

PROBLEM SET SIX: FORMAL LANGUAGES

Problems

Problem 1

Suppose A is finite and nonempty. Give an informal but persuasive argument that A^* is countably infinite. [Hint: this means showing there is a bijection from ω to A^* . So what you are asked to do is to describe *informally* a way of listing all the A -strings without repetitions. It is a lot of work to make this technically precise, but you are not asked to do that.]

Problem 2

Suppose A is finite and nonempty. Prove that the set of A -languages is nondenumerable. You can use any of the theorems or corollaries stated in Chapter Five, even if their proofs were not given. Also, you can take the countable infinitude of A^* to have been established in Problem 1.

Problem 3

Given an A -language L , use RT to formally define $\text{kl}(L)$. More specifically, tell how X , x , and F in the statement of RT must be instantiated so that the function h whose existence is guaranteed by RT has the property that $\text{kl}(L)$ is the union of all the languages $h(n)$, where $h(0)$ is x and for each $k \in \omega$, $h(k+1)$ is $F(h(k))$. [Hints: (a) The base clause in the informal definition should tell you what x has to be, and the recursion clause should tell you what F has to be. (b) Another informal, but perhaps more helpful way to define $\text{kl}(L)$ is as the union of all the languages $\text{kl}_n(L)$, where

1. $\text{kl}_0(L) = \text{l}_A$; and
2. $\text{kl}_{k+1}(L) = L \bullet \text{kl}_k(L)$ for all $k \in \omega$.]

Problem 4

In *propositional logic*, we have (1) a formal language PL whose strings are called (PL-)formulas, together with (2) a *semantics*, i.e. a function from PL to some other set whose members, called *propositions*, are thought of as the meanings expressed by the formulas; and (3) a *proof theory*, which is a precise way of formalizing what it means for one formula to *follow from* one or more other formulas. For now we're concerned only with the language

PL itself. We assume given a set Let whose members are usually called *propositional letters*. Informally, the set of formulas is defined as follows:

1. (The length-one string corresponding to) each propositional letter is a formula;
2. (The length-one strings corresponding to) \top and \perp are formulas;
3. if P and Q are formulas, so are $(\sim P)$, $(P \wedge Q)$, $(P \vee Q)$, $(P \rightarrow Q)$, and $(P \leftrightarrow Q)$; and
4. nothing else is a formula.

It's important to be aware that what we are talking about here are formulas—which are strings of ‘symbols’ (propositional letters, parentheses, \top , \perp , \sim , \wedge , \vee , \rightarrow , and \leftrightarrow), and *not* the propositions (whatever *they* are) that the formulas express. For example, in the informal definition, when we write “ $(\sim P)$ ” we mean the string obtained by concatenating together the following symbols in the specified order: (1) the left-paren symbol (; (2) the negation symbol \sim ; (3) the symbols in the string P ; and finally (4) the right paren symbol).

The problem is to define PL formally, using RT. As you should realize by now, this means telling how X , x , and F in the statement of RT should be instantiated. To keep things simple, suppose the only propositional letters are X , Y , and Z . [Hints: (a) clauses (1) and (2) are the bases clauses and clause (3) is the recursion clauses. (b) After you think you have the right definition, check to make sure it makes $(X \wedge Y) \wedge Z$ a formula. If it doesn't, you'll need to revise your definition.]

Problem 5

Given a set A , use RT to define $\text{Reg}(A)$. [Remember this is a set of A -languages. not a set of strings.] More specifically, tell how X , x , and F in RT must be instantiated so that the function h whose existence is guaranteed by RT is such that $\text{Reg}(A)$ is $\bigcup_{n \in \omega} h(n)$. [Hints: (a) as always, use the base clauses in the informal definition to figure out what x should be, and the recursion clauses to figure out what F should be. (b) There is a sense in which this definition resembles the one for PL, but not the one for $\text{Mir}(A)$ or for $\text{kl}(A)$. See hint (b) for Problem 4.]

Problem 6

In *autosegmental-metrical* (AM) theory, the phonological structure of the intonation of an English expression is represented by a string of **tones**.¹ (The question of how such a string is realized as a “tune” (pitch contour) is a matter of phonetics, which doesn’t concern us here.)

In one version of this theory, the set of tones has eleven members:

$$A = \{\%H, H^*, L^*, H^* + L, L^* + H, H + L^*, L + H^*, H^-, L^-, H\%, L\%\}$$

Note that each of these 11 elements is to be thought of as a single member of the tonal “alphabet”, no matter how many symbols it is written with. Of these, the first (%H) is called the **initial boundary tone** (IBT), the next six are called **pitch accents** (PAs)², the next two are called **intermediate-phrase final boundary tones** (IFBTs); and the last two are called **final boundary tones** (FBTs).

Certain sets of tone strings are defined informally as follows:

1. an **intermediate phrase** consists of one or more PAs followed by an IFBT.
2. an **intonation phrase** consists of an optional IBT followed by one or more intermediate phrases, followed by an FBT.
3. an **utterance** consists of one or more intonation phrases.

The problem is to *formally* define each of the following as regular languages:

1. the language IP of intermediate phrases;
2. the language IN of intonation phrases; and
3. the language UT of utterances.

You do not need to prove anything, you do not need to use RT, and you do not need to use PMI. All you need to do is express each of these three languages in terms of (1) the eleven tones; the underscore (that maps

¹The sense of the term *tone* here is “minimal unit of intonational phonology”. This is distinct from the notion of tone in the sense of pitch levels or contours within words that function as part of the word phonology and distinguish words from each other, e.g. Mandarin *ma* [high level] ‘mother’ vs. *ma* [high-falling-to-low] ‘scold’.

²The asterisks in the notation for the pitch accents have nothing to do with the asterisk of formal language theory. Instead, they designate a tone (or part of a tone) that has to be associated with an accented syllable.

each tone to the singleton language whose only string is the length-one string of that tone), the empty language \emptyset , the singleton language $\{\epsilon\}$ whose only member is the null string; and the operations union (\cup) of languages, concatenation (\bullet) of languages, Kleene closure (kl), and positive Kleene closure (kl^+). Note: the fact that this is possible means that these languages are all regular. [Hint: You will not necessarily need to use all the available operations. Do *not* use parentheses for optionality; you will have to express it some other way.]

Problem 7

Let T be a finite set. We say a T -language L is **context-free** if there exists a CFG $\langle T, N, D, P \rangle$ such that L is one of its syntactic categories, i.e. $L = C_A$ for some $A \in N$.

- a. Prove that $L = \text{Mir}(2)$ is context-free by presenting a CFG which has L as one of its syntactic categories.
- b. Same problem, but with $L = \text{PL}$, the language of propositional logic with three propositional letters in Problem 4.

Problem 8

Recall that if $\langle T, N, D, P \rangle$ is a CFG, then, in the informal style of recursive definition, the syntactic categories are defined as follows:

1. (Base Clause) If $A \rightarrow t \in D$, then $t \in C_A$.
2. (Recursion Clause) If $A \rightarrow A_0 \dots A_{n-1} \in P$ and for each $i < n$, $s_i \in C_{A_i}$, then $s_0 \dots s_{n-1} \in C_A$.

Once the syntactic categories C_A are simultaneously recursively defined using the RT, these two clauses become true assertions.

Now using the toy English grammar in Section 5, prove that *Mary believed the cat bit the dog* $\in C_S$.