Tutorial 6 CS 241

Graham Cooper

June 19th, 2015

Topics
- Non-deterministic Finite Automata (NFAs)
- Context Free Languages (CFLs)

NFAS

- Σ - input alphabet
- Q - finite set of states
- $q_0 \in Q$ - starting state
- A $\subseteq$ Q - set of accepting states
- δ: Q $\times\Sigma \rightarrow 2^Q$ - transition function
- δ: Q $\times\Sigma \rightarrow$ Q

CFGs

- Σ/T - finite set of terminals
- V/N - finite set of non-terminals
- P/R - finite set of rewrite/production rules
- S - starting non-terminal

NFA Problems

1)

$\Sigma\{a, b, c\}$
L = {Strings ending in abc or cab}

See page for image

2)

Convert the previous NFA to a DFA using subset construction

See page for image

CFL Problems

1)

Write a CFG that generates the same language as the NFA

S $\rightarrow$ abc | cab | TS
T $\rightarrow$ a|b|c

2)

$\Sigma\{0, 1\}$
L = $\{0^n 1^n | n \geq 0\}$

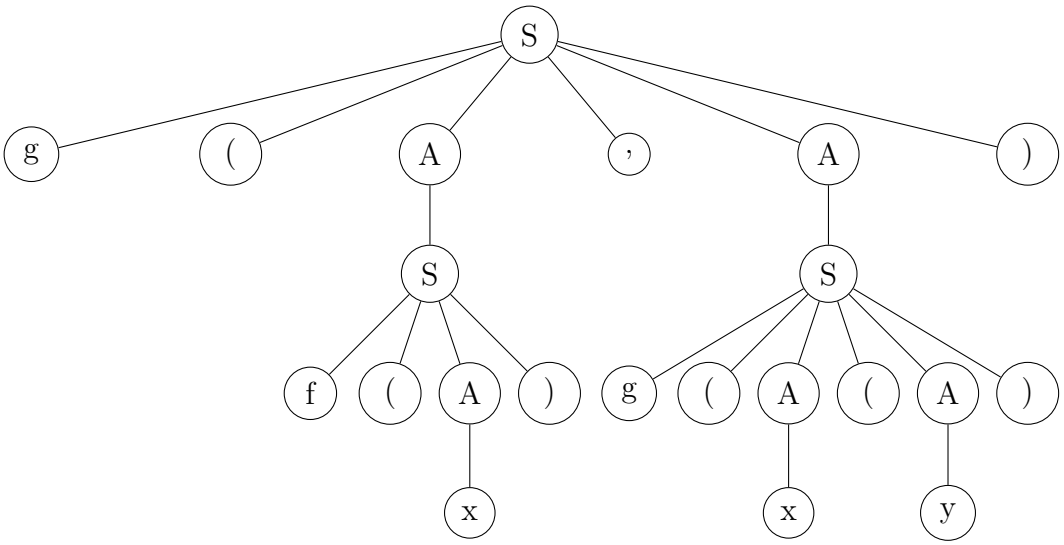
S $\rightarrow \epsilon$| 0S1

4)

$S \rightarrow f(A)$
$S \rightarrow g(A, A)$
$A \rightarrow x$
$A \rightarrow y$
$A \rightarrow S$
f(x)
g(y,f(x))
f(g(g(x,y,f(x))))

g(f(x), g(x,y))

$$
\begin{aligned}
S &\implies {}_2g(A, A) \\
&\implies {}_5g(S, A) \\
&\implies {}_1g(f(A), A) \\
&\implies {}_3g(f(x), A) \\
&\implies {}_5g(f(x), S) \\
&\implies {}_2g(f(x), g(A, A) \\
&\implies {}_3g(f(x), g(x, A) \\
&\implies {}_4g(f(x), g(x, y))
\end{aligned}
$$



5)

- S → (S)
- S → SS
- S → ϵ

Very helpful for assignment question!!!

```
eval(tree)
-- if (rule == s -> e) return 0
-- else if (rule == s -> (S)) return 1 + eval(S)
-- else if (rule == S -> S1S2) return max (eval(S1), eval(S2))
```