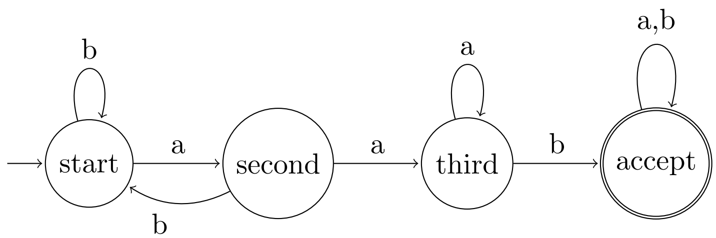
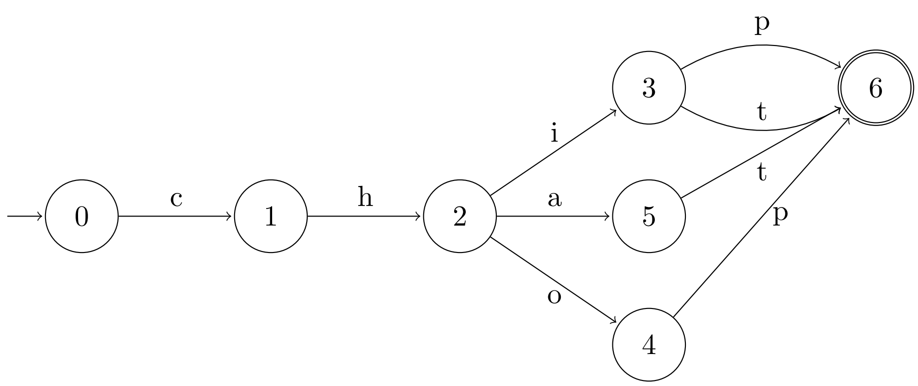
Tuesday 7/29: State Diagrams pt. 1

A **state diagram** is a special type of directed graph where nodes represent states and edges represent actions.

Let's begin with an example: a state diagram that reads in strings made up of a's and b's. It accepts strings that contain the substring "aab". This means it accepts all strings that contain the substring "aab" and rejects strings that do not.

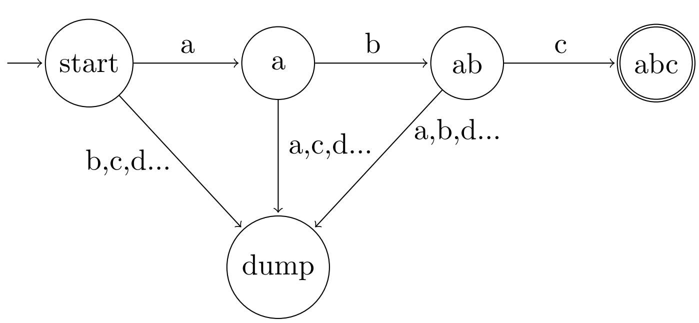


In this case, we have explicitly stated what happens when we see an "incorrect" character. For example, if we see a "b" as the first character, we return to the start state. This is because even if our string begins with a "b", it's still possible for it to contain "aab" as a substring. However, for other problems (especially in cases with larger alphabets) it might be wiser to assume that all "incorrect" characters go to a "dump" state. Here's an example.



This DFA (or, deterministic finite automata) accepts the strings chip, chit, chat, and chop. Since our alphabet includes all letters, it would be difficult to draw transitions to a "reject" state. Instead, we just state that we assume all arrows not pictured go to a dump state.

Here's an example with the dump state pictured. This DFA accepts all strings that begin with "abc".



Now let's talk about what a state diagram is defined by. We have:

- a set of states S (including start and end states)
- a set of actions A
- a transition function $\delta : S \times A \to \mathbb{P}(S)$

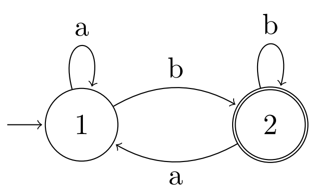
This means that given a state and an action, our state diagram should tell us what state (or set of states) we should go to. Because we are only working with *deterministic* state machines, our transition function actually takes this form: $\delta : S \times A \to S$.

And remember our rules about functions: for every element in the domain, we must have exactly one output in the co-domain. This means that every (state, action) pair should take us to exactly one state[1]. As you design your state diagrams, you should keep this in mind.

In the examples above, the state diagram's transition function was represented by arrows. However, we might also write it out as a table. Take the following example:

- states: $\{1, 2\}$; 1 start and 2 accept
- actions: $\{a, b\}$
- transitions: $\delta((1, a)) = 1, \delta((1, b)) = 2, \delta((2, a)) = 1, \delta((2, b)) = 2$

See how it looks as a state diagram:



[1]This is only true in the deterministic case. Take the non-deterministic case, where our function signature is still $\delta : S \times A \to \mathbb{P}(S)$. Here, a valid output is $\emptyset$, since this is an element of the co-domain. However, since $\emptyset$ is not an element of S (in most cases), this is not a valid deterministic output.