Kripke Machines and Duality for Automata

Monica Dinculescu and Prakash Panangaden
Pictures by Sophia Knight

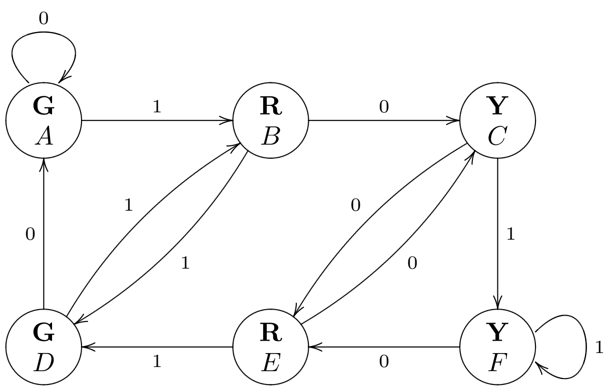
October 13, 2006

Kripke Machines

We can think of a Kripke machine as a black box, with push buttons and lights: it accepts an input and responds with an output. It is like a DFA in the sense that it is a finite automaton, however, it is paritally It is formally defined as $\mathcal{K} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, \delta, \gamma)$, where $\mathcal{S}$ is a set of hidden states $\mathcal{A}$ is a set of actions (or inputs), $\mathcal{O}$ is a set of observations (or outputs), and δ and γ are:

$$\begin{aligned} \delta : \mathcal{S} \times \mathcal{A} &\rightarrow S \ and \\ \gamma : \mathcal{S} \times \mathcal{O} &\rightarrow \{True, False\} = 2 \\ \text{or, equivalently} & \\ \gamma : \mathcal{S} &\rightarrow 2^{\mathcal{O}} \end{aligned}$$

Since pure theory is best left to llamas and emus, we will explain the concept of duality on a concrete example, using the finite automaton below, where $\mathcal{S} = \{A...F\}$, $\mathcal{A} = \{0, 1\}$ and $\mathcal{O} = \{G, R, B\}$.



Tests

We will define a ***test*** as a sequence of the form $a_1a_2...a_no$, where $a_1, a_2, ..., a_n \in \mathcal{A}$ and $o \in \mathcal{O}$. Formally a test is defined inductively as:

1. Any element of $\mathcal{O}$ is a test (i.e. Y is a test).
2. If t is a test, and $a \in \mathcal{A}$, then at is a test (i.e. 0Y is a test).

For example, in the above machine 010Y, R, and 110R are tests. Tests are very similar in conception to words in Σ^*. The only distinction is that since the Kripke machine is partially observable, you are not allowed to observe the state in which the word ends; instead, you are given an observation. Much like the words in Σ^*, there is an infinite number of tests for any machine.

We say that a state s **satisfies** a test t (or $s \models t$) if, starting at state s and following all the a actions in the test t, we observe the observation o at the end of the test. Inductively, this means:

1. $s \models t$ if $\gamma(s, t) = true$
2. $s \models at$ if $\delta(s, a) \models t$

Let us start with state A in our example. If we follow the sequence of actions 0010, we end up in state C, which gives an observation of Y. Thus, we can say that A $\models$ 0010Y. Check that D $\models$ 01R, A $\models$ 011G and F $\models$ 1100Y.

Equivalence Relations

Now that we have a relationship between states and tests, the next natural step is to ask the following questions:

1. What happens if two different tests are satisfied by exactly the same states?
2. What happens if two different states satisfy exactly the same tests?

We say that two tests are **equivalent** if they are satisfied by the same states. Specifically, we will define an **equivalence relation** $\sim$ on tests as follows: $t_1 \sim t_2$ if $\forall s \in S,\ s \models t_1 \Leftrightarrow s \models t_2$. Every equivalence class is thus identified by the set of states that satisfies the tests in the class, i.e. $[t] = \{s | s \models t\}$. Take the test 0G in the above example. We notice that only A and D satisfy it. In order to find other tests equivalent to it, we should look at tests that only A and D, and no other states, satisfy. Other such tests are G, 00G, 000G... etc. Thus, we can say that $[G] = [0G] = [00G]$.... Similarly, we find that the states B and E only satisfy the equivalent tests $[R] = [1R] = [101R]$..., and the states C and F satisfy $[Y] = [1Y] = [100Y]$....

As we have said before, an equivalence class is identified by the set of states that satisfy it. Then, our equivalence classes are: $t_1 = \{A, D\}$, $t_2 = \{B, E\}$ and

$t_3 = \{C, F\}$.

In the same way, we can say that two states are **equivalent** if they are satisfied by the same tests. In more alarming notation, we define an equivalence relation $\equiv$ on states: $s_1 \equiv s_2$ if $\forall t(a\ test) s_1 \models t \Leftrightarrow s_2 \models t$.

Since the set of tests for a machine is infinite, the "going through all of them" approach won't do. To avoid chaos, let's denote taking an action a in state s and ending up in state s' by $s \xrightarrow{a} s'$.

Notice that not only $\mathsf{A} \xrightarrow{0} \mathsf{A}$, and $\mathsf{D} \xrightarrow{0} \mathsf{A}$, but $\mathsf{A} \xrightarrow{1} \mathsf{B}$, and $\mathsf{D} \xrightarrow{1} \mathsf{B}$ as well. This means that A and D have the same behaviour, and thus any test satisfied by one must also be satisfied by the other. Thus, $\mathsf{A} \equiv \mathsf{D}$. Following the exact same logic, we can say that $\mathsf{B} \equiv \mathsf{E}$ and $\mathsf{C} \equiv \mathsf{F}$.

Executive summary: What happens if exactly the same set of states satisfy two different tests? The tests are equivalent. What happens if two different states satisfy all the same tests? The states are equivalent.

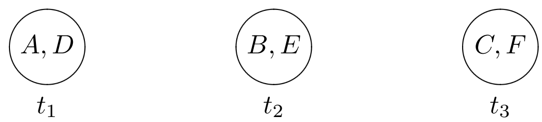
The Dual Construction

Armed with a bag of equivalence relations, we can construct the **dual machine** of that described above. Recall that $\mathcal{K} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, \delta, \gamma)$. The dual of $\mathcal{K}$ is $\mathcal{K}' = (\mathcal{S}', \mathcal{A}, \mathcal{O}', \delta', \gamma')$ where:

$$\begin{aligned} \mathcal{S}' &= T = \{[t]\} \\ \mathcal{O}' &= \mathcal{S} \\ \delta'([t], a] &= [at] \\ \gamma'([t], s) &= [t] \end{aligned}$$

The states of the new machine are the equivalence classes of the original machine, and the new transition function is defined to work with them; the observations are the states of the original machine. What observations do we see? Since the new states of the dual machine are the equivalence classes $[t]$ of the old one, then the observations that we see should be the states that satisfy t; this is just what is given by the new γ function.

Onwards and upwards, we shall now construct the dual machine. In the original machine, we had determined that we had 3 equivalence classes: t_1, t_2 and t_3. Thus, the new machine has those exact three states. In addition, the observations are the states of the original machine that satisfy each of the equivalence classes, so transition arrows aside, the dual looks like:



However, without any transition arrows, this is a pretty useless machine. So, let's see what happens if we apply the new δ function to these states:

$$\delta([t_1], 0) = [0t_1]$$

From our digression on equivalence classes, we know that $[t_1] = [\mathsf{G}]$, therefore $[0t_1] = [0\mathsf{G}]$. To find out what equivalence class this test belongs to, we have to go through the test backwards. In order to see an observation of G, we have to be in either state A or D. The action that produced the Y observation was 0, so let's see what happens if we take a 0 action backwards from A: $\mathsf{A} \overset{0}{\leftarrow} \mathsf{A}$ or $\mathsf{A} \overset{0}{\leftarrow} \mathsf{D}$. Note that there is no state s such that $\mathsf{D} \overset{0}{\leftarrow} \mathsf{s}$. Thus, whatever equivalence class contains $[0\mathsf{G}]$ is identified by the states $\{\mathsf{A}, \mathsf{D}\}$. Why, this is t_1! So we have discovered that there is a 0 transition arrow from t_1 to itself. What happens if a 1 action is taken?

$$\delta([t_1], 1) = [1t_1] = [1G].$$

Again, going backwards in the test, an observation of G is obtained in states A or D. Taking the 1 action that got us there, we see that: $\mathsf{A} \overset{1}{\leftarrow} \mathsf{B}$ and $\mathsf{D} \overset{1}{\leftarrow} \mathsf{E}$. Since $t_2 = \{\mathsf{A}, \mathsf{D}\}$, it means that there is a 1 transition from t_1 to t_2. By applying the exact same reasoning to the other two states, we obtain the dual machine: