Discussion of Common Problems – 1

Types

Language an Automaton Accepts

Sound of No Hands Clapping

Types

- Types or classes are not just important in programming; they are vital in mathematics and automata theory.
- We've seen distinctions among types:
  - Characters vs. strings.
  - Sets vs. elements.

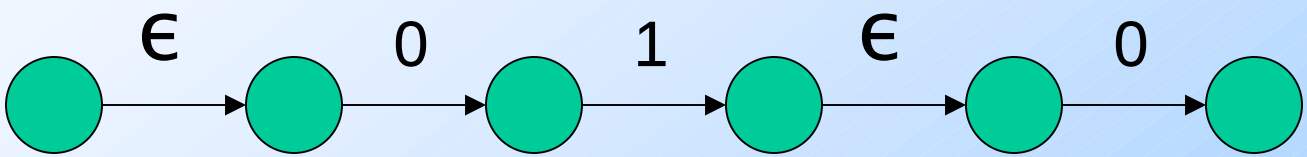
Strings Vs. Characters

- A common and important distinction in programming languages.
  - ”a” is of type String; ’a’ is of type char.
- Especially important: ϵ is of type String.
- Oddity: in the ϵ-NFA, we see arcs labeled by strings (ϵ in particular) and by characters (ordinary inputs like 0).

Strings Vs. Characters – (2)

- It's not a problem; characters can be coerced to strings.
  - Example: in Java "" + '0' = "0".
- Similarly, in an ϵ-NFA, you can mentally coerce the character labels to strings of length 1.
- And for any kind of finite automaton, labels of paths are strings.

Example: Label of a Path



Concatenation of the labels, each treated as a string is 010.

Sets Vs. Elements

- These are always different types.
- Especially, strings are elements, while sets of strings (e.g., languages) are sets.
- ϵ is a string.
- The empty set $\emptyset$ is a set.
- Sets can have “members”; elements never do.

Sets Vs. Elements – (2)

- The empty set is the only set in the world that does not have any members.
- Notice that strings like ϵ or 001 do not have members, but for a different reason:
  - Elements cannot have members.

States of an NFA

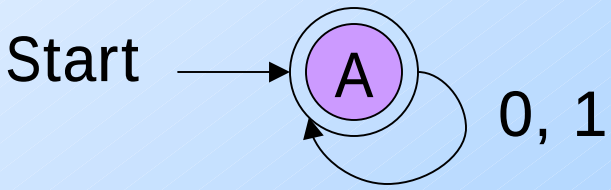
- States are always elements.
- The subset construction seems to construct DFA states that are sets of NFA states.
- Really, the DFA states *correspond* to sets of NFA states, but are elements with names like “Sally” or “q”.
  - Convenient to use something like {p, q} for the name of a DFA state.

Language of an Automaton

- Automata accept strings.
  - Labels of paths from the state state to an accepting state.
- They also accept languages.
  - EXACTLY the set of strings that the automaton accepts.
- Thus, many strings, but ONE language.

Fallacy

- We had a number of forum discussions where people took “automaton A accepts language L” to mean that all the strings of L are accepted by A.
- If that were the case, all languages would be “accepted” by:



Sound of No Hands Clapping

- People sometimes have trouble with the edge cases of general statements.
- Example: We know what the sum of integers is; but what if there are 0 integers?
- Example: we know what it means for a string to have an even number of 0's; but what if that string is empty?

Sum of Zero Integers

- We know what the sum of several integers is, e.g., 4 + 7 + 3.
- What is the sum of no integers?
- The only sensible choice is the identity for the operation +; i.e., 0.

Programming View

- If we wanted to sum integers a[i] for i = 0, 1,..., n-1, we would write, e.g.:

```
sum = 0;
for (i=0; i<n; i++) sum += a[i];
```

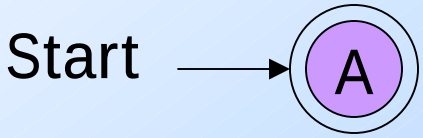
- What is the result if n=0? (Ans.: sum = 0).

Other Operators

- p OR q OR r? OR of zero propositions = false (the identity for OR).
- p AND q AND r? AND of zero propositions = true (the identity for AND).
- p*q*r? product of zero numbers = 1 (the identity for multiplication).
- ”w” ”x” ”y”? concatenation of zero strings = ϵ (the identity for concatenation).

Example: Concatenation of Zero Strings

- Suppose the start state is also accepting.



- The path from state A to A has, as label, the concatenation of zero strings.
- Implies that ϵ is accepted by this DFA.

Is 0 Odd or Even?

- Even, because the remainder of 0 divided by 2 is 0.
  - I.e., 0 = 2*0 + 0.
- The empty string has zero of every symbol.
- So ϵ has an even number of 0's an even number of 1's, and so on.

Automata – Not a Lady Automaton

- And let me add one more point – not of mathematics but of diction.
- "Automaton" is singular, and its plural is irregular: "automata."
- Oddly: the theory is called "automata theory," but other theories tend to be singular.
  - Examples: String theory, quantum theory.