

CONTENIDO TEMÁTICO

Ordenar temas utilizando codificación decimal

1. Introducción. Objeto de un compilador. Organización por módulos. Estructuras de datos usadas. Herramientas más frecuentes. Definición del lenguaje a implementar (Tiger).
2. Análisis léxico. Tokens. Expresiones regulares y autómatas finitos. Autómatas no determinísticos y determinísticos. Analizadores múltiples. Casos particulares. Herramientas más comunes: Lex, Flex, ML-Lex y Mosmllex.
3. Parsing. Gramáticas libres de contexto. Párseres recursivo-descendentes. Tratamiento de ambigüedades. Párseres bottom-up. Párseres predictivos. Gramáticas LR, SLR y LALR(n). Tratamiento de errores. Herramientas: YACC, Bison, ML-Yacc y Mosmlyacc. Casos particulares.
4. Sintaxis abstracta. Acciones semánticas. Árboles de sintaxis abstracta. Constructores. Integración con párseres. Discusión de casos particulares.
5. Análisis semántico. Tablas de símbolos. Funciones y bloques LET. Evaluación de expresiones tipadas. Declaraciones mutuamente recursivas. Tipos definidos por el usuario.
6. Marco de activación. Variables locales. Tratamiento de stacks. Prólogos y epílogos de funciones. Marcos anidados. Stack pointer y frame pointer. Tablas. Display. Implementaciones en arquitecturas Intel, Sparc, Mips y JVM (Java Virtual Machine).
7. Código intermedio. Árboles para representación intermedia. Traducción de AST a IRT. Declaraciones. Otros lenguajes intermedios. Lambda cálculo, Lambda enriquecido. Y combinadores. Lenguaje de Alto Nivel (HIR), Nivel Medio (MIR) y Bajo Nivel (LIR).
8. Bloques básicos y trazas. Árboles canónicos. Reescritura de IRT. Invariancia semántica. Transformaciones de saltos condicionales. Serialización y determinación de bloques básicos. Posibles trazas.
9. Selección de instrucciones. Algoritmos para selección de instrucciones. Pattern Matching. Bottom-Up Rewriting Systems (BURS). Implementaciones para las arquitecturas más comunes: Intel, Sparc, MIPS y JVM.
10. Análisis de tiempo de vida. Definición de tiempo de vida. Interferencias. Ecuaciones de flujos de datos. Algoritmos de resolución.
11. Alocación de registros. Objetivos. Algoritmo de Sethi-Ullman para alocación de registros en árboles. Algoritmo de Chaitin para grafos. Heurísticas: Briggs y George. Spilling.
12. Ensamblado y ambientes de ejecución. Emisión de código ensamblador. Formatos. Paquetes y rutinas de runtime. Ejemplos particulares. Paquetes de validación.

TRABAJOS PRÁCTICOS

a) Enumeración:

1. Scanner y parser para Tiger.
2. Análisis de tipos del AST y generación de código intermedio.
3. Canonización del código intermedio y linearización.
4. Emisión de código de máquina.
5. Data flow.
6. Enregistración mediante coloreo del grafo de interferencia.
7. Ensamblado y enlace con el runtime.

b) Guías de trabajos prácticos publicadas: (con su código de publicación)

BIBLIOGRAFÍA

a) Adecuada al programa. Ordenada por temas y con su codificación de biblioteca, incluidas las publicaciones de la Cátedra con su código de publicación.

Appel, A. Modern Compiler Implementation in ML. Cambridge University Press. 1.998.
Aho, A., R. Sethi and J. Ullman. Compilers: Principles Techniques and Tools. Addison-Wesley. 1.986.
Morgan, R. Building An Optimizing Compiler. Digital Press. 1.998.
Muchnick, S. Advanced Compiler Design and Implementation. Morgan Kauffman Pub. 1.997.
Fraser, c. and D. Hanson. A Retargetable C Compiler: Design and Implementation. Benjamin Cummings. 1.996.

b) Complementaria para profundización o extensión de temas.

Peyton-Jones, S. The Implementation of Functional Programming Languages. Prentice-Hall. 1.987.
Abelson, H. and G. Sussman. Structure and Interpretation of Programming Languages (2nd de). MIT Press. 1.996.
Baader F. and T. Nipkow. Term Rewriting and All That. Cambridge University Press. 1.998.
Kamin, S. Programming Languages: An Interpreter-Based Approach. Addison-Wesley. 1.988.
Gordon, M. Programming Language Theory and Its Implementation. Prentice-Hall. 1.988.
Hopcroft, J. Y J. Ullman. Introducción a la Teoría de Autómatas, Lenguajes y Computación.
Plasmeijer R. and M. van Eekelen. Functional Programming and Parallel Graph Rewriting. Addison-Wesley, 1.983.
Stoy, J. Denotational Semantics: The Scott-Strachey Approach. MIT Press, 1.977.
Slonnegger, K. And B. Kurtz. Formal Syntax and Semantics of Programming Languages. Addison-Wesley, 1.997.
Knuth, D. The Art of Computer Programming (3 vols). Addison-Wesley, 1.997.
Paulson, L. ML for the Working Programmer. Cambridge University Press, 1.998.
Stroustrup, B. The Design and Evolution of C++. Addison-Wesley, 1.995.
Milner, R. M. Tofte and R. Harper. The Definition of Standard ML. MIT Press, 1.990.