Student Information

Full Name: Berk Ulutaş
Id Number: 2522084

Answer 1

a)

G_1 represents a context-free grammar which generates n 0's followed by n 1's ($0^n 1^n$ where $n \geq 0$) or n 1's followed by n 0's ($1^n 0^n$ where $n \geq 0$).

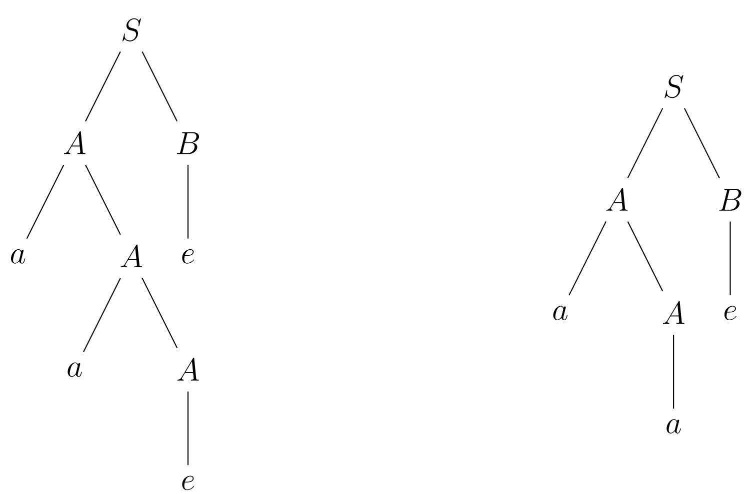
b)

Yes, there are 2 distinct parse trees for $e \in L(G_1)$. So, G_1 is ambiguous.

Answer 2

a)

There are two distinct parse trees for $aa \in L(G_2)$. So, G_2 is ambiguous.

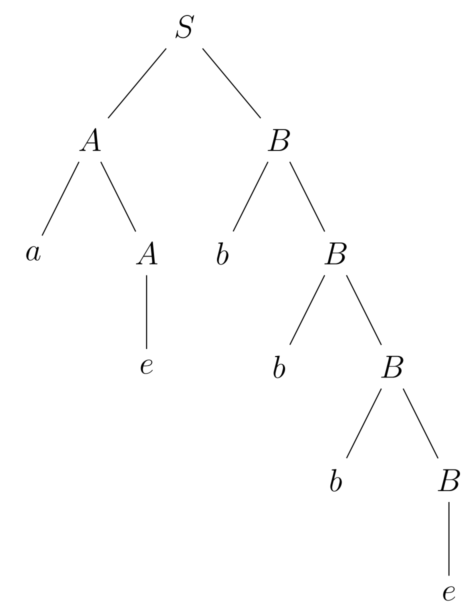


b)

$G_2 = \{V, \Sigma, R, S\}$ where $V = \{a, b, S, A, B\}$, $\Sigma = \{a, b\}$ and R:
$S \to AB$
$A \to aA|e$
$B \to bB|e$

c)

$$S \to AB \to aAB \to aB \to abB \to abbB \to abbbB \to abbb$$



Answer 3

a)

According to our textbook: "we call a language $L \subseteq \Sigma^*$ deterministic context-free if $L\$ = L(M)$ for some deterministic pushdown automaton M. Here \$ is a new symbol, not in Σ, which is appended to each input string for the purpose of marking its end." (page 159). So, for this question \$ will be represent end of string.

i)

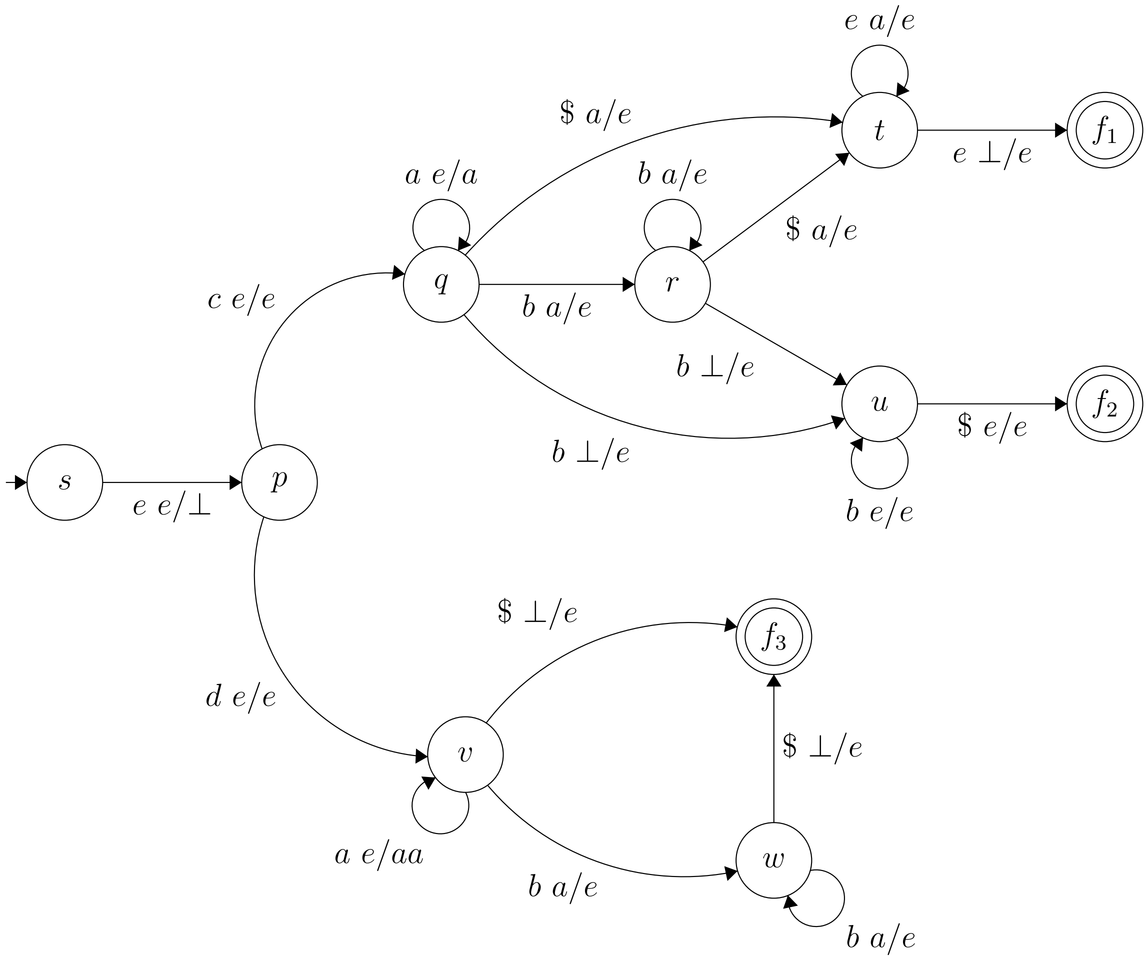
$$L_1 = \{ca^m b^n | m \neq n\} \cup \{da^m b^{2m} | m \geq 0\}$$

The language is deterministic context-free since it is accepted by the following DPDA.

- Start with pushing bottom of stack symbol the stack, and change state to p

- If machine reads c change state to q
  - For every a read from input push a to the stack.
  - If input completely read (read \$) and stack is not empty then change state to t, empty stack and accept.
  - If stack is empty (top of the stack is $\perp$) and input has not finished then change state to u finish reading input and accept.
  - If machine read all a's, there is remaining b's and stack is not empty. Read at least one b, change state to r. For every b read pop a from stack. If input completely read (read \$) and stack is not empty then change state to t, empty stack and accept. If stack is empty (top of the stack is $\perp$) and input has not finished then change state to u finish reading input and accept.
- If machine reads d change state to v
  - For every a read, push aa to the stack.
  - If input empty (read \$) and stack is empty (top of the stack $\perp$) then change state to f_3, and accept.
  - To process b's after reading first b and popping a, change state to w. For every b read, pop a from the stack. If input completely read (read \$) and stack is empty then change state to f_3 and accept.

$$M = \{K, \Sigma, \Gamma, s, F, \Delta\}$$
$$K = \{s, p, q, r, t, u, v, w, f_1, f_2, f_3\}$$
$$\Sigma = \{a, b, c, d, \$\}$$
$$\Gamma = \{a, \perp\}$$
$$F = \{f_1, f_2, f_3\}$$
$$\begin{aligned}\Delta = \{&((s,e,e),(p,\perp)),((p,c,e),(q,e)),\\ &((p,d,e),(v,e)),((q,a,e),(q,a)),\\ &((q,\$,a),(t,e)),((t,e,a),(t,e)),\\ &((t,e,\perp),(f_1,e)),((q,b,\perp),(u,e)),\\ &((u,b,e),(u,e)),((u,\$,e),(f_2,e)),\\ &((q,b,a),(r,e)),((r,b,a),(r,e)),\\ &((r,\$,a),(t,e)),((r,b,\perp),(u,e)),\\ &((v,a,e),(v,aa)),((v,\$,\perp),(f_3,e)),\\ &((v,b,a),(w,e)),((w,b,a),(w,e)),\\ &((w,\$,\perp),(f_3,e))\}\end{aligned}$$



ii)

$$L_2 = \{a^m cb^n | m \neq n\} \cup \{a^m db^{2m} | m \geq 0\}$$

The language is deterministic context-free since it is accepted by the following DPDA.

- Start with pushing bottom of stack symbol the stack, and change state to p
- For every a read from input push aa to the stack
- After this step if we read c from input, change state to q
  - For every b read, pop aa.
  - If input has finished (read \$) and stack not empty, change state to u and empty stack accept the input.
  - If stack is empty (top of stack $\bot$) and input has not finished, change state to t and finish reading input. When input has finished (read \$) change state to f_1 and accept.
- If we read d from input, change state to r
  - For every b read pop a.

– When input finished (read \$) and stack is empty (top of stack $\perp$), change state to f_2 and accept it.

$$M = \{K, \Sigma, \Gamma, s, F, \Delta\}$$

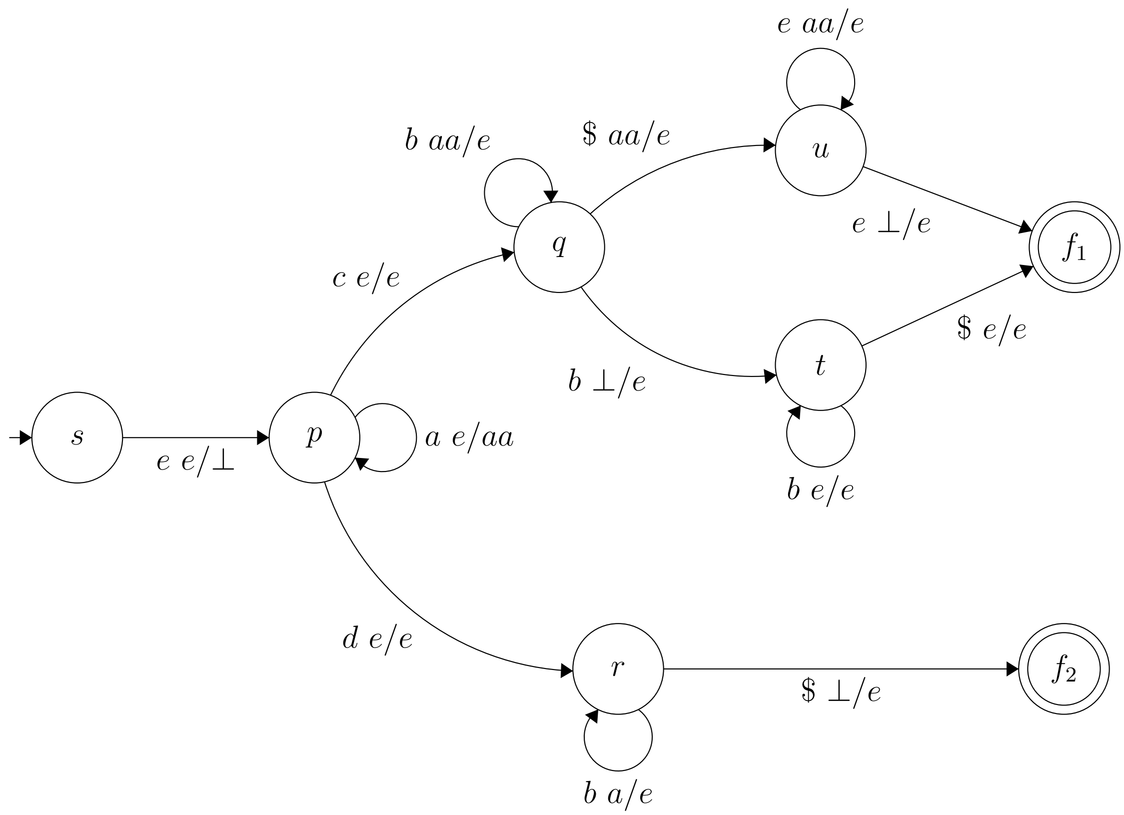
$$K = \{s, p, q, r, t, u, f_1, f_2\}$$

$$\Sigma = \{a, b, c, d, \$\}$$

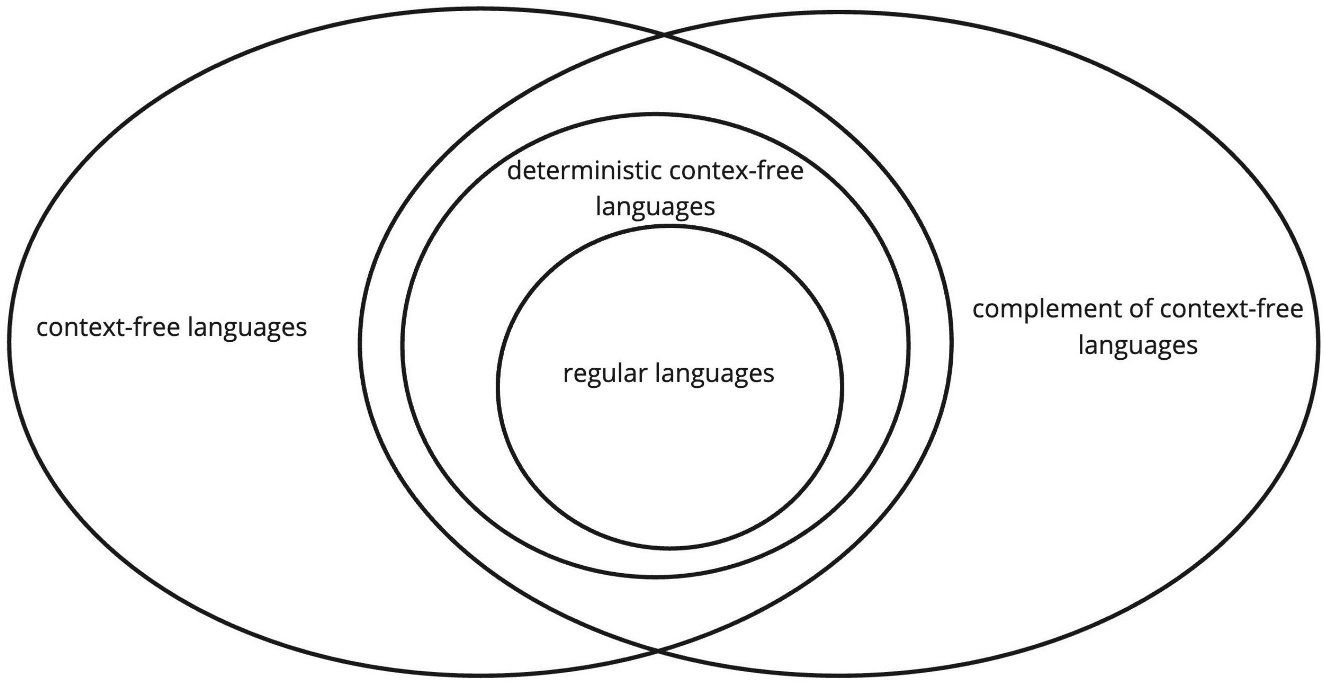
$$\Gamma = \{a, \perp\}$$

$$F = \{f_1, f_2\}$$

$$\begin{aligned} \Delta = \{ & ((s, e, e), (p, \perp)), ((p, a, e), (p, aa)), \\ & ((p, c, e), (q, e)), ((p, d, e), (r, e)), \\ & ((q, b, aa), (q, e)), ((q, \$, aa), (u, e)), \\ & ((u, e, aa), (u, e)), ((u, e, \perp), (f_1, e)), \\ & ((q, b, \perp), (t, e)), ((t, b, e), (t, e)), \\ & ((t, \$, e), (f_1, e)), ((r, b, a), (r, e)), \\ & ((r, \$, \perp), (f_3, e))\} \end{aligned}$$



b)


deterministic contex-free
languages
context-free languages
regular languages
complement of context-free
languages