Duality in Transition Systems

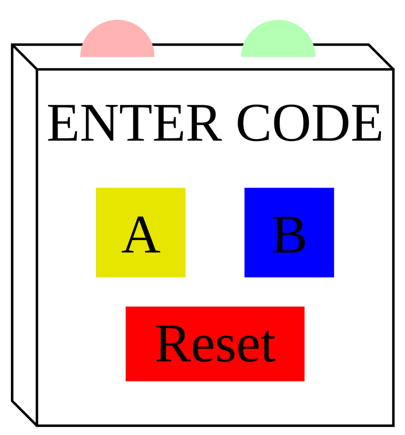
Christopher Hundt / Supervised by Prakash Panangaden and Doina Precup / christopher.hundt@mail.mcgill.ca

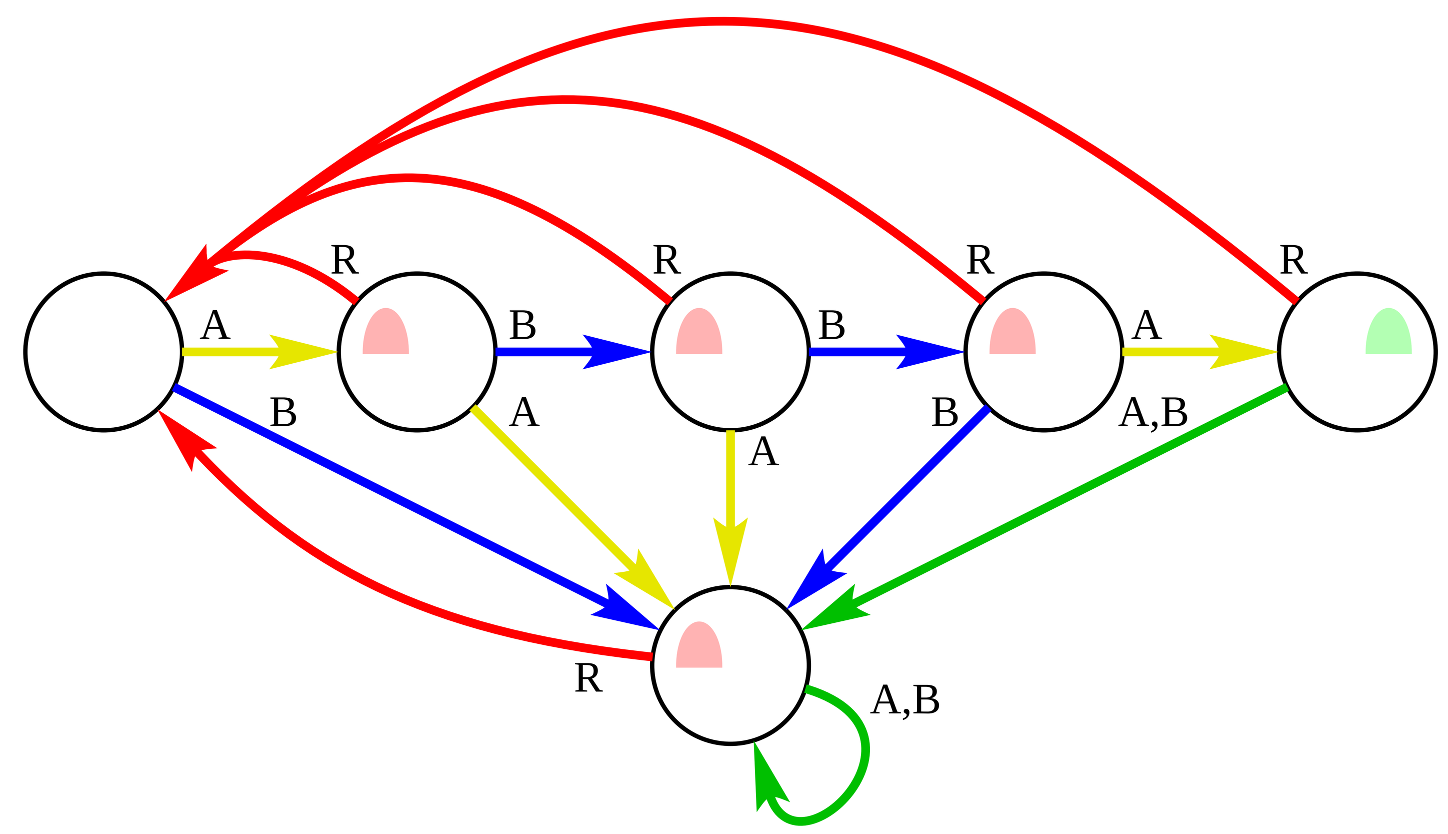
Transition Systems

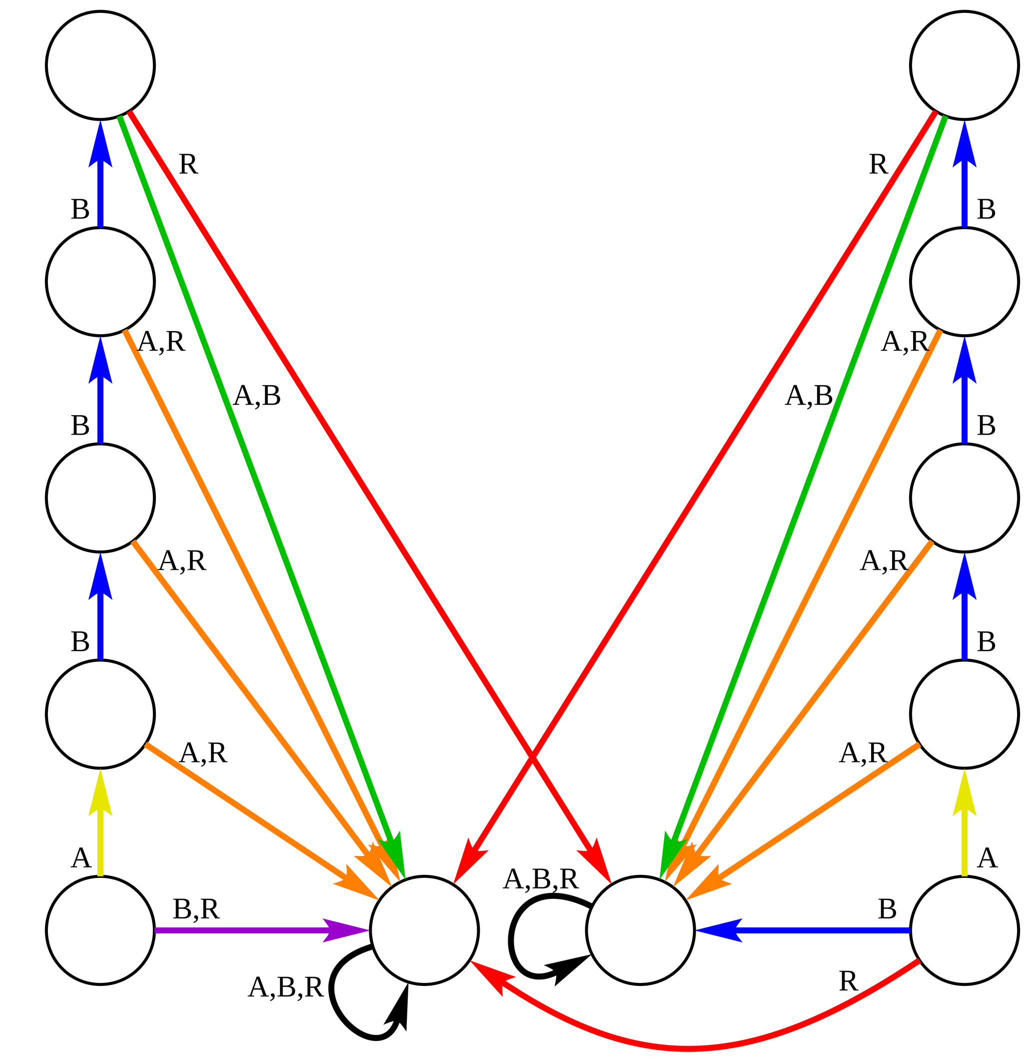
The lock below is designed to let someone in if they enter the correct code. The lock will open if the code ''A-B-B-A'' is entered. At any point, someone can press the Reset button and start over. A red light will go on once any buttons are pressed until either the reset button is pressed or the correct code is entered. A green light indicates that the correct code has been entered.

On the left is an automaton representation of this system. Each circle represents a *state*, and each of the three *actions* (A, B, or Reset) will cause a *transition* from one state to another. The arrows show what transition will occur at a given state with a given action. Each state also has some observations, which in this case are the lights.

On the right is another, apparently different, automaton. But in this poster it is shown that the left and right automatons are *duals* of one another, and that we can pass easily from one to the other.







Introduction

Transition systems are everywhere, and much work in computer science has been devoted to understanding them, working with them, and learning them from data. Most are represented as various types of automata. We'll show a novel concept of duality in automata, which can lead to greater understanding and easier learning of automata.

Definitions

A **non-deterministic automaton** is a 5-tuple $M = (Q, B, P, \delta : Q \times B \to 2^Q, \gamma : Q \to 2^P)$.

- Q is the set of states
- B is the set of actions
- P is the set of observations
- δ is the transition function
- γ is the observation function.

As a shortcut, we define $\delta(S,a) \stackrel{\text{def}}{=} \bigcup_{s\in S} \delta(s,a)$ and $\gamma(S) \stackrel{\text{def}}{=} \bigcup_{s\in S}\gamma(s)$ for any subset $S \subseteq Q$. We also define $\delta(s, a_1, \ldots, a_k)$ as the set of states which may be reached after the sequence $a_1 \cdots a_k$ of actions, when starting in state s, and likewise for a set S of states.

We now define **formulas** on automata, which will serve as a formal way to ask questions about the automaton's behavior, by this grammar:

$$\varphi ::== p \in P \mid (a)\varphi$$

Thus a formula is a sequence of actions followed by an observation: $\varphi = (a_1)(a_2)\cdots(a_k)p$. A formula like this is a way of asking, ''can I observe p after performing actions $a_1, \ldots, a_k$?'' To write this formally, we define a notion of **satisfaction of formulas**. For any $S \subseteq Q$ decide whether S satisfies φ, written $S \models \varphi$, according to the following rules:

$$S \models p \iff \exists s \in S : p \in \gamma(s)$$
$$S \models (a)\varphi \iff \delta(S,a) \models \varphi$$

Thus $S \models (a_1)(a_2)\cdots(a_k)p$ if and only if you can start from some state in S, do the actions $a_1, \ldots, a_k$, and observe p.

To define the dual we use a notion of **equivalence of formulas**. We write that two formulas φ_1 and φ_2 are M-equivalent, written $\varphi_1 \sim_M \varphi_2$, if and only if they are satisfied by exactly the same states. We write the M-equivalence class of φ as $[\![\varphi]\!]_M$.

We now use these equivalence classes define the **dual automaton** $M' = (Q', B, P', \delta', \gamma')$:

- $Q' = \{[\![\varphi]\!]_M\}$
- $P' = Q$
- $\delta'([\![\varphi]\!]_M, b) = [\![(b)\varphi]\!]_M \quad (\forall [\![\varphi]\!]_M \in Q', b \in B)$
- $\gamma'([\![\varphi]\!]_M, p) = \{q \in Q : \{q\} \models \varphi\} \quad (\forall [\![\varphi]\!]_M \in Q')$

Finally, we say that an automaton M_2 **simulates** another automaton M_1 if every M_1-formula is accepted either by both M_1 and M_2 or by neither.

Main result

THEOREM: Let M be an automaton. **The double-dual $M'' = (M')'$ simulates M.**

This means that the double-dual behaves in essentially the same way as the original: any formula on the original is satisfied by the original if and only if it is satisfied on the double-dual.

But does M also simulate M''? The answer is no, but ''almost.'' If you look back to the definition of the dual, you will see that the set of observations at every state (M-equivalence class) is the set of states satisfying the formulas in the class. But this set of states uniquely identifies the equivalence class, since formulas are equivalent if and only if they are satisfied by the same states. This means, in effect, that the states of the dual M' are observable, which means the states of the double-dual are also observable. If the states of the original are not observable then M'' may have more observations than M, and the formulas involving these ''extra'' observations will not translate into formulas on M.

However, if we were to add these new observations to M then it would simulate M''. This is because the double-dual really has the same structure as the original, even if the observations do not match exactly.

Proof

We define a helpful function $Sat : 2^Q \to 2^{\{[\varphi]_M\}}$, defined as follows for any $S \subseteq Q$:

$$Sat(S) \stackrel{\text{def}}{=} \{[\![\varphi]\!]_M : S \models \varphi\} = \bigcup_{s\in S} Sat(s).$$

Now we can show the following:

Let $[\![\vartheta]\!]_{M'}$ be any state in M''. Then $[\![\vartheta]\!]_{M'} = Sat(S)$ for some $S \subseteq Q$. Furthermore, $S = \delta(q, a_1, \ldots, a_k)$ for some state $q \in Q$ and some (possibly zero-length) sequence of actions $a_1, \ldots, a_k \in A$.

The proof is by induction on the length of ϑ. First suppose that ϑ has length 0, meaning that it is simply a proposition in P'. Then it is associated with a state $q \in Q$. So we write $\vartheta = [\![p_q]\!]_{M'}$. Then

$$[\![p_q]\!]_{M'} = \{[\![\varphi]\!]_M : [\![\varphi]\!]_M \models p_q\} = \{[\![\varphi]\!]_M : q \models \varphi\} = Sat(q)$$

and both statements of the lemma hold. Now suppose that $[\![\vartheta]\!]_{M'} = Sat(S)$ for some $S = \delta(q, a_1, \ldots, a_k)$ and we wish to consider the formula $(a)\vartheta$. Then

$$\begin{aligned}
&[\![(a)\vartheta]\!]_{M'} = \{[\![\varphi]\!]_M : \delta'([\![\varphi]\!]_M, a) \models \vartheta\}\\
&= \{[\![\varphi]\!]_M : [\![(a)\varphi]\!]_M \models \vartheta\} = \{[\![\varphi]\!]_M : [\![(a)\varphi]\!]_M \in [\![\vartheta]\!]_{M'}\}\\
&= \{[\![\varphi]\!]_M : [\![(a)\varphi]\!]_M \in Sat(S)\} = \{[\![\varphi]\!]_M : S \models (a)\varphi\}\\
&= Sat(\delta(S,a)),
\end{aligned}$$

and $Q \supseteq \delta(S,a) = \delta(q, a_1, \ldots, a_k, a)$, so both statements of the lemma are satisfied.

We now define simulation formally.

Given two NFAs $M_1 = (Q_1, B, P_1, \delta_1, \gamma_1)$ and $M_2 = (Q_2, B, P_2, \delta_2, \gamma_2)$, we say that M_2 simulates M_1 if there exist functions $f : Q_1 \to Q_2$ and $g : P_1 \to P_2$ such that for any M_1-formula $\varphi = a_1 \ldots a_k p$, the following holds:

$$\begin{aligned}
&\forall q \in Q,\ p \in \gamma_1(\delta_1(q, a_1, \ldots, a_k))\\
&\iff g(p) \in \gamma_2(\delta_2(f(q), a_1, \ldots, a_k)).
\end{aligned}$$

M is simulated by M''.

We use $f(q) = Sat(q)$ and $g(p) = [\![p]\!]_M$, remembering that the states of M' are the observations of M''. Then for any state $q \in Q$ and any sequence $a_1, \ldots, a_k \in A$ we have

$$\begin{aligned}
&g(p) \in \gamma''(\delta''(f(q), a_1, \ldots, a_k))\\
&\iff [\![p]\!]_M \in \gamma''(\delta''(Sat(q), a_1, \ldots, a_k))\\
&\iff [\![p]\!]_M \in \gamma''(\delta''([\![p_q]\!]_{M'}, a_1, \ldots, a_k))\\
&\iff [\![p]\!]_M \in \gamma''([\![(a_k)\cdots(a_1)p_q]\!]_{M'})\\
&\iff [\![p]\!]_M \in [\![(a_k)\cdots(a_1)p_q]\!]_{M'}\\
&\iff [\![p]\!]_M \models (a_k)\cdots(a_1)p_q\\
&\iff [\![(a_1)\cdots(a_k)p]\!]_M \models p_q\\
&\iff q \models (a_1)\cdots(a_k)p\\
&\iff p \in \gamma(\delta(q, a_1, \ldots, a_k)).
\end{aligned}$$

The probabilistic case

Many interesting transition systems used in applications such as artificial intelligence are probabilistic. That is, a given action may not always have the same effect, and you may not always make the same observations at the same states. Does the same kind of duality exist in these systems?

We have translated this notion of duality to probabilistic systems. We keep formulas of the same type, but instead of saying that a state ''satisfies'' a formula, we speak of the probability of the state satisfying the formula. For example, if $\varphi = (a_1)(a_2)\cdots(a_k)p$, we say that the probability that q satisfies φ is the probability that you will observe p after starting in q and taking the actions $a_1, \ldots, a_k$.

Two formulas φ_1 and φ_2 on a probabilistic automaton are equivalent if, for every q, $\Pr(q \models \varphi_1) = \Pr(q \models \varphi_2)$. Then we can construct a dual with the states as equivalence classes of formulas. Just as in the non-deterministic case, we make the states of the original the observations of the dual. We say that the probability that the equivalence class of φ emits q as an observation in M' is the probability that $q \models \varphi$ on M.

We take simulation to mean that any formula φ on M can be translated to a formula φ'' on M'' such that for any state q in M, $\Pr(q \models \varphi) = \Pr(q'' \models \varphi'')$, where q'' is the state on the double-dual associated with q. Then we proved that **M'' simulates M**.

However, unlike in the non-deterministic case, not all questions of interest related to probabilistic transition systems can be phrased using the language of formulas we developed.

Conclusions and future work

This new theory of duality has led to insight into automata theory in general and shows promise in learning: perhaps if a given system is hard to learn a model for, it may be easier to learn a model for its dual.

This is most interesting in the case of probabilistic automata, and this is where future work should lead. We would like to extend our duality more fully to probabilistic automata to make the double dual indistinguishable from the original; we want any question that you could ask about the original to have the same answer on the double dual. This would mean that the dual automaton has really captured all the dynamics of the original automaton, and would provide the most promise for learning based on the dual.

References

- J.A. Brzozowski. ''Canonical regular expressions and minimal state graphs for definite events.'' In *Mathematical Theory of Automata*, Vol. 12, 1962.
- R.L. Rivest and R. E. Schapire. ''Diversity-based inference of finite automata.'' JACM 41, 1994.
- Michael James, Satinder Singh and Matthew Rudary. ''Predictive State Representations: A New Theory for Modeling Dynamical Systems.'' UAI '04.