## The Big Picture: A Language Hierarchy

1. Consider the following problem: Given a digital circuit C, does C output 1 on all inputs? Describe this problem as a language to be decided.

L = {<C> : C is a digital circuit that outputs 1 on all inputs}. <C> is a string encoding of a circuit C.

1. Using the technique we used in Example 3.8 to describe addition, describe square root as a language recognition problem.

SQUARE-ROOT = {w of the form : <integer1>, <integer2>, where integer2 is the square root of integer1}.

1. Consider the problem of encrypting a password, given an encryption key. Formulate this problem as a language recognition problem.

L = {x; y; z : x is a string, y is the string encoding of an encryption key, and z is the string that results from encrypting x using y}.

1. Consider the optical character recognition (OCR) problem: Given an array of black and white pixels, and a set of characters, determine which character best matches the pixel array. Formulate this problem as a language recognition problem.

L = {<A, C-list, c> : A is an array of pixels, C-list is a list of characters, and c is an element of C-list with the property that A is a closer match to c than it is to any other element of C-list}.

1. Consider the language AnBnCn = {anbncn : n ≥ 0}, discussed in Section 3.3.3. We might consider the following design for a PDA to accept AnBnCn: As each a is read, push two a’s onto the stack. Then pop one a for each b and one a for each c. If the input and the stack come out even, accept. Otherwise reject. Why doesn’t this work?

This PDA will accept all strings in AnBnCn. But it will accept others as well. For example, aabccc.

1. Define a PDA-2 to be a PDA with two stacks (instead of one). Assume that the stacks can be manipulated independently and that the machine accepts iff it is in an accepting state and both stacks are empty when it runs out of input. Describe the operation of a PDA-2 that accepts AnBnCn = {anbncn : n ≥ 0}. (Note: we will see, in Section 17.5.2, that the PDA-2 is equivalent to the Turing machine in the sense that any language that can be accepted by one can be accepted by the other.)

M will have three states. In the first, it has seen only a’s. In the second, it has seen zero or more a’s, followed by one or more b’s. In the third, it has seen zero or more a’s, one or more b’s, and one or more c’s. In state 1, each time it sees an a, it will push it onto both stacks. In state 2, it will pop one a for each b it sees. In state 3, it will pop one a for each c it sees. It will accept if both stacks are empty when it runs out of input.