

- that b is a friend of a or that both a and b are its friends, and
- that the age of p is currently x ,

then i is

- committed to a to inform it (a) of p 's age, x , and
- committed to b to inform it (b) of p 's age, x .

(Note: a , b , p , and x are variables. The values of b and p are fixed in the message received from the other agent. The value of x is fixed by agent i 's beliefs.)

(ii) If some agent, a , informs i that tomorrow is the birthday of some other agent, b , or that tomorrow is the day some other agent, b , graduates, and i believes

- that the other agent (b) is his (i 's) friend and
- that the name of the other agent (b) is n ,

then i is

- committed to the second agent (b) to make a cake tomorrow on which is written "Happy birthday/graduation n ",
- committed to the first agent (a) to ask the second agent (b) tomorrow to have a happy birthday/graduation, and
- committed to the first agent (a) to tell him tomorrow that he sent the message.

(Note: a , b , and n are variables. The value of b is fixed in the message received from the other agent. The value of n is fixed by agent i 's beliefs. Assume that the time grain is one day. Making a cake is a private action with a parameter for what is written on the cake. You have to work out a way to remember whether it is a birthday or a graduation.)

4. The following Concurrent MetateM system has three agents. One, `hammerm`, makes hammers from the heads supplied by `headsup` and handles supplied by `handlesup`. Describe the agents in English and tabulate the first few steps in an example run using Figure 3.6 on page 60 of the text as an example. Proposition `make` is true when `hammerm` makes a hammer, `supply_head` is true when `headsup` supplies a head (to `hammerm`), and `supply_handle` is true when `handlesup` supplies a handle (to `hammerm`).

```

headsp(make)[supply_head,supply_handle]
  ( ¬ make S supply_head ) ∧ ( ¬ make S supply_handle ) ⇒
    ◇( make ∧ O( ¬ make U ( ◇supply_head ∧ ◇supply_handle ) )
headsup(make)[supply_head]
  ●make ⇒ ◇supply_head.
handlesup(make)[supply_handle]
  ●make ⇒ ◇supply_handle.

```

5. Specify a MetateM system with two agents that dig a hole. They take turns digging (one lifts out a shovel full of soil, then the other does, then the first does, etc.) until they hit bedrock.