PROGRAMMING LANGUAGES CONCEPTS AND IMPLEMENTATION

(Mandatory Exercise, Week 6, Student: Irina Alina Gabriela Luca)

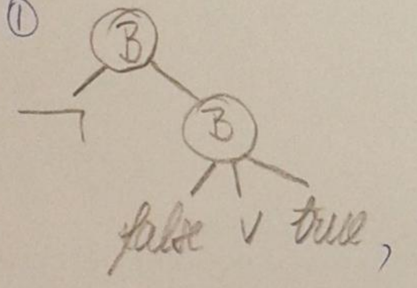
● THEORETICAL EXERCISE

Our ambiguous grammar is ⟹

$$
\begin{aligned}
B = \ & \text{true} \\
| \ & \text{false} \\
| \ & B \vee B \\
| \ & B \wedge B \\
| \ & \neg B \\
| \ & (B)
\end{aligned}
$$

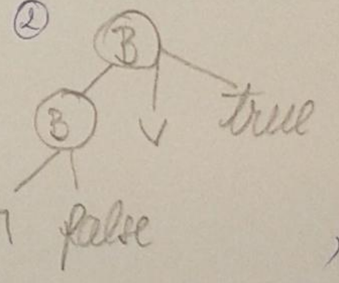
ⓐ The grammar is ambigious because for an expression like $\neg$ false $\vee$ true, for instance, we can create two parse trees, as it follows:

①



, interpreted as $\neg$(false $\vee$ true)

②



, interpreted as ($\neg$false) $\vee$ true

ⓑ In order to construct an unambiguous grammar from the above presented one, we have to take into account two aspects ⟹

⟹ - operators' precedence (see step 1)
- encode associativity (see step 2)

STEP 1 ⟹

$$
\begin{aligned}
B &= B \vee B \mid S \\
S &= S \wedge S \mid T \\
T &= \neg T \mid U \\
U &= (B) \mid \text{true} \mid \text{false}
\end{aligned}
$$

→ made sure operators' hierarchy is applied

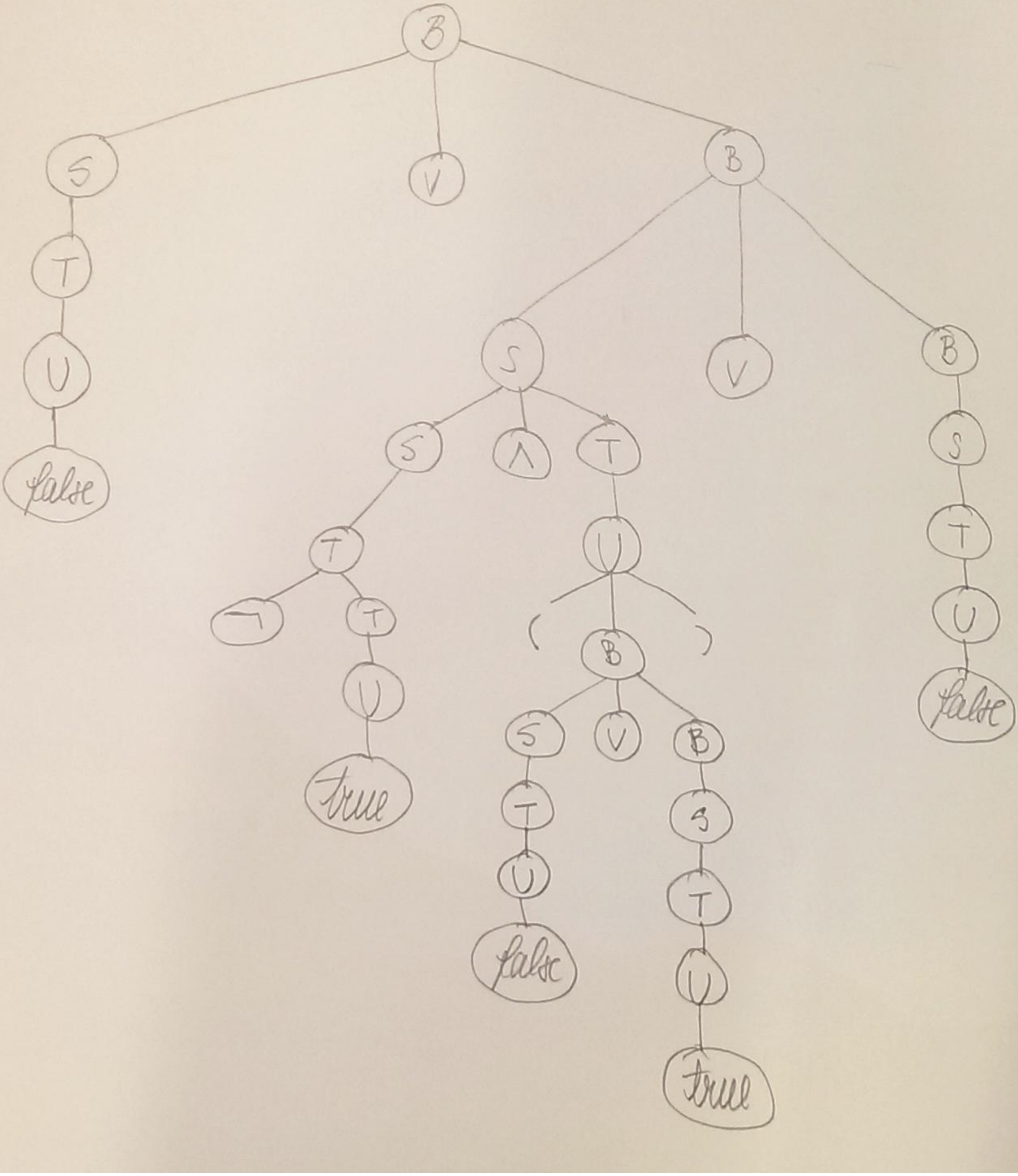
STEP 2 ⟹

$$
\begin{aligned}
B &= S \vee B \mid S \\
S &= S \wedge T \mid T \\
T &= \neg T \mid U \\
U &= (B) \mid \text{true} \mid \text{false}.
\end{aligned}
$$

→ encoded associativity, therefore obtaining an unambiguous form

ⓒ Considering the unambiguous grammar from ⓑ, we will construct the parse tree for the following expression:

$\boxed{false \lor \neg true \land (false \lor true) \lor false}$


B
S
∨
B
T
U
false
S
∨
B
S
∧
T
S
T
U
T
U
false
¬
T
(
B
)
U
true
S
∨
B
T
S
U
T
false
U
true