Compiler Construction

Syntax Definitions - Grammers

Mr. Usman Wajid

usman.wajid@nu.edu.pk

March 8, 2023

National University
of Computer & Emerging Sciences

What is Grammer?

Grammer

It is a set of rules and examples dealing with the syntax and word structure of a language

What is Grammer?

Grammer

It is a set of rules and examples dealing with the syntax and word structure of a language

- It is basically defined as a set of 4-tuple (V,T,P,S), where,
  1. **V** is set of nonterminals (variables)
  2. **T** is set of terminals (primitive symbols)
  3. **P** is set of productions (rules), which govern the relationship between non-terminal and terminals
  4. **S** is start symbol with which strings in grammar are derived

Grammar example in English Language

- Consider the following English grammar rules,

  $\langle \texttt{sentence} \rangle \rightarrow \langle \texttt{noun} \rangle \langle \texttt{verb} \rangle$

  $\langle \texttt{noun} \rangle \rightarrow \langle \texttt{com-noun} \rangle \mid \langle \texttt{prop-noun} \rangle$

  $\langle \texttt{verb} \rangle \rightarrow$ **ate** | **sat** | **ran**

  $\langle \texttt{prop-noun} \rangle \rightarrow$ **Ali** | **Mahad**

  $\langle \texttt{com-noun} \rangle \rightarrow$ **She** | **He**

- Using these rules, many sentences can be derived

Grammar example in English Language

- Consider the following English grammar rules,

  $\langle\, \texttt{sentence} \,\rangle \rightarrow \langle\, \texttt{noun} \,\rangle \langle\, \texttt{verb} \,\rangle$

  $\langle\, \texttt{noun} \,\rangle \rightarrow \langle\, \texttt{com-noun} \,\rangle \mid \langle\, \texttt{prop-noun} \,\rangle$

  $\langle\, \texttt{verb} \,\rangle \rightarrow$ **ate** | **sat** | **ran**

  $\langle\, \texttt{prop-noun} \,\rangle \rightarrow$ **Ali** | **Mahad**

  $\langle\, \texttt{com-noun} \,\rangle \rightarrow$ **She** | **He**

- Using these rules, many sentences can be derived

  1. Ali ate
  2. Mahad ran
  3. He sat
  4. He ran

Grammar ***notations***

Some of the ***notations*** used to represent grammar are,

1. **Terminal Symbols:** these can be represented by,
   - Lower-case letters of alphabet like a, c, z, etc
   - Operator symbols like +, **-**, etc
   - Punctuation symbols like (, {, ; etc
   - Digits 0-9
   - Bold face strings **int, main, if, else** etc

Grammar *notations*

Some of the ***notations*** used to represent grammar are,

1. **Terminal Symbols:** these can be represented by,
   - Lower-case letters of alphabet like a, c, z, etc
   - Operator symbols like +, -, etc
   - Punctuation symbols like (, {, ; etc
   - Digits 0-9
   - Bold face strings **int, main, if, else** etc
2. **Non-terminal Symbols:** there are represented by,
   - Upper case letters of alphabet like A,C,Z etc
   - Letter S is the start symbol
   - Lower case strings like expr, stmt, etc

Grammar ***notations***

Some of the ***notations*** used to represent grammar are,

1. **Terminal Symbols:** these can be represented by,
   - Lower-case letters of alphabet like a, c, z, etc
   - Operator symbols like +, -, etc
   - Punctuation symbols like (, $\{$, ; etc
   - Digits 0-9
   - Bold face strings **int, main, if, else** etc
2. **Non-terminal Symbols:** there are represented by,
   - Upper case letters of alphabet like A,C,Z etc
   - Letter S is the start symbol
   - Lower case strings like expr, stmt, etc
3. Grammar symbols can be terminals or non-terminals. The can be represented by Greek letters such as α, β

Grammar *notations*

Some of the ***notations*** used to represent grammar are,

1. **Terminal Symbols:** these can be represented by,
   - Lower-case letters of alphabet like a, c, z, etc
   - Operator symbols like +, -, etc
   - Punctuation symbols like (, {, ; etc
   - Digits 0-9
   - Bold face strings **int, main, if, else** etc
2. **Non-terminal Symbols:** there are represented by,
   - Upper case letters of alphabet like A,C,Z etc
   - Letter S is the start symbol
   - Lower case strings like expr, stmt, etc
3. Grammar symbols can be terminals or non-terminals. The can be represented by Greek letters such as α, β
4. Lower case letters u, v, w, x, y, z are generally used to represent a string of terminals

Type of Grammers —Chomsky Hierarchy

Chomsky Hierarchy

Linguist Noam Chomsky defined a four-level hierarchy, in terms of complexity, that correspond to four classes of machines

Type of Grammers —Chomsky Hierarchy

Chomsky Hierarchy

Linguist Noam Chomsky defined a four-level hierarchy, in terms of complexity, that correspond to four classes of machines

a. Type 0 Grammers—Unrestricted Grammer (URG)

Type of Grammers —Chomsky Hierarchy

Chomsky Hierarchy

Linguist Noam Chomsky defined a four-level hierarchy, in terms of complexity, that correspond to four classes of machines

a. Type 0 Grammers—Unrestricted Grammer (URG)
b. Type 1 Grammers—Context Sensitive Grammers (CSG)

Type of Grammers —Chomsky Hierarchy

Chomsky Hierarchy

Linguist Noam Chomsky defined a four-level hierarchy, in terms of complexity, that correspond to four classes of machines

a. Type 0 Grammers—Unrestricted Grammer (URG)
b. Type 1 Grammers—Context Sensitive Grammers (CSG)
c. Type 2 Grammers—Context Free Grammers (CFG)

Type of Grammers —Chomsky Hierarchy

Chomsky Hierarchy

Linguist Noam Chomsky defined a four-level hierarchy, in terms of complexity, that correspond to four classes of machines

a. Type 0 Grammers—Unrestricted Grammer (URG)
b. Type 1 Grammers—Context Sensitive Grammers (CSG)
c. Type 2 Grammers—Context Free Grammers (CFG)
d. Type 3 Grammers—Regular Grammers (RG)

Chomsky Hierarchy continued …

Type 0 Grammers—Unrestricted Grammers (URG)

All the productions are of the form $\alpha \rightarrow \beta$, where α and β may have any number of terminals and non-terminals, i.e., no restrictions on either side of the productions

Chomsky Hierarchy continued ...

Type 0 Grammers—Unrestricted Grammers (URG)

All the productions are of the form $\alpha \rightarrow \beta$, where α and β may have any number of terminals and non-terminals, i.e., no restrictions on either side of the productions

- The language specified by L(G) = $\{a^{2^i} | I >= 1\}$

Type 0 Grammers—Unrestricted Grammers (URG)

All the productions are of the form $\alpha \to \beta$, where α and β may have any number of terminals and non-terminals, i.e., no restrictions on either side of the productions

- The language specified by L(G) = $\{a^{2^i} | I >= 1\}$
- It can be represented by the following productions in URG,

S $\to$ACaB	aD$\to$Da
Ca$\to$aaC	AD$\to$AD
CB$\to$DB	aE$\to$Ea
CB$\to$E	AE$\to$ ϕ

Chomsky Hierarchy continued ...

Type 0 Grammers—Unrestricted Grammers (URG)

All the productions are of the form $\alpha \rightarrow \beta$, where α and β may have any number of terminals and non-terminals, i.e., no restrictions on either side of the productions

- The language specified by L(G) = $\{a^{2^i} | I >= 1\}$
- It can be represented by the following productions in URG,

S $\rightarrow$ACaB	aD$\rightarrow$Da
Ca$\rightarrow$aaC	AD$\rightarrow$AD
CB$\rightarrow$DB	aE$\rightarrow$Ea
CB$\rightarrow$E	AE$\rightarrow$ ϕ

- The ”aa” can be derived from the above grammar as follows:

Type 0 Grammers—Unrestricted Grammers (URG)

All the productions are of the form $\alpha \rightarrow \beta$, where α and β may have any number of terminals and non-terminals, i.e., no restrictions on either side of the productions

- The language specified by L(G) = $\{a^{2^i} | I >= 1\}$
- It can be represented by the following productions in URG,

S $\rightarrow$ACaB	aD$\rightarrow$Da
Ca$\rightarrow$aaC	AD$\rightarrow$AD
CB$\rightarrow$DB	aE$\rightarrow$Ea
CB$\rightarrow$E	AE$\rightarrow$ ϕ

- The ”aa” can be derived from the above grammar as follows:
  S $\Longrightarrow$ ACaB $\Longrightarrow$ AaaCB $\Longrightarrow$ AaaE $\Longrightarrow$ AaEa $\Longrightarrow$ AEaa $\Longrightarrow$ aa

Chomsky Hierarchy continued ...

Type 1 Grammers—Context Sensitive Grammars (CSG)

All the productions are of the form $\alpha \rightarrow \beta$ with $|\alpha| \leq |\beta|$, where α and β may have any number of terminals and non-terminals

Chomsky Hierarchy continued ...

Type 1 Grammers—Context Sensitive Grammars (CSG)

All the productions are of the form $\alpha \to \beta$ with $|\alpha| \leq |\beta|$, where α and β may have any number of terminals and non-terminals

- The language specified by L(G) = $\{a^n b^n \mid n \geq 1\}$

Chomsky Hierarchy continued …

Type 1 Grammers—Context Sensitive Grammars (CSG)

All the productions are of the form $\alpha \to \beta$ with $|\alpha| \leq |\beta|$, where α and β may have any number of terminals and non-terminals

- The language specified by L(G) = $\{a^n b^n \mid n \geq 1\}$
- This grammar is represented by the following productions in CSG,
  S $\to$ SBC | aC
  B $\to$ a
  CB $\to$ BC
  Ba $\to$ aa
  C $\to$ b

Chomsky Hierarchy continued ...

Type 1 Grammers—Context Sensitive Grammars (CSG)

All the productions are of the form $\alpha \rightarrow \beta$ with $|\alpha| \leq |\beta|$, where α and β may have any number of terminals and non-terminals

- The language specified by L(G) = $\{a^n b^n \mid n \geq 1\}$
- This grammar is represented by the following productions in CSG,
  S $\rightarrow$ SBC | aC
  B $\rightarrow$ a
  CB $\rightarrow$ BC
  Ba $\rightarrow$ aa
  C $\rightarrow$ b
- ”aaabbb” can be derived from the above grammar as follows,

Type 1 Grammers—Context Sensitive Grammars (CSG)

All the productions are of the form $\alpha \rightarrow \beta$ with $|\alpha| \leq |\beta|$, where α and β may have any number of terminals and non-terminals

- The language specified by L(G) = $\{a^n b^n \mid n \geq 1\}$
- This grammar is represented by the following productions in CSG,
  S $\rightarrow$ SBC $|$ aC
  B $\rightarrow$ a
  CB $\rightarrow$ BC
  Ba $\rightarrow$ aa
  C $\rightarrow$ b
- ”aaabbb” can be derived from the above grammar as follows,
  S $\Longrightarrow$ SBC $\Longrightarrow$ SBCBC $\Longrightarrow$ aCBCBC $\Longrightarrow$ aBCCBC $\Longrightarrow$ aaCCBC $\Longrightarrow$ aaCBCC $\Longrightarrow$ aaBCC $\Longrightarrow$ aaaCCC $\Longrightarrow$ aaabbb

Chomsky Hierarchy continued ...

Type 2 Grammers—Context Free Grammars (CFG)

All the productions are of the form $\alpha \rightarrow \beta$ with $|\alpha| \leq |\beta|$, $|\alpha|$ = 1 and is a non-terminal where β may have any number of terminals and non-terminals

Chomsky Hierarchy continued ...

Type 2 Grammers—Context Free Grammars (CFG)

All the productions are of the form $\alpha \to \beta$ with $|\alpha| \leq |\beta|$, $|\alpha|$ = 1 and is a non-terminal where β may have any number of terminals and non-terminals

- The language specified by L(G) = $\{a^n b^n \mid n \geq 1\}$

Chomsky Hierarchy continued ...

Type 2 Grammers—Context Free Grammars (CFG)

All the productions are of the form $\alpha \rightarrow \beta$ with $|\alpha| \leq |\beta|$, $|\alpha|$ = 1 and is a non-terminal where β may have any number of terminals and non-terminals

- The language specified by L(G) = $\{a^n b^n \mid n \geq 1\}$
- This grammar is represented by the following productions in CFG,
  S $\rightarrow$ aSb $\mid \phi$

Chomsky Hierarchy continued . . .

Type 2 Grammers—Context Free Grammars (CFG)

All the productions are of the form $\alpha \rightarrow \beta$ with $|\alpha| \leq |\beta|$, $|\alpha|$ = 1 and is a non-terminal where β may have any number of terminals and non-terminals

- The language specified by L(G) = $\{a^n b^n \mid n \geq 1\}$
- This grammar is represented by the following productions in CFG,
  S $\rightarrow$ aSb $|$ ϕ
- ”aabb” can be derived from the above grammar as follows,

Chomsky Hierarchy continued ...

Type 2 Grammers—Context Free Grammars (CFG)

All the productions are of the form $\alpha \rightarrow \beta$ with $|\alpha| \leq |\beta|$, $|\alpha|$ = 1 and is a non-terminal where β may have any number of terminals and non-terminals

- The language specified by L(G) = $\{a^n b^n \mid n \geq 1\}$
- This grammar is represented by the following productions in CFG,
  S $\rightarrow$ aSb | ϕ
- "aabb" can be derived from the above grammar as follows,

  S $\Longrightarrow$ aSb $\Longrightarrow$ aaSbb $\Longrightarrow$ aaϕbb $\Longrightarrow$ aabb

Chomsky Hierarchy continued . . .

Type 3 Grammers—Regular Grammars

These are the languages exactly decided by a finite state automaton. It can even be obtained by the regular expressions (RE)

Chomsky Hierarchy continued . . .

Type 3 Grammers—Regular Grammars

These are the languages exactly decided by a finite state automaton. It can even be obtained by the regular expressions (RE)

Example:

- Right linear grammer: A $\rightarrow$ a A | a
- Left linear grammer: A $\rightarrow$ A a | a

Type 3 Grammers—Regular Grammars

These are the languages exactly decided by a finite state automaton. It can even be obtained by the regular expressions (RE)

Example:

- Right linear grammer: A $\rightarrow$ a A $|$ a
- Left linear grammer: A $\rightarrow$ A a $|$ a
- An example of a regular grammer G with $V = \{S, A\}$, Σ = $\{a, b, c\}$, P consists of the following rules,
  S $\rightarrow$ aS $|$ bA
  A $\rightarrow$ ϕ
  A $\rightarrow$ cA

Chomsky Hierarchy continued ...

Type 3 Grammers—Regular Grammars

These are the languages exactly decided by a finite state automaton. It can even be obtained by the regular expressions (RE)

Example:

- Right linear grammer: A $\rightarrow$ a A | a
- Left linear grammer: A $\rightarrow$ A a | a
- An example of a regular grammer G with $V = \{S, A\}$, Σ = $\{a, b, c\}$, P consists of the following rules,
  S $\rightarrow$ aS | bA
  A $\rightarrow$ ϕ
  A $\rightarrow$ cA
- It describes the same language as by the regular expression a^*bc^*

Chomsky Hierarchy continued ...

A Demonstrative Example:

- Give a CSG but not CFG for ($a^n \mid n \geq 1$),

Chomsky Hierarchy continued ...

A Demonstrative Example:

- Give a CSG but not CFG for ($a^n \mid n \geq 1$),
  S $\rightarrow$ aS | X
  aS $\rightarrow$ aa
  X $\rightarrow$ a

- give a CFG but not an RG for ($a^n \mid n \geq 1$),

A Demonstrative Example:

- Give a CSG but not CFG for ($a^n \mid n \geq 1$),
  $S \rightarrow aS \mid X$
  $aS \rightarrow aa$
  $X \rightarrow a$

- give a CFG but not an RG for ($a^n \mid n \geq 1$),
  $S \rightarrow AS \mid a$
  $A \rightarrow a$

- give an RG for ($a^n \mid n \geq 1$),

Chomsky Hierarchy continued ...

A Demonstrative Example:

- Give a CSG but not CFG for ($a^n \mid n \geq 1$),
  S $\rightarrow$ aS | X
  aS $\rightarrow$ aa
  X $\rightarrow$ a

- give a CFG but not an RG for ($a^n \mid n \geq 1$),
  S $\rightarrow$ AS | a
  A $\rightarrow$ a

- give an RG for ($a^n \mid n \geq 1$),
  S $\rightarrow$ aS | a