## Solutions

**2.1**

- Six steps ✓.

**2.2**

- One string, two distinct parse trees ✓.

**2.3**

- Two independent counts; the CFG pumping argument handles only one ✓.

**3.1**

- The chain runs E, T, T x F, F x F, id x F, id x (E), id x (E + T), id x (T + T), id x (id + T), id x (id + id) – nine steps ✓.
- Each step is forced: the leftmost nonterminal with the only production whose FIRST matches the lookahead ✓.
- Tree: × at the root, left leaf id, right subtree the bracketed + – 5 leaves (**id**, ×, **id**, +, **id**) ✓.
- Verdict: the product is evaluated first – id x (id + id), not (id x id) + id ✓.

**3.2**

- Tree 1: root +, left id, right ×(id, id). Tree 2: root ×, left +(id, id), right id ✓.
- `id + id + id`: 2 trees (the two association orders) ✓.
- Danger: a code generator must pick one evaluation order – two trees mean two programs ✓.
- Fix: layer ($E/T/F$) so precedence is structural; the cost is a taller tree and re-association for left-recursive idioms ✓.



exercises.lol exercises.com.cn