

INDICE

1. Comentarios generales sobre la teoría de compiladores	1
1.1. tipos de sistemas de compilación	3
1.2. ambientes de compilación	4
1.3. análisis y síntesis	6
2. Aspectos formales	9
2.1. forma de Backus-Naur (BNF)	10
2.2. lenguajes formales	12
2.3. técnicas de análisis	22
2.4. grafos sintácticos	27
2.5. el lenguaje de programación PL/0	29
3. Análisis léxico y tablas de símbolos	32
3.1. autómatas finitos	33
3.2. el analizador léxico	43
3.3. tablas de símbolos	46
3.4. Análisis léxico de PL/0	52
4. Análisis sintáctico y construcción de analizadores sintáctico	57
4.1. análisis descendente	57
4.1.1. gramáticas LL	57
4.1.2. estrategia descendente recursiva	64
4.1.3. Análisis sintáctico tabular	73
4.2. análisis descendente	79
4.2.1. gramáticas LR(k)	79
4.2.2. análisis por desplazamiento y reducción	81
4.2.3. analizador sintáctico LR	83
4.3. analizador sintáctico descendente recursivo para PL/0	95
5. Semántica y análisis de tipos	102
5.1. códigos intermedios	102
5.2. traducción dirigida por la sintaxis	106
5.3. comprobación de tipos	112
5.4. generación de código intermedio para PL/0	114
6. Como manejar errores	119
6.1. clasificación de errores	120
6.2. efectos de los errores	122
6.3. manejo de errores en el análisis léxico	123
6.4. manejo de errores en el análisis sintáctico	124
6.5. errores semánticas	126
6.6. recuperación de errores de PL/0	126
7. Generación y optimización de código	131
7.1. asignación de memoria	132
7.1.1. asignación estática de memoria	132
7.1.2. asignación dinámica de memoria	133
7.2. paso de parámetros	140
7.3. direccionamiento de variables	141
7.4. generación de código	143
7.5. optimización de código	148
8. Impacto de los avances en hardware	154
8.1. arquitectura de computadores y lenguajes de programación	154

8.2. conjuntos de instrucciones y microcódigo	155
8.3. arquitecturas RISC	158
Ejercicios	160
Referencias	170
Vocabulario técnico bilingüe	173
Índice de materias	177