Introduction to Program Synthesis (WS 2024/25)

Chapter 2.4 - Foundations (Program Representation: Grammar)

Dr. rer. nat. Roman Kalkreuth

Chair for AI Methodology (**AIM**), Department of Computer Science,
RWTH Aachen University, Germany

Computer Programs: Representations

Grammar-based representation

- **Grammar** $\rightarrow$ Set of rules commonly used to describe the syntax of sentences and expressions (in a certain language)
- **Context Free Grammar (CFG)** $\rightarrow$ Used to describe context-free languages (CFL)
  - CFL $\rightarrow$ many programming languages
  - Production rules can be applied to non-terminal symbols regardless of the context (i.e. surrounding symbols)
- CFG can be used to represent computer programs
  - CFG based representation models are used for program synthesis $\rightarrow$ **grammar-guided program synthesis**

Computer Programs: Representations
Grammar-based representation

Definition (Context Free Grammar)

A context free grammar (CFG) can be defined a four-tuple $\{N, \Sigma, P, S\}$:

- $N \rightarrow$ Finite set of non-terminal symbols or variables
- $\Sigma \rightarrow$ Finite set of terminals
- $P \rightarrow$ Set of production rules
  - Finite relation in $V \times (V \cup \Sigma)^\star$
- $S \in N \rightarrow$ Starting symbol

A grammar is context free if every production R is in form $R \rightarrow (V \cup \Sigma)^\star$ with $G \in V$

- Contains only substitution rules in which exactly one non-terminal symbol is always derived from an arbitrarily long sequence of non-terminal and terminal symbols

Computer Programs: Representations
Grammar-based representation

Example (Context-free Grammar)

```
1  S ⇒ < exp >
2  < exp > ⇒ < var > | < exp >< op >< exp >
3  < op > ⇒ + | − | ∗ | /
4  < var > ⇒ x
```

Listing: Simple context-free grammar

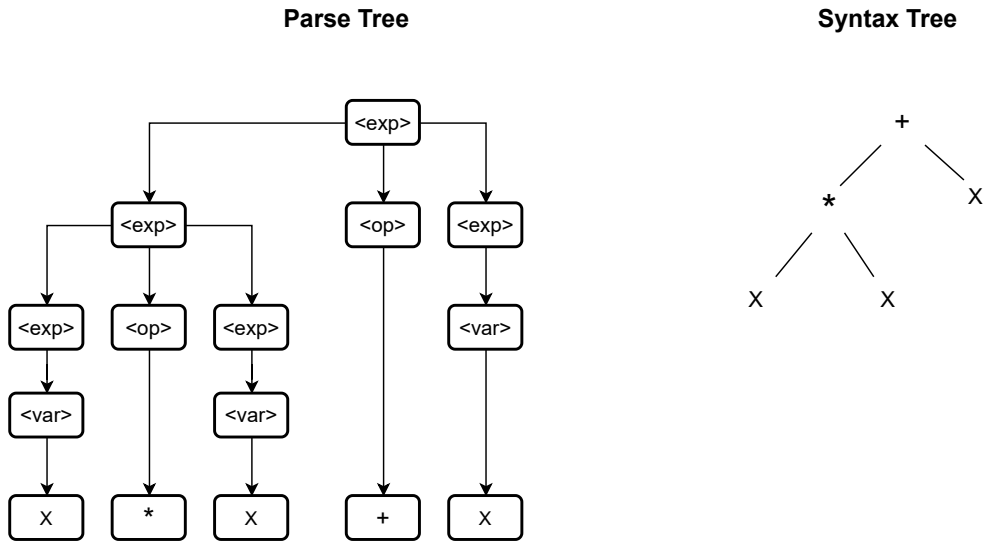

Figure: The expression $x * x + x$ that can be produced by the CFG

Computer Programs: Representations
Grammar-based representation

- Integer-based encoding of programs $\rightarrow$ Enables a simple and straight-forward way to variate programs
- **Grammatical Evolution** [RCO98] $\rightarrow$ Grammar-based program synthesis with artificial evolution mechanisms
- Encodes and decodes programs represented as grammar

Computer Programs: Representations

Grammar-based representation

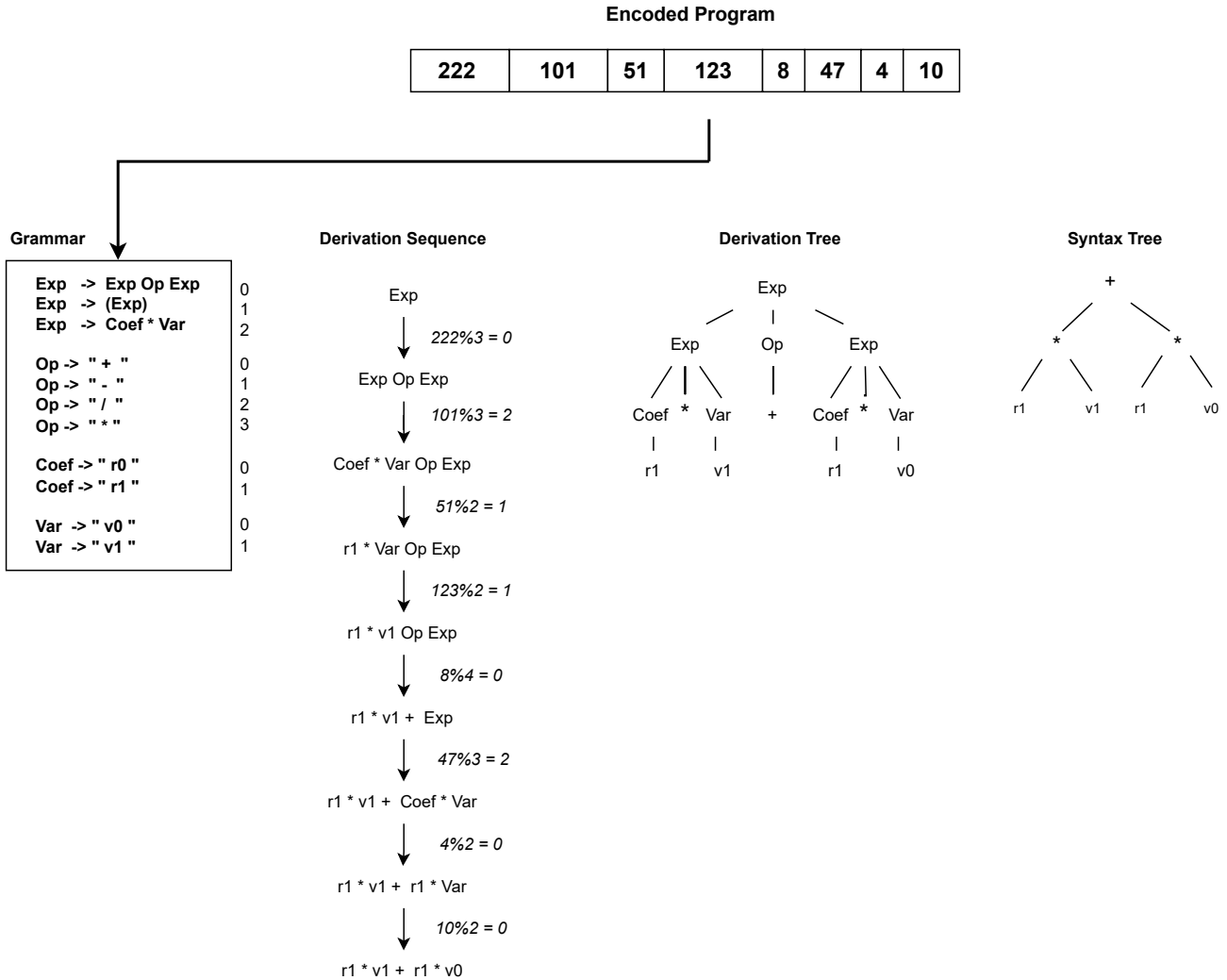

Figure: Grammar-based encoding of programs

References

[RCO98] Conor Ryan, J. J. Collins, and Michael O'Neill. "Grammatical Evolution: Evolving Programs for an Arbitrary Language". In: *Genetic Programming, First European Workshop, EuroGP'98, Paris, France, April 14-15, 1998, Proceedings*. Ed. by Wolfgang Banzhaf et al. Vol. 1391. Lecture Notes in Computer Science. Springer, 1998, pp. 83–96. DOI: 10.1007/BFB0055930. URL: https://doi.org/10.1007/BFb0055930.