Systems Design Laboratory

Hybrid Automata

[1]Department of Mathematics, University of Padova, ITALY

[2]Department of Computer Science, University of Verona, ITALY

Hybrid Automata

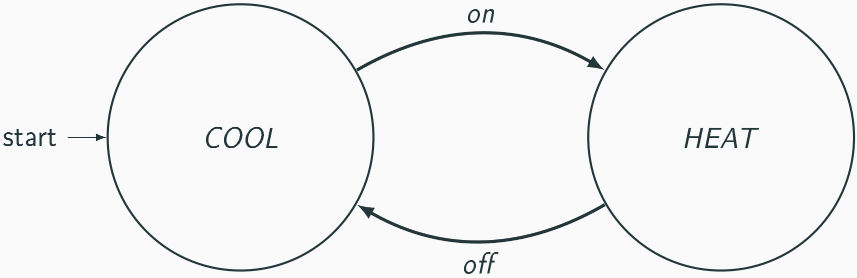
Hybrid = Discrete + *Continuous*

Discrete part - Locations

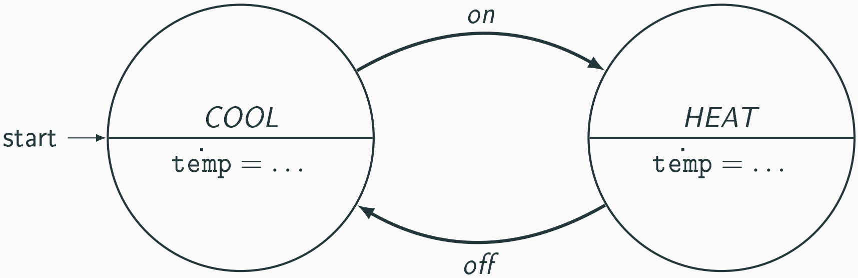
Here, we have two locations: $COOL$ and $HEAT$.

Discrete part - Events



Here, we have two events: *on* and *off*.

Continuous part - continuous variables



Here, we have a continuous variable $\texttt{temp} \in \mathbb{R}$ modeling the temperature of the room.

Hybrid Automata - differential equations

- The dynamics of variables are expressed in terms of differential equations. We use Ordinary Differential Equations (ODEs).
- An ODE is an equation involving an unknown function $y(x)$ and its derivatives $y'(x), y''(x), \ldots, y^n$.
- The unknown function $y(x)$, if it exists, is the solution of the ODE. Moreover,
  - if no initial condition $y(0) :=?$ is given, then $y(x)$ actually represents a family of functions;
  - if the initial condition $y(0) := y_0$ is given, then $y(x)$ is unique (Initial Value Problem).

Hybrid Automata - differential equations - example

Suppose that $\texttt{temp}(t)$ is an unknown function modeling how the temperature of a room changes (continuously) over time t.

Even if we do not know the expression of $\texttt{temp}(t)$ we might know a differential *evolution law* such as:

$$\overbrace{\texttt{temp}'(t)}^{\text{1st derivative of } \texttt{temp}(t)} = (30 - \overbrace{\texttt{temp}(t)}^{\text{Unknown function}})$$

Such an ODE can be solved analytically leading to the family of functions:

$$\texttt{temp}(t) = c_1 \cdot e^{-t} + 30$$

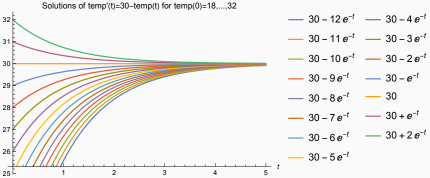
By adding initial conditions, then our $\texttt{temp}$ is no longer unknown.

$$\begin{cases} \texttt{temp}'(t) = 30 - \texttt{temp}(t) \\ \texttt{temp}(0) = 10 \end{cases} \quad \Rightarrow \quad \overbrace{\texttt{temp}(t)}^{\text{Known function}} = 30 - 20e^{-t}$$

Some dynamics are more equal than others

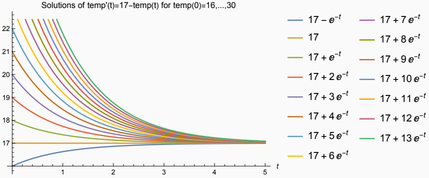
Heating dynamic

$$\mathtt{temp}'(t) = 30 - \mathtt{temp}(t)$$



Cooling dynamic

$$\mathtt{temp}'(t) = 17 - \mathtt{temp}(t)$$

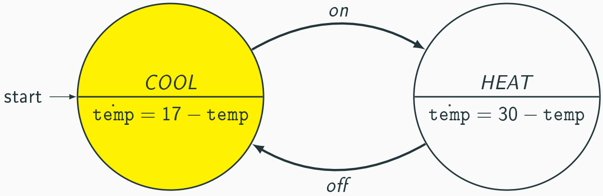


The dynamic $\mathtt{temp}'(t) = x - \mathtt{temp}(t)$ reaches the threshold x exponentially fast, where $\mathtt{temp}(t)$ is such that:

- If $\mathtt{temp}(0) < x$, then $\mathtt{temp}(t)$ is monotone strictly increasing;
- If $\mathtt{temp}(0) > x$, then $\mathtt{temp}(t)$ is monotone strictly decreasing;
- If $\mathtt{temp}(0) = x$, then $\mathtt{temp}(t) = x$ is constant.

Continuous part - the concept of state

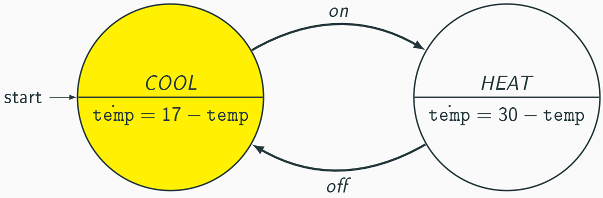
State = (Location, values of the variables)



- In general, an HA has infinite states
- Going from one state to the next defines a **trajectory**.

Continuous part - dynamics

At the beginning the temperature is $\mathtt{temp}(0) = 18$ degrees and the current location is *COOL*.



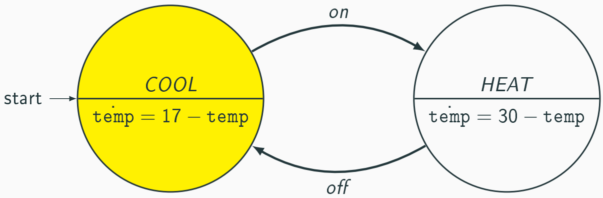
The temperature evolves according to $\mathtt{temp}(t)$ computed as follows:

$$\begin{cases} \mathtt{temp}'(t) = 17 - \mathtt{temp}(t) \\ \mathtt{temp}(0) = 18 \end{cases} \quad \Rightarrow \quad \mathtt{temp}(t) = 17 + e^{-t}$$

The current state is $(COOL, 18)$ since $\mathtt{temp}(0) = 17 + e^0 = 18$.

Continuous part - example of run

Suppose that the HA stays for 0.69 hours in COOL.



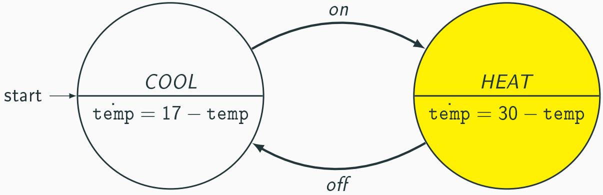
The temperature lowers to 17.5 degrees since

$$\texttt{temp}(0.69) = 17 + e^{-0.69} \approx 17.5$$

Thus, the current state is $(COOL, 17.5)$.

Continuous part - example of run

After staying for 0.69 hours in *COOL*, we immediately execute the transition labeled by *on* and move to location *HEAT* where the HA starts heating up the room.



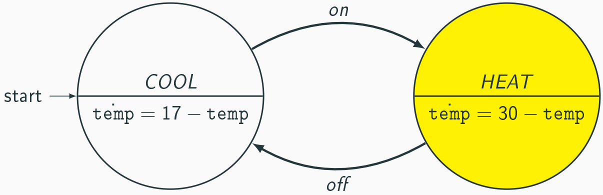
The temperature evolves according to $\mathtt{temp}(t)$ computed as follows:

$$\begin{cases} \mathtt{temp}'(t) = 30 - \mathtt{temp}(t) \\ \mathtt{temp}(0) = 17.5 \end{cases} \Rightarrow \quad \mathtt{temp}(t) = 30 - 12.5e^{-t}$$

The current state is $(HEAT, 17.5)$ since $\mathtt{temp}(0) = 30 - 12.5e^0 = 17.5$

Continuous part - example of run

Suppose that the HA stays 1 hour in *HEAT*.



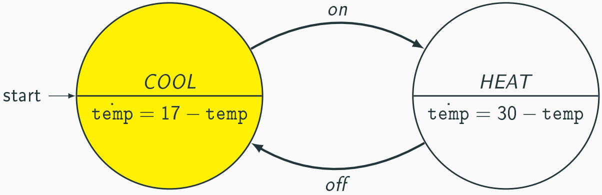
The temperature raises to 25.4 degrees since

$$\texttt{temp}(1) = 30 - 12.5e^{-1} \approx 25.4$$

Thus, the current state is $(HEAT, 25.4)$.

Continuous part - example of run

After staying for 1 hour in *HEAT*, we immediately execute the transition labeled by *off* and move to location *COOL* where the HA starts cooling down the room.



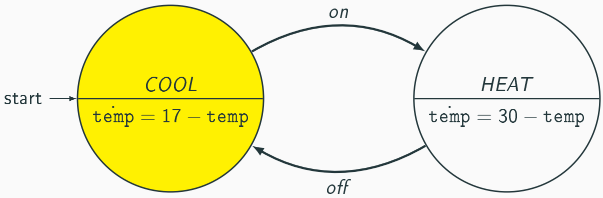
The temperature evolves according to $\texttt{temp}(t)$ computed as follows:

$$\begin{cases} \texttt{temp}'(t) = 17 - \texttt{temp}(t) \\ \texttt{temp}(0) = 25.4 \end{cases} \Rightarrow \quad \texttt{temp}(t) = 17 + 8.4e^{-t}$$

The current state is $(COOL, 25.4)$ since $\texttt{temp}(0) = 17 + 8.4e^0 = 25.4$

Continuous part - example of run

Suppose that the HA stays 2 hours in *COOL*.

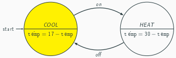
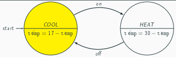
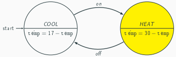
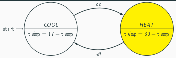
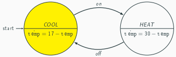
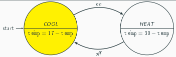


The temperature decreases to 18.13 since

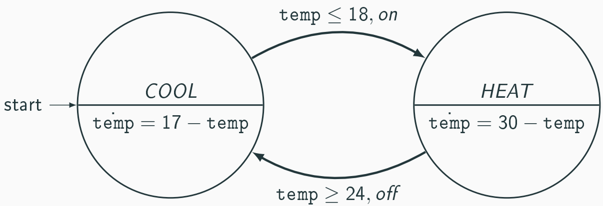
$$\texttt{temp}(2) = 17 + 8.4e^{-2} \approx 18.13$$

Thus, the current state is $(COOL, 18.13)$.

Continuous part - summary of the previous example run

Location	State	Dynamic
	$(COOL, 18)$	$\mathtt{temp}(t) = 17 + e^{-t}$
Delay transition	$\downarrow 0.69$	
	$(COOL, 17.5)$	
Discrete transition	$\downarrow$ *on*	
	$(HEAT, 17.5)$	$\mathtt{temp}(t) = 30 - 12.5e^{-t}$
Delay transition	$\downarrow 1$	
	$(HEAT, 25.4)$	
Discrete transition	$\downarrow$ *off*	
	$(COOL, 25.4)$	$\mathtt{temp}(t) = 17 - 8.4e^{-t}$
Delay transition	$\downarrow 2$	
	$(COOL, 18.13)$	

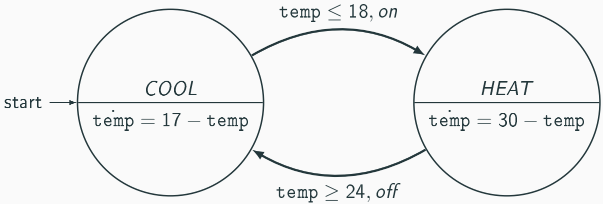
Continuous part - transition guards



Guards are **predicates** over the variables.

- $(COOL) \xrightarrow{\texttt{temp} \leq 18, \mathit{on}} (HEAT)$ says that the value of the temperature must not be greater than 18 for the transition to be taken.
- $(HEAT) \xrightarrow{\texttt{temp} \geq 24, \mathit{off}} (COOL)$ says that the value of the temperature must not be lower than 24 for the transition to be taken.

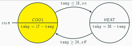
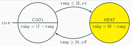
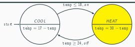
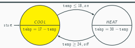
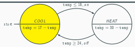
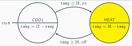
Continuous part - urgent transitions



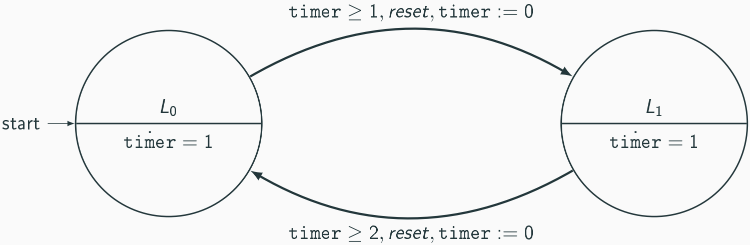
- Convention: from now on, we consider all transitions urgent. That is, transitions are taken as soon as the values of the variables satisfy their guards.
- This way, non-determinism only arises when more transitions are executable at the same time instant.

This choice is because we are going to work in CIF, where all events are urgent.

Example of urgent run

Location	State	Dynamic
	$(COOL, 18)$	$\texttt{temp}(t) = 17 + e^{-t}$
Discrete transition	$\downarrow on$	
	$(HEAT, 18)$	$\texttt{temp}(t) = 30 - 12e^{-t}$
Delay transition	$\downarrow \approx 0.694$	
	$(HEAT, 24)$	
Discrete transition	$\downarrow off$	
	$(COOL, 24)$	$\texttt{temp}(t) = 17 + 7e^{-t}$
Delay transition	$\downarrow \approx 0.55962$	
	$(COOL, 18)$	
Discrete transition	$\downarrow on$	
	$(HEAT, 18)$	$\texttt{temp}(t) = 30 - 12e^{-t}$

Continuous part - transition updates



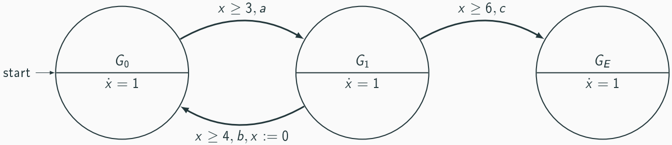
Updates are functions over the variables.

- $L_0 \xrightarrow{\texttt{timer}\geq 1, \mathit{reset}, \texttt{timer}:=0} L_1$ says that the value of the timer must be set to 0 when taking the transition.
- $L_1 \xrightarrow{\texttt{timer}\geq 2, \mathit{reset}, \texttt{timer}:=0} L_0$ says that the value of the timer must be set to 0 when taking the transition.

$$(L_0, 0) \xrightarrow{1} (L_0, 1) \xrightarrow{\mathit{reset}} (L_1, 0) \xrightarrow{2} (L_1, 2) \xrightarrow{\mathit{reset}} (L_0, 0) \xrightarrow{1} \ldots$$

Limitations to keep in mind with urgent transitions

Consider this HA.



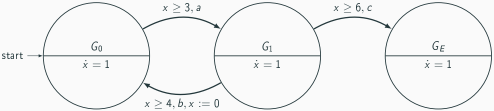
Considering urgency of transitions, we have the trajectory:

$$\begin{tikzcd}
(G_0, 0) \arrow[r, "3"] & (G_0, 3) \arrow[r, "a"] & (G_1, 3) \arrow[r, "1"] & (G_1, 4) \arrow[lll, bend right, "b"']
\end{tikzcd}$$

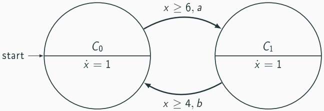
If G_E is an error location, we will never enter G_E by simulating this way.

However...

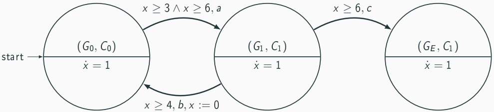
Consider this HA.



Consider this other HA.



The parallel composition is



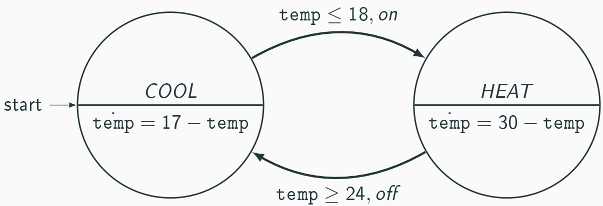
Now, we have a different trajectory:

$$((G_0, C_0), 0) \xrightarrow{6} ((G_0, C_0), 6) \xrightarrow{a} ((G_1, C_1), 6) \xrightarrow{c} ((G_E, C_1), 6)$$

b

Now we can enter G_E.

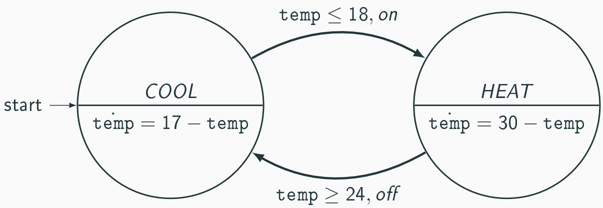
CIF Basics - Hybrid Automata - continuous variables




```
automaton HA:
    cont temp = 18;
    location COOL: initial;
        ...

    location HEAT:
        ...
end

...
```

- Continuous variables are specified by the keywork "`cont`"
- Their initial value is 0 if not specified.

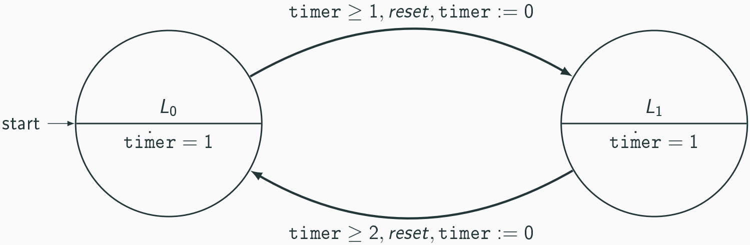
CIF Basics - Hybrid Automata - Dynamics




```
automaton HA:
    cont temp = 18;
    location COOL: initial;
      equation temp' = 17 - temp;
      ...

    location HEAT:
      equation temp' = 30 - temp;
        ...
end

...
```

- Dynamics can be specified in terms of ODEs by the keyword "equation"
- If the dynamic of a continuous variable changes according to the location of the HA, we must specify the form of the ODE in every location.

CIF Basics - Hybrid Automata - Fixed Dynamics



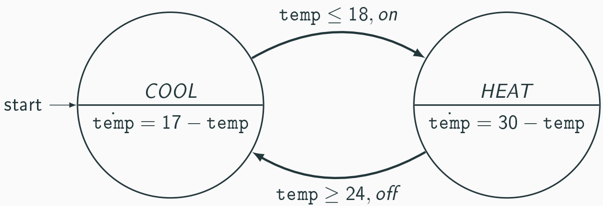

```
automaton HA:
    cont timer = 0 der 1;
    location L0: initial;
        ...

    location L1:
        ...
end
```

```
automaton HA:
    cont timer = 0;
    equation timer' = 1;
    location L0: initial;
        ...

    location L1:
        ...
end
```

```
automaton HA:
    cont timer = 0;
    location L0: initial;
      equation timer' = 1;
        ...

    location L1:
      equation timer' = 1;
        ...
end
```

If the dynamic of a continuous variable never changes it can be specified once at the beginning (first two cases).

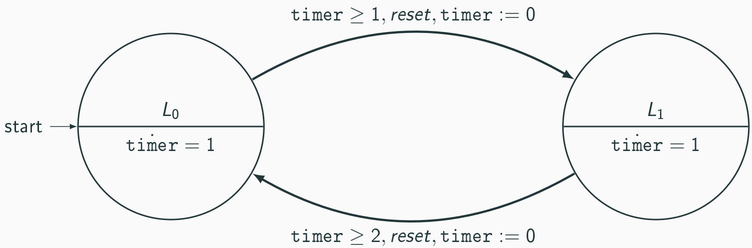
CIF Basics - Hybrid Automata - Transition guards




```
automaton HA:
  event on, off;
  cont temp = 18;
  location COOL: initial;
    equation temp' = 17 - temp;
    edge on when temp <= 18 goto HEAT;

  location HEAT:
    equation temp' = 30 - temp;
    edge off when temp >= 24 goto COOL;
end
```

- Transition guards are specified by the keyword "`when`"

CIF Basics - Hybrid Automata - Transition updates




```
automaton HA:
  event reset;
  cont timer = 0 der 1;
  location L0: initial;
    edge reset when timer >= 1 do timer := 0 goto L1;

  location L1:
    edge reset when timer >= 2 do timer := 0 goto L0;
end
```

- Transition updates are specified by the keyword “do”

A programmable thermostat

1. A programmable thermostat is parametrized on 4 times $0 < t_1 < t_2 < t_3 < t_4 < 24$ (for a 24-hour cycle) and the corresponding setpoint temperatures $temp_1$, $temp_2$, $temp_3$, $temp_4$.
2. Each $temp_i$ is the temperature that we want to reach after the timer hits t_i. That is, at t_i, the system starts heating or cooling the room so that the current temperature of the room reaches $temp_i$.
3. For each $i = 1, \ldots, 4$, if the temperature of the room reaches $temp_i$ before the timer hits $t_{(i+1 \mod 4)}$ the system keeps the temperature stable (until the timer hits $t_{(i+1 \mod 4)}$).

t_i	$\mathtt{temp}_i$
06.00	23°
09.00	20°
18.00	24°
23.00	18°

Assume:

- Initial temperature 18°
- Heating dynamic $\mathtt{temp}'(t) = 30 - \mathtt{temp}(t)$
- Cooling dynamic $\mathtt{temp}'(t) = 17 - \mathtt{temp}(t)$

A programmable thermostat

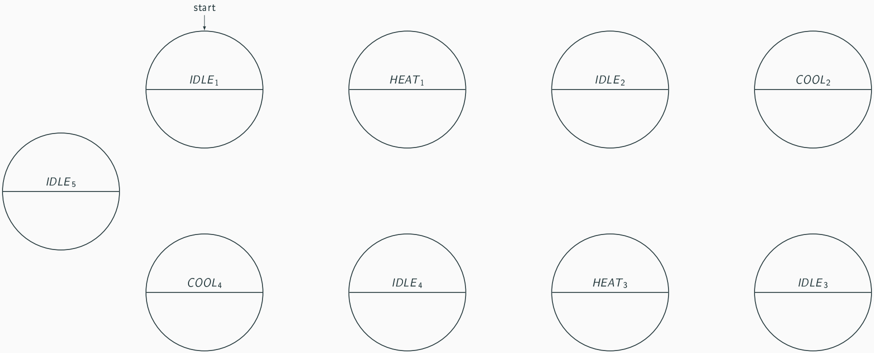
- Locations?

t_i	$\texttt{temp}_i$
06.00	23°
09.00	20°
18.00	24°
23.00	18°

Assume:

- Initial temperature 18°
- Heating dynamic $\texttt{temp}'(t) = 30 - \texttt{temp}(t)$
- Cooling dynamic $\texttt{temp}'(t) = 17 - \texttt{temp}(t)$

A programmable thermostat



A programmable thermostat

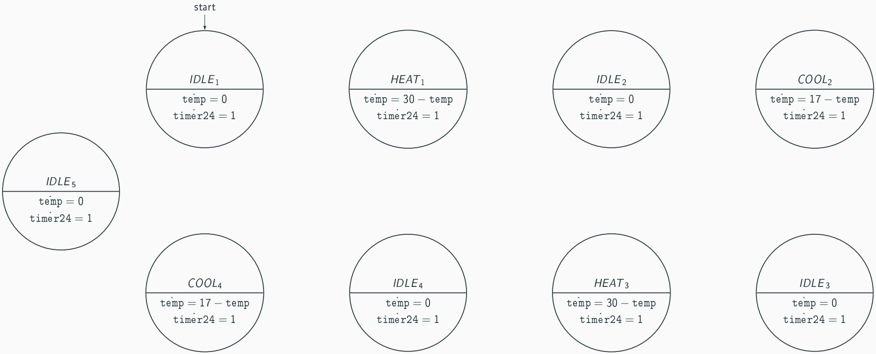
- Continuous variables and dynamics?

t_i	$\mathtt{temp}_i$
06.00	23°
09.00	20°
18.00	24°
23.00	18°

Assume:

- Initial temperature 18°
- Heating dynamic $\mathtt{temp}'(t) = 30 - \mathtt{temp}(t)$
- Cooling dynamic $\mathtt{temp}'(t) = 17 - \mathtt{temp}(t)$

A programmable thermostat



A programmable thermostat

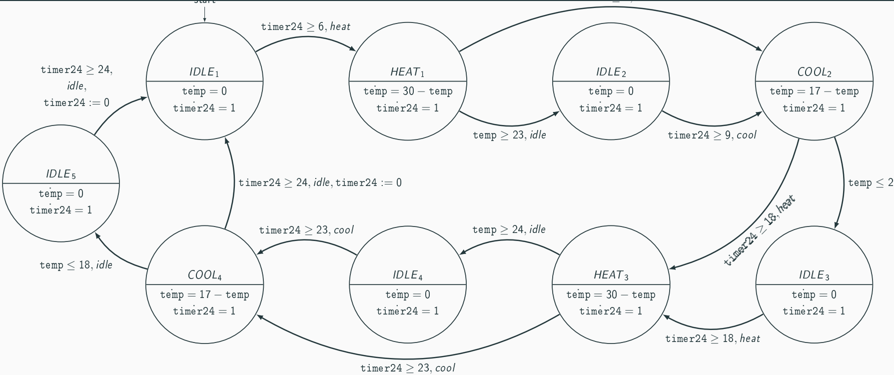
- Transitions?
- Suppose events *heat*, *cool*, *idle* (even though in this example they are not really needed);
- Recall that we might not reach the desired temperatures in time (=need to handle those cases)

t_i	$\texttt{temp}_i$
06.00	23°
09.00	20°
18.00	24°
23.00	18°

Assume:

- Initial temperature 18°
- Heating dynamic $\texttt{temp}'(t) = 30 - \texttt{temp}(t)$
- Cooling dynamic $\texttt{temp}'(t) = 17 - \texttt{temp}(t)$

A programmable thermostat

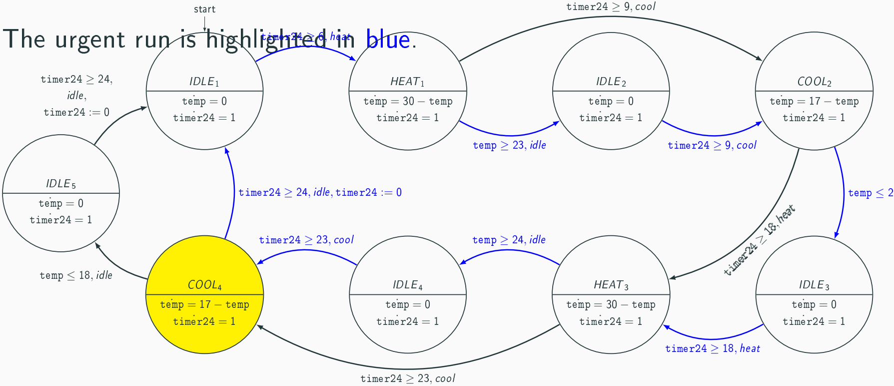


Considering urgency of transitions, does there exist a situation in which the HA cannot reach the desired temperature in time?

Try simulating the HA.

A programmable thermostat

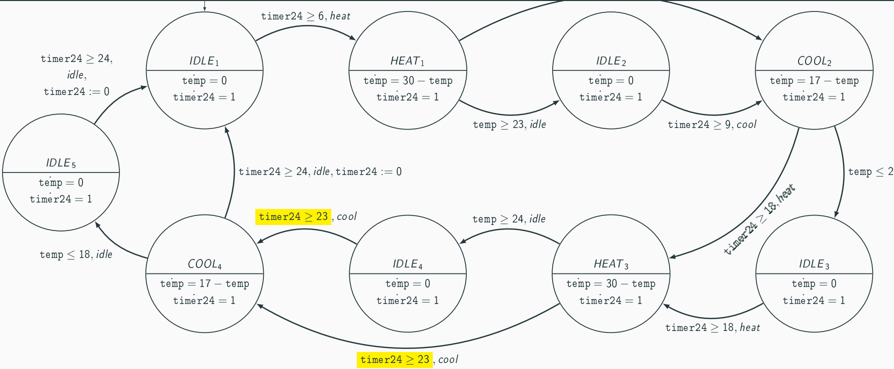
The urgent run is highlighted in blue.



When the HA is in $COOL_4$, 1 hour is a time too short to lower the temperature from 23° to 18°.

Indeed, when $\texttt{timer24} = 24$, we have that $\texttt{temp} \approx 19.57°$

A programmable thermostat

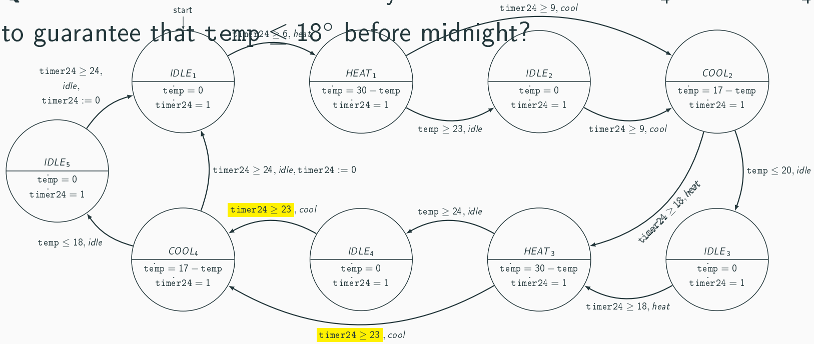


Invariant to take for granted: All variations of temperature are always reached in time before entering $IDLE_4$ (even from the second day on).

Question: At what time of the day should we enter $COOL_4$ from $IDLE_4$ to guarantee that $temp \leq 18°$ before midnight?

A programmable thermostat

Question: At what time of the day should we enter $COOL_4$ from $IDLE_4$ to guarantee that $\texttt{temp} \leq 18°$ before midnight?



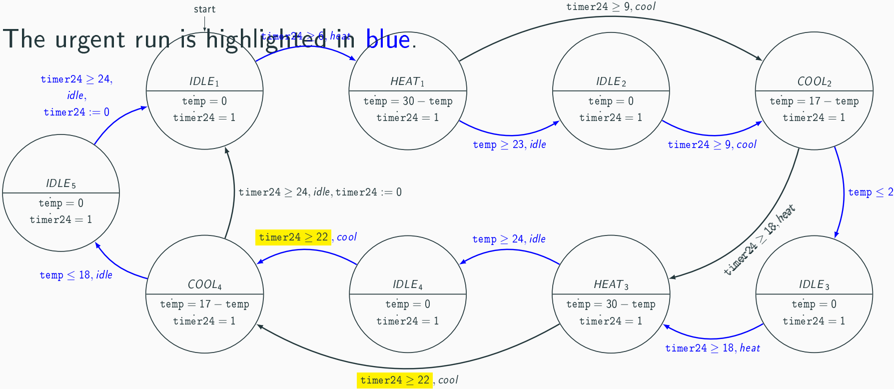
Solve the ODE with respect to entering $COOL_4$.

$$\begin{cases} \texttt{temp}'(t) = 17 - \texttt{temp}(t) \\ \texttt{temp}(0) = 24 \end{cases} \quad \Rightarrow \quad \texttt{temp}(t) = 17 + 7e^{-t}$$

Solve $17 + 7e^{-t} = 18$. It takes $t \approx 1.9459$ hours to lower $\texttt{temp}$ to $18°$.

A programmable thermostat

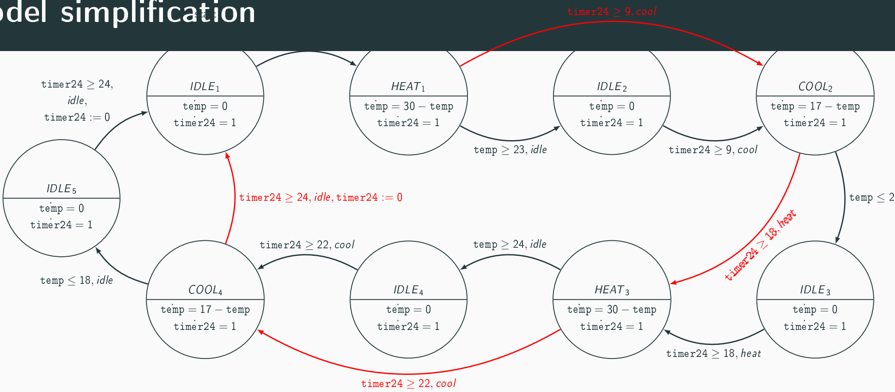
The urgent run is highlighted in blue.



When the HA is in $COOL_4$, it's 2 hours to midnight and we need slightly less of that amount of time to lower the temperature from 24° to 18°.

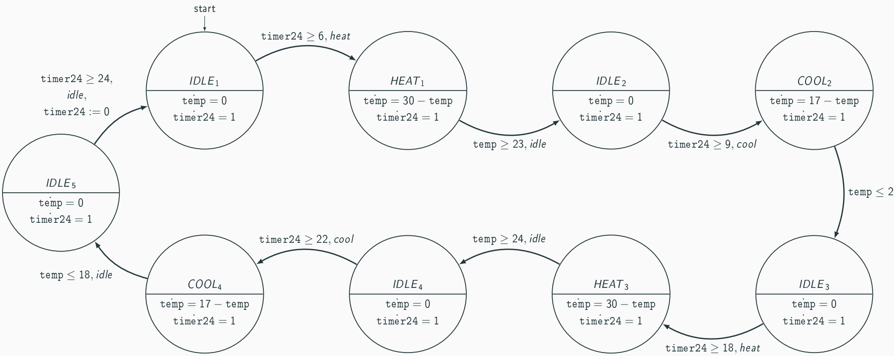
Try setting $t_4 = 22$ and do some simulation.

Model simplification



- Let's remove all transitions to get to the next *HEAT* or *COOL* locations in case the required temperature is not reached in time as we know that this is no longer the case.
- This is not necessary but helps to keep the rest simple.
- Recall that we do so because we assume event urgency.

Model simplification



Already verified invariant: All temperatures are always reached in time before entering all *IDLE* states.

Question: Can we avoid expressing `temp` dynamics in terms of ODEs?

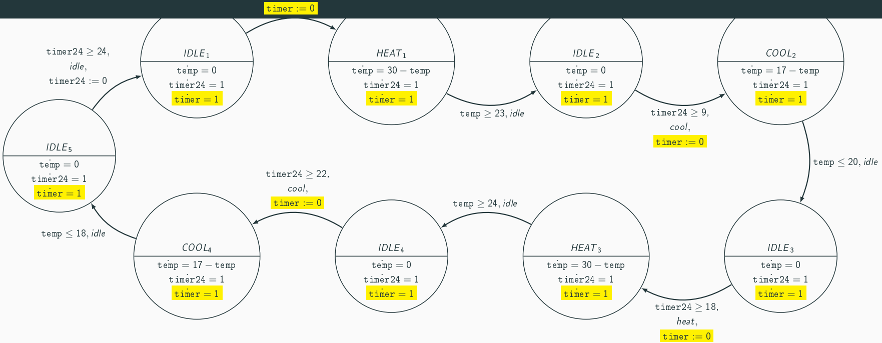
Algebraic variables

- Algebraic variables can be used to give a name to an expression (computation), similar to how constants can be used to give a fixed values to a name.
- The benefits of using an algebraic variable are similar to the benefits of using constants.
- Both can be used to improve readability, and to make it easier to consistently change the model.

```
...
alg type var_name = expression;
...
```

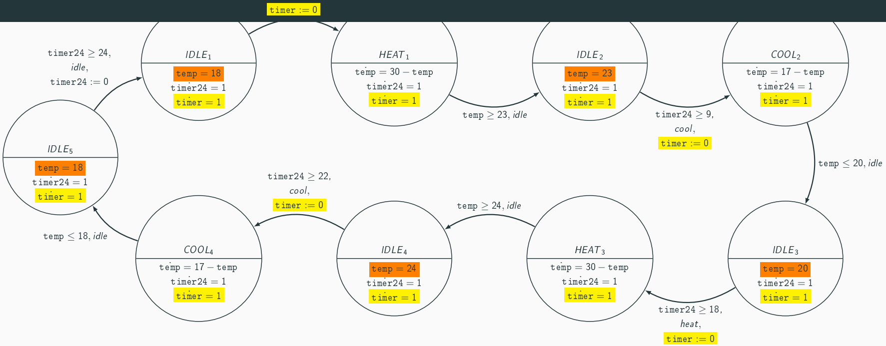
Can we use algebraic variables to hardcode temperature dynamics as solutions of ODEs (considering that we can compute them analytically)?

Hardcoded dynamics - Extra timer to model the evolution



- Add `timer` as a new clock (i.e., continuous variable) which is always reset upon entering all *HEAT* and *COOL* locations.
- `timer` will be used as the parameter for varying the value of the algebraic variables modeling temperature dynamics
- `timer24` will keep working the same.

Hardcoded dynamics - IDLE locations



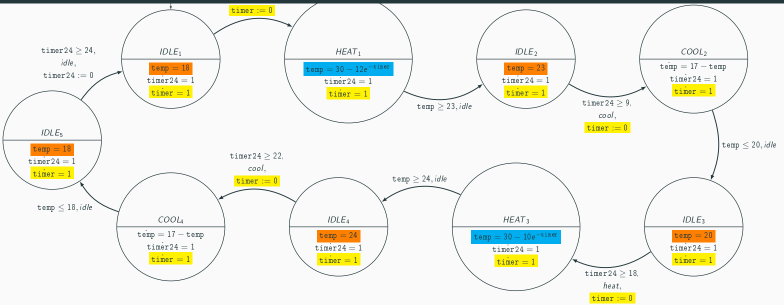
Compute the solution to the temperature ODE with respect to its initial conditions t_i (the temperature value upon entering $IDLE_i$).

$$\begin{cases} \mathtt{temp}'(t) = 0 \\ \mathtt{temp}(0) = t_i \end{cases} \Rightarrow \mathtt{temp}(t) = t_i$$

$IDLE$	1	2	3	4	5
$\mathtt{temp}(t)$	18	23	20	24	18

Replace $\mathtt{temp}'(t) = 0$ with $\mathtt{temp} = t_i$ ($\mathtt{temp}$ is now an algebraic variable).

Hardcoded dynamics - HEAT locations



Compute the solution to the temperature ODE with respect to its initial conditions t_i (the temperature value upon entering $HEAT_i$).

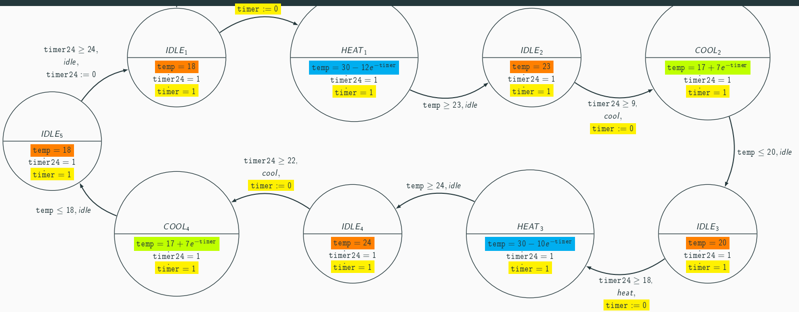
$$\begin{cases} \texttt{temp}'(t) = 30 - \texttt{temp}(t) \\ \texttt{temp}(0) = t_i \end{cases}$$

$HEAT$	1	3
$\texttt{temp}(t)$	$30 - 12e^{-t}$	$30 - 10e^{-t}$

Replace $\texttt{temp}'(t) = 30 - \texttt{temp}(t)$ in $HEAT_1$ with $\texttt{temp} = 30 - 12e^{-\texttt{timer}}$

Replace $\texttt{temp}'(t) = 30 - \texttt{temp}(t)$ in $HEAT_3$ with $\texttt{temp} = 30 - 10e^{-\texttt{timer}}$

Hardcoded dynamics - COOL locations



Compute the solution to the temperature ODE with respect to its initial conditions t_i (the temperature value upon entering $COOL_i$).

$$\begin{cases} \mathtt{temp}'(t) = 17 - \mathtt{temp}(t) \\ \mathtt{temp}(0) = t_i \end{cases}$$

$COOL$	2	4
$\mathtt{temp}(t)$	$17 + 6e^{-t}$	$17 + 7e^{-t}$

Replace $\mathtt{temp}'(t) = 17 - \mathtt{temp}(t)$ in $COOL_2$ with $\mathtt{temp} = 17 + 6e^{-\mathtt{timer}}$

Replace $\mathtt{temp}'(t) = 17 - \mathtt{temp}(t)$ in $COOL_4$ with $\mathtt{temp} = 17 + 7e^{-\mathtt{timer}}$