Bahir Dar University
Institute of Technology /BiT

Principles of Compiler Design

Syntax Analysis – Individual Assignment

Jan 4, 2026
Yonatan Ayisheshim [BDU1508377]

Table of Contents

Introduction

Syntax analysis is a critical phase of compiler design that verifies whether a sequence of tokens generated by the lexical analyzer conforms to the grammatical structure of the programming language. This assignment focuses on understanding syntactic structures through grammar-based problem solving.

1. Problem Solving

1.1 Grammar Analysis and Parse Trees

Given Grammar: `S → aS | bS | ε`

Where:

- Terminals: `{a, b}`
- Non-terminal: `S`
- Start symbol: `S`
- `ε` denotes the empty string

Derivations:

1. **String "aa":** `S → aS → aaS → aaε = aa`
2. **String "ab":** `S → aS → abS → abε = ab`
3. **String "ba":** `S → bS → baS → baε = ba`
4. **String "bb":** `S → bS → bbS → bbε = bb`

The list of strings of length 2 is: `aa`, `ab`, `ba`, `bb`.

1.2 Parse Trees

Below are the textual representations of the parse trees for each generated string.

A. Parse Tree for "aa"

```
      S
     / \
    a   S
       / \
      a   S
          |
          ε
```

B. Parse Tree for "ab”

```
   S
  / \
 a   S
    / \
   b   S
       |
       ε
```

C. Parse Tree for "ba"

```
   S
  / \
 b   S
    / \
   a   S
       |
       ε
```

D. Parse Tree for "bb"

```
   S
  / \
 b   S
    / \
   b   S
       |
       ε
```

1.3 Observation

The grammar is **right-recursive** and generates **all possible strings** of `a` and `b`, including the empty string. Each character corresponds to a recursive expansion of `S` until termination via `ε`.

Conclusion

This assignment demonstrated grammar-driven string generation and parse tree construction

Understanding this concept is essential for mastering compiler front-end design and lays the foundation for advanced topics such as semantic analysis and code generation.

References

- Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. *Compilers: Principles, Techniques, and Tools* (2nd ed.)
- Compiler Design Lecture Notes