# 一种用于Java程序验证编译的标签类型

陈 晖, 陈意云, 茹祥民

陈 晖, 陈意云, 茹祥民

(中国科学技术大学 计算机科学系,安徽 合肥 230027)

作者简介: 陈晖(1979－),女,福建永春人,博士生,主要研究领域为程序语言设计与实现,类型论,基于语言的代码安全;陈意云(1946－),男,教授,博士生导师,主要研究领域为形式描述,程序设计语言的理论、设计与实现,编译理论与技术,软件安全;茹祥民(1978－),男,硕士生,主要研究领域为程序语言设计与实现.

联系人: 陈意云 Phn: +86-551-3607043, E-mail: yiyun@ustc.edu.cn

Abstract

In the area of language-based security, programs written in typed high-level languages need to be translated into those written in typed low-level languages. This work is not trivial when type dispatch constructs are involved and implemented with tag mechanism. This paper proposes a new type that can deal with a special type dispatch construct occurring in the interface invokation of Java. A low-level language with this type is able to efficiently implement the interface invokation. This implementation approach is adopted in a Just-In-Time compiler with a typed low-level language as a certifying language.

Chen H, Chen YY, Ru XM. A tag type for certifying compilation of Java program. *Journal of Software*, 2005,16(3):346-354.
http://www.jos.org.cn/1000-9825/16/346.htm

摘要

在基于语言考虑代码安全性的工作中,往往需要将高级语言程序翻译成类型化低级语言的程序进行类型检查.许多高级语言具有类型调度结构,在向低级语言的编译过程中需要用标签机制来实现.针对具有多继承接口的Java程序包含的一种特殊的类型调度结构,提出了一种新的标签类型.包含这种标签类型的低级语言能够有效地实现Java程序中的接口调用.这种对接口调用的编译方法被用在一个以类型化低级语言为验证语言的Java字节码即时编译器中.

References:

[1] Schneider FB, Morrisett G, Harper R. A language-based approach to security. LNCS: Informatics-10 Years Back. 10 Years Ahead. Springer-Verlag, 2001. 86-101.

[2] Necula G. Proof-Carrying code. In: ACM Symp. on Principles of Programming Language. New York: ACM Press, 1997. 106-119.

[3] Morrisett G, Walker D, Crary K, Glew N. From system F to typed assembly language. ACM Trans. on Programming Languages and Systems, 1999,21(3):528-569.

[4] Morrisett G, Crary K, Glew N. TALx86: A realistic typed assembly language. In: ACM SIGPLAN Workshop on Compiler Support for System Software. Atlanta, 1999. 25-35.

[5] Shao Z. Typed common intermediate format. In: USENIX Conf. on Domain-Specific Languages. Santa Barbara, 1997. 82-102. http://flint.cs.yale.edu/flint/publications/tcif.html

[6] Glew N. Type dispatch for named hierarchical types. In: Proc. of the 1999 ACM SIGPLAN Int'l Conf. on Functional Programming. Paris: ACM Press, 1999. 172-182.

[7] Glew N. An efficient class and object encoding. Technical Report STAR-TR-00. 07-02, STAR Laboratory, 2000.

[8] Chen YY, Zhang Y. Theory of Compiler. Beijing: Higher Education Press, 2003. 344-351 (in Chinese).

[9] Intel Microprocessor System Lab. ORP Online Resource. http://intel.com/research/mrl/ORP

[10] Kamin S. Inheritance in Smalltalk-80: A denotational definition. In: Proc. of the 15th ACM SIGACT-SIGPLAN Symp. on Principles of Programming Languages. San Diego: ACM Press, 1988. 80-87.

[11] Crary K, Weirich S. Flexible type analysis. In: Int'l Conf. on Functional Programming. Paris: ACM Press, 1999. 233-248.

[12] Crary K, Weirich S, Morrisette. Intensional polymorphism in type-erasure semantics. Journal of Functional Programming, 2002,12(6):567-600.

[13] League C, Shao Z, Trifonov V. Type-Preserving compilation of featherweight Java. ACM Trans. on Programming Languages and Systems, 2002,24(2):112-152.

附中文参考文献:
[8] 陈意云,张昱.编译原理.北京:高等教育出版社,2003.344-351.