LR(k)

Canonical name	LRk
Date of creation	2013-03-22 19:00:31
Last modified on	2013-03-22 19:00:31
Owner	CWoo (3771)
Last modified by	CWoo (3771)
Numerical id	11
Author	CWoo (3771)
Entry type	Definition
Classification	msc 03D10
Classification	msc 68Q42
Classification	msc 68Q05
Synonym	$LR(k)$
Related topic	LLk

Given a word u and a context-free grammar G, how do we determine if $u \in L(G)$?

One way is to look for any subword v of u such that there is a production $A \to v$. If this is successful, we may replace v with A in u to obtain a word w, so that $w \Rightarrow u$. We may then repeat the process on w to obtain another word x such that $x \Rightarrow w$ (if successful). In the end, if everything works successfully, we arrive at the starting non-terminal symbol σ, and get a derivation $\sigma \Rightarrow^* u$ as a result, so that $u \in L(G)$. This procedure is known as the bottom-up parsing of the word u.

In general, unless one is very lucky, successfully finding a derivation $\sigma \Rightarrow^* u$ requires many trials and errors, since at each stage, for a given word u, there may be several words w such that $w \Rightarrow u$.

Nevertheless, there is a particular family of context-free grammars, called the $LR(k)$ grammars, which make the bottom-up parsing described above straightforward in the sense that, given a word u, once a word w is found such that $w \Rightarrow_R u$, any other word w' such that $w' \Rightarrow_R u$ forces $w' = w$. Here, $\Rightarrow_R$ is known as the rightmost derivation (meaning that u is obtained from w by replacing the rightmost non-terminal in w). The L in $LR(k)$ means scanning the symbols of u from left to right, R stands for finding a rightmost derivation for u, and k means having the allowance to look at up to k symbols ahead while scanning.

The details are as follows:

Definition. Let $G = (\Sigma, N, P, \sigma)$ be a context-free grammar such that $\sigma \to \sigma$ is not a production of G, and $k \geq 0$ an integer. Suppose U is any sentential form over Σ with the following setup: $U = U_1U_2U_3$ where

- U_3 is a terminal word,
- $X \to U_2$ a production, and
- $\sigma \Rightarrow_R^* U_1XU_3 \Rightarrow_R U$.

Let $n = |U_1U_2| + k$, and Z the prefix of U of length n (if $|U| < n$, then set $Z = U$).

Then G is said to be $LR(k)$ if W is another sentential form having Z as a prefix, with the following setup: $W = W_1W_2W_3$, where

- W_3 is a terminal,
- $Y \to W_2$ is a production, and

- $\sigma \Rightarrow_R^* W_1 Y W_3 \Rightarrow_R W$,

implies that

$$W_1 = U_1, \qquad Y = X, \qquad \text{and} \qquad W_2 = U_2.$$

Simply put, if D_U and D_W are the rightmost derivations of U and W respectively, and if the prefix of U obtained by including k symbols beyond the last replacement in D_U is also a prefix of W, then the prefix of U' obtained by including k symbols beyond the last replacement in D_U is also a prefix of W', where U' and W' are words at the next to the last step in D_U and D_W respectively. In particular, if $U = W$, then $U' = W'$. This implies that any derivable in an $LR(k)$ grammar has a unique rightmost derivation, hence

Proposition 1. *Any $LR(k)$ grammar is unambiguous.*

Examples.

- Let G be the grammar consisting of one non-terminal symbol σ (which is also the final non-terminal symbol), two terminal symbols a, b, with productions

  $$\sigma \to a\sigma b, \qquad \sigma \to \sigma b \qquad \text{and} \qquad \sigma \to b.$$

  Then G is not $LR(k)$ for any $k \geq 0$. For instance, look at the following two derivations of $U = a^2\sigma b^3$:

  $$\sigma \Rightarrow^* a\sigma b^2 \Rightarrow a^2\sigma b^3 \qquad \text{and} \qquad \sigma \Rightarrow^* a^2\sigma b^2 \Rightarrow a^2\sigma b^3$$

  Here, $U_1 = a$, $U_2 = \sigma b$. Let $k = 1$. Then the criteria in the definition are satisfied. Yet, $W_1 = a^2 \neq U_1$. Therefore, G is not $LR(1)$.

- Note that the grammar G above generates the language $L = \{a^m b^n \mid n > m\}$, which can also be generated by the grammar with three non-terminal symbols σ, X, Y, with σ the final non-terminal symbol, where the productions are given by

  $$\sigma \to XY, \quad X \to aXb, \quad X \to \lambda, \quad Y \to Yb, \quad \text{and} \quad Y \to b.$$

  However, this grammar is $LR(1)$.

Determining whether a context-free grammar is $LR(k)$ is a non-trivial problem. Nevertheless, an algorithm exists for determining, given a context-free grammar G and a non-negative integer k, whether G is $LR(k)$. On the other hand, without specifying k in advance, no algorithms exist that determine if G is $LR(k)$ for *some* k.

Definition. A language is said to be $LR(k)$ if it can be generated by an $LR(k)$ grammar.

Theorem 1. *Every $LR(k)$ language is deterministic context-free. Every deterministic context-free language is $LR(1)$.*

Hence, deterministic context-free languages are the same as $LR(1)$ languages.

References

[1] A. Salomaa, *Formal Languages*, Academic Press, New York (1973).

[2] J.E. Hopcroft, J.D. Ullman, *Formal Languages and Their Relation to Automata*, Addison-Wesley, (1969).