

## 3.5 Solution

Here's the answer to the problem I posed earlier, of reformatting a file of city and country names.

```

1  my %table;

2  while (<>) {
3      chomp;
4      my ($city, $country) = split /, /;
5      $table{$country} = [] unless exists $table{$country};
6      push @{$table{$country}}, $city;
7  }

8  foreach $country (sort keys %table) {
9      print "$country: ";
10     my @cities = @{$table{$country}};
11     print join ', ', sort @cities;
12     print ".\n";
13 }
```

The program has two pieces: Lines 2–7 read the input and build a data structure, and lines 8–13 analyze the data and print out the report. We're going to have a hash, %table, whose keys are country names, and whose values are references to arrays of city names. The data structure will look like this:

```

      %table
+-----+-----+
|      |      | +-----+-----+
|Germany| *---->| Frankfurt | Berlin |
|      |      | +-----+-----+
+-----+-----+
|      |      | +-----+
|Finland| *---->| Helsinki |
|      |      | +-----+
+-----+-----+
|      |      | +-----+-----+-----+
|  USA  | *---->| Chicago | Washington | New York |
|      |      | +-----+-----+-----+
+-----+-----+
```

We'll look at output first. Supposing we already have this structure, how do we print it out?

```

8  foreach $country (sort keys %table) {
9      print "$country: ";
10     my @cities = @{$table{$country}};
11     print join ', ', sort @cities;
12     print ".\n";
13 }
```

%table is an ordinary hash, and we get a list of keys from it, sort the keys, and loop over the keys as usual. The only use of references is in line 10. \$table{\$country} looks up the key \$country in the hash and gets the value, which is a reference to an array of cities in that country. **Use Rule 1** says that we can recover the array by saying @{\$table{\$country}}. Line 10 is just like

```
@cities = @array;
```