CS 121 Section 3

Code as Data, Data as Code
& Deterministic Finite Automaton

Big Ideas

Big Idea 6: A program is a piece of text, and so it can be fed as input to other programs.

Big Idea 7: Some functions $f:\{0,1\}^n \to \{0,1\}$ cannot be computed by a Boolean circuit using fewer than exponential (in n) gates.

Big Idea 8: $F:\{0,1\}^* \to \{0,1\}^*$ specifies the computational task mapping an input $x \in \{0,1\}^*$ into the output $F(x)$.

Tuple Representation

Let P be a $\mathrm{NAND\text{-}CIRC}$ program on n inputs, m outputs, and s lines, and let t be the number of distinct variables used by P. The list of tuples representation of P is the triple (n, m, L) where L is a list of triples of the form (i, j, k) for $i, j, k \in [t]$.

We assign a number for variable of P as follows:

- For every $i \in [n]$, the variable $\mathrm{X}[i]$ is assigned the number i.
- For every $j \in [m]$, the variable $\mathrm{Y}[j]$ is assigned the number $t - m + j$.
- Every other variable is assigned a number in $\{n, n + 1, \ldots , t - m - 1\}$ in the order in which the variable appears in the program P.

Example 1

How would you represent a AND gate in tuple representation?

Example 2

What does the function does the following tuple represent?

$$(2, 1, ((2, 0, 0), (3, 1, 1), (4, 0, 1), (5, 2, 3), (6, 4, 5)))$$

$\mathrm{EVAL}_{s,n,m}$ function

For every natural number $s, m, n > 0$ we define the function $\mathrm{EVAL}\_\{s, n, m\} : \{0, 1\}^{S(s) + n} \rightarrow \{0, 1\}^m$ as follows. $\mathrm{EVAL}_{s,n,m}(px) = P(x)$ if $p \in \{0,1\}^{S(s)}$ represents a size-s program P with n inputs and m outputs. Otherwise it outputs 0^m (Some junk output).

Circuit that computes $EVAL_{s, n, m}$

```
<EVAL(x, L)>
for i in [n]:
    table = UPDATE(table, i, X[i])
for (i, j, k) in L:
    a = LOOKUP(table, j)
    b = LOOKUP(table, k)
    c = NAND(a, b)
    table = UPDATE(table, i, c)
for j in [m]:
    Y[j] = LOOKUP(table, t - m + j)
```

```
<UPDATE(table, i, b)>
for j in [|table|]:
    a = EQUALS(j, i)
    c = LOOKUP(table, j)
    Y[0] = IF(a, b, c)
```

Deterministic Finite Automaton (DFA)

A deterministic finite automaton (DFA) with C states over $\{0,1\}$ is a pair (T, S) with $T:[C] \times \{0,1\} \rightarrow [C]$ and $S \subseteq [C]$. The finite function T is known as the transition function of the DFA. The set S is known as the set of accepting states.

Let $F:\{0,1\}^* \rightarrow \{0,1\}$ be a Boolean function with the infinite domain $\{0,1\}^*$. We say that (T, S) *computes* a function $\mathrm{F}:\{0,1\}^* \rightarrow \{0,1\}$ if for every $n \in \mathbb{N}$ and $x \in \{0,1\}^n$, if we define $s_0 = 0$ and $s_{i+1} = T(s_i, x_i)$ for every $i \in [n]$, then:

$$s_n \in S \Leftrightarrow F(x) = 1$$

Example

Consider a DFA with the set of states $\{0,1,2\}$, the set of accepting states $\{0\}$, and the transition function shown below. Run this DFA on the string 10101011101. What is the result? What function does this DFA compute?

State	Input Bit	Resulting State
0	0	0
0	1	1
1	0	1
1	1	2
2	0	2
2	1	0

Practice Problems

Practice Problem 1

I. Write a tuple representation for a program that computes the following functions: NAND, OR, XOR, ONE. (If you like building circuits check out http://nandgame.com/)

II. What common boolean circuits do the following tuple representations of a NAND-CIRC program correspond to.

A. (1, 1, ((1, 0, 0)))

B. (1, 1, ((1, 0, 0), (2, 1, 0)))

C. (3, 1, ((3, 2, 2), (4, 1, 1), (5, 3, 4), (6, 2, 1), (7, 6, 6), (8, 0, 0), (9, 7, 8), (10, 5, 0), (11, 9, 10)))

Practice Problem 2

I. For every $k \in \mathbb{N}$, show that there is an $O(k)$ line $\mathrm{NAND\text{-}CIRC}$ program that computes the function $\mathrm{EQUALS}_k : \{0,1\}^{2k} \rightarrow \{0,1\}$ where $\mathrm{EQUALS}_k(x, x') = 1$ if and only if $x = x'$.

II. For every $k \in \mathbb{N}$ and $x' \in \{0,1\}^{\mathrm{k}}$, show that there is an $O(k)$ line $\mathrm{NAND\text{-}CIRC}$ program that computes the function $\mathrm{EQUALS}_{x'} : \{0,1\}^{k} \rightarrow \{0,1\}$ that on input $\mathrm{x} \in \{0,1\}^{\mathrm{k}}$ outputs 1 if and only if $x = x'$.

Practice Problem 3

Design a DFA that computes the following functions.

I. Outputs 1 if and only if the input length is divisible by 3.
II. Outputs 1 if and only if the input starts and ends with 01.