Formal Languages and Computability 5

Ragnar Björn Ingvarsson, rbi3

1. október 2024

1 The DFA depicted below determines whether the remainder after dividing a *binary number* by three is one. Convert it to a corresponding context-free grammar.

We will use the algorithm described in lecure 5.

There are three states so we will get variables R_0, R_1 and R_2. Then we get the rules

$R_0 \to 0R_0 \mid 1R_1$
$R_1 \to 1R_0 \mid 0R_2 \mid \epsilon$
$R_2 \to 1R_2 \mid 0R_1$

Where the start variable is R_0.

2 Convert the following CFG into an equivalent CFG in Chomsky normal form, using the procedure given in Theorem 2.9

We start with

$$A \to BAB \mid B \mid \epsilon$$

$$B \to 00 \mid \epsilon$$

And then we first add a start state S_0 and rule $S_0 \to A$. Then we remove all ϵ-rules by adding a case to all rules which contain the nullable variable on the right hand side where the nullable variable does not occur.

We start with B and get

$$S_0 \to A$$

$$A \to BAB \mid BA \mid AB \mid B \mid \epsilon$$

$$B \to 00$$

And then do A as well:

$$S_0 \to A \mid \epsilon$$

$$A \to BAB \mid BA \mid AB \mid B$$

$$B \to 00$$

We then get rid of unit rules:

$$S_0 \to BAB \mid BA \mid AB \mid 00 \mid \epsilon$$

$$A \to BAB \mid BA \mid AB \mid 00$$
$$B \to 00$$

We then remove the long rules:

$$S_0 \to BA_1 \mid BA \mid AB \mid 00 \mid \epsilon$$
$$A \to BA_1 \mid BA \mid AB \mid 00$$
$$B \to 00$$
$$A_1 \to AB$$

And then at last we remove the excess terminal symbols with

$$S_0 \to BA_1 \mid BA \mid AB \mid U_1U_1 \mid \epsilon$$
$$A \to BA_1 \mid BA \mid AB \mid U_1U_1$$
$$B \to U_1U_1$$
$$A_1 \to AB$$
$$U_1 \to 0$$

And now we are done. As we can see, the CFG is in Chomsky normal form as all rules are of the form $A \to BC$ or $A \to a$ or $S \to \epsilon$.

3

a) We can give the CFG with the start state S:

$$S \to 0BA0$$
$$A \to 0A0 \mid \geq$$
$$B \to 0B \mid \epsilon$$

b) We define the start state to be S again:

$$S \to 0A0$$
$$A \to B0C0 \mid 0A0$$
$$B \to 0B \mid B0 \mid +$$
$$C \to 0C0 \mid \geq$$

c) If $i_A = i_B$ and so forth, then $A \cup B$ is not CF since then it would be impossible to ensure that in both A and B there are the same amount of 0's in one place and not in some other, but if i_A is unrelated to i_B and so on, the language $C = A \cup B$ is indeed CF since we can just add the start state S_0 and rule $S_0 \to S_AS_B$ where S_A is the start state for A and S_B is the start state for B.