Simplification of CFGs

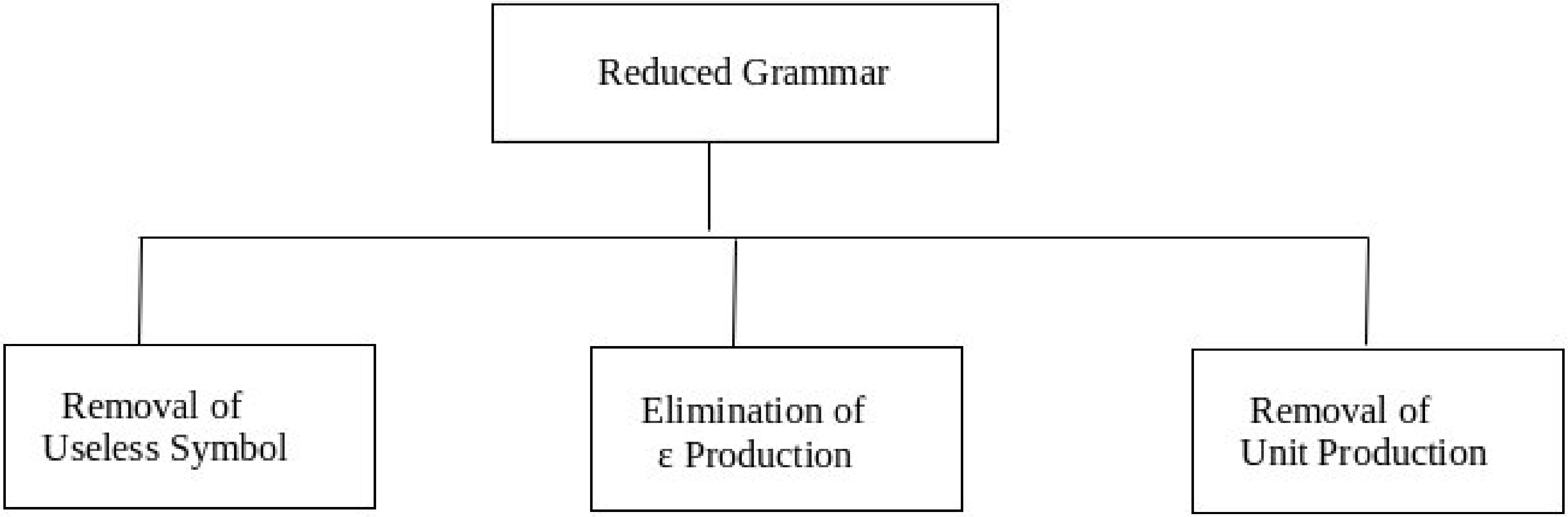
Elimination of Null/Epsilon Production
Elimination of Unit Production
Removal of Useless Productions

Simplification of CFG

- Various languages can efficiently be represented by a context-free grammar.
- All the grammar are not always optimized that means the grammar may consist of some extra symbols(non-terminal).

Simplification of CFG

- Having extra symbols, unnecessary increase the length of grammar.
- Simplification of grammar means reduction of grammar by removing useless symbols.



Null Production

Null Production-

A production of the form A->ε is called as ε-production

Nullable Non-terminal-

If A is ~~a non termi~~nal and If $A \underset{*}{\rightarrow} \varepsilon$, i.e. If A derives to an empty string in Zero, one or more derivations, then A is said to be nullable non-terminal

Elimination of Null Production

To eliminate ε-productions from a grammar, we use the following technique:-

- If A->ε is an ε-production to be eliminated ,
- then we look for all those productions in the grammar whose right side contain A,
- [replace each occurrence of A by ε in each of these productions to obtain the non-ε productions to be added to the grammar to keep the language generated same.]
  =>Addition

Elimination of Null Production

Eg 1) Consider the following grammar

S->aA

A->~~b|ε~~

Eliminate all the ε-productions from the grammar without changing the language

Null productions

$A \to \epsilon$

Lets check for productn with A on RHS

$S \to aA$, substitute ϵ

Grammar

$S \to aA \mid a$

$A \to b$

without

Λ

Elimination of Null Production

Soln 1)

S->aA

A->b|ε

To eliminate A-> ε ,

Replace A on the right side of the production S->aA by ε

To obtain a non ε-production S->a,

ε-free grammar is:-

S->aA|a

A->b

Elimination of Null Production

Eg 2) Consider the following grammar

S->ABAC

A->aA|ε

B->bB| ε

C->c

Eliminate all the ε-productions from the grammar without changing the language

Find all ε productions ⇒

$A \to \epsilon$
$B \to \epsilon$ } 2 productions

Lets remove $A \to \epsilon$ prodⁿ

search for prodⁿ with A on RHS

$S \to ABAC$
$A \to aA$ } RHS

substitute Null

$S \to ABAC \mid BAC \mid ABC \mid BC$
$A \to aA \mid a$, $B \to bB \mid \epsilon$, $C \to c$

Lets remove $B \to \epsilon$

B on RHS

② $B \to bB$

① $S \to ABAC \mid BAC \mid ABC \mid BC$

substitute with Null

$S \to ABAC \mid BAC \mid ABC \mid BC \mid AAC \mid AC \mid C$
$B \to bB \mid b$
$A \to aA \mid a$
$C \to c$

Elimination of Null Production

Eg 2) Consider the following grammar

S->ABAC

A->aA|ε

B->bB| ε

C->c

Eliminate all the ε-productions from the grammar without changing the language

Elimination of Null Production

Soln 2)
S->ABAC
A->aA|ε
B->bB|ε
C->c
To Eliminate A-> ε ,
Non- ε productions to be added
List of Productions containing A on RHS:-
S->ABAC
A->aA
Replace each occurrence of A by ε in each of these productions to obtain the non ε productions
S->BAC|ABC|BC
A->a
Add these productions to the grammar and eliminate A-> ε from the grammar to
S->ABAC|BAC|ABC|BC
A->aA|a
B->bB|ε
C->c

Elimination of Null Production

Soln 2)
S->ABAC|BAC|ABC|BC
A->aA|a
B->bB|ε
C->c

- To Eliminate B-> ε ,
- Non- ε productions to be added
- List of Productions containing B on RHS:-
- S->ABAC|BAC|ABC|BC
- B->bB

- Replace each occurrence of B by ε in each of these productions to obtain the non ε productions
- S->AAC|AC|C
- B->b
- Add these productions to the grammar and eliminate B-> ε from the grammar to
- **S->ABAC|BAC|ABC|BC|AAC|AC|C**
- **A->aA|a**
- **B->bB|b**
- **C->c**

Elimination of Null Production

EXERCISE) Consider the following grammar

$S \rightarrow ABCd$

$A \rightarrow BC$

$B \rightarrow bB \mid \lambda$

$C \rightarrow cC \mid \lambda$

Eliminate all the ε-productions from the grammar without changing the language

Unit Production

A production of the form A->B where both A and B are the non-terminals is called unit production

Unit production increases the cost of derivations

$$\boxed{A \rightarrow B}$$

LHS RHS

Elimination of Unit Production

Algorithm :-

while(there exists a unit production A->B in the grammar?) do

{

select a unit production A->B, such that there exists non-unit production

B->α

for (every non-unit production B-> α) do

add production A-> α to the grammar

eliminate A->B from the grammar

}

Elimination of Unit Production

Eg 3) Given the grammar, Remove Unit Production:

S->AB

A->a

B->C|b

C->D

~~D->E~~

~~E->a~~

Lets find all Unit Prodn

(1) B → C

(2) C → D

(3) D → E

Lets check for B → C

If there is a non-unit prodn for C? No

Let check for (2) C → D

If there is non-unit prodn for D? No

Lets check for D → E

If there is a non-unit prodn for E? Yes

E → a

D → a

E → a

D → a

C → a

B → a

S → AB

A → a

B → a/b

C → a

D → a

~~E → a~~

cannot be derived from S useless variable remove them

Start symbol = S

S → AB

$\boxed{\begin{array}{l} S \to AB \\ A \to a \\ B \to a/b \end{array}}$

Simplified Grammar

Elimination of Unit Production

Eg 3) Given the grammar:

S->AB

A->a

B->C|b

C->D

D->E

E->a

Elimination of Unit Production

Eg 3) Given the grammar:

S->AB
A->a
B->C|b
C->D
D->E
E->a

Eliminate all the unit productions from the grammar
B->C
C->D
D->E

Elimination of Unit Production

Eg 3) Given the grammar:

B->C

C->D

D->E

- For B->C, since there exists no non-unit C production in the grammar
- For C->D, since there exists no non-unit D production in the grammar
- For D->E, there exists a non-unit production for E, E->a, thus
- Add production D->a to the grammar
- Eliminate D->E

Elimination of Unit Production

Eg 3) Given the grammar:
B->C
C->D
D->E

- Check For B->C, since there exists no non-unit C production in the grammar
- Check For C->D, since there exists a non-unit production in the grammar for D,D->a, **<u>thus add production C->a</u>**
- **<u>Eliminate C->D</u>**

- Again Check For C->D, since there exists a non-unit production in the grammar for C, C->a, **<u>thus add production C->a</u>**
- **<u>Eliminate B->C</u>**

Elimination of Unit Production

Eg 3) Given the grammar:

S->AB

A->a

B->C|b

C->D

D->E

E->a

The final grammar without Null Production:-

S->AB

A->a

B->a|b

C->a

D->a

E->a

Elimination of Unit Production

The final grammar without Unit Production:-
S->AB
A->a
B->a|b
C->a
D->a
E->a

Symbols C,D,E become useless as a result of elimination of unit productions, because they will not be used in the derivation of any w in L(G)

Eliminate C,D,E to obtain the grammar:-
S->AB
A->a
B->a|b

Elimination of Unit Production

Exercise-Given the grammar, Remove Unit Production:

$S \rightarrow 0A \mid 1B \mid C$

$A \rightarrow 0S \mid 00$

$B \rightarrow 1 \mid A$

$C \rightarrow 01$

$S \rightarrow C$, $\boxed{C \rightarrow 01}$

$B \rightarrow A$

$S \rightarrow 01$

for $B \rightarrow A$

$A \rightarrow 0S \mid 00$. $\boxed{B \rightarrow 1 \mid 0S \mid 00}$

$S \rightarrow 0A \mid 1B \mid 01$

$A \rightarrow 0S \mid 00$

$B \rightarrow 1 \mid 0S \mid 00$

$C \rightarrow 01$

Remove C

final grammar

Reduced grammar

$S \rightarrow 0A \mid 1B \mid 01$

$A \rightarrow 0S \mid 00$

$B \rightarrow 1 \mid 0S \mid 00$

Elimination of Unit Production

Exercise-Given the grammar,
Remove Unit Production:

S → 0A | 1B | C
A → 0S | 00
B → 1 | A
C → 01

Unit Productions:-
S->C
B->A

Lets remove B->A
B->1|0S|00

Lets remove S->C
S->01

Thus, Grammar free of Unit production
S->0A|1B|01
A->0S|00
B->1|0S|00
C->01

Removal of Useless Productions

Eg 4) Given the grammar:

$S \to 0 \mid A$

~~$A \to AB$~~

~~$B \to 1$~~

$S \to 0$ deriving to $w \in T^*$

$B \to 1$ derive w or a string $\in T^*$

$A \to AB$

RHS A B

? derives to $w \in T^*$

Cannot derive a w from A $\Rightarrow$ Useless var $\Rightarrow$ Remove all the productions containing A

$S \to 0$

$S \Rightarrow A$ using $S \to A$

$\Rightarrow AB$ using $A \to AB$

$\Rightarrow A1$ using $B \to 1$

$\boxed{\begin{array}{l} S \to 0 \\ B \to 1 \end{array}}$ $\Rightarrow$ $\boxed{S \to 0}$ Reduced grammar

? B is not in RHS of S

Removal of Useless Productions

Eg 4) Given the grammar:

S->0|A

A->AB

B->1

- Since S->0 and B->1 are the productions of the form A->w, where w is in T*,
- Non terminal S and B are capable of deriving to w in T*

Removal of Useless Productions

Eg 4)S->0|A

A->AB

B->1

- Production A->AB, the RHS contains non-terminals A and B, even though B is known to be capable of deriving to w in T*, Non-terminal A is not deriving to w in T*.
- Eliminate the productions containing A,

S->0

B->1

- S is the start symbol.
- B doesn’t occur on the RHS of S production, it will not be used in the derivation of any w.
- Eliminating Non Terminal B, we get S->0
- Reduced Grammar Equivalent to the given grammar, containing no useless grammar

Removal of Useless Productions

Eg 5) Find the reduced grammar that is equivalent to the CFG given below:-

S->aC|SB

A->bSCa

B->aSB|bBC

C->aBC|ad

Removal of Useless Productions

Eg 5) Find the reduced grammar that is equivalent to the CFG given below:-

S->aC|SB

A->bSCa

B->aSB|bBC

~~C->aBC|ad~~

C → ad ⇒ C derives to w ∈ T*

S → aC S derives to w ∈ T*

A → b S C a

b: terminal; S: derives to w ∈ T*; C: derives to w ∈ T*; a: terminal

A derives to w ∈ T*

B → aSB | bBC ⇒ B cannot lead to a w ∈ T*

⇒ Remove all prodn with B on RHS

S → aC
A → bSCa
C → ad

⇒

S → aC
C → ad

Reduced grammar

Removal of Useless Productions

Eg 5) Find the reduced grammar that is equivalent to the CFG given below:-

S->aC|SB

A->bSCa

B->aSB|bBC

C->aBC|ad

- Since C->ad is the productions of the form A->w, where w is in T*,
- Non terminal C is capable of deriving to w in T*
- Production S->aC, the RHS contains a terminal a and Non terminal C which is capable of deriving w in T*,
- Thus, S is also capable of deriving w in T*

Removal of Useless Productions

Eg 5) S->aC|SB

A->bSCa

B->aSB|bBC

C->aBC|ad

- Production A->bSCa, the RHS contains a terminal b,a and Non terminal S,C capable of deriving w in T*,
- Thus, A is also capable of deriving w in T*
- Production B->aSB|bBC, the RHS contains non-terminals S,B and C
- Though S and C are capable of deriving to w in T*
- Non-terminal B is not capable of deriving w in T*
- Eliminating B Terminal

S->aC

A->bSCa

C->ad

Removal of Useless Productions

Eg 5) S->aC
A->bSCa
C->ad

- S is the start symbol,
- Since S->aC , terminal a and non terminal C will also be used in derivation.
- Terminal A doesn’t occur on RHS of S or C, Thus A will not be used in derivation of w in T*
- Eliminating A

S->aC
C->ad

- Reduced Grammar Equivalent to the given grammar, containing no useless grammar