

Motivating Examples

January 22, 2018

Hannah Catabia

(These lecture notes are **not** proofread and proof-checked by the instructor.)

1 Transition Systems

1.1 What is a transition system?

A **transition system** is an object with six elements, each represented by a variable in the 6-tuple $(S, Act, \rightarrow, I, AP, L)$. Here, we will use the example of a traffic light to help clarify what each component represents.

- S : a set of states. For a traffic light, the states would be the color of the lights, so $S = \{\text{red, yellow, green}\}$. States are conventionally denoted with letters from the Roman alphabet, so let's say $S = \{r, y, g\}$.
- Act : a set of actions. For a traffic light, there would only be one action in $A = \{\text{change_light_color}\}$. Actions are conventionally denoted with letters from the Greek alphabet, like α . Sometimes, an action represents an internal process of the transition system. Since a user cannot communicate with it, it is not important to give it a name. In these cases, we call it a “hidden” action and denote it with the Greek letter τ . The action of changing light color is a great example of an action we could call τ .
- \rightarrow : transition relation. It is a subset or equal to $\{S \times A \times S\}$. For a traffic light:

$$\{S \times A \times S\} = \{(r \xrightarrow{\tau} g), (r \xrightarrow{\tau} y), (g \xrightarrow{\tau} r), (g \xrightarrow{\tau} y), (y \xrightarrow{\tau} r), (y \xrightarrow{\tau} g)\}$$

However, since the sequence of colors of a traffic light repeats the pattern *green to yellow to red*, the transition relation of a traffic light is a subset of $\{S \times A \times S\}$:

$$\rightarrow = \{(g \xrightarrow{\tau} y), (y \xrightarrow{\tau} r), \{(r \xrightarrow{\tau} g)\}$$

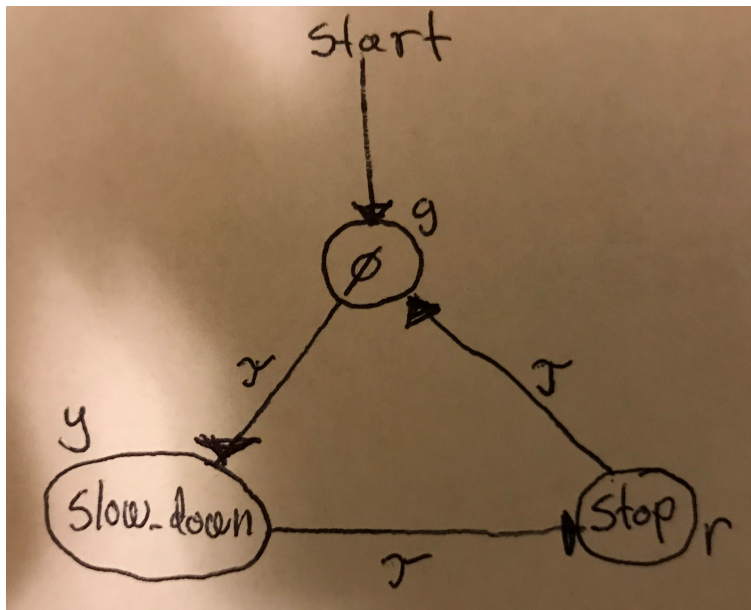
- I : a set of initial states. This will be a subset or equal to all the possible states S . When we first switch on our traffic light, let's say that it will always start out with the color green, so $I = \{g\}$.
- AP : a set of atomic propositions. An atomic proposition is a formula that can be either True or False. It is called “atomic” because it can't be broken down any smaller components (ie: The formula $\neg a \wedge b$ can be evaluated as True or False, but it has two components so it is not atomic. On their own, the formulae $\neg a$ and b are each atomic propositions because they may each evaluate to True or False but cannot be broken down any further.) In a transition system, we could simply make an atomic proposition for each state: red, green, and yellow. Then, when we are in state r , AP would evaluate to red=True, green=False,

yellow=False, etc. However, since we have three states, we only really need two Boolean operators to differentiate between them. Let's name these two atomic propositions *stop* and *slow_down*. When both of them are False, we can assume that means "go". In this case, in state *r*, stop=True and slow_down=False. In state *y*, stop=False and slow_down=True. In state *g*, stop=False and slow_down=False.

- *L*: a labeling function. The argument to this function is a state, and the output of the function is a list of the atomic propositions that are True in that state. In our traffic light example, $L(y) = \{\text{slow_down}\}$, $L(r) = \{\text{stop}\}$, $L(g) = \{\emptyset\}$. Notice that all labels will be in 2^{AP} (the power set of *AP*), because this is the set of all possible True/False combinations of the atomic propositions.

1.2 Graphs of transition systems

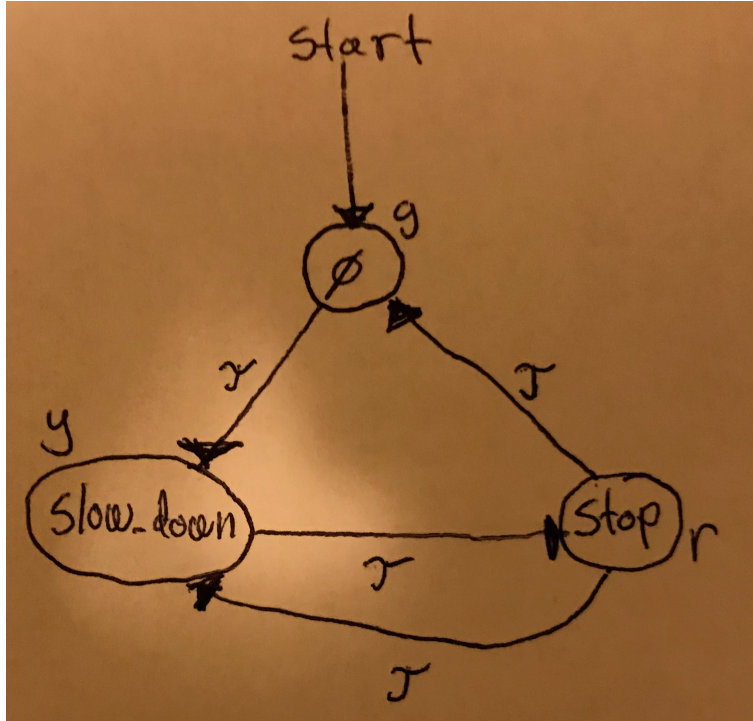
As professor Kfoury mentioned in lecture, transition systems can be pictured as graphs. Here's a graph of the traffic light transition system example discussed above. Each node represents a state (*r*, *g*, or *y*). The edges represent transitions τ . Our initial state *g* has an edge leading into it from *start*. Inside each node, there is a list of the atomic propositions that evaluate to True in that state.



1.3 Transition systems are nondeterministic

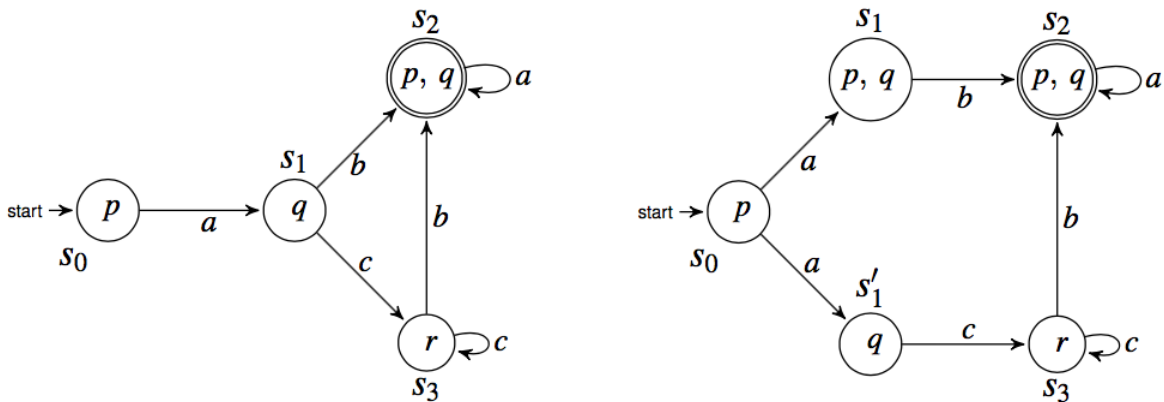
In a deterministic system, if you insert the same input into the system twice, the same series of actions will be executed both times. If the system has an output, the outputs will also match.

Transition systems do not have to be deterministic. It is possible to put the same input in twice, but have two different behaviors and outputs of the system. Unfortunately, the traffic light is a bad example of this, because once you enter the initial state *g*, you cycle from *g* to *y* to *r* and back to *g* in an infinite loop. However, imagine that when the light was red, it could make a non-deterministic decision to become either yellow or green. This is allowed behavior for a transition system, and its graph would look like this:



1.4 “A transition System is more than an automaton”

Let's reexamine the two graphs that Professor Kfoury presented in lecture:



If we say that these two graphs represent finite automata, then they are two representations of the same automaton because both evaluate regular expressions of the form $aba^* + acc^*ba^*$. However, if we say these two graphs represent transition systems, they do *not* represent the same system. This is because they have a different number of states, and the states of the two systems have different atomic propositions evaluating to True within them.