Lecture 6 – Chomsky Normal Form, Pumping lemma for context-free languages

NTIN071 Automata and Grammars

Jakub Bulín (KTIML MFF UK)

Spring 2024

** Adapted from the Czech-lecture slides by Marta Vomlelová with gratitude. The translation, some modifications, and all errors are mine.*

Recap of Lecture 5

- Grammars: general, context-sensitive, context-free, right-linear (regular) – Chomsky hierarchy
- The language of a grammar, derivation
- Right-linear grammars correspond to FA (and so do left/linear)
- Linear grammars are stronger
- Context-free grammars: parse tree and its yield
- (un)ambiguous grammars, inherently ambiguous languages

2.6 Chomsky Normal Form

Chomsky normal form

The Chomsky normal form (ChNF) of a context-free grammar:

- all rules of the form $A \to BC$ or $A \to a$ ($A, B, C \in V$, $a \in T$)
- no useless symbols

Theorem

For every context-free language L such that $L \setminus \{\epsilon\} \neq \emptyset$ there exists a grammar in ChNF that generates $L \setminus \{\epsilon\}$.

Applications:

- Test membership in L: the CYK algorithm (Sakai 1962)
- Prove the Pumping lemma for context-free languages

Converting to ChNF

Take any context-free grammar for L and simplify (in this order!):

1. eliminate ϵ-productions $A \to \epsilon$ [here we lose $\epsilon \in L$]
2. eliminate unit productions $A \to B$
3. eliminate useless symbols
   - 3a. unreachable [from the start symbol]
   - 3b. nongenerating [a word over terminals]

Now we have a reduced grammar. To get to ChNF, we further:

4. separate terminals from bodies
5. break up longer bodies

Step 1: Eliminate ϵ-productions

A variable $A \in V$ is nullable if $A \Rightarrow^* \epsilon$. An algorithm to find them:

basis: for every ϵ-production $A \to \epsilon$ mark A as nullable

induct: if $B \to C_1 \dots C_k \in \mathcal{P}$ where all C_i are nullable, B is nullable

To eliminate ϵ-productions: 1. find nullable variables, 2. remove ϵ-productions, 3. process every production $A \to X_1 \dots X_k \in \mathcal{P}$:

- let $J \subseteq \{1, \dots, k\}$ be the positions of all nullable variables
- for every $J' \subseteq J$ create a copy of the production where X_j for $j \in J'$ are deleted, except if $J = \{1, \dots, k\}$ require $J' \neq \emptyset$

Example: $\mathcal{P} = \{S \to AB, A \to aAB \mid \epsilon, B \to ABBA \mid \epsilon\}$

$S \to AB \mid A \mid B$ $A \to aAB \mid aA \mid aB \mid a$
$B \to ABBA \mid ABA \mid ABB \mid BBA \mid AA \mid AB \mid BA \mid BB \mid A \mid B$

Step 2: Eliminate unit productions

Idea: for a unit production $A \to B$ copy rules for B with head A, but unit productions can be composed, we need transitive closure:

Unit pairs $\mathcal{U} \subseteq V \times V$ are defined as follows:

- $(A, B) \in \mathcal{U}$ for every unit production $A \to B \in \mathcal{P}$
- if $(A, B) \in \mathcal{U}$ and $(B, C) \in \mathcal{U}$, then $(A, C) \in \mathcal{U}$

To eliminate unit productions:

1. find all unit pairs $\mathcal{U}$
2. remove all unit productions
3. for every unit pair $(A, B) \in \mathcal{U}$ and production $B \to \beta \in \mathcal{P}$ add the production $A \to \beta$ to $\mathcal{P}$

Step 2: Eliminate unit productions – an example

$E \to T \mid E + T$

$F \to I \mid (E)$

$I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$T \to F \mid T * F$

unit pairs:

$(E, E), (E, F), (E, I), (E, T),$

$(F, F), (F, I),$

$(I, I),$

$(T, F), (T, I), (T, T)$

the result:

$E \to E + T \mid T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$F \to (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$T \to T * F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

Step 3: Eliminate useless symbols

- $X \in V \cup T$ is a useful symbol (in G) if there exists a derivation of the form $S \Rightarrow^* \alpha X \beta \Rightarrow^* w$ for some $w \in T^*$
- X is useless if it is not useful
- X is generating if $X \Rightarrow^* w$ for some $w \in T^*$
- X is reachable if $S \Rightarrow^* \alpha X \beta$ for some $\alpha, \beta \in (V \cup T)^*$

Observe:

- useful $\Leftrightarrow$ generating and reachable
- useless $\Leftrightarrow$ nongenerating or unreachable (we eliminate both)
- all terminals are generating

Step 3: Eliminate useless symbols – the algorithm

1. Find all generating symbols:

   basis: mark all terminals $a \in T$ as generating

   induct: for every production $A \to \beta$ where every symbol in the body β is generating, mark the head A as generating (incl. $A \to \epsilon$)

2. Remove all nongenerating symbols and rules containing them
3. Find all reachable symbols

   basis: mark S as reachable

   induct: for every production $A \to \beta$ where the head A is reachable mark every symbol in the body β as reachable

4. Remove all unreachable symbols and rules containing them

- The order is important! Eliminating unreachable symbols can create new nongenerating symbols, but not vice versa
- **Example:** eliminate nongenerating B, then unreachable A

$$\begin{aligned} S &\to AB \mid a \\ A &\to b \end{aligned} \qquad \begin{aligned} S &\to a \\ A &\to b \end{aligned} \qquad S \to a$$

Step 4: Separate terminals from bodies

TODO

Step 5: Break up longer bodies

TODO