12 April 2019 — Turing machines vs SJAVA.

Announcement: no homework this week.

Last time: defined Turing machine & what it computes.

ASCII ←	alphabet	configuration
Registers ←	state set	initialization
Memory ←	tape set	transitions
Program ←	transition function	

This time: ① Small example of something interesting (?) that a TM can compute.

② Introduce computationally equivalent model, "Simplified Java" aka. SJAVA, that looks like ordinary computer programming.

The substring problem: given two strings $x_1, x_2 \in \{0,1\}^*$ is x_1 a (contiguous) substring of x_2?

```
for i = 0, ..., |x2| - |x1|
    match = true
    for j = 0, ..., |x1| - 1
        if x1[j] ≠ x2[i+j] { match = false }
    endfor
    if (match = true) { return true }
endfor
return false
```

Plan: Represent i by modifying location i on the x_2 input tape.

Represent j by position of read-write head on the x_1 input tape.

Plan: Represent i by modifying location i on the x_2 input tape.
Represent j by position of read-write head on the x_1 input tape.

My original plan:

What I'll do:

Alphabet	$\{\_, 0, 1, \overline{0}, \overline{1}\}$
Alphabet	$\{\_, 0, 1\}$
Tapes	{input 1, input 2, output}
States	$\{s, t, c, r\}$

bookmark for keeping track of "i", the pos'n where we started looking for a match in x_2.

s: start, t: terminate, c: check (analogous to match == true), r: return (analogous to match = false)

Ingredients of JAVA:

Data types

boolean	{true, false}
char	ASCII
int	$\mathbb{Z} \cup \{\perp\}$
String	ASCII* (ε denotes empty string)
array	int*

Functions

Program is a seq of functions.
First one is the "base function"
(like main in a C or Java program)

Statements

Body of a function

- Assignment: `x = expr;` or `x[n] = expr` (pad x with ∅ or _ as necessary)
- if (expr) {...}
- while (expr) {...}
- return expr

Expressions:

- variable
- constant
- binary operation applied to 2 expressions
- built-in function
  e.g. length(x) * elements in a string or array
- accessing element of string or array
  e.g. x[n]
- function application (calling another func in the same prog)

Semantics?

1. Configurations
2. Initialization
3. Transitions.

2 terms: function state & execution state
(local state of one function call) (global state of program)

Def. A **function state** is an ordered pair (c, ϕ) where

c is a program counter (natural number indicating position of next statement to be executed, as an index in the string encoding the entire program).

ϕ is a **dictionary**: a function mapping variable names to values.

= "configuration"

An **execution state** is a finite sequence of $\emptyset$ or more function states called the **stacks**.

$(c_0, \phi_0), (c_1, \phi_1), \ldots, (c_L, \phi_L)$ if program is running

The execution state of a program that terminated is a string representing its output.

The exec state is initialized with initial state of base function. (Program counter points to 1st statement in function body. Function arguments initialized with input values. Every other variable initialized with default value.)

Defaults	
boolean	false
int	0
char	''
string	""
array	[]

Transitions: ① Ordinary statement that doesn't call a function: only (c_h, ϕ_h) changes. (According to ordinary rules of variable assignment & control flow.)

② Statement includes an expression that calls another function: no change in (c_h, ϕ_h). Instead push (c_{h+1}, ϕ_{h+1}) onto stack. (Namely, the initial state of the func that was called.)

③ Return statement. Evaluate expr in the return statement. If that involves calling a function, push onto stack as in ②. But if the expr can be evaluated immediately, pop stack. New top of stack is paused on a line of code containing an expression that called the function that just finished. Substitute function's return value for that expression and execute the statement. If that statement is a "return" statement, pop stack again, recursively.

If stack becomes empty, return value is program output. Put it into the exec state and halt execution.

(Closing curly brace of function body is interpreted as implicit "return default value" command.)