Solutions

2.1

- The production whose FIRST contains the lookahead (FOLLOW, for an ε production) ✓.

2.2

- Two productions land in one cell: one lookahead cannot choose – the grammar is not LL(1) ✓.

2.3

- The leftmost nonterminal expands to itself: the stack never shrinks and no input is consumed – an infinite loop ✓.

3.1

- Table: on lookahead id, S expands to id = E and E expands to T Ep and T expands to id; on +, Ep expands to + T Ep; at end of input, Ep expands to empty ✓.
- Snapshots, written stack | lookahead: S | id; then id = E | id (apply S); = E | = (match); E | id (match); T Ep | id (apply E); id Ep | id (apply T); Ep | + (match); + T Ep | + (apply Ep); T Ep | id (match); id Ep | id (apply T); Ep | dollar (match; apply Ep to empty) ✓.
- Eleven moves; accept (stack empty at end of input) ✓.

3.2

- Transformed: $A \to \mathbf{c}A'$; $A' \to \mathbf{a}A' \mid \mathbf{b}A' \mid \varepsilon$ ✓.
- Derivation: A, then c A-prime, then c a A-prime, then c a b A-prime, then c a b a A-prime, then c a b a ✓.
- Original language: **c** followed by any string of **a**'s and **b**'s – $\mathbf{c}(\mathbf{a} \mid \mathbf{b})^*$; the transformation generates exactly that ✓.
- Cost: right-recursive trees lean right; left associativity must be restored when building the AST ✓.

exercises.lol exercises.com.cn