CS 5000: Theory of Computability
Assignment 1

Vladimir Kulyukin
Department of Computer Science
Utah State University

July 8, 2017

Learning Objectives

1. Deterministic Finite State Machines
2. Sussman's Anomaly
3. Induction

Problem 1 (1 point)

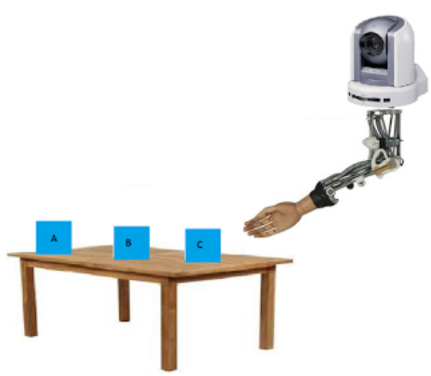

Figure 1: Camera-arm unit.

Recall the camera-arm unit discussed in Lecture 2 that consists of one camera and one arm. For this problem, we will extend the blocks world by adding another block to it. Design a deterministic finite state automaton (DFA) for controlling the robot camera-arm unit in the three-block world shown in Figure 1. The blocks **A**, **B**, and **C** are on top of the table **T**. You may assume that the unit's API has the following functions.

1. **puton(X, Y)** - grab **X**, place **X** on top of **Y**, and release it, where **X** is a block and **Y** is either a block or a table;
2. **clear(X)** - predicate that evaluates to true if the top of **X** is clear, where **X** is a block;
3. **on(X, Y)** - predicate that evaluates to true if **X** is on top of **Y**, where **X** is a block and **Y** is either a block or a table.

Problem 2 (1 point)

The Sussman Anomaly is a famous AI problem in automated planning named after its inventor Gerald Sussman. Dr. Sussman used this problem to illustrate a weakness of linear planning algorithms. Read about the Sussman anomaly at `https://en.wikipedia.org/wiki/Sussman_Anomaly` and design a DFA for the 3-block camera-arm unit that solves it. You do not have to be too formal when specifying your DFA. Careful and clear drawings are sufficient. Also, illustrate how your DFA work on a few test cases.

Problem 3 (2 points)

Imagine a table top divided into 6 square cells, as shown below.

```
------_------------
|     |     |     |
-------_-----------
|     |     |     |
-------------------
```

Five objects (3 cups, 1 tea pot, and 1 milk pitcher) are placed on the table, as shown below. C stands for *cup*, T stands for *tea pot*, and P stands for *pitcher*.

```
-------------------
| C |     | T |
-------------------
| C | C | P |
-------------------
```

The objects on the table can be moved according to the following rules:

1. An object can be moved only into the neighboring free cell.
2. An object cannot be lifted and placed into a cell over any other object.
3. A cell may contain at most 1 object.

Design a finite state machine that allows to swap the tea pot and the pitcher. In other words, the goal state should look as follows:

```
-------------------
| C |     | P |
-------------------
| C | C | T |
-------------------
```

Your finite state machine can be a drawing similar to the one we did in Lecture 2 when solving the 3-animal puzzle. Your states can be the arrangements of the objects in the cells at a particular point in time.

Clearly identify the moves that connect one state to another state. For example, you may want to name the cells as (0, 0), (0, 1), (0, 2), (1, 0), (1, 1), and (1, 2). Then you can have the operator $move(X, (i, j), (k, l))$ that moves the object X from cell (i, j) to cell (k, l). For example, $move(T, (0, 2), (0, 1))$.

Problem 4 (1 point)

Prove the following equalities and inequalities by induction:

1. $\sum_{i=1}^{n} i(i+1) = \frac{n(n+1)(n+2)}{3}$.
2. $\sum_{i=0}^{n} i!i = (n+1)! - 1$.
3. $n! > 2^n, n \geq 4$.

4. Let A be a set and let A^c be the complement of A. Show that, for $n > 1$, $(A_1 \cup A_2 \cup ... \cup A_n)^c = A_1^c \cap A_2^c \cap ... \cap A_n^c$.