

Índice general

Capítulo 1	Introducción a la compilación	1
1.1	Compiladores	1
1.2	Análisis del programa fuente	4
1.3	Las fases de un compilador	10
1.4	Programas de sistemas relacionados con un compilador	16
1.5	El agrupamiento de las fases	20
1.6	Herramientas para la construcción de compiladores	22
	Notas bibliográficas	24
Capítulo 2	Un compilador sencillo de una pasada	25
2.1	Perspectiva	25
2.2	Definición de la sintaxis	26
2.3	Traducción dirigida por la sintaxis	33
2.4	Análisis sintáctico	40
2.5	Traductor de expresiones simples	49
2.6	Análisis léxico	54
2.7	Incorporación de una tabla de símbolos	61
2.8	Máquinas de pila abstractas	64
2.9	Reunión de las técnicas	69
	Ejercicios	79
	Ejercicios de programación	81
	Notas bibliográficas	82
Capítulo 3	Análisis léxico	85
3.1	Función del analizador léxico	86
3.2	Manejo de los <i>buffers</i> de entrada	90
3.3	Especificación de los componentes léxicos	94
3.4	Reconocimiento de componentes léxicos	100
3.5	Un lenguaje para la especificación de analizadores léxicos ...	109
3.6	Autómatas finitos	115
3.7	Paso de una expresión regular a un AFN	123
3.8	Diseño de un generador de analizadores léxicos	131
3.9	Optimación de buscadores por concordancia de patrones basados en AFD	136
	Ejercicios	149
	Ejercicios de programación	159
	Notas bibliográficas	159

Capítulo 4	Análisis sintáctico	163
4.1	El papel del analizador sintáctico	164
4.2	Gramáticas independientes del contexto	169
4.3	Escritura de una gramática	176
4.4	Análisis sintáctico descendente	186
4.5	Análisis sintáctico ascendente	200
4.6	Análisis sintáctico por precedencia de operadores	209
4.7	Analizadores sintácticos LR	221
4.8	Uso de gramáticas ambiguas	254
4.9	Generadores de analizadores sintácticos	264
	Ejercicios	274
	Notas bibliográficas	284
Capítulo 5	Traducción dirigida por la sintaxis	287
5.1	Definiciones dirigidas por la sintaxis	288
5.2	Construcción de árboles sintácticos	295
5.3	Evaluación ascendente de definiciones con atributos sintetizados	302
5.4	Definiciones con atributos por la izquierda	305
5.5	Traducción descendente	311
5.6	Evaluación ascendente de los atributos heredados	318
5.7	Evaluadores recursivos	326
5.8	Consideraciones de espacio para valores de atributos en el momento de la compilación	329
5.9	Asignación de espacio en el momento de la construcción del compilador	333
5.10	Análisis de definiciones dirigidas por la sintaxis	340
	Ejercicios	347
	Notas bibliográficas	351
Capítulo 6	Comprobación de tipos	355
6.1	Sistemas de tipos	356
6.2	Especificación de un comprobador de tipos sencillo	360
6.3	Equivalencia de expresiones de tipos	364
6.4	Conversiones de tipos	371
6.5	Sobrecarga de funciones y operadores	373
6.6	Funciones polimórficas	376
6.7	Un algoritmo para la unificación	388
	Ejercicios	393
	Notas bibliográficas	398
Capítulo 7	Ambientes para el momento de la ejecución	401
7.1	Aspectos del lenguaje fuente	401
7.2	Organización de la memoria	408

7.3	Estrategias para la asignación de memoria	413
7.4	Acceso a nombres no locales	424
7.5	Paso de parámetros	437
7.6	Tablas de símbolos	443
7.7	Instrumentos de los lenguajes para la asignación dinámica de la memoria	454
7.8	Técnicas para la asignación dinámica de la memoria	457
7.9	Asignación de memoria en FORTRAN	460
	Ejercicios	469
	Notas bibliográficas	475
Capítulo 8	Generación de código intermedio	477
8.1	Lenguajes intermedios	478
8.2	Declaraciones	487
8.3	Proposiciones de asignación	492
8.4	Expresiones booleanas	502
8.5	Proposiciones case	511
8.6	Relleno de retroceso	515
8.7	Llamadas a procedimientos	521
	Ejercicios	523
	Notas bibliográficas	526
Capítulo 9	Generación de código	529
9.1	Aspectos del diseño de un generador de código	530
9.2	La máquina objeto	535
9.3	Administración de la memoria durante la ejecución	538
9.4	Bloques básicos y grafos de flujo	545
9.5	Información sobre el siguiente uso	550
9.6	Un generador de código simple	552
9.7	Distribución y asignación de registros	558
9.8	Representación de bloques básicos por medio de GDA	563
9.9	Optimación mediante «mirilla»	570
9.10	Generación de código a partir de los GDA	574
9.11	Algoritmo para generación de código con programación dinámica	584
9.12	Generadores de generadores de código	589
	Ejercicios	596
	Notas bibliográficas	599
Capítulo 10	Optimación de código	603
10.1	Introducción	604
10.2	Las principales fuentes para la optimación	610
10.3	Optimación de bloques básicos	616
10.4	Lazos en los grafos de flujo	620
10.5	Introducción al análisis global del flujo de datos	626

10.6	Solución iterativa de las ecuaciones de flujo de control	641
10.7	Transformaciones para mejorar el código	651
10.8	Tratamiento con sinónimos (alias)	666
10.9	Análisis de flujo de datos de grafos de flujo estructurados	678
10.10	Algoritmos eficientes para el flujo de datos	690
10.11	Una herramienta para el análisis del flujo de datos	698
10.12	Estimación de tipos	713
10.13	Depuración simbólica de código optimado	721
	Ejercicios	729
	Notas bibliográficas	736
Capítulo 11	¿Quiere escribir un compilador?	741
11.1	Proyectando un compilador	741
11.2	Aspectos del desarrollo de compiladores	743
11.3	El entorno para desarrollo de compiladores	747
11.4	Pruebas y mantenimiento	749
Capítulo 12	Una mirada a algunos compiladores	751
12.1	EQN, un preprocesador para tipografía de matemáticas	751
12.2	Compiladores para Pascal	752
12.3	Los compiladores para C	753
12.4	Los compiladores de FORTRAN H	755
12.5	El compilador BLISS/11	758
12.6	Compilador optimador de MODULA-2	760
Apéndice	Un proyecto de programación	763
A.1	Introducción	763
A.2	Estructura del programa	763
A.3	Sintaxis de un subconjunto de Pascal	763
A.4	Convenciones lexicográficas	766
A.5	Ejercicios propuestos	767
A.6	Evolución del intérprete	768
A.7	Ampliaciones	769
Bibliografía		771
Índice de materias		791
Vocabulario bilingüe de términos técnicos		805