Principles of Programming Languages

CS 314

Recitation 2

RUTGERS

Topics Today

Deterministic finite automaton (DFA)

Context-free grammar

Derivation

Parse tree

Ambiguous grammars

Topics Today

Deterministic finite automaton (DFA)

Context-free grammar

Derivation

Parse tree

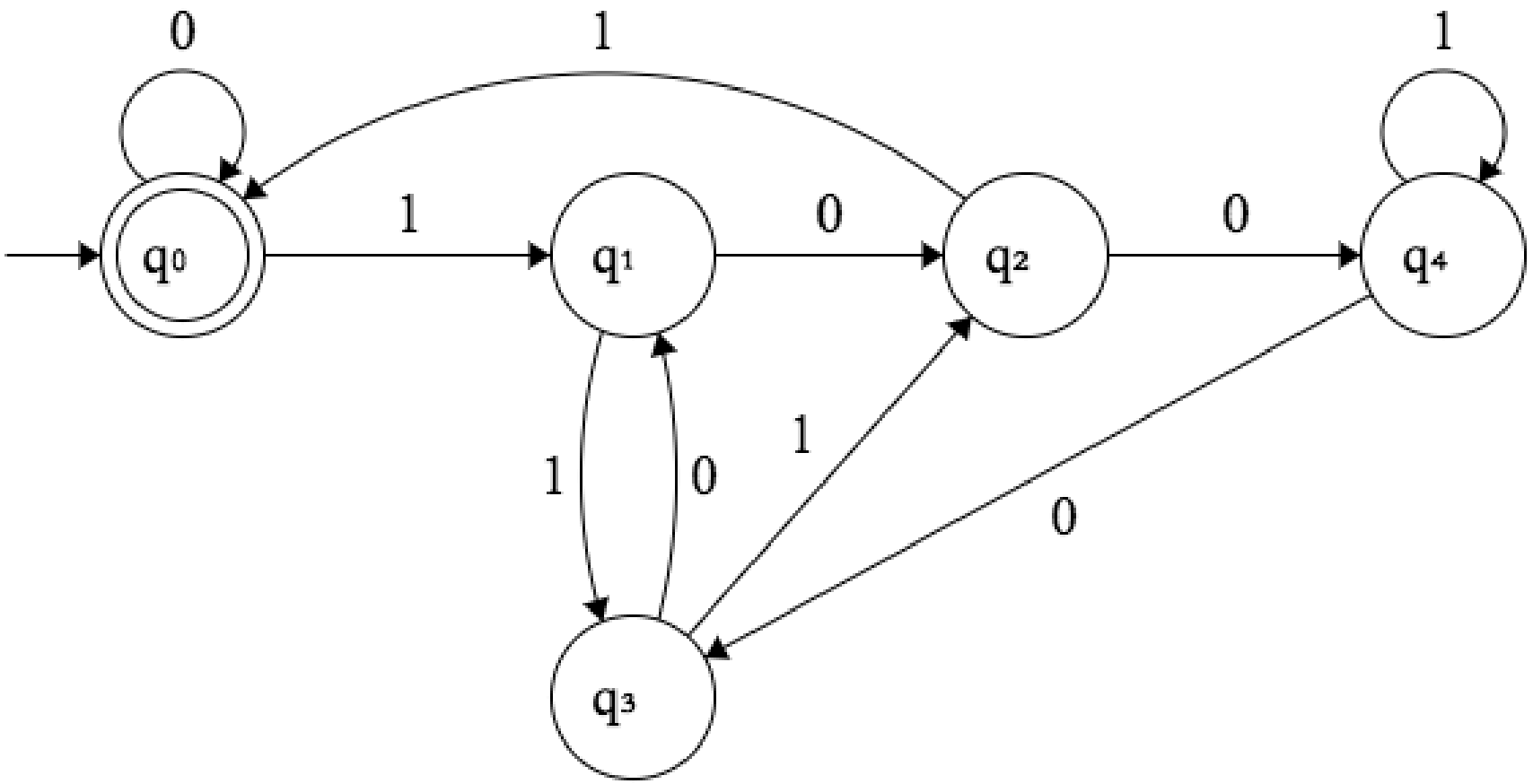
Ambiguous grammars

DFA - Example

Specify a DFA using a transition diagram and a formal FSA specification <S, s, F, T> that recognizes the following language:

“All strings of 0’s and 1’s that, when interpreted as a binary number, are divisible by 5. In other words, value(binary number) modulo 5 = 0.”

DFA - Solution



Topics Today

Deterministic finite automaton (DFA)

Context-free grammar

Derivation

Parse tree

Ambiguous grammars

CFG – Example 1

Specify a context-free grammar in BNF notation that generates the following language. (With alphabet $\sum=\{0,1\}$)

$\{ w \mid w$ has at least three $1\}$

CFG – Solution 1

Specify a context-free grammar in BNF notation that generates the following language. (With alphabet ∑={0,1})

{ w | w has at least three 1}

CFG:

<S> ::= <A>1<A>1<A>1<A>

<A>::=0<A>|1<A>|ε

CFG – Example 2

Specify a context-free grammar in BNF notation that generates the following language. (With alphabet $\sum=\{0,1\}$)

{ w | |w| is odd, and the symbol in the middle of w is 0 }

CFG – Solution 2

Specify a context-free grammar in BNF notation that generates the following language. (With alphabet ∑={0,1})

{ w | |w| is odd, and the symbol in the middle of w is 0 }

CFG:

<S>::=<A><S><A>|0

<A>::=0|1

CFG – Example 3

Specify a context-free grammar in BNF notation that generates the following language.

$\{ a^{3n}b^{3n}c^{3n}d^{4n} \mid n \geq 0\}$, with alphabet $\sum=\{a, b, c, d\}$

CFG – Solution 3

Specify a context-free grammar in BNF notation that generates the following language.

$\{ a^{3n}b^{3n}c^{3n}d^{4n} \mid n \geq 0\}$, with alphabet $\sum=\{a, b, c, d\}$

Not a context-free grammar!

Topics Today

Deterministic finite automaton (DFA)

Context-free grammar

Derivation

Parse tree

Ambiguous grammars

Derivation – Example

Given a CFG:

<E>::=<E>+<T>|<E>-<T>|<T>

<T>::=<T>*<F>|<T>/<F>|<F>

<F>::=(<E>)|i

Give a leftmost and a rightmost derivation for the sentence: i+i*i

Derivation – Solution

<E>	
⇒L	<E>+<T>
⇒L	<T>+<T>
⇒L	<F>+<T>
⇒L	i+<T>
⇒L	i+<T>*<F>
⇒L	i+<F>*<F>
⇒L	i+i*<F>
⇒L	i+i*i

<E>	
⇒R	<E>+<T>
⇒R	<E>+<T>*<F>
⇒R	<E>+<T>*i
⇒R	<E>+<F>*i
⇒R	<E>+i*i
⇒R	<T>+i*i
⇒R	<F>+i*i
⇒R	i+i*i

Given a CFG:

<E>::=<E>+<T>|<E>-<T>|<T>

<T>::=<T>*<F>|<T>/<F>|<F>

<F>::=(<E>)|i

sentence: i+i*i

Topics Today

Deterministic finite automaton (DFA)

Context-free grammar

Derivation

Parse tree

Ambiguous grammars

Parse tree - Example

Show the corresponding parse trees for the derivations we got in last example

Given a CFG:

<E>::=<E>+<T>|<E>-<T>|<T>

<T>::=<T>*<F>|<T>/<F>|<F>

<F>::=(<E>)|i

sentence: i+i*i

<E>

⇒L <E>+<T>

⇒L <T>+<T>

⇒L <F>+<T>

⇒L i+<T>

⇒L i+<T>*<F>

⇒L i+<F>*<F>

⇒L i+i*<F>

⇒L i+i*i

Parse tree - Solution

Show the corresponding parse trees for the derivations we got in last example

Given a CFG:

<E>::=<E>+<T>|<E>-<T>|<T>

<T>::=<T>*<F>|<T>/<F>|<F>

<F>::=(<E>)|i

sentence: i+i*i

<E>

⇒L <E>+<T>

⇒L <T>+<T>

⇒L <F>+<T>

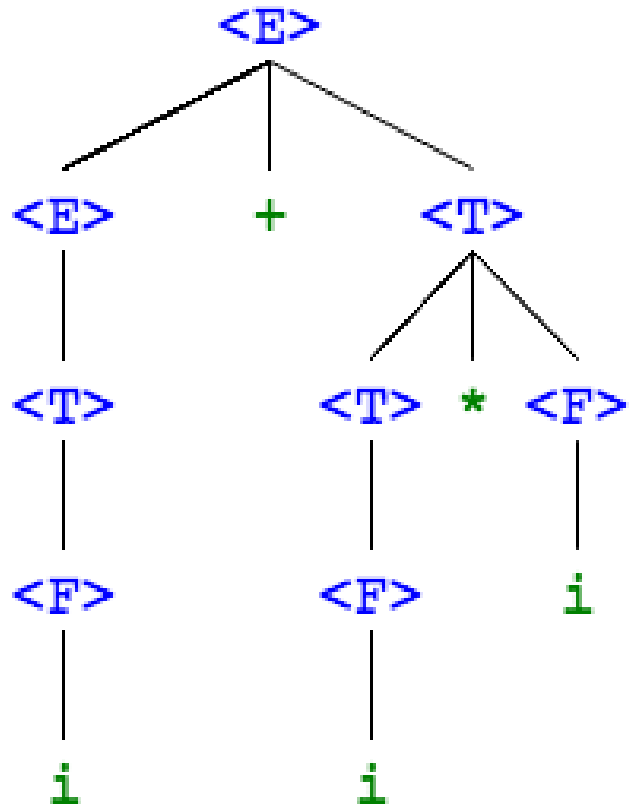
⇒L i+<T>

⇒L i+<T>*<F>

⇒L i+<F>*<F>

⇒L i+i*<F>

⇒L i+i*i



Topics Today

Deterministic finite automaton (DFA)

Context-free grammar

Derivation

Parse tree

Ambiguous grammars

Ambiguous Grammars - Example

Given grammar:

<exp> ::= <exp> <oper><exp> |(<exp>)|i

<oper>::= +|-|*|/

Show that the above grammar is ambiguous.

Ambiguous Grammars - Solution

For the sentence i+i*i, it has two distinct rightmost derivations.

So the grammar is ambiguous.

<exp> ::= <exp> <oper><exp> |(<exp>)|i
<oper>::= +|-|*|/

(1)

<exp>
⇒R <exp><oper><exp>
⇒R <exp><oper>i
⇒R <exp>*i
⇒R <exp><oper><exp>*i
⇒R <exp><oper>i*i
⇒R <exp>+i*i
⇒R i+i*i

(2)

<exp>
⇒R <exp><oper><exp>
⇒R <exp><oper><exp><oper><exp>
⇒R <exp><oper><exp><oper>i
⇒R <exp><oper><exp>*i
⇒R <exp><oper>i*i
⇒R <exp>+i*i
⇒R i+i*i