Equivalence with context-free grammars I

- Language context free $\Leftrightarrow$ recognized by pushdown automata
- For the left-hand side, recall that by definition a language is context-free if it is constructed by some CFG
- For the proof, one direction is easier, while the other is harder
- As usual, we do the easier one first

CFL $\rightarrow$ PDA I

- Given a CFG, we find a PDA to simulate this grammar
- Two keys:

  stack

  nondeterminism: different substitutions
- We do the proof by an example
- Suppose we are given the following CFG

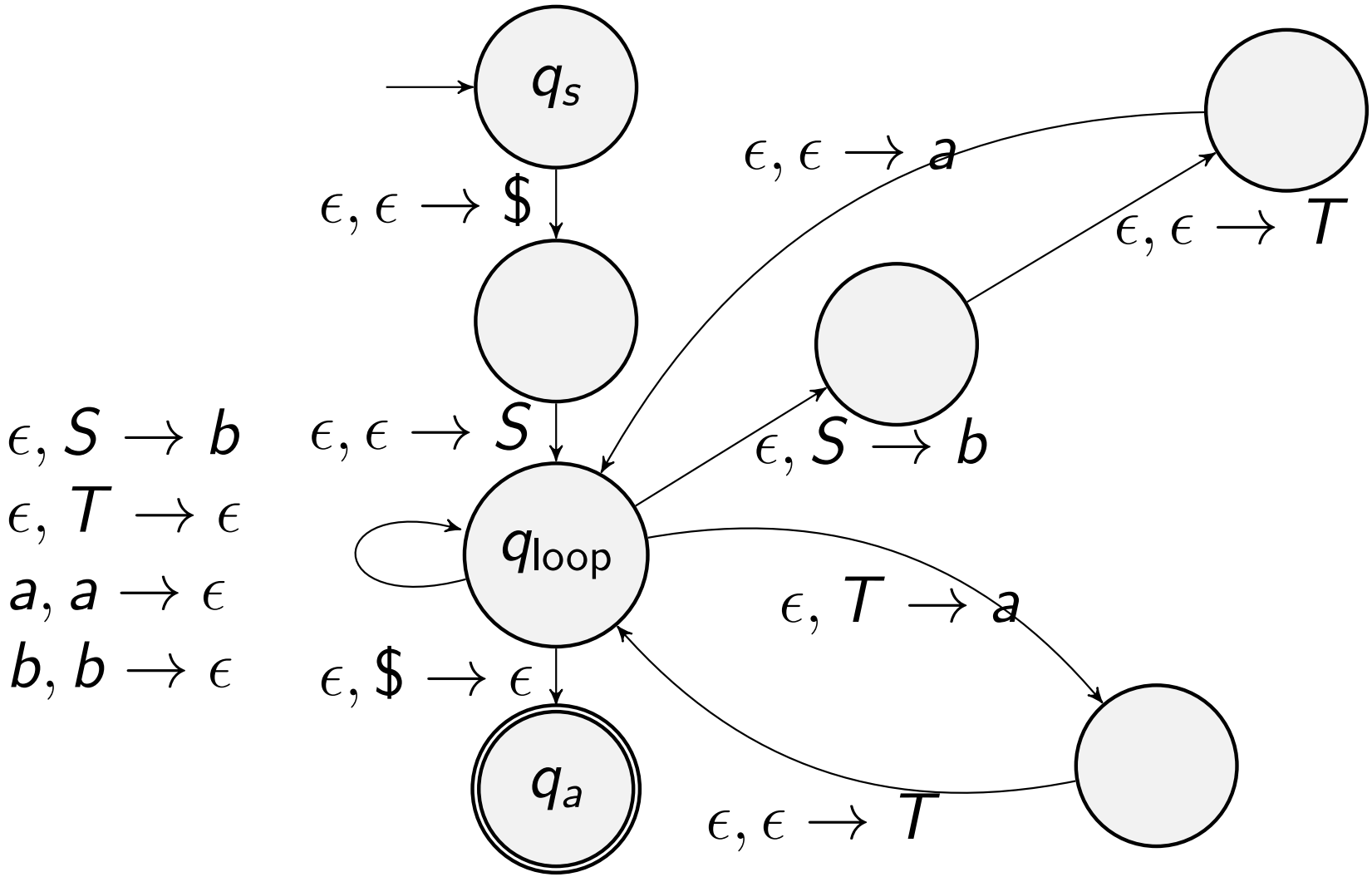
$$S \rightarrow aTb \mid b$$
$$T \rightarrow Ta \mid \epsilon$$

CFL $\rightarrow$ PDA II

- Idea: for rule substitution, push right-hand side to stack

  For example, aTb is pushed to stack in a reversed way
- A PDA can be as follows

CFL $\rightarrow$ PDA III



CFL $\rightarrow$ PDA IV

- Consider an example sequence *aaaab*

CFL $\rightarrow$ PDA V

$$q_{\text{start}} \xrightarrow{\epsilon} q_{\text{loop}}, \{S, \$\} \xrightarrow{\epsilon} q_1, \{b, \$\} \xrightarrow{\epsilon} q_2, \{T, b, \$\}$$
$$\xrightarrow{\epsilon} q_{\text{loop}}, \{a, T, b, \$\} \xrightarrow{a} q_{\text{loop}}, \{T, b, \$\}$$
$$\xrightarrow{\epsilon} q_3, \{a, b, \$\} \xrightarrow{\epsilon} q_{\text{loop}}, \{T, a, b, \$\}$$
$$\xrightarrow{\epsilon} q_3, \{a, a, b, \$\} \xrightarrow{\epsilon} q_{\text{loop}}, \{T, a, a, b, \$\}$$
$$\xrightarrow{\epsilon} q_3, \{a, a, a, b, \$\} \xrightarrow{\epsilon} q_{\text{loop}}, \{T, a, a, a, b, \$\}$$
$$\xrightarrow{\epsilon} q_{\text{loop}}, \{a, a, a, b, \$\} \xrightarrow{a} q_{\text{loop}}, \{a, a, b, \$\}$$
$$\xrightarrow{a} q_{\text{loop}}, \{a, b, \$\} \xrightarrow{a} q_{\text{loop}}, \{b, \$\}$$
$$\xrightarrow{b} q_{\text{loop}}, \{\$\} \xrightarrow{\epsilon} q_{accept}$$

CFL $\rightarrow$ PDA VI

- Even with a non-deterministic setting, we ensure that only strings generated by this CFG can be accepted by the PDA
  1. A string is accepted only if all characters are processed
  2. We have \$ to ensure that the stack is empty in the end