

Table 7.2: *In the 'In credit' state, the event 'withdraw money' can have different results*

<i>Event</i>	<i>Guard</i>	<i>Result</i>
money is withdrawn	the new balance is greater than or equal to zero	the account remains in the 'In credit' state
money is withdrawn	the new balance is less than zero and within the overdraft limit	the account moves into the 'Overdrawn' state

Table 7.3: *In the 'Overdrawn' state, the event 'deposit money' can have different results*

<i>Event</i>	<i>Guard</i>	<i>Result</i>
deposit money	the new balance is still less than zero	the account remains in the 'Overdrawn' state
deposit money	the new balance is zero or more	the account returns to the 'In credit' state

Another event that can occur in the 'Overdrawn' state is 'withdraw money'. This has a guard '[new balance < 0 and within overdraft limit]', which means that you can still withdraw money while in the 'Overdrawn' state as long as you do not exceed your overdraft limit.

Constructing a state diagram

In our second example we show you how to build a state diagram.¹ You can find a list of all the steps involved in the summary at the end of the chapter. This example concerns a Human Resources system, where one class, Job Application, is complex enough to justify drawing a state diagram. The diagram will illustrate all the different possible behaviours of objects of the Job Application class.

1. It is also possible to draw a state diagram starting from the interaction diagrams; for details of how to do this, see Bennett et al., (2002).