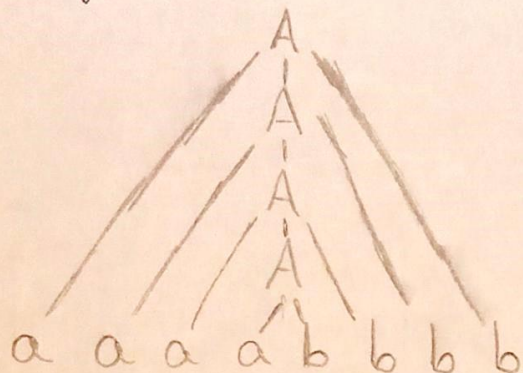


1) Considere la siguiente gramática:  $A \rightarrow aAb \mid ab$

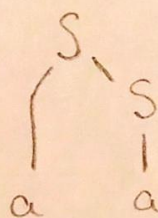
a) Mostrar una derivación de  $aaaaabbbb$

$$A \rightarrow aAb \rightarrow aaAbb \rightarrow aaaAbbb \rightarrow \underline{aaaaabbbb} /$$

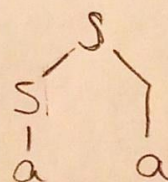
b) Dibuje el árbol de análisis sintáctico para la entrada  $aaaaabbbb$



2) Demostrar que la siguiente gramática es ambigua  $S \rightarrow aS \mid Sa \mid a$  con la cadena  $aa$



Dos árboles distintos  
para la misma  
cadena:  $aa$   
ambigua



3) Elimine la recursividad por la izquierda  $A \rightarrow 1 \mid AO$

Observamos que el lenguaje generado por la gramática es

$10^m$

$$A \rightarrow 1C$$

$$C \rightarrow \epsilon \mid OC$$

}  $10^m$  lenguaje generado



4) Considere la siguiente gramática  $A \rightarrow a \mid b \mid (A)$   
 Escriba el analizador sintáctico predictivo recursivo

```
void pareja(clexico t) {
    if (preara == t)
        preara = seglexico();
    else error();
}
```

```
void main() {
    preara = segToken();
    A();
}
```

```
void A() {
    if (preara == 'a') {
        pareja('a');
    } else if (preara == 'b') {
        pareja('b');
    } else if (preara == '(') {
        pareja('('); A(); pareja(')');
    } else error();
}
```

5) Considere la siguiente gramática

$L \rightarrow L, D \mid D$   
 $D \rightarrow 0 \mid 1 \mid 2$

a) Escriba una sección de la especificación de YACC para la gramática

b) Escriba las acciones gramaticales para que se imprima el número de 2's

```
% %
%%
{ printP("\n Numero de 2's: %d", $2); }
```

```
L: L ',' D { $$ = $1 + $3; }
    | D { $$ = $1; }
```

```
D: '0' { $$ = 0; }
    | '1' { $$ = 0; }
    | '2' { $$ = 1; }
    ;
%%
```

c) Escriba yylex()

```
int yylex()
int c;

while( (c = getchar()) == ' ' || c == '\n') ;

if (c == EOF)
    return 0;
else if (c == '0' || c == '1' || c == '2' || c == ',')
    return c;
else if (c == '\n') {
    lineo++;
    return c;
}
else error();
}
```