Appunti lezione sulla trasformazione di una CFG G=(V,Σ,R,S) in una equivalente in forma normale d Chomsky.

Si precisa che questi appunti non sostituiscono il libro di testo che dà le regole per la conversione in forma normale di Chomsky in un modo meno formale. Dare le regole in forma algoritmica serve per illustrare in modo approfondito come si possono implementare le idee relative ad ogni passo di trasformazione.

Una grammatica context-free G=(V,Σ,R,S) è in forma norma di Chomsky, CNF, se

1. Le regole di riscrittura sono della forma A → BC, con B,C in V oppure A → a con a in Σ.
2. S non compare nella parte destra di alcuna regola
3. S → ε è l'unica regola "cancellante" eventualmente presente.

Algoritmo per la trasformazione in CNF
input una CFG G=(V,Σ,R,S)
output una CFG G'=(V',Σ,R',S') equivalente a G, in NFC

1. Se Il simbolo iniziale S compare nella parte destra di una regola si aggiunge un nuovo simbolo iniziale S' e la regola S' → S, altrimenti S'= S.
2. eliminazione delle regole cancellanti, cioè del tipo A → ε
3. eliminazione regole unitarie, cioè del tipo A → B con B in V
4. eliminazione variabili inutili, cioè improduttive o non derivabili da S e relative regole. Una variabile si dice produttiva se esiste una parola di terminali derivabili da essa, mentre una variabile si dice derivabile da S se occorre in una parola derivabile da S.

5. eliminazione delle regole del tipo A → $u_1u_2…u_k$, dove k≥3, che vengono sostituite da regole che rispettano la forma A → BC, con B,C in V oppure A → a con a in Σ. Se al passo 2 S è risultato cancellante, si aggiunge la regola S'→ ε per generare la parola vuota.

Punto 2. Algoritmo eliminazione regole cancellanti.
input una CFG G=(V,Σ,R,S)
output una CFG G'=(V,Σ,R',S) equivalente a G, a meno della parola vuota, e priva di regole cancellanti

passo 1 Calcoliamo l'insieme delle variabili che possono cancellarsi in uno o più passi:

$NULL_0 = \{ A \mid A$ è in V e $A \rightarrow \varepsilon$ è in R$\}$
$NULL_{i+1} = NULL_i \cup \{A \mid A$ è in V, $A \rightarrow A_1 \dots A_k$ è in R e ogni $A_j, 1 \le j \le k$, è in $NULL_i\}$

Certamente esiste un i tale che $NULL_{i+1} = NULL_i$, nel caso peggiore se $NULL_i = V$.
Poniamo $NULL = NULL_i$, per tale i.

passo 2. Per ogni regola $A \rightarrow uBv$ dove u,v sono in $(V \cup \Sigma)^*$ e per ogni B in NULL, aggiungiamo la regola $A \rightarrow uv$, tranne nel caso che uv sia la parola vuota

Esempio 1: data la grammatica
$S \rightarrow AB$
$A \rightarrow aAA \mid \varepsilon$
$B \rightarrow bBB \mid \varepsilon$

$NULL_0 = \{ A,B \}$
$NULL_1 = \{ A,B,S \} = V$

Quindi tutti le variabili sono cancellabili, la grammatica equivalente a meno della parola vuota:
$S \rightarrow AB \mid A \mid B$ la prima aggiunta perché B è in NULL, la seconda perché A è in NULL ma non aggiungiamo $S \rightarrow \varepsilon$
$A \rightarrow aAA \mid aA \mid a$ aggiunte perché A è in NULL
$B \rightarrow bBB \mid bB \mid b$ aggiunte perché B è in NULL

Esempio 2: data la grammatica
$S \rightarrow 0A0 \mid 1B1 \mid BB$
$A \rightarrow C$
$B \rightarrow S \mid A$
$C \rightarrow S \mid \varepsilon$

$NULL_0 = \{C \}$
$NULL_1 =\{C\} \cup \{ A\}$
$NULL_2 = \{A,C \} \cup \{ B\}$
$NULL_3 = \{A,C,B\} \cup \{ S\}$

Quindi NULL = V
Tutte le variabili sono cancellabili, la grammatica equivalente a meno della parola vuota:
$S \rightarrow 0A0 \mid 1B1 \mid BB \mid 00 \mid 11 \mid B$
$A \rightarrow C$

$B \rightarrow S \mid A$
$C \rightarrow S$

Algoritmo eliminazione regole unitarie:
input una CFG $G=(V,\Sigma,R,S)$
output una CFG $G'=(V,\Sigma,R',S)$ equivalente a G, priva di regole unitarie

passo 1 Calcoliamo le coppie di variabili che si riscrivono in uno o più passi cioè le coppie (A,B) tali che $A \Rightarrow^* B$, con A e B in V.

$UNIT_0 = \{ (A,A) \mid A \text{ è in } V\}$
$UNIT_{i+1} = UNIT_i \cup \{(A,C) \mid (A,B) \text{ in } UNIT_i \text{ e } B \rightarrow C \text{ è in } R\}$

Certamente esiste un i tale che $UNIT_{i+1} = UNIT_i$, nel caso peggiore se $UNIT_i = V^2$.
Poniamo $UNIT= UNIT_i$, per tale i.

passo 2. Per ogni coppia (A,B) in UNIT e per ogni regola $B \rightarrow u$ dove u è in $(V \cup \Sigma)^*$ aggiungiamo la regola $B \rightarrow u$, a meno che u sia una variabile

Esempio:

$S \rightarrow 0A0 \mid 1B1 \mid BB \mid 00 \mid 11 \mid B$
$A \rightarrow C$
$B \rightarrow S \mid A$
$C \rightarrow S$

$UNIT_0 = \{ (A,A),(B,B),(C,C),(S,S)\}$
$UNIT_1 = UNIT_0 \cup \{(S,B),(A,C),(B,S),(B,A),(C,S)\}$
$UNIT_2 = UNIT_0 \cup \{(S,A),(A,S),(B,C),(C,B)\}$
$UNIT_3 = UNIT_2 \cup \{(S,C),(A,B),(C,A)\}$
$UNIT_4 = UNIT_3$

Quindi $UNIT= UNIT_3$.

La grammatica deve diventare:

$S \rightarrow 0A0 \mid 1B1 \mid BB \mid 00 \mid 11$
Qui dovrei aggiungere le stesse produzioni al posto di B perché (B,S) è in UNIT
$A \rightarrow 0A0 \mid 1B1 \mid BB \mid 00 \mid 11$ perché (A,S) è in UNIT
$B \rightarrow 0A0 \mid 1B1 \mid BB \mid 00 \mid 11$ perché (B,S) è in UNIT

Qui si dovrebbero aggiungere le stesse produzioni al posto di A perché (B,A) è in UNIT
C → 0A0 | 1B1 | BB | 00 | 11 perché (C,S) è in UNIT

Algoritmo eliminazione variabili improduttive e relative regole:
input una CFG G=(V,Σ,R,S)
precondizione L(G) ≠ ∅
output una CFG G'=(V',Σ,R',S) equivalente a G, priva di variabili improduttive e relative regole

passo 1 Calcoliamo le variabili produttive in uno o più passi

$PROD_0$ = { A | A è in V e A → x è in R, dove x è in Σ*}

$PROD_{i+1}$ = $PROD_i$ ∪ {(A | A → $A_1 \ldots A_k$ è in R e ogni A_j, 1≤j≤k, è in $PROD_i$ o è un terminale}

Certamente esiste un i tale che $PROD_{i+1}$ = $PROD_i$, nel caso peggiore se $PROD_i$ = V.

passo 2. Per ogni variabile B che non è in PROD eliminiamo B da V e ogni regola che coinvolge B da R.

Esempio:

S → AB | b
A → aA | a
B → AB

$PROD_0$ = { S,A}
$PROD_1$ = { S,A}

Quindi la grammatica è
S → b
A → aA | a

Algoritmo eliminazione variabili inderivabili e relative regole:
input una CFG G=(V,Σ,R,S)
output una CFG G'=(V',Σ,R',S) equivalente a G, priva di variabili inderivabili e relative regole improduttive

passo 1 Calcoliamo le variabili derivabili in uno o più passi

$DER_0 = \{S\}$
$DER_{i+1} = DER_i \cup \{(A_j \mid A_j$ è in V, $A \rightarrow A_1 \dots A_k$ è in R, A è in DER_i, $1 \le j \le k\}$

Certamente esiste un i tale che $DER_{i+1} = DER_i$, nel caso peggiore se DER_i = V.
Sia DER = DER_i , per tale i.

passo 2. Eliminiamo ogni B non in DER_i da V e ogni regola che lo coinvolge da R.

Esempio:

$S \rightarrow b$
$A \rightarrow aA \mid a$

$DER_0 = \{ S\}$
$DER_1 = \{ S\}$

Quindi la grammatica è
$S \rightarrow b$

5.
1. Trasformazione delle regole in cui nella parte destra compare una o più variabile e uno o più terminali

Introduciamo una nuova variabile, D_a, per ogni terminale a e aggiungiamo la regola $D_a \rightarrow a$.
Per ogni occorrenza di a in una parte destra della regola di lunghezza maggiore o uguale a 2 sostituiamo a con D_a.

2. Eliminazione delle regole la cui parte destra ha lunghezza maggiore o uguale a 3

Se $A \rightarrow A_1 \dots A_k$ è in R, con $k \ge 3$ allora creiamo nuove variabili $B_1, \dots, B_{k-2}$, e sostituiamo la regola per A con le seguenti

$A \rightarrow A_1B_1$
$B_1 \rightarrow A_2B_2$
...
$B_{k-3} \rightarrow A_{k-2}B_{k-2}$
$B_{k-2} \rightarrow A_{k-1}A_k$

Esempio:

$S \rightarrow ASA \mid aB$
$A \rightarrow BbA$
$B \rightarrow b$

Dopo il passo 1
$S \rightarrow ASA \mid D_aB$
$A \rightarrow BD_bA$
$B \rightarrow b$
$D_a \rightarrow a$
$D_b \rightarrow b$

Dopo il passo 2
$S \rightarrow AB_1 \mid D_aB$
$B_1 \rightarrow SA$
$A \rightarrow BB_2$
$B_2 \rightarrow D_bA$
$B \rightarrow b$
$D_a \rightarrow a$
$D_b \rightarrow b$