quad philip \mapsto 0113, \\
\quad frank \mapsto 0110, \\
\quad frank \mapsto 6190, \\
\quad \vdots \\
\quad \}
\end{array}
$$

9

Z Formal Specification

Structure

Domain and range

- **The domain of phone includes every employee who can be reached by telephone: aki, doug, and the others.**
- **The range of phone includes all the numbers that have been assigned to telephones: 4117, 4017, and so forth.**
- dom R = { x : X ; y : Y | x ↦ y ∈ R • x }
- ran R = { x : X ; y : Y | x ↦ y ∈ R • y }

$$\mathrm{dom}\, phone = \{\ldots, aki, philip, doug, frank, \ldots\}$$

$$\mathrm{ran}\, phone = \{\ldots, 4117, 4107, 4136, 0113, 0110, 6190, \ldots\}$$

  - **(Again, the dots are not part of the Z notation, but indicate that many elements are not shown.)**
- **The domain and range of a relation are not necessarily the same as the sets that appear in its declaration, which are called the source set and the target set. In this example the source set is NAME, and the target set is PHONE; the domain and range are subsets of the source and target sets.**

10

Z Formal Specification

Structure

Binary relations

If X and Y are sets, then X $\boxed{\leftrightarrow}$ Y denotes the set of all relations between X and Y . The relation symbol may be defined by generic abbreviation: $X \leftrightarrow Y == \mathbb{P}(X \times Y)$

- Any element of X $\boxed{\leftrightarrow}$ Y is a set of ordered pairs in which the first element is drawn from X , and the second from Y : that is, a subset of the Cartesian product set X × Y .

Z Formal Specification

Structure

Binary relations

Example The set of relations {a, b} $\boxed{\leftrightarrow}$ {0, 1} is the set of sets of pairs

{∅, {(a, 0)}, {(a, 1)}, {(b, 0)}, {(b, 1)}, {(a, 0), (a, 1)}, {(a, 0), (b, 0)},
{(a, 0), (b, 1)}, {(a, 1), (b, 0)}, {(a, 1), (b, 1)}, {(b, 0), (b, 1)},
{(a, 0), (a, 1), (b, 0)}, {(a, 0), (a, 1), (b, 1)}, {(a, 0), (b, 0), (b, 1)},
{(a, 1), (b, 0), (b, 1)}, {(a, 0), (a, 1), (b, 0), (b, 1)}}

- The expression x ↦ y is another way of writing (x, y).

Z Formal Specification

Structure

Binary relations

Example The relation drives is used to record which makes of car are driven by the members of a small group of people. If the group of people is defined by

Drivers == {helen, indra, jim, kate}

and the choice of cars is defined by

Cars == {alfa, beetle, cortina, delorean}

then drives is an element of Drivers $\boxed{\leftrightarrow}$ Cars, and the statement 'Kate drives a cortina' could be formalized as kate $\mapsto$ cortina $\in$ drives.

13

Z Formal Specification

Structure

Example The relation drives could be defined by

$$\begin{array}{|l} \_drives\_ : Drivers \leftrightarrow Cars \\ \hline drives = \{helen \mapsto beetle, indra \mapsto alfa, jim \mapsto beetle, kate \mapsto cortina\} \end{array}$$

Example The set of people that drive is the domain of drives:

dom drives = {helen, indra, jim, kate}

The set of cars that are driven is the range:

ran drives = {alfa, beetle, cortina}

14

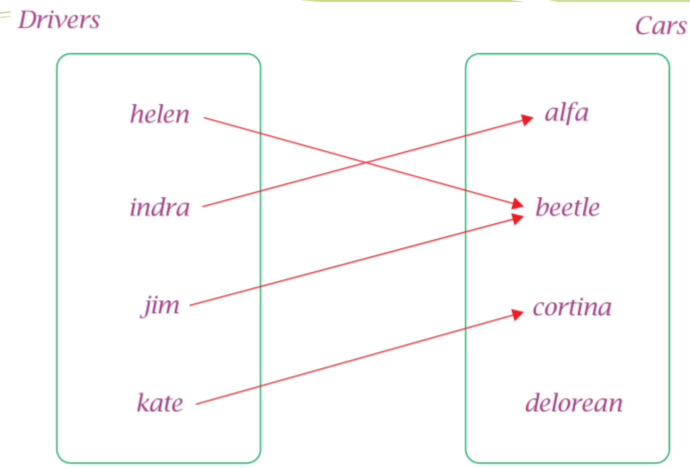

Figure Who drives what?

- Simple relations can be illustrated using diagrams with arrows, or graphs.

Z Formal Specification

Operators for relations: lookups, queries, updates, and inverses

- **Relations are important in computing.**
- **Z provides a rich collection of operators for binary relations.**
- **The Z relational operators behave like typical database operations.**

Z Formal Specification

Operators for relations: lookups, queries, updates, and inverses

Relational image operator (R(| A |)

Can model table lookup.

Its first argument is a relation, and its second argument is a set of elements from the domain, and its value is the set of corresponding elements from the range.

drives (| {indra, jim} |) = {alfa, beetle}

$phone(\!|\{doug, philip\}|\!) = \{4107, 4136, 0113\}$

17

Z Formal Specification

Operators for relations: lookups, queries, updates, and inverses

Domain restriction(◁) operator

Can model database queries

Selects tuples based on the values of their first elements: Its first argument is a set of elements from the domain of a relation,

Its second argument is a relation,

and its value is the matching tuples from the relation.

18

- To retrieve all the tuples for Doug and Philip from the phone relation, we apply domain restriction

$$\{doug, philip\} \lhd phone =$$

$$\begin{aligned}&\{philip \mapsto 4107,\\ &doug \mapsto 4107,\\ &doug \mapsto 4136,\\ &philip \mapsto 0113\}\end{aligned}$$

- The value of this expression is another relation of the same type as phone.

19

Z Formal Specification

Operators for relations: lookups, queries, updates, and inverses

Range restriction (▷) operator

Selects tuples based on the values of their second elements.

Its first argument is a relation,

its second argument is a set of elements from the range, and its value is the matching tuples.

20

- To retrieve all the tuples that have numbers in the 4000s from the phone relation, we apply range restriction:

$$
\begin{aligned}
&phone \rhd (4000 \mathinner{..} 4999) = \{ \\
&\quad \vdots \\
&\quad aki \mapsto 4117, \\
&\quad philip \mapsto 4107, \\
&\quad doug \mapsto 4107, \\
&\quad doug \mapsto 4136, \\
&\quad \vdots \\
&\quad \}
\end{aligned}
$$

21

- We can combine domain and range restriction. This expression finds the numbers for Doug and Philip in the 4000s:

$$\{doug, philip\} \lhd phone \rhd (4000 \mathinner{..} 4999) =$$

$$
\begin{aligned}
&\{philip \mapsto 4107, \\
&doug \mapsto 4107, \\
&doug \mapsto 4136\}
\end{aligned}
$$

22

Z Formal Specification

Operators for relations: lookups, queries, updates, and inverses

Range (⩤) and Range antirestriction (⩥) operator

S ⩤ R Selects tuples based on the values which are not in S

R ⩥ T Selects tuples based on the values which are not in T.

23

Z Formal Specification

Operators for relations: lookups, queries, updates, and inverses

The override operator ⊕

Can model database updates. Both of its arguments are relations.

Its value is a relation that contains the tuples from both relations, update existing tuple in first relation and add new if no replacement found.

24

$$phone \oplus \{heather \mapsto 4026, aki \mapsto 4026\} = \{$$

$$\begin{array}{l} \vdots \\ aki \mapsto 4026, \\ philip \mapsto 4107, \\ doug \mapsto 4107, \\ doug \mapsto 4136, \\ philip \mapsto 0113, \\ frank \mapsto 0110, \\ frank \mapsto 6190, \\ heather \mapsto 4026, \\ \vdots \\ \} \end{array}$$

25

Z Formal Specification

Operators for relations: lookups, queries, updates, and inverses

The inverse operator ~

The inverse operator reverses the direction of a binary relation by exchanging the first and second components of each pair.

26

- The inverse of the phone relation is a reverse directory from telephone numbers to names:

$$
\begin{aligned}
phone^{\sim} = \{ \\
&\vdots \\
&4117 \mapsto aki, \\
&4107 \mapsto philip, \\
&4107 \mapsto doug, \\
&4136 \mapsto doug, \\
&0013 \mapsto philip, \\
&0110 \mapsto frank, \\
&6190 \mapsto frank, \\
&\vdots \\
&\}
\end{aligned}
$$

Z Formal Specification

Composing relations

If the target type of one relation matches the source type of another, then they may be combined to form a single relation.

If R is an element of X $\boxed{\leftrightarrow}$ Y , and S is an element of Y $\boxed{\leftrightarrow}$ Z, then we write R ⨾ S to denote the relational composition of R and S. This is the element of X $\boxed{\leftrightarrow}$ Z such that

$x \mapsto z \in R \mathbin{⨾} S \Leftrightarrow \exists\, y : Y \bullet x \mapsto y \in R \land y \mapsto z \in S$

That is, two elements x and z are related by the composition R ⨾ S if there is an intermediate element y such that x is related to y and y is related to z.

- S

$$\begin{array}{|l} phone : NAME \leftrightarrow PHONE \\ \hline phone = \{ \\ \qquad \vdots \\ \qquad aki \mapsto 4117, \\ \qquad philip \mapsto 4107, \\ \qquad doug \mapsto 4107, \\ \qquad doug \mapsto 4136, \\ \qquad philip \mapsto 0113, \\ \qquad frank \mapsto 0110, \\ \qquad frank \mapsto 6190, \\ \qquad \vdots \\ \qquad \} \end{array}$$

$$\begin{array}{|l} dept : PHONE \leftrightarrow DEPARTMENT \\ \hline dept = \{ \\ \qquad 0000 \mapsto administration, \\ \qquad \vdots \\ \qquad 0999 \mapsto administration, \\ \qquad 4000 \mapsto research, \\ \qquad \vdots \\ \qquad 4999 \mapsto research, \\ \qquad 6000 \mapsto manufacturing, \\ \qquad \vdots \\ \qquad 6999 \mapsto manufacturing\} \end{array}$$

-

29

$$\begin{array}{l} phone \mathbin{;} dept = \{ \\ \qquad \vdots \\ \qquad aki \mapsto research, \\ \qquad philip \mapsto research, \\ \qquad doug \mapsto research, \\ \qquad philip \mapsto administration, \\ \qquad frank \mapsto administration, \\ \qquad frank \mapsto manufacturing, \\ \qquad \vdots \\ \qquad \} \end{array}$$

30

Z Formal Specification

Binary relations and linked data structures

- **Relations are not just for modelling tables and flat databases.**
- **Can model linked data structures as well.**
- **Linked data structures are often pictured as graphs: networks of nodes connected by arcs. (e.g. Data flow diagrams, state transition diagrams, and syntax trees are all examples of graphs.)**

31

Z Formal Specification

Binary relations and linked data structures

- **We can model any graph as a binary relation where both the domain and range are drawn from the same set: the set of nodes in the graph.**
- **Each arc in the graph is a pair in the relation.**

32

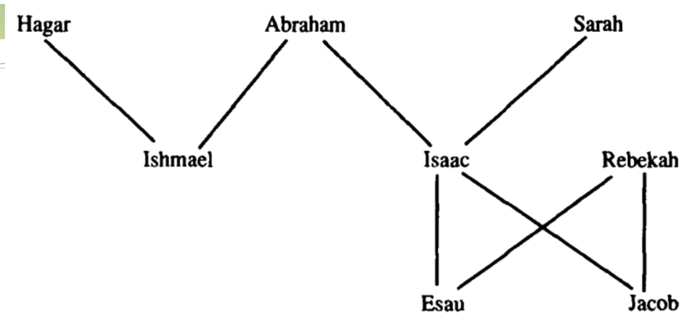

Figure 9.1: A genealogy.

$$PERSON ::= hagar \mid abraham \mid sarah \mid ishmael \mid isaac \mid rebekah \mid esau \mid jacob$$

$$\begin{aligned} child == \{ & hagar \mapsto ishmael, abraham \mapsto ishmael, abraham \mapsto isaac, \\ & sarah \mapsto isaac, isaac \mapsto esau, isaac \mapsto jacob, rebekah \mapsto esau, \\ & rebekah \mapsto jacob\} \end{aligned}$$

$$child(\!|\{abraham, sarah\}|\!) =$$

33

$$child(\!|\{abraham, sarah\}|\!) = \{ishmael, isaac\}$$

34