CHAPTER 7

Functions

Aims

To introduce the concept of the function as a specialised sort of relation, to introduce function operators and to show how functions may be used in Z specifications.

Learning objectives

When you have completed this chapter, you should be able to:

- recognise whether a given relation is a function, and whether a function is total, injective, surjective or bijective;
- decide where functions are needed within your specifications;
- understand and use the operators introduced earlier in this book, and those introduced in this chapter, in creating function-valued expressions.

7.1 Introduction

The concept of the function is a very useful and powerful one in computer science. Procedural programming languages such as C and Pascal support the use of functions as algorithms for transforming input parameters or arguments into output values. The functional programming paradigm, as represented by languages such as ML and Miranda, is based on the idea of constructing programs from side-effect-free functions. The expressive power of such languages facilitates the production of succinct, elegant, understandable and modifiable code. As we shall see, functions are also an extremely important tool in writing Z specifications.

7.2 Functions in Z

A function is a special sort of relation. A relation $f: A \leftrightarrow B$ is a *partial function* iff it satisfies the following condition:

$$\forall x : A; y, z : B \bullet x \mapsto y \in f \land x \mapsto z \in f \Rightarrow y = z$$

In other words, in the picture of f, there is *at most* one arrow emerging from each element of the source set, or to put it another way, there are no diverging arrows emerging from any element of dom f. Note that there is not necessarily an arrow emerging from every source set element. A partial function f is declared as

$$f: A \rightarrow\!\!\!\!\!+ B$$ partial

A *total function,* declared as

$$f: A \rightarrow B$$ total

is a partial function for which

$$\operatorname{dom} f = A$$

that is, every source element has an arrow emerging. The use of the declarations $f: A \rightarrow\!\!\!\!\!+ B$ and $f: A \rightarrow B$ is therefore a shorthand for the declaration $f: A \leftrightarrow B$ together with the appropriate predicates from above. This means that if f is part of a state schema, we must ensure that any state-changing operations do not violate these predicates, which are an implicit part of the state invariant. In other words, if we have declared an object as a total or partial function, it must remain one!

For a function f as above, if we have

$$x \mapsto y \in f$$

then we may write

$$f x = y$$

The notation fx is read as 'the function f applied to the argument x'. The concept is analogous to the notion of relational image discussed in the last chapter.

Exercises 7.1

The following types are given:

cols ::= *red* | *blue* | *green*
plants ::= *pansy* | *geranium*

1. State whether each of the following is a partial function, a total function or neither.
   (i) $f = \{red \mapsto geranium\}$ P
   (ii) $g = \{blue \mapsto pansy,\ blue \mapsto geranium\}$ N
   (iii) $h = \{red \mapsto pansy,\ blue \mapsto pansy,\ green \mapsto pansy\}$ T
   (iv) $i = \{\}$ P (if i : cols ⇸ plants).
2. What can you say about the type of h^{-1} (the inverse of h)? N
3. The *backward composition* of relations $S : B \leftrightarrow C$ and $R : A \leftrightarrow B$ is denoted by $S \circ R$ and is the relation of type $A \leftrightarrow C$ defined as follows:

$$S \circ R = \{a : A; c : C \mid (\exists b : B \bullet a \mapsto b \in R \wedge b \mapsto c \in S) \bullet a \mapsto c\}$$

   (i) What is the relationship between $S \circ R$ and $R ; S$?
   (ii) If R and S are functions, write down an alternative expression for $(S \circ R)x$.
   (iii) Given the function $double : \mathbb{N} \rightarrow \mathbb{N}$, where

$$\forall x : \mathbb{N} \bullet double\ x = 2 * x$$

   how would you describe the function $double \circ double$?

7.3 Function overriding

The overriding operator, $\oplus$, may be applied to any two functions f and g, of the same type, and the result is another function. The expression

$$f \oplus g$$

read as 'f overridden by g', defines a new function with the following properties:

$$\mathrm{dom}(f \oplus g) = \mathrm{dom}\, f \cup \mathrm{dom}\, g$$

$$\forall x : \mathrm{dom}(f \oplus g) \bullet ((x \in \mathrm{dom}\, g \Rightarrow (f \oplus g)x = gx) \\ \wedge (x \in \mathrm{dom}\, f \wedge x \notin \mathrm{dom}\, g \Rightarrow (f \oplus g)x = fx))$$

In other words, $f \oplus g$ behaves the same as f when applied to objects not in the domain of g, and behaves as g otherwise. The term 'overriding' refers to the fact that, when $f \oplus g$ is applied to objects in the intersection of the domains of f and g, it is the maplets from g which take precedence.

For example, let f and g be as follows:

$$f, g : \mathbb{N} \rightarrow\!\!\!\!\!\!+\ \mathbb{N}$$
$$f = \{3 \mapsto 9, 4 \mapsto 16, 5 \mapsto 25\}$$
$$g = \{2 \mapsto 7, 3 \mapsto 16, 4 \mapsto 17\}$$

This situation is illustrated in Figure 7.1

From Figure 7.1, we see that 3 is mapped to 9 by f and to 16 by g. Therefore 3 is mapped to 16 by $f \oplus g$. Similarly, 4 is mapped to 16 by f and to 17 by g. Therefore 4 is mapped to 17 by $f \oplus g$. The resultant function is therefore

$$f \oplus g = \{2 \mapsto 7, 3 \mapsto 16, 4 \mapsto 17, 5 \mapsto 25\}$$

This is illustrated in Figure 7.2.

As stated above, a function is a restricted sort of relation, which in turn is a restricted sort of set. This means that all the relation operators and set operators introduced in earlier chapters may be applied to functions, although as we discovered in Exercises 7.1 the result of such applications is not necessarily a function.

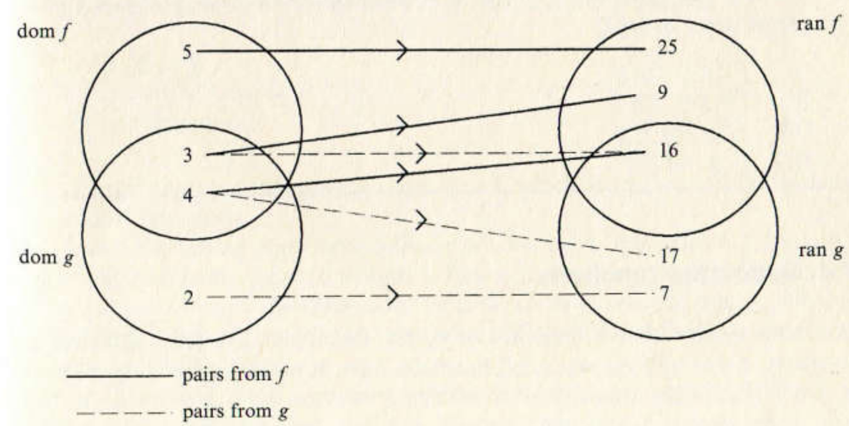

Figure 7.1 The functions *f* and *g*

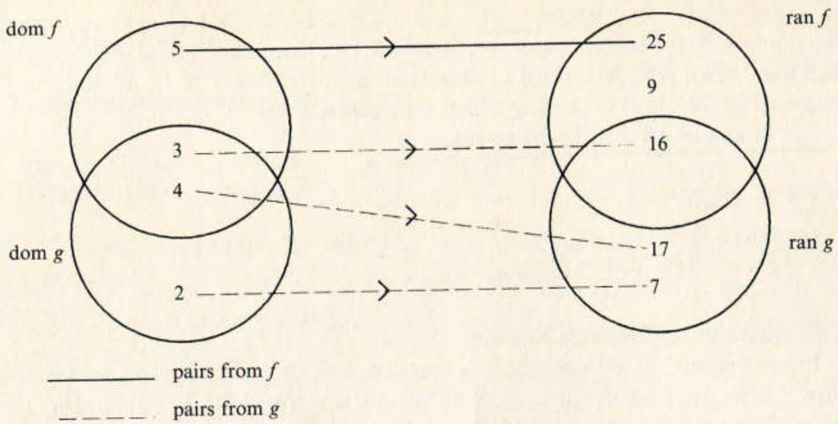

Figure 7.2 Function f overridden by function g ($f \oplus g$)

Exercises 7.2

1. For the above functions f and g, what is the value of the following expressions?

   (i) $g \oplus f$
   (ii) $f^{-1} \oplus g^{-1}$
   (iii) $\{5\} \lhd f \oplus g \rhd \{17, 7\}$
   (iv) $f \cap g \oplus f \cup g$
   (v) $(f^{-1} ; g) \oplus g$

2. For *any* two functions f and g, in what circumstances could the following expressions be true?

   (i) $f \cup g = f \oplus g$
   (ii) $f \oplus g = g \oplus f$
   (iii) $f \cap g = f \oplus g$
   (iv) $f \setminus g = f \oplus g$

7.4 Restricted functions

We may specify that a function is to be restricted to satisfy some given property. Some of these restricted functions have names and special symbols in Z (variations of the arrow symbol defining a function type). However, they are not often required, and the symbols will not be listed here. The special functions are as follows.

A function $f: A \twoheadrightarrow B$ is an *injection* iff every value in ran f occurs in precisely one maplet in f. In other words, there are no converging arrows in the picture of f. An injective function is also called a *one-to-one* function. To specify that f has this property, we could include the following predicate in the schema in which f is defined:

$$\forall x, y : A; z : B \bullet fx = z \land fy = z \Rightarrow x = y$$

The function $f: A \twoheadrightarrow B$ is a *surjection* iff *every* value in B is mapped to by the function, that is

$$\text{ran } f = B$$

A surjective function is said to be *onto* its target.

The function $f: A \twoheadrightarrow B$ is a *bijection* iff f is an injection, a surjection and is total.

Exercises 7.3

1. Give an alternative predicate to specify that the function $f: A \twoheadrightarrow B$ is a surjection.
2. What can you say about the inverse of a function which is an injection?
3. What can you say about the inverse of a function which is a surjection?
4. What can you say about the inverse of a function which is a bijection?
5. We can define a global constant function in a specification using an *axiomatic description* which, you will recall from Chapter 4, is a Z construct for defining a global object. Describe the following function in English, and state which of the above properties it possesses.

$$f : \mathbb{Z} \rightarrow \mathbb{Z}$$
$$\forall x : \mathbb{Z} \bullet fx = x$$

6. Give examples of functions of type $\mathbb{N} \twoheadrightarrow \mathbb{N}$ which are injections, surjections and/or bijections.
7. Recall the fishing game specification in Chapter 4, Exercises 4.7. Extend the state schema there to include a function which maps each fish in the pond to its weight rounded to the nearest whole number of grams. Write a schema to specify an operation to return the weight of the heaviest fish currently in the net.