WIKIPEDIA
The Free Encyclopedia

Context-free language

In formal language theory, a **context-free language** (**CFL**), also called a **Chomsky type-2 language**, is a language generated by a context-free grammar (CFG).

Context-free languages have many applications in programming languages, in particular, most arithmetic expressions are generated by context-free grammars.

Background

Context-free grammar

Different context-free grammars can generate the same context-free language. Intrinsic properties of the language can be distinguished from extrinsic properties of a particular grammar by comparing multiple grammars that describe the language.

Automata

The set of all context-free languages is identical to the set of languages accepted by pushdown automata, which makes these languages amenable to parsing. Further, for a given CFG, there is a direct way to produce a pushdown automaton for the grammar (and thereby the corresponding language), though going the other way (producing a grammar given an automaton) is not as direct.

Examples

An example context-free language is $L = \{a^n b^n : n \geq 1\}$, the language of all non-empty even-length strings, the entire first halves of which are a's, and the entire second halves of which are b's. L is generated by the grammar $S \to aSb \mid ab$. This language is not regular. It is accepted by the pushdown automaton $M = (\{q_0, q_1, q_f\}, \{a, b\}, \{a, z\}, \delta, q_0, z, \{q_f\})$ where δ is defined as follows:[note 1]

$$\begin{aligned}
\delta(q_0, a, z) &= (q_0, az) \\
\delta(q_0, a, a) &= (q_0, aa) \\
\delta(q_0, b, a) &= (q_1, \varepsilon) \\
\delta(q_1, b, a) &= (q_1, \varepsilon) \\
\delta(q_1, \varepsilon, z) &= (q_f, \varepsilon)
\end{aligned}$$

Unambiguous CFLs are a proper subset of all CFLs: there are inherently ambiguous CFLs. An example of an inherently ambiguous CFL is the union of $\{a^n b^m c^m d^n | n, m > 0\}$ with $\{a^n b^n c^m d^m | n, m > 0\}$. This set is context-free, since the union of two context-free languages is always context-free. But there is no way to unambiguously parse strings in the (non-context-free) subset $\{a^n b^n c^n d^n | n > 0\}$ which is the intersection of these two languages.[1]

Dyck language

The language of all properly matched parentheses is generated by the grammar $S \to SS \mid (S) \mid \varepsilon$.

Properties

Context-free parsing

The context-free nature of the language makes it simple to parse with a pushdown automaton.

Determining an instance of the membership problem; i.e. given a string w, determine whether $w \in L(G)$ where L is the language generated by a given grammar G; is also known as *recognition*. Context-free recognition for Chomsky normal form grammars was shown by Leslie G. Valiant to be reducible to boolean matrix multiplication, thus inheriting its complexity upper bound of $O(n^{2.3728596})$.[2][note 2] Conversely, Lillian Lee has shown $O(n^{3-\varepsilon})$ boolean matrix multiplication to be reducible to $O(n^{3-3\varepsilon})$ CFG parsing, thus establishing some kind of lower bound for the latter.[3]

Practical uses of context-free languages require also to produce a derivation tree that exhibits the structure that the grammar associates with the given string. The process of producing this tree is called *parsing*. Known parsers have a time complexity that is cubic in the size of the string that is parsed.

Formally, the set of all context-free languages is identical to the set of languages accepted by pushdown automata (PDA). Parser algorithms for context-free languages include the CYK algorithm and Earley's Algorithm.

A special subclass of context-free languages are the deterministic context-free languages which are defined as the set of languages accepted by a deterministic pushdown automaton and can be parsed by a LR(k) parser.[4]

See also parsing expression grammar as an alternative approach to grammar and parser.

Closure properties

The class of context-free languages is closed under the following operations. That is, if L and P are context-free languages, the following languages are context-free as well:

- the union $L \cup P$ of L and P[5]
- the reversal of L[6]
- the concatenation $L \cdot P$ of L and P[5]
- the Kleene star L^* of L[5]
- the image $\varphi(L)$ of L under a homomorphism φ[7]

- the image $\varphi^{-1}(L)$ of L under an inverse homomorphism φ^{-1}[8]
- the circular shift of L (the language $\{vu : uv \in L\}$)[9]
- the prefix closure of L (the set of all prefixes of strings from L)[10]
- the quotient L/R of L by a regular language R[11]

Nonclosure under intersection, complement, and difference

The context-free languages are not closed under intersection. This can be seen by taking the languages $A = \{a^n b^n c^m \mid m, n \geq 0\}$ and $B = \{a^m b^n c^n \mid m, n \geq 0\}$, which are both context-free.[note 3] Their intersection is $A \cap B = \{a^n b^n c^n \mid n \geq 0\}$, which can be shown to be non-context-free by the pumping lemma for context-free languages. As a consequence, context-free languages cannot be closed under complementation, as for any languages A and B, their intersection can be expressed by union and complement: $A \cap B = \overline{\overline{A} \cup \overline{B}}$. In particular, context-free language cannot be closed under difference, since complement can be expressed by difference: $\overline{L} = \Sigma^* \setminus L$.[12]

However, if L is a context-free language and D is a regular language then both their intersection $L \cap D$ and their difference $L \setminus D$ are context-free languages.[13]

Decidability

In formal language theory, questions about regular languages are usually decidable, but ones about context-free languages are often not. It is decidable whether such a language is finite, but not whether it contains every possible string, is regular, is unambiguous, or is equivalent to a language with a different grammar.

The following problems are undecidable for arbitrarily given context-free grammars A and B:

- Equivalence: is $L(A) = L(B)$?[14]
- Disjointness: is $L(A) \cap L(B) = \emptyset$?[15] However, the intersection of a context-free language and a *regular* language is context-free,[16][17] hence the variant of the problem where B is a regular grammar is decidable (see "Emptiness" below).
- Containment: is $L(A) \subseteq L(B)$?[18] Again, the variant of the problem where B is a regular grammar is decidable, while that where A is regular is generally not.[19]
- Universality: is $L(A) = \Sigma^*$?[20]
- Regularity: is $L(A)$ a regular language?[21]
- Ambiguity: is every grammar for $L(A)$ ambiguous?[22]

The following problems are *decidable* for arbitrary context-free languages:

- Emptiness: Given a context-free grammar A, is $L(A) = \emptyset$?[23]
- Finiteness: Given a context-free grammar A, is $L(A)$ finite?[24]
- Membership: Given a context-free grammar G, and a word w, does $w \in L(G)$? Efficient polynomial-time algorithms for the membership problem are the CYK algorithm and Earley's Algorithm.

According to Hopcroft, Motwani, Ullman (2003),[25] many of the fundamental closure and (un)decidability properties of context-free languages were shown in the 1961 paper of Bar-Hillel, Perles, and Shamir[26]

Languages that are not context-free

The set $\{a^n b^n c^n d^n | n > 0\}$ is a context-sensitive language, but there does not exist a context-free grammar generating this language.[27] So there exist context-sensitive languages which are not context-free. To prove that a given language is not context-free, one may employ the pumping lemma for context-free languages[26] or a number of other methods, such as Ogden's lemma or Parikh's theorem.[28]

Notes

1. meaning of δ's arguments and results: $\delta(\mathbf{state}_1, \mathbf{read}, \mathbf{pop}) = (\mathbf{state}_2, \mathbf{push})$
2. In Valiant's paper, $O(n^{2.81})$ was the then-best known upper bound. See Matrix multiplication#Computational complexity for bound improvements since then.
3. A context-free grammar for the language *A* is given by the following production rules, taking *S* as the start symbol: $S \to Sc \mid aTb \mid \varepsilon$; $T \to aTb \mid \varepsilon$. The grammar for *B* is analogous.

References

1. Hopcroft & Ullman 1979, p. 100, Theorem 4.7.
2. Valiant, Leslie G. (April 1975). "General context-free recognition in less than cubic time" (https://figshare.com/articles/journal_contribution/General_context-free_recognition_in_less_than_cubic_time/6605915/1/files/12096398.pdf) (PDF). *Journal of Computer and System Sciences*. **10** (2): 308–315. doi:10.1016/s0022-0000(75)80046-8 (https://doi.org/10.1016%2Fs0022-0000%2875%2980046-8).
3. Lee, Lillian (January 2002). "Fast Context-Free Grammar Parsing Requires Fast Boolean Matrix Multiplication" (http://www.cs.cornell.edu/home/llee/papers/bmmcfl-jacm.pdf) (PDF). *J ACM*. **49** (1): 1–15. arXiv:cs/0112018 (https://arxiv.org/abs/cs/0112018). doi:10.1145/505241.505242 (https://doi.org/10.1145%2F505241.505242). S2CID 1243491 (https://api.semanticscholar.org/CorpusID:1243491). Archived (https://web.archive.org/web/20030427152836/http://www.cs.cornell.edu/home/llee/papers/bmmcfl-jacm.pdf) (PDF) from the original on 2003-04-27.
4. Knuth, D. E. (July 1965). "On the translation of languages from left to right". *Information and Control*. **8** (6): 607–639. doi:10.1016/S0019-9958(65)90426-2 (https://doi.org/10.1016%2FS0019-9958%2865%2990426-2).
5. Hopcroft & Ullman 1979, p. 131, Corollary of Theorem 6.1.
6. Hopcroft & Ullman 1979, p. 142, Exercise 6.4d.
7. Hopcroft & Ullman 1979, p. 131-132, Corollary of Theorem 6.2.
8. Hopcroft & Ullman 1979, p. 132, Theorem 6.3.
9. Hopcroft & Ullman 1979, p. 142-144, Exercise 6.4c.
10. Hopcroft & Ullman 1979, p. 142, Exercise 6.4b.
11. Hopcroft & Ullman 1979, p. 142, Exercise 6.4a.

12. Stephen Scheinberg (1960). "Note on the Boolean Properties of Context Free Languages" (https://core.ac.uk/download/pdf/82210847.pdf) (PDF). *Information and Control*. **3** (4): 372–375. doi:10.1016/s0019-9958(60)90965-7 (https://doi.org/10.1016%2Fs0019-9958%2860%2990965-7). Archived (https://web.archive.org/web/20181126005901/https://core.ac.uk/download/pdf/82210847.pdf) (PDF) from the original on 2018-11-26.
13. Beigel, Richard; Gasarch, William. "A Proof that if L = L1 ∩ L2 where L1 is CFL and L2 is Regular then L is Context Free Which Does Not use PDA's" (http://www.cs.umd.edu/~gasarch/BLOGPAPERS/cfg.pdf) (PDF). *University of Maryland Department of Computer Science*. Archived (https://web.archive.org/web/20141212060332/http://www.cs.umd.edu/~gasarch/BLOGPAPERS/cfg.pdf) (PDF) from the original on 2014-12-12. Retrieved June 6, 2020.
14. Hopcroft & Ullman 1979, p. 203, Theorem 8.12(1).
15. Hopcroft & Ullman 1979, p. 202, Theorem 8.10.
16. Salomaa (1973), p. 59, Theorem 6.7
17. Hopcroft & Ullman 1979, p. 135, Theorem 6.5.
18. Hopcroft & Ullman 1979, p. 203, Theorem 8.12(2).
19. Hopcroft & Ullman 1979, p. 203, Theorem 8.12(4).
20. Hopcroft & Ullman 1979, p. 203, Theorem 8.11.
21. Hopcroft & Ullman 1979, p. 205, Theorem 8.15.
22. Hopcroft & Ullman 1979, p. 206, Theorem 8.16.
23. Hopcroft & Ullman 1979, p. 137, Theorem 6.6(a).
24. Hopcroft & Ullman 1979, p. 137, Theorem 6.6(b).
25. John E. Hopcroft; Rajeev Motwani; Jeffrey D. Ullman (2003). *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley. Here: Sect.7.6, p.304, and Sect.9.7, p.411
26. Yehoshua Bar-Hillel; Micha Asher Perles; Eli Shamir (1961). "On Formal Properties of Simple Phrase-Structure Grammars". *Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung*. **14** (2): 143–172.
27. Hopcroft & Ullman 1979.
28. "How to prove that a language is not context-free?" (https://cs.stackexchange.com/q/265).

Works cited

- Hopcroft, John E.; Ullman, Jeffrey D. (1979). *Introduction to Automata Theory, Languages, and Computation* (https://archive.org/details/introductiontoau00hopc) (1st ed.). Addison-Wesley. ISBN 9780201029888.
- Salomaa, Arto (1973). *Formal Languages*. ACM Monograph Series.

Further reading

- Autebert, Jean-Michel; Berstel, Jean; Boasson, Luc (1997). "Context-Free Languages and Push-Down Automata". In G. Rozenberg; A. Salomaa (eds.). *Handbook of Formal Languages* (http://www-igm.univ-mlv.fr/~berstel/Articles/1997CFLPDA.pdf) (PDF). Vol. 1. Springer-Verlag. pp. 111–174. Archived (https://web.archive.org/web/20110516030515/http://www-igm.univ-mlv.fr/%7Eberstel/Articles/1997CFLPDA.pdf) (PDF) from the original on 2011-05-16.
- Ginsburg, Seymour (1966). *The Mathematical Theory of Context-Free Languages*. New York, NY, USA: McGraw-Hill.
- Sipser, Michael (1997). "2: Context-Free Languages". *Introduction to the Theory of Computation* (https://archive.org/details/introductiontoth00sips). PWS Publishing. pp. 91–122. ISBN 0-534-94728-X.