

CS 172 Spring 2007 — Discussion Handout 1

1. Simpleton machines: DFAs

Design DFAs to recognize the following languages:

- (a) $\{w \mid w \text{ is any string not in } a^*b^*\}$ with $\Sigma = \{a, b\}$.
- (b) $\{w \mid w \text{ has length at least 3 and its third symbol is 0}\}$ with $\Sigma = \{0, 1\}$.
- (c) $\{w \mid w \text{ contains an even number of a's and an odd number of b's and does not contain the substring } ab\}$,
 $\Sigma = \{a, b\}$.
- (d) $B_n = \{a^k \mid n \text{ divides } k\}$ for $\Sigma = \{a\}$.

2. Getting Moody: NFAs

Design NFAs to recognize the following languages:

- (a) The set of all binary strings (of length at least 10) such that at least one of the last 10 characters is a 1.
- (b) The set of all decimal numbers such that the final digit has not appeared before.

3. Once a regular language, always a regular language

In the lecture you saw certain operations like union, intersection, star etc., which when applied to a regular language (or two languages), still give a regular language. Here we define some more operations on a single language. Prove that if A is a regular language, then $Op(A)$ is also a regular language, for each of the operations defined below.

- (a) **Complement:** $A^c = \{w \in \Sigma^* \mid w \notin A\}$.
- (b) **NOPREFIX:** $NOPREFIX(A) = \{w \in A \mid \text{No proper prefix of } w \text{ is in } A\}$.
- (c) **DROP-OUT:** $DROP - OUT(A) = \{xz \mid x, z \in \Sigma^* \text{ and } \exists y \in \Sigma \text{ such that } xyz \in A\}$.

4. Laconic NFAs

Show that every NFA can be converted to another NFA which accepts exactly the same language, but has just one accepting state.