



Linear grammar

In computer science, a **linear grammar** is a context-free grammar that has at most one nonterminal in the right-hand side of each of its productions.

A **linear language** is a language generated by some linear grammar.

Example

An example of a linear grammar is G with $N = \{S\}$, $\Sigma = \{a, b\}$, P with start symbol S and rules

$$S \rightarrow aSb$$

$$S \rightarrow \varepsilon$$

It generates the language $\{a^i b^i \mid i \geq 0\}$.

Relationship with regular grammars

Two special types of linear grammars are the following:

- the **left-linear** or left-regular grammars, in which all rules are of the form $A \rightarrow \alpha w$ where α is either empty or a single nonterminal and w is a string of terminals;
- the **right-linear** or right-regular grammars, in which all rules are of the form $A \rightarrow w\alpha$ where w is a string of terminals and α is either empty or a single nonterminal.

Each of these can describe exactly the regular languages. A regular grammar is a grammar that is left-linear or right-linear.

Observe that by inserting new nonterminals, any linear grammar can be replaced by an equivalent one where some of the rules are left-linear and some are right-linear. For instance, the rules of G above can be replaced with

$$S \rightarrow aA$$

$$A \rightarrow Sb$$

$$S \rightarrow \varepsilon$$

However, the requirement that *all* rules be left-linear (or all rules be right-linear) leads to a strict decrease in the expressive power of linear grammars.

Expressive power

All regular languages are linear; conversely, an example of a linear, non-regular language is $\{a^n b^n\}$, as explained above. All linear languages are context-free; conversely, an example of a context-free, non-

linear language is the Dyck language of well-balanced bracket pairs. Hence, the regular languages are a proper subset of the linear languages, which in turn are a proper subset of the context-free languages.

While regular languages are deterministic, there exist linear languages that are nondeterministic. For example, the language of even-length palindromes on the alphabet of 0 and 1 has the linear grammar $S \rightarrow 0S0 \mid 1S1 \mid \epsilon$. An arbitrary string of this language cannot be parsed without reading all its letters first which means that a pushdown automaton has to try alternative state transitions to accommodate for the different possible lengths of a semi-parsed string.^[1] This language is nondeterministic. Since nondeterministic context-free languages cannot be accepted in linear time, linear languages cannot be accepted in linear time in the general case. Furthermore, it is undecidable whether a given context-free language is a linear context-free language.^[2]

A language is linear iff it can be generated by a one-turn pushdown automaton – a pushdown automaton that, once it starts popping, never pushes again.

Closure properties

Positive cases

Linear languages are closed under union. Construction is the same as the construction for the union of context-free languages. Let L_1, L_2 be two linear languages, then $L_1 \cup L_2$ is constructed by a linear grammar with $S \rightarrow S_1 \mid S_2$, and S_1, S_2 playing the role of the linear grammars for L_1, L_2 .

If L is a linear language and M is a regular language, then the intersection $L \cap M$ is again a linear language; in other words, the linear languages are closed under intersection with regular sets.

Linear languages are closed under homomorphism and inverse homomorphism.^[3]

As a corollary, linear languages form a full trio. Full trios in general are language families that enjoy a couple of other desirable mathematical properties.

Negative cases

Linear languages are not closed under intersection. For example, let $L_1 = \{a^n b^n c^m \mid n, m \geq 0\}$, $L_2 = \{a^n b^m c^m \mid n, m \geq 0\}$, then their intersection is not only not linear, but also not context-free. See pumping lemma for context-free languages.

As a corollary, linear languages are not closed under complement (as intersection can be constructed by de Morgan's laws out of union and complement).

References

1. Hopcroft, John; Rajeev Motwani; Jeffrey Ullman (2001). *Introduction to automata theory, languages, and computation 2nd edition*. Addison-Wesley. pp. 249–253.
2. Greibach, Sheila (October 1966). "The Unsolvability of the Recognition of Linear Context-Free Languages" (<https://doi.org/10.1145%2F321356.321365>). *Journal of the ACM*. **13** (4): 582–587. doi:10.1145/321356.321365 (<https://doi.org/10.1145%2F321356.321365>)

21365). S2CID 37003419 (<https://api.semanticscholar.org/CorpusID:37003419>).

3. John E. Hopcroft and Jeffrey D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley Publishing, Reading Massachusetts, 1979.
ISBN 0-201-02988-X., Ex. 11.1, pp. 282f

Retrieved from "https://en.wikipedia.org/w/index.php?title=Linear_grammar&oldid=1152703213"