



Figure C.9: Controller that keeps the nondeterministic cat alive.

C.4.2 Independent section

The state machine for the controller is shown in figure C.9. Notice that the output that is produced depends only on the state, not on the input. Here is the *output* function of the controller:

```

% CONTROLLEROUTPUT - Given the state of the controller,
% return its output.
function output = controllerOutput(state)

switch(state)
case 'happy'
    output = 'time passes';
otherwise
    output = 'feed';
end

```

The state update of the controller, of course, does depend on the input. The *update* function for the controller is given below:

```

% CONTROLLER - A function representing the state update of
% a state machine providing inputs to keep a virtual pet alive.
% The first argument must be in {'happy', 'hungry'}
% The second argument can be output from the pet.
% The returned value is a 1x2 cell array with the
% next state and the output.

function r = controller(state, in)

if ~strcmp(in, 'absent')
    switch(state)
    case 'happy'

```

```

        if strcmp(in, 'rubs')
            r = {'hungry', 'time passes'};
        else
            r = {'happy', 'time passes'};
        end
    otherwise
        if strcmp(in, 'rubs')
            r = {'hungry', 'feed'};
        else
            r = {'happy', 'feed'};
        end
    end
end
else
    % The default behavior is to stutter.
    r = {state, 'absent'};
end
end

```

The program that drives the controller and the nondeterministic cat in a feedback loop is:

```

% DRIVELOOP - Execute the virtual pet state machine composed
% in a feedback loop with the controller state machine.

% Set the initial states.
controllerstate='happy';
petstate='happy';

% loop 10 times, since this is automatically driven.
for i=1:10,
    % Determine the output of the controller.
    petinput = controllerOutput(controllerstate);

    % update the state of the pet and get its output.
    r = chooserow(petUpdates(petstate, petinput));
    petstate = r{1};
    petoutput = r{2};

    % Update the state of the controller.
    % Ignore its output since we've already processed it.
    r = controller(controllerstate, petoutput);
    controllerstate = r{1};

    % Display the output of the pet.
    disp(petoutput)
end

```

Here is a typical output:

```
>> driveloop  
rubs  
purrs  
rubs  
purrs  
rubs  
purrs  
rubs  
rubs  
purrs  
rubs
```