# 15-424/15-624 Recitation 2
# Logic and transition relations

## 1. Satisfiability, validity and those sneaky quantifiers!

Let $\phi$ be a formula. Recall that $\phi$ is

- *Satisfiable* if there is a state $\nu$ such that $\nu \models \phi$.
- *Valid* if that for all $\nu$, $\nu \models \phi$
- *Falsifiable*, or *not valid*, if there is a $\nu$ such that $\nu \not\models \phi$
- *Unsatisfiable* if there is no $\nu$ such that $\nu \models \phi$

Notice how these notions critically depend on $\nu$, which states the *initial* values for variables. So, if we look at simple formula $x < y$, then to figure out whether it is valid, satisfiable or unsatisfiable we simply try to find a state - an assignment of variables - that satisfies or falsifies it.

In this case, $x > y$ is satisfiable but not valid. That's because if we consider the state $\nu = \{x \mapsto 2, y \mapsto 1\}$, then $\nu \models x > y$, but we can also find $\omega = \{x \mapsto 1, y \mapsto 2\}$ where $\omega \not\models x > y$.

How do quantifiers affect satisfiability and validity? Let's change our formula to $\forall x. x > y$. The semantics state that

$$\nu \models \forall x. x > y \text{ iff } \nu[x \mapsto d] \models x > y \textit{ for all } d \in \mathbb{R}$$

Really, the quantifier is overwriting whatever value $\nu$ originally assigned to $x$. The same happens for the existential quantifier!

$$\nu \models \exists x. x > y \text{ iff } \nu[x \mapsto d] \models x > y \textit{ for some } d \in \mathbb{R}$$

Is this existential formula $\exists x. x > y$ satisfiable? It is, because we can find an assignment of variables, like $\nu = \{x \mapsto 1, y \mapsto 2\}$ that satisfies it. Even though $\nu \not\models x > y$, the real question, because of the quantifier, is whether we can overwrite $x$ with a new value that is larger than $y$. The answer is yes, since we can choose $x$ to be 3.

In fact, given any $\nu$, to satisfy the existential quantifier all we need to do is assign to $x$ the value or $y + 1$. Since no matter what the original $\nu$ is, we can always find a value for $x$ that satisfies $x > y$, then the $\exists x. x > y$ is actually *valid*!

Strangely, $\forall x. x > y$ turns out to be unsatisfiable because it is impossible to find an initial value for $y$ such that all real numbers are greater...

Now imagine that there aren't any free variables, i.e. that all variables are quantified, like for example:

$$\nu \models \forall x. \forall y. \phi(x, y)$$

Above, $\phi(x, y)$ is any formula that depends on $x$ and $y$. This formula can never be satisfiable but not valid. That's because it doesn't really matter what the initial values $\nu$ assigns to variables, since they will always be overwritten by the quantifiers.

So formulas without free variables are always either valid or unsatisfiable.

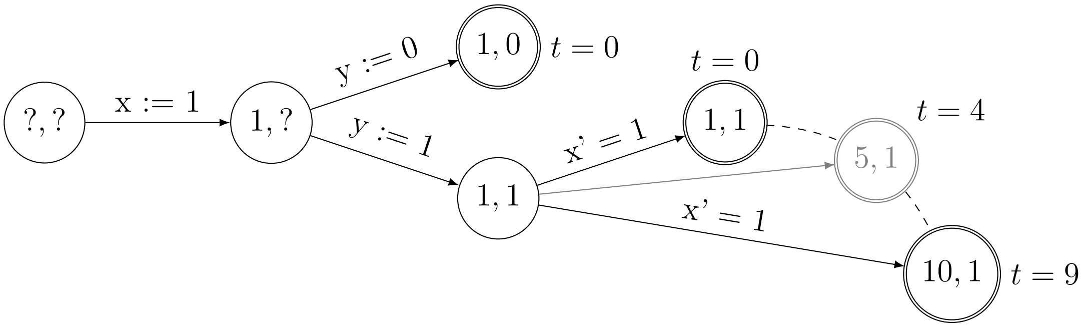
2. **The transition relation**

The transition relation is actually pretty weird, but also super important since it defines exactly what your programs/CPS are supposed to be doing. If you don't understand what your hybrid program is doing, the chances that it's actually a good model for a real CPS are really low.

Consider the following program:

$$\alpha = x := 1; y := 0 \cup (y := 1; x' = 1 \,\&\, x \leq 10)$$

If you think about starting in a specific state, $\nu_0$, it's not too hard to look at the transition relation as if it was a tree.



In the tree above, each node is a state, and the two numbers inside each node are the possible values for $x, y$, respectively. Notice how the tree doesn't all have to be the same depth: that's because certain paths along the program are simply shorter!

Branching represents non-determinism, and you can see that the tree branches out in two different ways:

(a) There is structural non-determinism, which comes from non-deterministic choice $\alpha \cup \beta$ and from non-deterministic repetition $\alpha^*$. The branching comes from deciding which of finitely many choices the program can take. *None of these choices implies the passage of time.*

(b) There is the non-determinism of differential equations. In this case, there aren't finitely many choices, because this non-determinism presents the choice of a *duration* throughout which to evolve according to the differential equation.

This is a *continuous* choice! As long as the differential equation doesn't go out of the domain, it can stop at *any* time. This is the only way in which time passes.

Notice how the leaf nodes, which are double-circled, represent possible end-states of the program. The program can't *end* at any of the single-circled nodes, since there is something else that it needs to execute.

Notice also how double-circled nodes can represent completely different times. The depth of the tree is not necessarily tied to the notion of time. This is simply a program which might take anywhere from 0 to 9 time units to execute, with the world acting accordingly. All of those states are end states! Pretty shifty, huh?

If you were to remove the domain of the differential equation, that transition would basically be able to evolve for *any* duration!

3. **Alternative transition relates**

So we just looked at how, starting from a single state, we can obtain a multitude of end states for a given problem. What if that's the intuition we used to define the transition relation itself?

$$R(\alpha) : S \to 2^S$$

$S$ is, in this case, the set of all states. So in the above definition, $R(\alpha)(\nu)$ is the set of states we can reach using program $\alpha$ from state $\nu$.

Let's try to define it. Assignment is easy, since it's deterministic. Given one state, you get to one state!

$$R(x := \theta)(\nu) = \{\nu[x \mapsto [\![\theta]\!]_\nu]\}$$

The test is pretty similar, in that it's basically either a singleton set or an empty set!

$$R(?H)(\nu) = \{\nu : \nu \models H\}$$

When we have a non-deterministic choice $\alpha \cup \beta$, then the transition relation must include the behaviour of both $\alpha$ and $\beta$.

$$R(\alpha \cup \beta)(\nu) = R(\alpha)(\nu) \cup R(\beta)(\nu)$$

Continuing this crazy non-deterministic trend, we have non-deterministic repetition $\alpha^*$. We will use a technique that is similar to the one we used for the original transition relation - yay, less work! We define $\alpha^0$ as ?true, $\alpha^1 = \alpha$, $\alpha^2 = \alpha; \alpha$, and so forth. Then we just need to consider as possible behaviour the behaviours of $\alpha^n$ for all $n \in \mathbb{N}_{\geq 0}$, because we can choose to repeat any number of times.

$$R(\alpha^*)(\nu) = \bigcup_{n \in \mathbb{N}_{\geq 0}} R(\alpha^n)(\nu)$$

Finally, we have the dreaded differential equations, $x' = f(x) \ \& \ H$! But fear not, brave students, for it is kinda somewhat maybe more of the same! And you already understand “the same”, right?!

We will assume we have a solution $\varphi : \mathbb{R}_{\geq 0} \to S$ to the *initial value problem* we get from considering the initial state $\nu$. Starting from $\nu$, which states will we be able to reach? The idea here is to follow the solution $\varphi$ for as long as possible, until we fail to satisfy $H$. As we evolve, we collect all the states that we’ve been passing through and add them to the transition relation. Since the we allow the differential equation to stop at any time, then any state we pass through could’ve been an end state!

$$R(x' = f(x) \ \& \ H)(\nu) = \{\varphi(t) : \varphi \text{ is a sol. } \wedge \ \varphi(0) = \nu \wedge \forall_{0 \leq r \leq t}.\ \varphi(r) \models H\}$$

So really, as long as we are along the $\varphi$ path, we add our current location to the set of end states as long as we’ve been within the domain $H$ since the start!