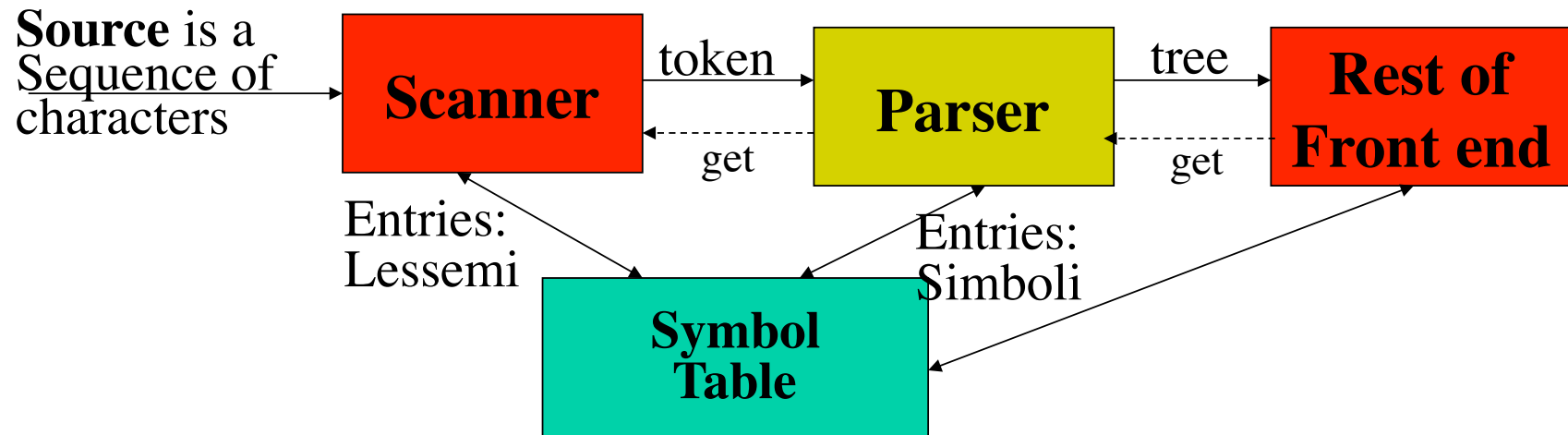


SYNTACTIC ANALYSIS

- **How to define Syntax;**
- **How to do Analysis;**
- **At what extent Analysis can be done in Linear Time;**
- **How to build (Linear) Parser Generators;**
- **Relationship between Analysis and Internal Representation (Tree Representation)**

One Pass Structure (Two phase pipeline)



Syntactic Analysis (Parser) is driven from
Semantics Analysis which is asking for visiting a subtree not built yet

How to define Syntax

- Syntactic Analysis and Syntactic Languages
- Syntactic Languages and Grammars
- Classification of Grammars
- Classification of Languages
- Foundations: Derivation, Sentential Form, Ambiguity, Tarski's Fixpoint Iteration

Syntactic Analysis

- It scans sequences of tokens to check for *phrase structures* that belong to the Syntax of the Language
- Syntax, just like Lexics, is expressed by a Language: The Syntactic Language
- Syntactic Langs are much more complicate than Lexical ones

$\{u^n v^n \mid n \geq 0\}$ has been proved not be regular

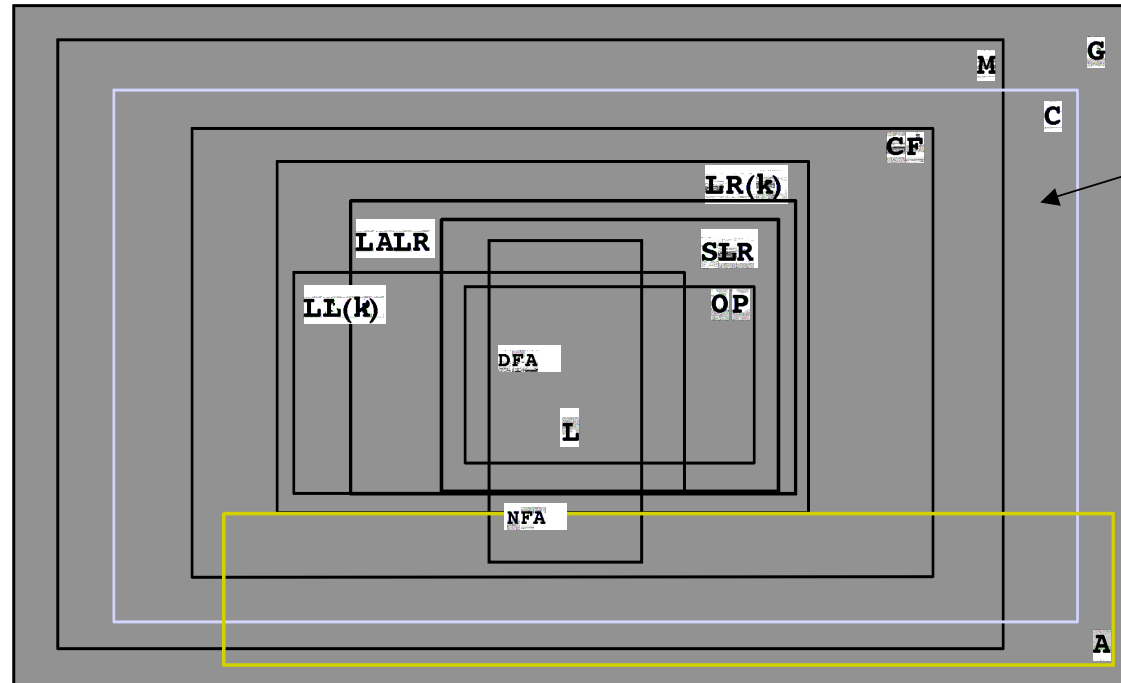
but

$\text{num}+(3-((\text{id}*\text{id})+\text{num})/\text{id}) \in \{(\alpha u)^n (\alpha v)^n \alpha \mid n \geq 0, \alpha \in L, u="(", v=")"\}$

4

Grammar Classification (Chomsky)

Grammars Inclusion



Non-monotone C

G=General [Recursive Enumerable but Non-Recursive - $\{u^n v^k \text{akermann}(n)\}$]

A=Ambiguous

M=Monotone [Recursive Languages - $\{u^n v^n\}$]

C=Contextual [$\{u^n v^n z^n\}$]

LR(k)=Context-Free [$\{u^n v^n\}$]

LALR(k)=Context-Free [$\{u^n v^n\}$]

SLR(k)=Simple Left-to-right rightmost reversed [Viable-Prefix; Bottom-Up/k symbols $\{u^n v^n\}$]

LL(K)=Leftmost-Left Left-to-right [Predictive; Top-Down/k symbols $\{u^n v^n\}$]

OP=Operator-Precedence [$\{u^n v^n\}$]

L=Linear [Recursive Grammars; Regular Languages]

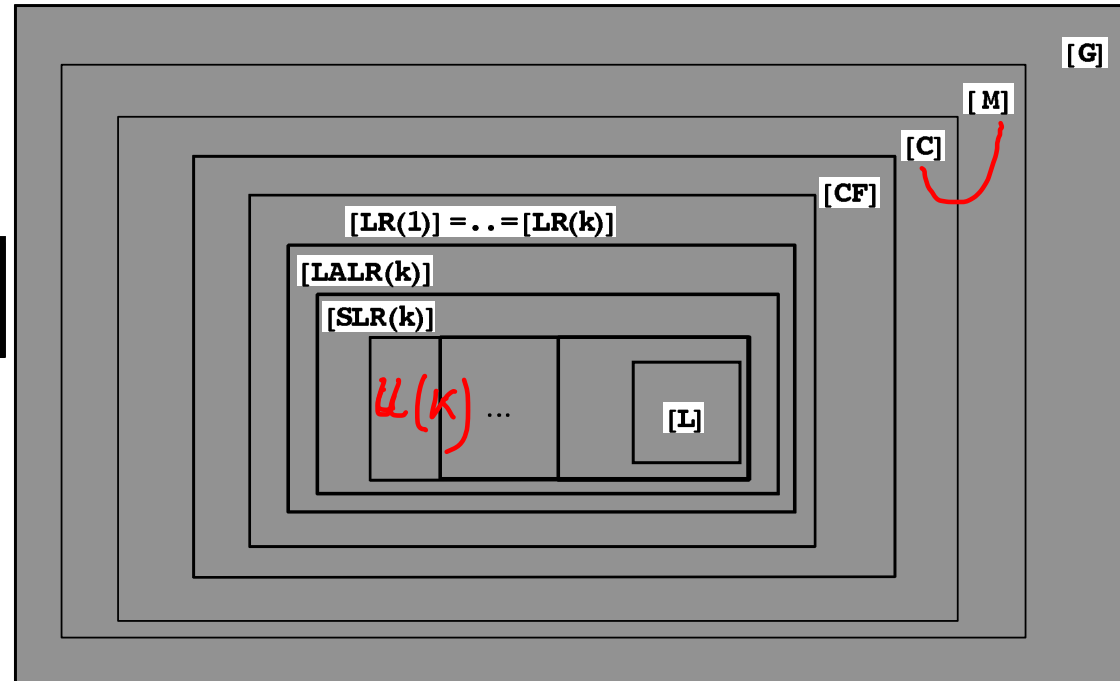
DFA= -- [Regular Grammars/Expressions; Regular Languages]

NFS= -- [Regular Grammars/Expressions; Regular Languages]

**Kinds of Grammar
[Defined Language Features]**

Language Classification

Language
Inclusion



[G] = Recursively Enumerable Languages

[M] = Recursive Languages

[C] = Contextual Languages: $\{u^n v^n z^n \mid n \geq 0\}$

[CF] = Context-Free Languages: $\{u^n v^m z^k \mid n, m, k \geq 0 \text{ and } (n=m \text{ or } m=k)\}$

[LR(k)] = LR/k symbols Languages: $\{u^m v^n \mid m > n \geq 0\}$

[LALR(k)] = LALR/k symbols Languages

[SLR(k)] = SLR/k symbols Languages

[LL(k)] = LL/k symbols Languages: $\{u^n v^n \mid n \geq 0\}$

[L] = Regular Languages: $\{u^n v^m \mid n \geq 0, m \geq 0\}$

Productions: No *-operator

Let $G = \langle V, \Sigma, s \in V, P \rangle$

Productions

$A ::= \gamma \in P$ only if:

- $A \in V$

- $\gamma \in \{\gamma_1 | \dots | \gamma_n : n \geq 1, \gamma \in (\Sigma \cup V)^*\}$

$A ::= a A | c$

but:

$A ::= a^* | b$

Canonical

P contains exactly one production $A ::= \gamma$
for each nonterminal $A \in V$

$A ::= a A | c$

Normal

No production right part γ can contain
occurrences of |-operator

$A ::= a A$

$A ::= c$

From Grammars to Languages

A methodology for finding $L(G)$, given G

$A_0 ::= e_0$
 $A_1 ::= e_1$
 \dots
 $A_n ::= e_n$

1) Partial ordering \geq^G on non-Terminals

$\forall i \geq 0, e_i \equiv f(A_{i_1}, \dots, A_{i_{n_i}})$ then: $A_{i_1}, \dots, A_{i_{n_i}} \geq^G A_i$

Removal of Mutual Recursion, when possible

$A_j ::= g(A_{j_1}, \dots, A_i, \dots, A_{j_{n_j}})$ con $A_{j_1}, \dots, A_{j_{n_j}} \geq^G A_i \geq^G A_j$
 $A_i ::= f(A_{i_1}, \dots, A_j, \dots, A_{i_{n_i}})$ con $A_{i_1}, \dots, A_{i_{n_i}} \geq^G A_i$



$A_j ::= g(A_{j_1}, \dots, A_i, \dots, A_{j_{n_j}})$
 $A_i ::= f(A_{i_1}, \dots, g(A_{j_1}, \dots, A_i, \dots, A_{j_{n_j}}), \dots, A_{i_{n_i}})$

2

Example

$S ::= u A B v \mid B u$
 $B ::= u S v \mid u v u$

$S ::= u A B v \mid B u$
 $A ::= v u v \mid B$
 $B ::= u S v \mid u v u$

$B \geq^G A \geq^G S$

$S ::= u A B v \mid B u$
 $B ::= u (u A B v \mid B u) v \mid u v u$

$S ::= u A B v \mid B u$
 $A ::= v u v \mid B$
 $B ::= uu A B vv \mid u B uv \mid u v u$

$A ::= v u v \mid B$
 $B ::= uu A B vv \mid u B uv \mid u v u$

$S ::= u A B v \mid B u$
 $A ::= v u v \mid B$
 $B ::= uu vuv B vv \mid uu B B vv \mid u B uv \mid u v u$

$A ::= v u v \mid B$
 $B ::= uu (v u v \mid B) B vv \mid u B uv \mid u v u$

2) Productions as equations on languages $L(A_i)$

$\forall i \geq 0, L(A_i) = L(e_i)$

- $L(e)$ is an expression on 2^{Σ^*} containing only:
 X (finite products)
 \cup (possibly, denumerable unions)
- $L(e)$ is continuous on 2^{Σ^*}

Whenever $L(A_i) = L(e_i)$ is recursive: $L(e_i) \equiv E(L(A_i))$

Recursive equations $X = E(X)$, have to be solved in the variable $X \equiv L(A_i)$ on 2^{Σ^*} using the **Tarski's (Fixpoint) Iteration** below:

$$\begin{aligned} X &= \bigcup_{i \in \mathbb{N}} E(\perp)^i \\ E(\perp)^0 &= E(X \leftarrow \perp) \\ E(\perp)^{k+1} &= E(X \leftarrow E(\perp)^k) \end{aligned}$$

What about a system of equations

A system of Recursive equations:

$$\{X_1 = E_1(X_1, \dots, X_n), \dots, X_n = E_n(X_1, \dots, X_n)\}$$

$X_i \equiv L(A_i)$ on 2^{Σ^*} using the **Tarski's (Fixpoint) Iteration** below:

$$\begin{aligned} X_j &= \bigcup_{i \in \mathbb{N}} E_j(\perp, \dots, \perp)^i \\ E_j(\perp, \dots, \perp)^0 &= E_j(X_1 \leftarrow \perp, \dots, X_n \leftarrow \perp) \\ E_j(\perp, \dots, \perp)^{k+1} &= E_j(X_1 \leftarrow E_1(\perp)^k, \dots, X_n \leftarrow E_n(\perp)^k) \end{aligned}$$

Example 1: Tarski's Fixpoint Iteration

$$S ::= u S \mid \varepsilon \quad \longrightarrow \quad X = \{u\} \times X \cup \{\lambda\}$$

$$S ::= S u \mid \varepsilon \quad \boxed{E(X)}$$

$$E(X \leftarrow \perp)^0 = \{u\} \times \perp \cup \{\lambda\} = \perp \cup \{\lambda\} = \{\lambda\}$$

$$E(X \leftarrow \perp)^1 = \{u\} \times \{\lambda\} \cup \{\lambda\} = \{u, \lambda\}$$

$$E(X \leftarrow \perp)^2 = \{u\} \times \{u, \lambda\} \cup \{\lambda\} = \{uu, u, \lambda\}$$

$$E(X \leftarrow \perp)^3 = \{u\} \times \{uu, u, \lambda\} \cup \{\lambda\} = \{u^3, u^2, u, \lambda\}$$

$$\begin{aligned}
 E(X \leftarrow \perp)^{n+1} &= \{u\} \times \{u^k \mid k \in [0..n]\} \cup \{\lambda\} \\
 &= \{u^{k+1} \mid k \in [0..n]\} \cup \{u^0\} \\
 &= \{u^k \mid k \in [1..n+1]\} \cup \{u^0\} \\
 &= \{u^k, u^0 \mid k \in [1..n+1]\} \\
 &= \{u^k \mid k \in [0..n+1]\} = \{u^{n+1}, u^n, \dots, u, \lambda\}
 \end{aligned}$$

$$E(X \leftarrow \perp)^n = \{u^n, u^{n-1}, \dots, u, \lambda\}$$

$$L(S) = \{u^n \mid n \in \mathbb{N}\}$$

$$\begin{aligned}
 &= \bigcup_{n \in \mathbb{N}} \{u^n\} \\
 &= u^*
 \end{aligned}$$

Example2: Tarski's Fixpoint Iteration

$S ::= u S v \mid z$



$X = \{u\} \times X \times \{v\} \cup \{z\}$

$E(X)$

$$E(X \leftarrow \perp)^0 = \{u\} \times \perp \times \{v\} \cup \{z\} = \perp \cup \{z\} = \{z\}$$

$$E(X \leftarrow \perp)^1 = \{u\} \times \{z\} \times \{v\} \cup \{z\} = \{uzv, z\}$$

$$E(X \leftarrow \perp)^2 = \{u\} \times \{uzv, z\} \times \{v\} \cup \{z\} = \{u^2zv^2, uzv, z\}$$



$$E(X \leftarrow \perp)^n = \{u^n z v^n, u^{n-1} z v^{n-1}, \dots, z\}$$



$$L(S) = \{u^n z v^n \mid n \geq 0\}$$

7

Example3: Tarski's Fixpoint Iteration

$A ::= A + A$

$A ::= A * A$

$A ::= \text{id}$

$\longrightarrow X = \{x+y \mid x,y \in X\} \cup \{x*y \mid x,y \in X\} \cup \{\text{id}\}$

$E(X)$

$$E(X \leftarrow \perp)^0 = \{x+y \mid x,y \in \perp\} \cup \{x*y \mid x,y \in \perp\} \cup \{\text{id}\} \\ = \perp \cup \perp \cup \{\text{id}\} = \{\text{id}\}$$

$$E(X \leftarrow \perp)^1 = \{x+y \mid x,y \in \{\text{id}\}\} \cup \{x*y \mid x,y \in \{\text{id}\}\} \cup \{\text{id}\} \\ = \{\text{id}+\text{id}\} \cup \{\text{id}*\text{id}\} \cup \{\text{id}\} = \{\text{id}, \text{id}+\text{id}, \text{id}*\text{id}\}$$

$$E(X \leftarrow \perp)^2 = \{x+y \mid x,y \in E(X \leftarrow \perp)^1\} \cup \{x*y \mid x,y \in E(X \leftarrow \perp)^1\} \cup \{\text{id}\} \\ = \{\text{id}, \text{id}+\text{id}, \text{id}*\text{id}, \text{id}+\text{id}+\text{id}, \text{id}+\text{id}*\text{id}, \text{id}*\text{id}+\text{id}, \text{id}*\text{id}*\text{id}, \dots, \text{id}*\text{id}*\text{id}*\text{id}\}$$

$$E(X \leftarrow \perp)^n = \{\text{id}, \text{id } t^k \mid t \in \{+\text{id}, *\text{id}\}, k \in [1..2^n-1]\}$$

$$L(A) = \{\text{id } t^n \mid t \in \{+\text{id}, *\text{id}\}, n \in \mathbb{N}\}$$

8

How to do Syntactic Analysis

- Top-Down and leftmost derivation
 $LL(k)$
- Bottom-Up and reversed rightmost derivation
 $LR(k), LALR(k), \&LR(k)$

Definitions: Derivation, SF

Let $G = \langle V, \Sigma, s \in V, P \rangle$

Derivation is a binary relation \Rightarrow_G su $(\Sigma \cup V)^* \times (\Sigma \cup V)^*$

$$\alpha A \beta \Rightarrow \alpha \gamma \beta \quad \text{sse } A ::= \gamma \in P$$

Subscript, G , is omitted, in \Rightarrow_G , when the grammar G is clearly stated from the context

\Rightarrow^* : Transitive and Reflexive Closure of \Rightarrow

- $\alpha \Rightarrow^* \alpha$
- if $\alpha_1 \Rightarrow \dots \Rightarrow \alpha_n$
then $\alpha_1 \Rightarrow^* \alpha_n$

\Rightarrow^+ : Transitive Closure of \Rightarrow

$$\text{if } \alpha_1 \Rightarrow \alpha_2 \Rightarrow \dots \Rightarrow \alpha_n \\ \text{then } \alpha_1 \Rightarrow^+ \alpha_n$$

Sentential form of G
 $SF = \{ \gamma \mid s \Rightarrow^* \gamma \}$

L(G): Language Generated by a Grammar

Let $G = \langle V, \Sigma, s \in V, P \rangle$

$$L(G) = \{w \in \Sigma^* \mid s \Rightarrow^+ w\}$$

(where \Rightarrow is \Rightarrow_G)

Example: Let G below

$p1: E ::= E + E$

$p2: E ::= E * E$

$p3: E ::= id$

Then

$id + id \in L(E)$

A proof (Leftmost Derivation):

$E \Rightarrow_{(p1)} E + E \Rightarrow_{(p3)} id + E \Rightarrow_{(p3)} id + id$

A different proof (Rightmost Derivation):

$E \Rightarrow_{(p1)} E + E \Rightarrow_{(p3)} E + id \Rightarrow_{(p3)} id + id$

11

Ambiguous Grammars are Bad Definitions for Lang. Syntax

Example: Let G below

$p1: E ::= E + E$

$p2: E ::= E * E$

$p3: E ::= id$

Then

$id + id * id \in L(E)$

A proof (Leftmost Derivation):

$E \Rightarrow_{(p1)} E + E \Rightarrow_{(p3)} id + E \Rightarrow_{(p2)} id + E * E \Rightarrow_{(p3)} id + id * E \Rightarrow_{(p3)} id + id * id$

A different proof (another Leftmost Derivation):

$E \Rightarrow_{(p2)} E * E \Rightarrow_{(p1)} E + E * E \Rightarrow_{(p1)} id + E * E \Rightarrow_{(p1)} id + id * E \Rightarrow_{(p1)} id + id * id$

Different Leftmost (Rightmost) Derivations lead to different Parse Trees

12

Derivations (on the P-tree domain)

Let $G = \langle V, \Sigma, s \in V, P \rangle$

$(\Sigma \cup V)_T^*$

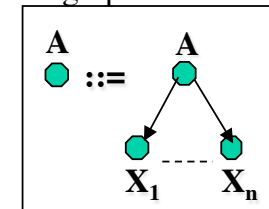
Smallest set such that, $\forall a \in \Sigma \cup V$:

- leaf: $\langle [a, -] \rangle \in (\Sigma \cup V)_T^*$
- $\forall \langle t_1, \dots, t_n \rangle \in (\Sigma \cup V)_T^*$
 - Tree: $\langle [a, \langle t_1, \dots, t_n \rangle] \rangle \in (\Sigma \cup V)_T^*$
 - Forest: $\langle t_1, \dots, t_n, u_1, \dots, u_m \rangle \in (\Sigma \cup V)_T^*, \forall \langle u_1, \dots, u_m \rangle \in (\Sigma \cup V)_T^*$

Productions on $(\Sigma \cup V)_T^*$

$A ::= X_1 \dots X_n \in P \quad \text{sse} \quad \langle [A, -] \rangle ::= \langle [A, \langle \langle [X_1, -] \rangle, \dots, \langle [X_n, -] \rangle \rangle] \rangle \in P_T$

A graphical view



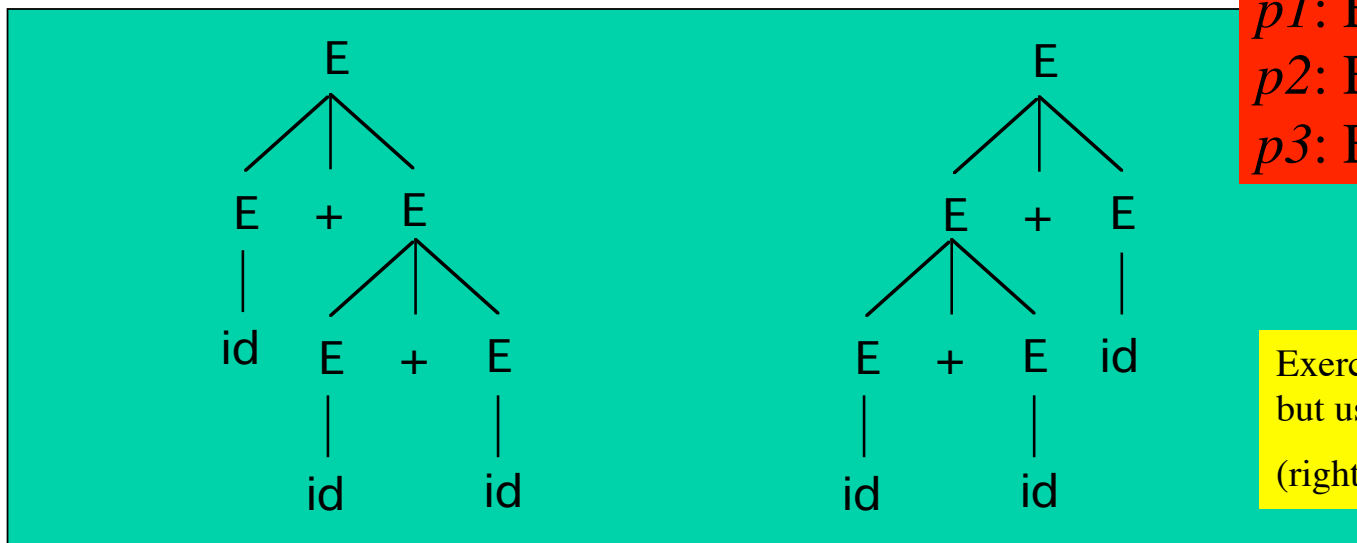
Relation \Rightarrow on $(\Sigma \cup V)_T^* \times (\Sigma \cup V)_T^*$

$\alpha A \beta \Rightarrow \alpha \gamma \beta \quad \text{sse} \quad A ::= \gamma \in P_T$

Ambiguous Grammars

A Graphical View

A different proof of ambiguity that uses: The *last trees* of two different Tree-derivations



$p1: E ::= E + E$
 $p2: E ::= E * E$
 $p3: E ::= id$

Exercise: Show the same but using leftmost (rightmost) derivations

TOP-DOWN and BOTTOM-UP Parsers

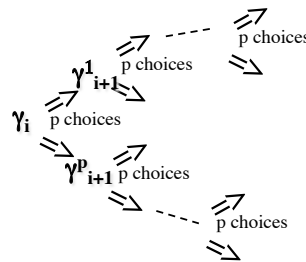
Let $G = \langle V, \Sigma, s \in V, P \rangle$ be a (context free) grammar. Let w be a sequence of words in Σ .

- Analysis has to answer to the following question:
is $w \in L(G)$ or not ?
- or, equivalently:
is $s \Rightarrow^* w$ or not ?
- **Membership:** is this Decision Problem, computable?
-- **Yes.** It is decidable for all classes of **Monotone Grammars.**
- The solution consists in defining a procedure (**The Parser Core**)
able to construct a derivation $s \Rightarrow \gamma_1 \Rightarrow \dots \Rightarrow \gamma_k \equiv w$, if one exists.

Construction of a Derivation

- The solution consists in defining a procedure (**The Parser Core**) able to construct a derivation $s \Rightarrow \gamma_1 \Rightarrow \dots \Rightarrow \gamma_k \equiv w$, if one exists.
- The construction of a derivation could be done in a non-efficient way, and even worse, at a non-linear, up to exponential, complexity time (/space) cost.

Trying p optional productions at each γ_i leads to:



construction of (exponential) ($O(p^n)$) derivations to find the one right or to answer “no-accept”.

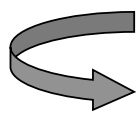
Top-Down

Simple for Handmade Constructions,
Few Grammars

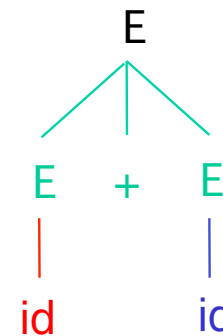
$E \Rightarrow E+E \Rightarrow id +E \Rightarrow id+id$

Leftmost non-terminal of Left-Sentential-Form

First Applicable Production



Failure: Backward to the last alternative



Step 1
Step 2
Step 3

Top-Down = Leftmost

$p1: E ::= E+E$
 $p2: E ::= E * E$
 $p3: E ::= id$

LSF forms a Complete Base for Context-Free Grammars

$G = \langle V, \Sigma, s \in V, P \rangle$

Left Sentential Form (of G): LSF_G

$\alpha\beta\gamma \in LSF_G \quad \text{iff} \quad s \Rightarrow^+ \alpha\beta\gamma$

$\alpha A \beta \Rightarrow \alpha \gamma \beta \quad \text{iff} \quad A ::= \gamma \in P \ \& \ \alpha \in \Sigma^*$

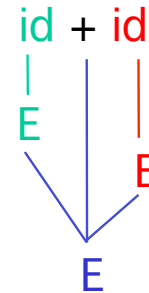
Only LSF_G

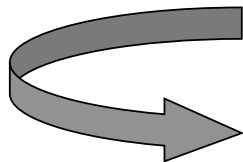
$L(G) = \{w \in \Sigma^* \mid s \Rightarrow^+ w\}$
 $= \{w \in \Sigma^* \mid s \Rightarrow^+ w\}$

Bottom-Up

More Complicated Techniques
Many More Grammars - Many More Languages

$E \Rightarrow E+E \Rightarrow E+id \Rightarrow id+id$



 Looking for Handle
reduction
failure: backward for "true" Handle

Step 1
Step 2
Step 3

Bottom-Up = Rightmost Reversed

$p1: E ::= E+E$
 $p2: E ::= E * E$
 $p3: E ::= id$

RSF forms a Complete Base for Context-Free Grammars

$G = \langle V, \Sigma, s \in V, P \rangle$ Right Sentential Form (of G): RSF_G

$\alpha\beta\gamma \in RSF_G$ iff $s \xrightarrow{r} \alpha\beta\gamma$

$\alpha A \beta \xrightarrow{r} \alpha \gamma \beta$ iff $A ::= \gamma \in P$ & $\beta \in \Sigma^*$

Only RSF

$L(G) = \{w \in \Sigma^* \mid s \Rightarrow^+ w\}$
 $= \{w \in \Sigma^* \mid s \xrightarrow{r} w\}$

$B ::= \beta \in P$ is **Handle** of $\alpha\beta\gamma \in RSF_G$
if and only if $\alpha B \gamma \in RSF_G$

20