



planetmath.org

Math for the people, by the people.

derivation tree of a derivation

Canonical name	DerivationTreeOfADerivation
Date of creation	2013-03-22 19:00:25
Last modified on	2013-03-22 19:00:25
Owner	CWoo (3771)
Last modified by	CWoo (3771)
Numerical id	9
Author	CWoo (3771)
Entry type	Definition
Classification	msc 68Q42
Classification	msc 68Q45

Constructing a Derivation from a Derivation Tree

Let G be a context-free grammar. Given a (finite) derivation tree T of G , we can construct a derivation D of G . There are two ways of doing this: top-down, or bottom-up.

The idea with the top-down approach is to build the derivation one derivation step at a time, starting from the root of the tree. During each step, a node with children is picked so that the next derivation step is formed, thus creating a new derivation with the new derivation step attached. Do this until all nodes of the tree are used.

To facilitate the discussion, for each node n of T , denote $[n]$ the label of n .

The top-down approach works as follows:

1. Set $X_0 = Y_0 =: \{r\}$, and D_0 the derivation $\sigma = [r]$, where r is the root of T .
2. Suppose X_i, Y_i and D_i are defined for $i \geq 0$, and $Y_i = \{n_1, \dots, n_k\}$ with $n_1 < \dots < n_k$. It is easy to see that D_i is the derivation $\sigma \Rightarrow^* [n_1] \cdots [n_k]$.

If none of the elements of Y have children, then set $D := D_i$. Otherwise, pick a node $n_j \in Y_i$ with children n'_1, \dots, n'_m such that $n'_1 < \dots < n'_m$. Then set

- $X_{i+1} := X_i \cup \{n_j\}$,
- $Y_{i+1} := (Y_i - \{n_j\}) \cup \{n'_1, \dots, n'_m\}$, and
- $D_{i+1} := D_i \Rightarrow [n_1] \cdots [n_{j-1}][n'_1] \cdots [n'_m][n_{j+1}] \cdots [n_k]$.

Since T is finite, D can be constructed in a finite number of steps above.

Definition. A derivation tree T is said to *correspond to* a derivation D if D can be constructed from T from the top-down.

From the construction above, one sees that every derivation tree corresponds to at least one derivation. But it may correspond to more than one derivation. If, in the second step of the construction above, one always picks the smallest node with children in each step, then the corresponding derivation is the leftmost derivation. Similarly, consistently picking the largest node in each step results in the rightmost derivation. This shows that a derivation tree T corresponds to a unique leftmost derivation (written $LD(T)$) as well as a unique rightmost derivation (written $RD(T)$).

The bottom-up approach starts out with the result u of the derivation tree T . The result corresponds to leaves of T . Partition the set of leaves into blocks, so that two leaves belonging to the same block if they have the same parent. Pick a block B of leaves, and form a new word u_1 from u by replacing the labels of leaves in B by the label of their parent. Then $u_1 \Rightarrow u$ is a derivation step. Next, form a new tree T_1 from T by removing the leaves of T in B . So u_1 is now the result of T_1 . Repeat the process just described, until all of the nodes are removed ($T_i = \emptyset$ for some i), which is possible since T is finite. What we end up with is a derivation. We will leave the formal detail to the reader.

Proposition 1. *A derivation can be constructed from the top-down iff it can be constructed from the bottom-up.*

Constructing a Derivation Tree from a Derivation

Conversely, given a derivation $D = \sigma \Rightarrow W_1 \Rightarrow \dots \Rightarrow W_n$, one can construct a derivation tree T such that D corresponds to T . The way this works is as follows: start with the tree with a single node whose label is σ . At each derivation step, additional nodes are added from the tree constructed so far, until the last derivation step is processed. The tree so constructed has a natural linear ordering, and hence is a derivation tree, which D corresponds to. This construction can be formalized as follows:

1. Defining nodes: (The nodes of the tree T will be tuples of positive integers)
 - (a) Set node (1) whose label is σ .
 - (b) Suppose nodes are defined for symbols occurring in W_i . Look at the derivation step $W_i \Rightarrow W_{i+1}$. Suppose it is based on production $P \rightarrow Q$, and (p_1, \dots, p_m) is the node whose label is P .
 - If $Q = N_1 \dots N_k$, where each $N_j \in \Sigma$, then define nodes

$$(p_1, \dots, p_m, 1), \dots, (p_1, \dots, p_m, k)$$

whose labels are N_1, \dots, N_k respectively.

- If $Q = \lambda$, then define one node $(p_1, \dots, p_m, 1)$ whose label is λ .

- (c) Continue (b) until nodes whose labels are symbols forming word W_n are defined.
- 2. Defining arrows: Given two nodes $p = (p_1, \dots, p_r)$ and $q = (q_1, \dots, q_s)$, an arrow is constructed from p to q iff $s = r + 1$ and $p_1 = q_1, \dots, p_r = q_r$.
- 3. Extending the partial order on T to a linear order based on the nodes: see the entry on ordered tree for detail.

The algorithm shows that the derivation tree is uniquely constructed.

Definition. Given a derivation D , the derivation tree of D , denoted by T_D , is the ordered tree constructed above from D .

It is easy to see that T_D corresponds to D .

Remark. Suppose u is a word derived by a derivation D . One can find the leftmost derivation D' of u using the methods provided above: first find T_D of D , then find $LD(T_D)$, and set $D' = LD(T_D)$.

References

- [1] H.R. Lewis, C.H. Papadimitriou, *Elements of the Theory of Computation*. Prentice-Hall, Englewood Cliffs, New Jersey (1981).
- [2] A. Salomaa, *Formal Languages*, Academic Press, New York (1973).
- [3] J.E. Hopcroft, J.D. Ullman, *Formal Languages and Their Relation to Automata*, Addison-Wesley, (1969).