## Contents

**Top Down Processing** **1**

DBBCCDC . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

## Top Down Processing

- Considering the Grammar

1. `<s>` $\rightarrow$ D`<s><A>`
2. `<s>` $\rightarrow$ B`<A>`C
3. `<A>` $\rightarrow$ D`<A>`
4. `<A>` $\rightarrow$ C

- This is an S-Grammar
  - S stand for simple

### DBBCCDC

- `<s>` $\overset{1}{\Rightarrow}$ D`<s><A>`
- D`<s><A>` $\overset{2}{\Rightarrow}$ DB`<A>`C`<A>`
- DB`<A>`C`<A>` $\overset{4}{\Rightarrow}$ DBCC`<A>`
- DBCC`<A>` $\overset{3}{\Rightarrow}$ DBCCD`<A>`
- DBCCD`<A>`$\overset{4}{\Rightarrow}$ DBCCDC

The parse is deterministic. Can only be parsed correctly.

- Inputs = {B, C, D, $\dashv$}
- Stack Symbols = {`<s>`, `<A>`, C, $\triangledown$}
- Starting Stack = $\triangledown$S

| | B | C | D | $\dashv$ |
|---|---|---|---|---|
| `<s>` | #2 | | #1 | |
| `<A>` | | #4 | #3 | |
| C | | POP,ADVANCE | | |
| $\triangledown$ | | | | ACCEPT |

All blank non table entries represent reject

- `<A>` $\rightarrow$ B $\alpha$

1. REPLACE(`<A><s>`), ADVANCE
2. REPLACE(C`<A>`), ADVANCE
3. -REPLACE(A),- ADVANCE
4. POP, ADVANCE

At each step during the parse there is an assertion that the input is correct if and only if the string of remaining terminals can be derived from the sequence of symbols on the stack

| Stack | Inputs |
|---|---|
| $\nabla$`<s>` | DBCCBD$\dashv$ |
| $\nabla$`<A><s>` | BCCDC$\dashv$ |
| $\nabla$`<A>`C`<A>` | CCDC$\dashv$ |
| $\nabla$`<A>`C | CDC$\dashv$ |
| $\nabla$`<A>` | DC$\dashv$ |
| $\nabla$`<A>` | C$\dashv$ |
| $\nabla$ | $\dashv$ |

**ACCEPT**

| Past Inputs | Stack | String |
|---|---|---|
| | `<s>`$\nabla$ | `<s>` |
| D | `<s><A>`$\nabla$ | D`<s><A>` |
| DB | `<A>`C`<A>`$\nabla$ | DB`<A>`C`<A>` |
| DBC | C`<A>`$\nabla$ | DBCC`<A>` |
| DBCC | `<A>`$\nabla$ | DBCC`<A>` |
| DBCCDC | $\nabla$ | DBCCDC |

Stack from right to left

Intermediate string concatenation of the past inputs and the symbols on the stack