CS 132 Compiler Construction, Fall 2014

Instructor: Jens Palsberg

Multiple Choice Exam, Nov 20, 2014

ID

Name

This exam consists of 22 questions. Each question has four options, exactly one of which is correct, while the other three options are incorrect. For each question, you can check multiple options.

I will grade each question in the following way. If you check *none* of the options, you get 0 points. If you check all *four* options, you get 0 points.

Check one option. If you check *one* option, and that option is correct, you get 2 points. If you check *one* option, and that option is wrong, you get –0.667 points (yes, negative!).

Check two options. If you check *two* options, and one of those options is correct, you get 1 point. If you check *two* options, and both of them are wrong, you get –1 point (yes, negative!).

Check three options. If you check *three* options, and one of those options is correct, you get 0.415 points. If you check *three* options, and all three of them are wrong, you get –1.245 points (yes, negative!).

The maximum point total is $22 \times 2 = 44$ points. I will calculate a percentage based on the points in the following way:

$$\frac{\max(0, \text{point total})}{44} \times 100$$

Notice that if your point total is negative, you will get 0 percent

Example
Consider the grammar

$A \ ::= \ x \ B \ y \ \mid \ C \ z$
$B \ ::= \ x \ A \ \mid \ \epsilon$
$C \ ::= \ y \ B \ z$

where $\{A, B, C\}$ is the set of nonterminal symbols, A is the start symbol, $\{x, y, z\}$ is the set of terminal symbols, and ϵ denotes the empty string.

Question 1
Which nonterminals are nullable?
- a☐ A
- b☐ B
- c☐ C
- d☐ A and B

Question 2
What is First(A)?
- a☐ $\{y\}$
- b☐ $\{x, y\}$
- c☐ $\{y, z\}$
- d☐ $\{x, y, z\}$

Question 3
What is First(B)?
- a☐ $\{x\}$
- b☐ $\{y\}$
- c☐ $\{z\}$
- d☐ $\{x, y\}$

Question 4
What is First(C)?
- a☐ $\{x\}$
- b☐ $\{y\}$
- c☐ $\{z\}$
- d☐ $\{x, y\}$

Question 5
What is Follow(A)?
- a☐ $\{x\}$
- b☐ $\{y\}$
- c☐ $\{z\}$
- d☐ $\{y, z\}$

Question 6
What is Follow(B)?
- a☐ $\{x\}$
- b☐ $\{y\}$
- c☐ $\{z\}$
- d☐ $\{y, z\}$

Question 7
What is Follow(C)?
- a☐ $\{x\}$
- b☐ $\{y\}$
- c☐ $\{z\}$
- d☐ $\{y, z\}$

Question 8
Is the grammar LL(1)?
- a☐ Yes
- b☐ No
- c☐ The question cannot be answered with the information provided
- d☐ The LL(1)-checker would go into an infinite loop

Example
Consider the grammar

$$
\begin{array}{lll}
A & ::= & x\ B\ C \\
B & ::= & y \mid x\ A \mid \epsilon \\
C & ::= & z\ A
\end{array}
$$

where $\{A, B, C\}$ is the set of nonterminal symbols, A is the start symbol, $\{x, y, z\}$ is the set of terminal symbols, and ϵ denotes the empty string. The grammar is LL(1). The predictive parsing table is a two-dimensional table called *table*.

Question 9
What does $table(A, x)$ contain?
- a☐ error
- b☐ ϵ
- c☐ $x\ B\ C$
- d☐ $x\ A$

Question 10
What does $table(A, y)$ contain?
- a☐ error
- b☐ ϵ
- c☐ $x\ B\ C$
- d☐ $x\ A$

Question 11

What does $table(A, z)$ contain?

- a ☐ error
- b ☐ ϵ
- c ☐ $x\ B\ C$
- d ☐ $x\ A$

Question 12

What does $table(B, x)$ contain?

- a ☐ error
- b ☐ ϵ
- c ☐ y
- d ☐ $x\ A$

Question 13

What does $table(B, y)$ contain?

- a ☐ error
- b ☐ ϵ
- c ☐ y
- d ☐ $x\ A$

Question 14

What does $table(B, z)$ contain?

- a ☐ error
- b ☐ ϵ
- c ☐ y
- d ☐ $z\ A$

Question 15

What does $table(C, x)$ contain?

- a ☐ error
- b ☐ ϵ
- c ☐ $x\ A$
- d ☐ $x\ B\ C$

Question 16

What does $table(C, y)$ contain?

- a ☐ error
- b ☐ ϵ
- c ☐ y
- d ☐ $z\ A$

Question 17

What does $table(C, z)$ contain?

a ☐ error
b ☐ ϵ
c ☐ y
d ☐ $z\ A$

Example

Consider the grammar

$$
\begin{array}{lcl}
A & ::= & x\ B \mid C\ x \\
B & ::= & z\ A \mid \epsilon \\
C & ::= & y\ A \mid z\ C
\end{array}
$$

where $\{A, B, C\}$ is the set of nonterminal symbols, A is the start symbol, $\{x, y, z\}$ is the set of terminal symbols, and ϵ denotes the empty string. The grammar is LL(1). Assume that a recursive-descent parser for the above grammer declares a variable `next` of type `token`, and that the program has three procedures `A()`, `B()`, `C()`, and the following main part:

```
void main() {
  next = getnexttoken();
  A();
}
```

The procedure `getnexttoken()` gets the next token from an input file. Assume also we have the following helper procedure, written in pseudo-code:

```
void eat(token t) {
  if (t == next) {
     next = getnexttoken();
  }
  else {
     error();
  }
```

Question 18
The procedure `A()` looks like:

```
if (next == x) {
   eat(x);
   B()
}
else {
   ????
}
```

What is "`????`" ?

a ☐ `/* do nothing */`
b ☐ `error();`
c ☐ `C(); eat(x);`
d ☐ `if (next == y) { C(); eat(x); } else { error(); }`

Question 19
The procedure `B()` looks like:

```
if (next == z) {
   eat(z);
   A()
}
else {
   ????
}
```

What is "`????`" ?

a ☐ `/* do nothing */`
b ☐ `error();`
c ☐ `if (next == x) { error(); } else { /* do nothing */ }`
d ☐ `if (next == x) { /* do nothing */ } else { error(); }`

Question 20
The procedure `C()` looks like:

```
if (next == y) {
   eat(y);
   A()
}
else {
   ????
}
```

What is "`????`" ?

a☐ `/* do nothing */`
b☐ `error();`
c☐ `if (next == z) { eat(z); C() } else { eat(x); B() }`
d☐ `if (next == z) { eat(z); C() } else { error(); }`

Example
Consider the grammar:

$$A \;::=\; A\ x\ B \;\mid\; A\ y \;\mid\; z$$
$$B \;::=\; z\ B \;\mid\; \epsilon$$

where $\{A, B\}$ is the set of nonterminal symbols, A is the start symbol, and $\{x, y, z\}$ is the set of terminal symbols, and ϵ denotes the empty string.

Question 21
Which grammar generates the same language as the above grammar?

a☐

$$A \;::=\; z\ D$$
$$B \;::=\; z\ B \;\mid\; \epsilon$$
$$C \;::=\; x\ B \;\mid\; y$$
$$D \;::=\; C\ D \;\mid\; \epsilon$$

b☐

$$A \;::=\; A\ C \;\mid\; z$$
$$B \;::=\; z\ B \;\mid\; \epsilon$$
$$C \;::=\; x\ C \;\mid\; y$$

c☐

$$A \;::=\; z\ D$$
$$B \;::=\; z\ B \;\mid\; \epsilon$$
$$C \;::=\; x\ B \;\mid\; z$$
$$D \;::=\; C\ D \;\mid\; \epsilon$$

d☐ None of the other cases.

Example
Consider the grammar:

$A \;::=\; y\ B \;\mid\; x\ y\ B$
$B \;::=\; x\ y \;\mid\; x\ A$

where $\{A, B\}$ is the set of nonterminal symbols, A is the start symbol, $\{x, y\}$ is the set of terminal symbols, and ϵ denotes the empty string.

Question 22
Which grammar generates the same language as the above grammar?

a☐

$A \;::=\; y\ B \;\mid\; x\ y\ B$
$B \;::=\; x\ C$
$C \;::=\; y\ D \;\mid\; x\ y\ B$
$D \;::=\; \epsilon \;\mid\; B$

b☐

$A \;::=\; y\ B \;\mid\; x\ y\ B$
$B \;::=\; x\ y \;\mid\; x\ y\ B \;\mid\; x\ x\ A$

c☐

$A \;::=\; y\ x\ y\ C \;\mid\; x\ y\ C$
$B \;::=\; x\ C$
$C \;::=\; y\ D \;\mid\; x\ y\ D$
$D \;::=\; \epsilon \;\mid\; B$

d☐ None of the other cases.