Script **generated by TTT**

Title: Petter: Compiler Construction (07.05.2020)
- 10: Semantics of Grammars

Date: Fri Apr 24 17:24:01 CEST 2020

Duration: 24:20 min

Pages: 13

Pair of grammars:

E	$\rightarrow$	$E+E$ 0	|	$E*E$ 1	|	(E) 2	|	name 3	| int 4

E	$\rightarrow$	$E+T$ 0	|	T 1	
T	$\rightarrow$	$T*F$ 0	|	F 1	
F	$\rightarrow$	(E) 0	|	name 1	| int 2

Both grammars describe the same language

Derivation

Grammars are term rewriting systems. The rules offer feasible rewriting steps. A sequence of such rewriting steps $\alpha_0 \rightarrow \ldots \rightarrow \alpha_m$ is called derivation.

... for example:

$$\begin{aligned} \underline{E} &\rightarrow \underline{E} + T \\ &\rightarrow \underline{T} + T \\ &\rightarrow T * \underline{F} + T \\ &\rightarrow \underline{T} * \text{int} + T \\ &\rightarrow \underline{F} * \text{int} + T \\ &\rightarrow \text{name} * \text{int} + \underline{T} \\ &\rightarrow \text{name} * \text{int} + \underline{F} \\ &\rightarrow \text{name} * \text{int} + \text{int} \end{aligned}$$

Definition

The rewriting relation $\rightarrow$ is a relation on words over $N \cup T$, with

$$\alpha \rightarrow \alpha' \quad \text{iff} \quad \alpha = \alpha_1\ A\ \alpha_2 \ \wedge\ \alpha' = \alpha_1\ \beta\ \alpha_2 \ \text{ for an } \ A \rightarrow \beta \in P$$

Derivation

Remarks:

- The relation $\rightarrow$ depends on the grammar
- In each step of a derivation, we may choose:
  - a spot, determining where we will rewrite.
  - a rule, determining how we will rewrite.
- The language, specified by G is:

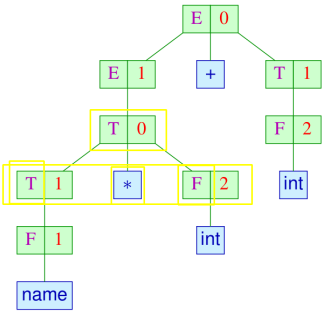
$$\mathcal{L}(G) = \{w \in T^* \mid S \rightarrow^* w\}$$

Derivation Tree

Derivations of a symbol are represented as derivation trees:

... for example:

$$
\begin{aligned}
\underline{E} &\to^0 \underline{E} + T \\
&\to^1 \underline{T} + T \\
&\to^0 T * \underline{F} + T \\
&\to^2 \underline{T} * \text{int} + T \\
&\to^1 \underline{F} * \text{int} + T \\
&\to^1 \text{name} * \text{int} + \underline{T} \\
&\to^1 \text{name} * \text{int} + \underline{F} \\
&\to^2 \text{name} * \text{int} + \text{int}
\end{aligned}
$$



A derivation tree for $A \in N$:
inner nodes: rule applications
root: rule application for A
leaves: terminals or ϵ
The successors of (B, i) correspond to right hand sides of the rule

Special Derivations

Attention:
In contrast to arbitrary derivations, we find special ones, always rewriting the leftmost (or rather rightmost) occurance of a nonterminal.

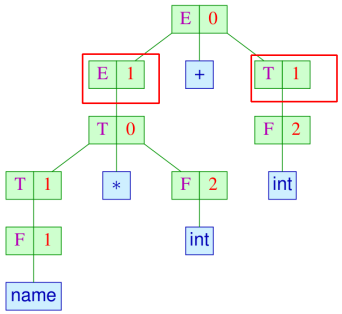
- These are called leftmost (or rather rightmost) derivations and are denoted with the index L (or R respectively).
- Leftmost (or rightmost) derivations correspondt to a left-to-right (or right-to-left) preorder-DFS-traversal of the derivation tree.
- Reverse rightmost derivations correspond to a left-to-right postorder-DFS-traversal of the derivation tree

Derivation Tree

Derivations of a symbol are represented as derivation trees:

... for example:

$$
\begin{aligned}
\underline{E} &\to^0 \underline{E} + T \\
&\to^1 \underline{T} + T \\
&\to^0 T * \underline{F} + T \\
&\to^2 \underline{T} * \text{int} + T \\
&\to^1 \underline{F} * \text{int} + T \\
&\to^1 \text{name} * \text{int} + \underline{T} \\
&\to^1 \text{name} * \text{int} + \underline{F} \\
&\to^2 \text{name} * \text{int} + \text{int}
\end{aligned}
$$



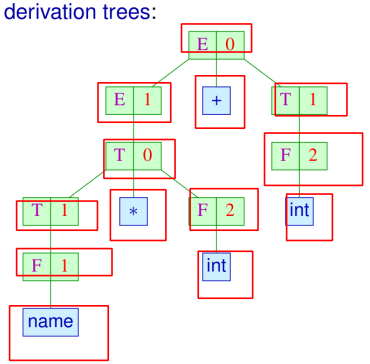
A derivation tree for $A \in N$:
inner nodes: rule applications
root: rule application for A
leaves: terminals or ϵ
The successors of (B, i) correspond to right hand sides of the rule

Derivation Tree

Derivations of a symbol are represented as derivation trees:

... for example:

$$
\begin{aligned}
\underline{E} &\to^0 \underline{E} + T \\
&\to^1 \underline{T} + T \\
&\to^0 T * \underline{F} + T \\
&\to^2 \underline{T} * \text{int} + T \\
&\to^1 \underline{F} * \text{int} + T \\
&\to^1 \text{name} * \text{int} + \underline{T} \\
&\to^1 \text{name} * \text{int} + \underline{F} \\
&\to^2 \text{name} * \text{int} + \text{int}
\end{aligned}
$$



A derivation tree for $A \in N$:
inner nodes: rule applications
root: rule application for A
leaves: terminals or ϵ
The successors of (B, i) correspond to right hand sides of the rule

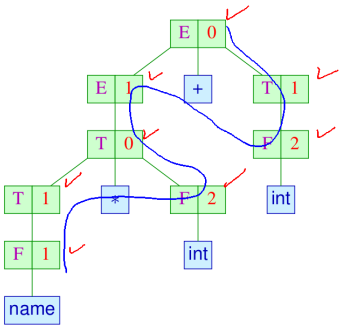
Special Derivations

Attention:
In contrast to arbitrary derivations, we find special ones, always rewriting the leftmost (or rather rightmost) occurance of a nonterminal.

- These are called leftmost (or rather rightmost) derivations and are denoted with the index L (or R respectively).
- Leftmost (or rightmost) derivations correspondt to a left-to-right (or right-to-left) preorder-DFS-traversal of the derivation tree.
- Reverse rightmost derivations correspond to a left-to-right postorder-DFS-traversal of the derivation tree

Special Derivations

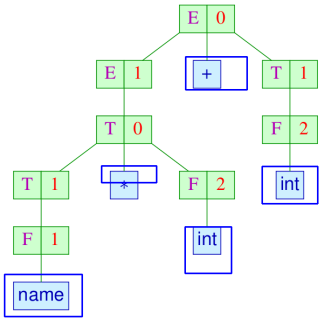
... for example:



Unique Grammars

The concatenation of leaves of a derivation tree t are often called $\text{yield}(t)$.

... for example:



gives rise to the concatenation: $\texttt{name} * \texttt{int} + \texttt{int}$.

Unique Grammars

Definition:
Grammar G is called unique, if for every $w \in T^*$ there is maximally one derivation tree t of S with $\text{yield}(t) = w$.

... in our example:

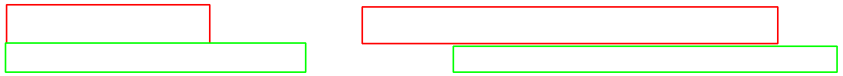
$$E \to E{+}E\,^0 \mid E{*}E\,^1 \mid (\ E\ )\,^2 \mid \texttt{name}\,^3 \mid \texttt{int}\,^4$$

$$\begin{array}{lll} E & \to & E{+}T\,^0 \mid T\,^1 \\ T & \to & T{*}F\,^0 \mid F\,^1 \\ F & \to & (\ E\ )\,^0 \mid \texttt{name}\,^1 \mid \texttt{int}\,^2 \end{array}$$

The first one is ambiguous, the second one is unique

Conclusion:

- A derivation tree represents a possible hierarchical structure of a word.
- For programming languages, only those grammars with a unique structure are of interest.
- Derivation trees are one-to-one corresponding with leftmost derivations as well as (reverse) rightmost derivations.



Chapter 2:
Basics of Pushdown Automata