

Compiladores - Tarea 2

1. Construya cualquier gramática libre de contexto que reconozca el mismo lenguaje que el siguiente fragmento de código en C++:

```
bool T(char* c)
{
    return c[0] == 'y' && c[1] == 'y';
}

bool U(char* c)
{
    return c[0] == 'y' && c[1] == 'z';
}

bool S(char* c)
{
    return c[0] == 'x' && (T(c + 1) || U(c + 1));
}
```

2. Argumente si la gramática que usó para la pregunta anterior es LL(1). En caso de que la respuesta sea negativa, construya una gramática LL(1) equivalente.
3. ¿Qué diferencia hay entre un lenguaje LL(1) y una gramática LL(1)?
4. El siguiente fragmento de código C es sintácticamente ambiguo:

T * a;

debido a que puede denotar una multiplicación o una declaración de apuntador dependiendo del significado de T (y por lo mismo se podrían generar dos árboles sintácticos diferentes). Explique una manera de resolver esto (hay al menos dos maneras).

5. Suponga que cuenta con una gramática libre de contexto para C. ¿Lo anterior significa que C es libre de contexto? Argumente su respuesta.