Turno 1

Esercizio 1.1

Descrivere a parole il linguaggio generato dalla CFG G:

$S \rightarrow 0SA \mid \varepsilon$
$A \rightarrow 1 \mid 11$

Esercizio 1.2

Si costruisca la grammatica equivalente, o anche equivalente a meno della parola vuota, in forma normale di Chomsky, mettendo in evidenza i singoli passi.

Turno 2

Esercizio 2.1

Descrivere a parole il linguaggio generato dalla CFG G:

$S \rightarrow A \mid B$
$A \rightarrow 0A0 \mid 0A1 \mid B \mid 1$
$B \rightarrow 1B1 \mid 1B0 \mid A \mid 1$

Esercizio 2.2

Si costruisca la grammatica equivalente priva di regole unitarie, facendo vedere i singoli passi di eliminazione.

Turno 3

Esercizio 3.1

Descrivere a parole il linguaggio generato dalla CFG G:

$S \rightarrow 1AC \mid DAB \mid \varepsilon$
$A \rightarrow DA \mid 1A \mid \varepsilon$
$B \rightarrow 0D$
$C \rightarrow 1D$
$D \rightarrow 01$

Esercizio 3.2

Si costruisca la grammatica equivalente priva di regole cancellanti, facendo vedere i singoli passi di eliminazione.

Sol. Turno 1

Esercizio 1.1

Descrivere a parole il linguaggio generato dalla CFG G:

$S \to 0SA \mid \varepsilon$
$A \to 1 \mid 11$

Sol.3.1. La prima regola per S è tale da consentire di derivare tanti 0 a sinistra di S di quante A a destra:

$S \Rightarrow^* 0^n S A^n$

eliminando S con la seconda regola per S si ottiene
$S \Rightarrow^* 0^n A^n$

Ora ogni A può essere riscritta o con un solo 1 o con due 1, quindi la parola più corta generata sarà ottenuta riscrivendo A sempre con un solo 1, ed è quindi 0^n1^n, la più lunga si ottiene riscrivendo A sempre con due 1, ed è 0^n1^{2n}, sono possibili tutti i casi intermedi, in cui una parte delle A è rimpiazzata da un 1 e altre da due 1, e sono le parole 0^n1^m, con $n \leq m \leq 2n$. Inoltre si genera la parola vuota.
Concludendo L(G) è l'insieme delle parole in cui gli 0 sono seguiti da 1, e il numero degli 1 è compreso tra il numero degli 0 e il doppio del numero degli 0, oltre alla parola vuota. Formalmente $L(G)=\{ 0^n1^m, \text{ con } 0 \leq n \leq m \leq 2n\}$

Esercizio 3.2

Si costruisca la grammatica equivalente in forma normale di Chomsky, mettendo in evidenza i singoli passi.

Per prima cosa si deve aggiungere una nuova variabile iniziale perché nella CNF non si vogliono occorrenze del simbolo iniziale nella parte destra di una regola:

G':
$S' \to S$
$S \to 0SA \mid \varepsilon$
$A \to 1 \mid 11$

Poi si devono eliminare le regole cancellanti, in questo caso abbiamo:

$NULL_0 = \{S\}$
$NULL_1 = \{S'\} \cup NULL_0$
$NULL_2 = NULL_1$

Quindi dobbiamo aggiungere le regole nella cui parte destra si cancella S o

S', a meno che risulti una regola cancellante:

$S' \rightarrow S$
$S \rightarrow 0SA \mid 0A$
$A \rightarrow 1 \mid 11$

Come secondo passo si devono eliminare le regole unitarie, in questo caso la regola è una sola $S' \rightarrow S$ e l'aggiunta delle regole con S' come parte sinistra e quelle di S come parte destra produce la seguente grammatica:

$S' \rightarrow 0SA \mid 0A$
$S \rightarrow 0SA \mid 0A$
$A \rightarrow 1 \mid 11$

Ora bisogna eliminare le variabili improduttive o inderivabili, e le relative regole. Il risultato per le prime si ottiene considerando le variabili produttive, notando che la grammatica genera almeno la parola 01 e quindi L(G') è non vuoto.

$PROD_0 = \{A\}$
$PROD_1 = \{S\} \cup PROD_0$

$PROD_2 = \{S'\} \cup PROD_1$
Poiché $PROD_2$ è l'insieme di tutte le variabili della grammatica il procedimento termina e si può concludere che non ci sono variabili improduttive.

Il risultato per le seconde si ottiene considerando le variabili derivabili.
$DER_0 = \{S'\}$
$PROD_1 = \{S,A\} \cup PROD_0$
Poiché $PROD_1$ è l'insieme di tutte le variabili della grammatica il procedimento termina e si può concludere che non ci sono variabili inderivabili.

A questo punto si fa in modo che le parti destre delle regole siano formate da sole variabili a meno che la parte destra sia un singolo terminale:

$S' \rightarrow BSA \mid BA$
$S \rightarrow BSA \mid BA$
$A \rightarrow 1 \mid CC$
$B \rightarrow 0$
$C \rightarrow 1$

Ora si riduce la lunghezza delle parti destre delle regole in modo che sia sempre pari a due, introducendo opportunamente nuove variabili:

S’ → BD | BA
S → BF | BA
A → 1 | CC
B → 0
C → 1
D → SA
F → SA

E’ evidente che la grammatica ottenuto può essere semplificata, ma scopo dell’esercizio era illustrare i passi dell’algoritmo di trasformazione CNF.

Sol. Turno 2

Esercizio 2.1

Descrivere a parole il linguaggio generato dalla CFG G:

S → A | B
A → 0A0 | 0A1 | B |1
B → 1B1 | 1B0 | A |1

Sol. 1.1 Usando la prima regola per A si ottengono le parole della forma $0^n A O^n$, con n≥1, poi A potrebbe essere invece sostituita usando la seconda regola portando a parole del tipo $0^n\ 0^m A 1^m O^n$, con n,m≥1, ma si potrebbe invertire anche l’ordine di applicazione delle regole portando a parole del tipo $0^n\ 0^m A 0^m 1^n$, con n,m≥1, e si potrebbero applicare in alternanza più volte. Se poi A va in B si possono ottenere parole del tipo $0^n\ 0^m 1^p B 1^p 0^m 1^n$, o $0^n\ 0^m 1^p B 0^p 0^m 1^n$ con n,mp≥1. Visto che usando le regole di A posso scrivere in corrispondenza di uno 0 uno 0 o un 1 e usando le regole per B in corrispondenza di un 1 analogamente posso scrivere uno 0 o un 1, concludo che l’unico legame tra la parola derivata a sinistra dell’occorrenza di A o B, dopo un certo numero di applicazioni delle regole per A e per B mescolate, è la lunghezza di queste parole, che infatti è la stessa. In sintesi da S posso generare qualsiasi parola del tipo xAy o xBy, con x,y in {0,1}* e |x|=|y|. Poi A e B si possono riscrivere solo con 1, quindi il linguaggio generato da G è quello delle parole di lunghezza dispari con un 1 al centro.

Esercizio 2.2

Si costruisca la grammatica equivalente priva di regole unitarie, facendo vedere i singoli passi di eliminazione.

Sol.Applicando l’algoritmo dobbiamo innanzi tutto individuare le variabili che, anche in più passi, si riscrivono con una variabile.
$UNIT_0$= {(A,A),(B,B),(S,S)}

$UNIT_1 = UNIT_0 \cup \{(A,B),(B,A),(S,A),(S,B)\}$

$UNIT_2 = UNIT_1$

Ora sostituiamo per ogni coppia in $UNIT_2$ la parte destra della regola unitaria con a sinistra il primo membro della coppia e con parte destra la parte destra non unitaria di una regola con parte sinistra il secondo membro della coppia:

per (S,A) abbiamo

$S \rightarrow 0A0 \mid 0A1 \mid 1 \mid B$
$A \rightarrow 0A0 \mid 0A1 \mid B \mid 1$
$B \rightarrow 1B1 \mid 1B0 \mid A \mid 1$

per (S,B) abbiamo

$S \rightarrow 0A0 \mid 0A1 \mid 1 \mid 1B1 \mid 1B0$
$A \rightarrow 0A0 \mid 0A1 \mid B \mid 1$
$B \rightarrow 1B1 \mid 1B0 \mid A \mid 1$

per (A,B) abbiamo

$S \rightarrow 0A0 \mid 0A1 \mid 1 \mid 1B1 \mid 1B0$
$A \rightarrow 0A0 \mid 0A1 \mid 1 \mid 1B1 \mid 1B0 \mid$
$B \rightarrow 1B1 \mid 1B0 \mid A \mid 1$

per (B,A) abbiamo

$S \rightarrow 0A0 \mid 0A1 \mid 1 \mid 1B1 \mid 1B0$
$A \rightarrow 0A0 \mid 0A1 \mid 1 \mid 1B1 \mid 1B0 \mid$
$B \rightarrow 1B1 \mid 1B0 \mid 0A0 \mid 0A1 \mid 1$

La grammatica ottenuta non ha regole unitarie ed è equivalente alla grammatica data, poiché ogni variabile ha le stesse regole nella parte destra possiamo semplificarla riducendo le variabili a una sola:

$S \rightarrow 0S0 \mid 0S1 \mid 1S1 \mid 1S0 \mid 1 \mid$

In questa versione appare più evidente che le parole derivate a sinistra e a destra di S sono qualunque, ma della stessa lunghezza.

Turno 3

Esercizio 3.1

Descrivere a parole il linguaggio generato dalla CFG G:

$S \rightarrow 1AC \mid DAB \mid \varepsilon$
$A \rightarrow DA \mid 1A \mid \varepsilon$

$B \rightarrow 0D$
$C \rightarrow 1D$
$D \rightarrow 01$

Sol 3.1.
Dobbiamo partire dalle derivazioni più semplici:
Sia G_D, la grammatica in cui consideriamo solo le regole per D:
$D \rightarrow 01$ allora $L(G_D)$ è il linguaggio {01}
Sia G_C, la grammatica in cui consideriamo solo le regole per C, sostituendo D con ciò che genera:
$C \rightarrow 101$ allora $L(G_C)$ è il linguaggio {101}
Sia G_B, la grammatica in cui consideriamo solo le regole per B, sostituendo D con ciò che genera:
$B \rightarrow 001$
allora $L(G_B)$ è il linguaggio {001}
Sia G_A, la grammatica in cui consideriamo solo le regole per A, sostituendo D con ciò che genera:
$A \rightarrow 1A \mid 01A \mid \varepsilon$
allora $L(G_A)$ è il linguaggio $\{01,1\}^*$, cioè l'insieme delle stringhe binarie in cui ogni occorrenza di 0 è seguita immediatamente da almeno un'occorrenza di 1. Questo linguaggio è regolare e può essere descritto quindi da un'espressione regolare $(01+1)^* = ((01)^*1^*)^*$.
Infine sostituiamo, nelle regole per S, le variabili che generano linguaggi finiti, ottenendo:
$S \rightarrow 1A101$ (rimpiazzando C)
$S \rightarrow 01A001$ (rimpiazzando D e B)
otteniamo:
$S \rightarrow \varepsilon \mid 1A101 \mid 01A001$

Concludendo, il linguaggio generato da S, oltre alla parole vuota, comprende le parole del tipo 1x101 o 01x001, dove x è una stringa binaria in cui ogni occorrenza di 0 è seguita immediatamente da almeno un'occorrenza di 1. Formalmente il linguaggio può essere descritto da un'espressione regolare $\varepsilon + 1(01+1)^*101 + 01(01+1)^*001$ oppure insiemisticamente come il linguaggio $L = \{xyz \mid y \text{ è in } \{01,1\}^* \text{ e } x = 1 \text{ e } z = 101 \text{ oppure } x = 10 \text{ e } z = 001\} \cup \{\varepsilon\}$.

Esercizio 3.2
Si costruisca la grammatica equivalente, a meno della parola vuota, priva di regole cancellanti, facendo vedere i singoli passi di eliminazione.

Seguendo l'algoritmo dato si deve costruire l'insieme delle variabili cancellanti cioè quelle che derivano ε in uno o più passi:
$NULL_0 = \{A,S\}$

$NULL_1 = NULL_0$

Infatti non ci sono regole con parte destra fatta di parole su {A,S}

Quindi la grammatica diventa, aggiungendo le regole nella cui parte destra si cancella A o S:

$$
\begin{aligned}
S &\rightarrow 1AC \mid DAB \mid 1C \mid DB \\
A &\rightarrow DA \mid 1A \mid D \mid 1 \mid \\
B &\rightarrow 0D \\
C &\rightarrow 1D \\
D &\rightarrow 01
\end{aligned}
$$

La nuova grammatica non genera più la parola vuota e anche A ora genera ogni parola non vuota su {01,1}. Il linguaggio è lo stesso a meno della parola vuota.