$\Sigma = \{\&, A, B, C\}$ $\quad L_{source} = \Sigma^*$

$\&\&A\&CB \longrightarrow \&A\ Cmd(C)\ B$

$AB\&\&\&BA\& \longrightarrow AB\&\ Cmd(B)\ A$

$\&$ is a quote of the next character

if $A, B, C$ are quoted then they have to be translated into commands $\&A \longrightarrow Cmd(A)$

if $\&$ quotes $\&$ then they must be translated as one $\&$

if $A, B, C$ are not quoted, they are translated as is

if there is a trailing single $\&$ it is not translated

1) Give an SDT suitable to be implemented during TOP-DOWN PARSING

and that computes as an attribute of the starting symbol the translated sequence.

Let's use an LL(1) grammar and then let's define the attributes

$S' \to \{ S.i = false \}\ S\ \{ S'.s = S.s \}$

$S \to L\ \{ S_1.i = \underline{if}\ (S.i)\ \underline{then}\ false\ \underline{else}\ \underline{if}\ (L.s == '\&')\ \underline{then}\ true\ \underline{else}\ false; \}\ S_1\ \{ S.s = \underline{if}\ (S.i)\ \underline{then}\ \underline{if}\ (L.s == '\&')\ \underline{then}\ '\&' \,||\, S_1.s\ \underline{else}\ 'Cmd('\,||\,L.s\,||\,')'\,||\,S_1.s\ \underline{else}\ \underline{if}\ (L.s == '\&')\ \underline{then}\ S_1.s\ \underline{else}\ L.s \,||\, S_1.s \}$

$S \to \varepsilon\ \{ S.s = '' \}$

$L \to \&\ \{ L.s = '\&' \}$

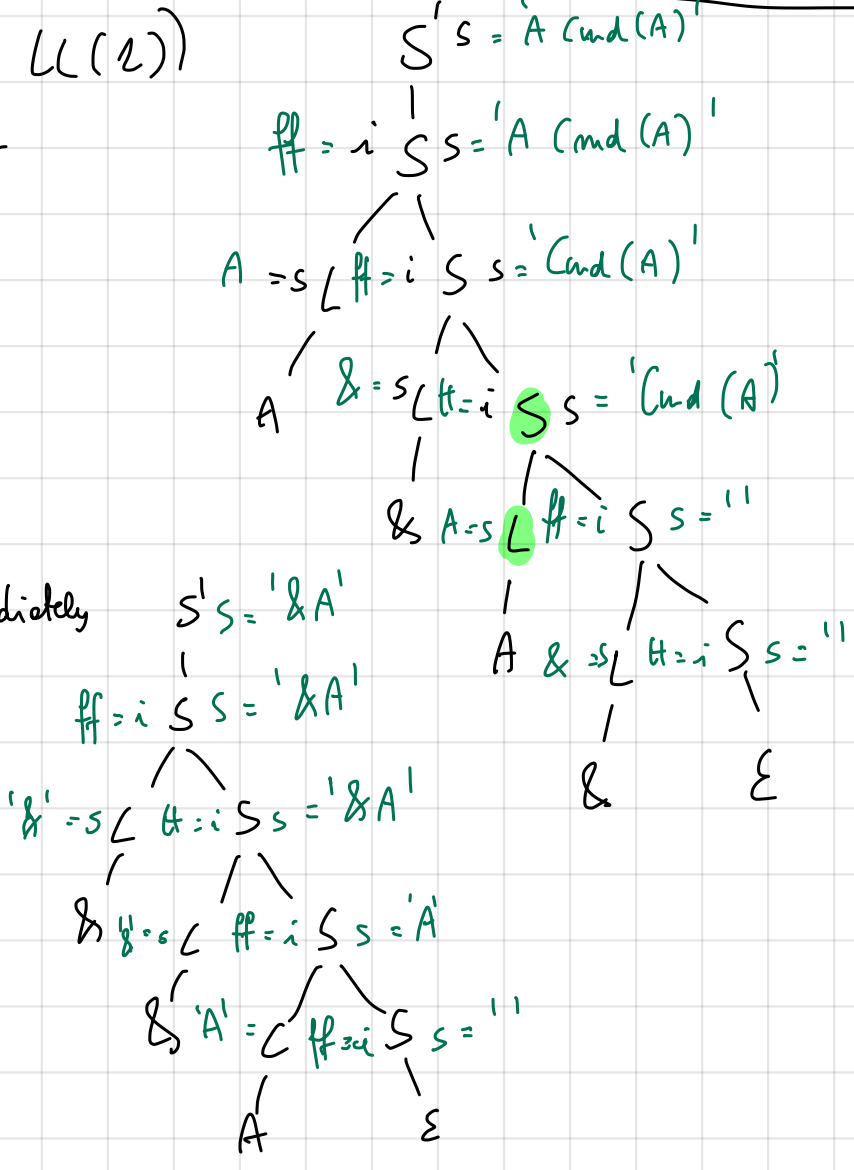
$L \to A\ \{ L.s = 'A' \}$

$L \to B\ \{ L.s = 'B' \}$

$L \to C\ \{ L.s = 'C' \}$

Attributes (it is LL(1))

- symbols S, S' and L have a synthesised attribute s of type string
- symbol S has an inherited attribute i of type boolean meaning "not immediately before there was an unquoted &"



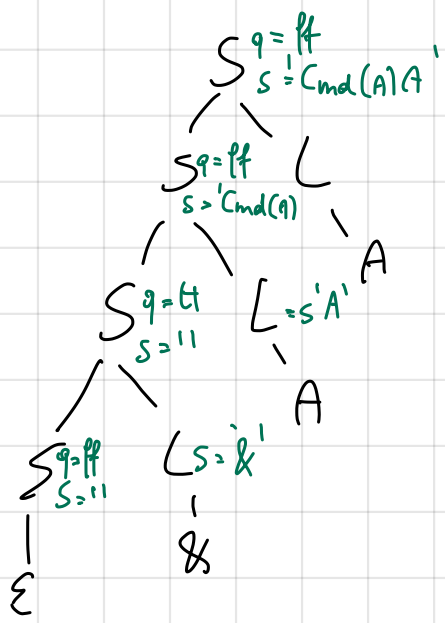
2) Do the same but suitable for being implemented during BOTTOM-UP parsing

- we can only use synthesised attributes
- the grammar must be LR(1)

$S' \to S$ (0)

$S \to SL$ (1) | $\varepsilon$ (2)

$L \to \& \mid A \mid B \mid C$ (3, 4, 5, 6)




FIRST(S) = {&, A, B, C, ε}

FIRST(L) = {&, A, B, C}

FOLLOW(S') = {$}

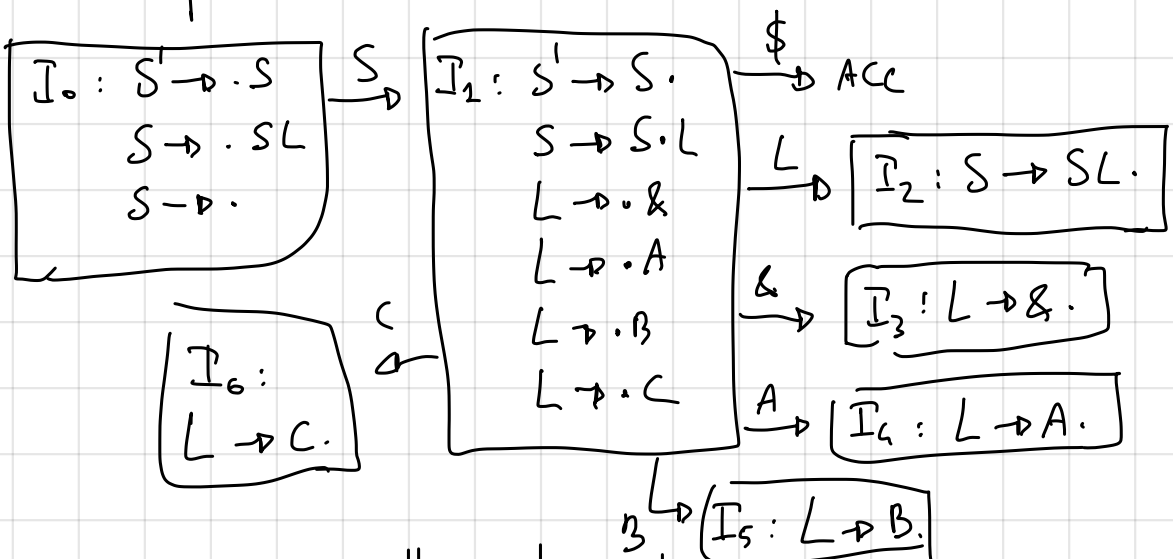
FOLLOW(S) = {$, &, A, B, C}

FOLLOW(L) = {$, &, A, B, C}

Attributes

s of type string for S, L

q of type boolean for S




| | & | A | B | C | $ | S | L |
|---|---|---|---|---|---|---|---|
| 0 | r2 | r2 | r2 | r2 | r2 | 1 | |
| 1 | s3 | s4 | s5 | s6 | ACC | | 2 |
| 2 | r1 | r1 | r1 | r1 | r1 | | |
| 3 | r3 | r3 | r3 | r3 | r3 | | |
| 4 | r4 | r4 | r4 | r4 | r4 | | |
| 5 | r5 | r5 | r5 | r5 | r5 | | |
| 6 | r6 | r6 | r6 | r6 | r6 | | |

→ the grammar is SLR(1) so it is LR(1)

$S' \to S \quad \{ S'.s = S.s \}$

$S \to S_1\ L$

$$\begin{cases} S.q = \text{if } (S_1.q) \text{ then false else } (\text{if } (L.s == '\&') \text{ then true else false}); \\ S.s = \text{if } (S_1.q) \text{ then } (\text{if } (L.s = '\&') \text{ then } S_1.s \,||\, '\&' \\ \quad \text{else } (S_1.s \,||\, 'Cmd(' \,||\, L.s \,||\, ')') \text{ else if } (L.s == '\&' \text{ then } S_1.s \text{ else } S_1.s \,||\, L.s); \end{cases}$$

$S \to \varepsilon \quad \{ S.q = ff ;\ S.s = '' \}$

$L \to \& \quad \{ L.s = '\&' \}$

$L \to A \quad \{ L.s = 'A' \}$

$L \to B \quad \{ L.s = 'B' \}$

$L \to C \quad \{ L.s = 'C' \}$