Developing Formal Predicates from Informal Statements: A Checklist

Variables occurring free should be disjoint from those occurring bound

In order to minimize the liklihood of confusion, formal predicates should be written so that no variable occurs both free and bound in the same expression.

To illustrate this, consider the informal statement

> The value 37 occurs at location i of array b, and -2 occurs somewhere in b.

A translation that fails to follow this recommendation (yet is technically correct) is

$$(b.i = 37) \ \wedge \ (\exists i \mid 0 \leq i < \#b \ : \ b.i = -2)$$

The occurrence of i in the first conjunct is free (and hence the truth value of that conjunct depends upon the value of i in the evaluation state), whereas all occurrences of i in the second conjunct are bound (and hence the truth value of that conjunct does not depend upon the value of i in the evaluation state). (In case it is not clear why the second conjunct's value does not depend upon i, consider that, if you "expand" the existential quantification, you get the disjunction

$$b.0 = -2 \ \vee \ b.1 = -2 \ \vee \ \cdots \ \vee \ b.(\#b - 1) = -2$$

nowhere in which i appears.)

In other words, the i used in the first conjunct has nothing to do with the i used in the second!

To lessen the chance of confusion, we should rename the dummy in the second conjunct (see **(8.21) Axiom, Dummy Renaming** in Gries and Schneider). If we choose j as the new name, we get

$$(b.i = 37) \ \wedge \ (\exists j \mid 0 \leq j < \#b \ : \ b.j = -2)$$

Suppose that, instead of renaming i to j in the second conjunct, we did it in the first. Then our formal predicate would correspond to the informal statement

> The value 37 occurs at location j of array b, and -2 occurs somewhere in b.

This is not equivalent to the original informal statement, because its truth value depends upon the value of j (and not upon i's value) whereas the original statement's truth value depends upon i's value (but not j's).

A variable should occur free if it is mentioned in the informal statement

As an example, consider the statement

Location i of array b contains x.

Each of i, b, and x occurs in this statement; hence, there should be at least one free occurrence of each one in the corresponding formal predicate. For this reason, we can immediately reject the following two formal predicates:

$$(\exists i \mid 0 \leq i < \#b \ : \ b.i = x) \quad \text{(no free occurrences of } i\text{)}$$

$$(\exists x \mid : \ b.i = x) \quad \text{(no free occurrences of } x\text{)}$$

An exception to the rule applies when the informal statement is written in a tortured, unnecessarily formal way that obviously suggests a quantification. As an example, take

At some location i, b contains x.

(Or replace "some" by "every".) Here, even though i is mentioned explicitly, it should be clear that it is being used as a dummy. To convince yourself of this, it's necessary only to rename i as j (or any other name not already appearing in the statement) and to consider whether the resulting statement means the same thing as the original. If it does, the variable in question is serving as a dummy for a quantification.

Or, perhaps even better, try to re-word the statement without using the variable at all. If you can, then clearly that variable's value does not influence the meaning of the statement and hence there is no need for it to occur free in the formalized version. In this case, we can re-word the statement as

x occurs (somewhere) in b.

A corresponding formal predicate is

$$(\exists i \mid 0 \leq i < \#b \ : \ b.i = x)$$

A variable should occur free only if it is mentioned in the informal statement

As an example, consider the statement

Addition is commutative.

No variable occurs in this statement; hence there should be no free occurrences of variables in the formal equivalent. For this reason, we reject

$$x + y \ = \ y + x$$

in which there are free occurrences of x and y. The correct way to write it is

$$(\forall x, y \mid :\ x + y\ =\ y + x)$$

Or, using the *everywhere* operator (which implicitly universally quantifies over all variables having free occurrences), this could be written as

$$[x + y\ =\ y + x]$$

Check compatibility of data types

Consider the informal statement

> The value 23 occurs at least twice in b.

Obviously, this statement is either true or false. Thus, any corresponding formal statement should be of type boolean. That excludes an expression such as

$$(\#i \mid 0 \leq i < \#b\ :\ b.i = 23)$$

from being a correct translation, as it is of type integer.

Or, consider the following attempt to formalize the statement

> k is the product of two prime numbers.

$$(\exists p, q \mid :\ k = \text{isPrime}.p \cdot \text{isPrime}.q)$$

This is incorrect because the multiplication operator $\cdot$ is being applied to two boolean values.