

Estratto per riassunto della tesi di dottorato

Studente: Francesco Di Giacomo

matricola: 831569

Dottorato: Corso di dottorato in Informatica

Ciclo: 30

Titolo della tesi¹: Metacasanova: a High-performance Meta-compiler for Domain-specific Languages

Abstract:

English:

Programming languages are at the foundation of computer science, as they provide abstractions that allow the expression of the logic of a program independent from the underlying hardware architecture. In particular scenarios, it can be convenient to employ Domain-Specific Languages, which are capable of providing an even higher level of abstraction to solve problems which are common in specific domains. Examples of such domains are database programming, text editing, 3D graphics, and game development. The use of a domain-specific language for the development of particular classes of software may drastically increase the development speed and the maintainability of the code, in comparison with the use of a general-purpose programming language. While the idea of having a domain-specific language for a particular domain may be appealing, implementing such a language tends to come at a heavy cost: as it is common to all programming languages, domain-specific languages require a compiler which translates their programs into executable code. Implementing a compiler tends to be an expensive and time-consuming task, which may very well be a burden which overshadows the advantages of having a domain-specific language.

To ease the process of developing compilers, a special class of compilers called "meta-compilers" has been created. Meta-compilers have the advantage of requiring only the definition of a language in order to generate executable code for a program written in that language, thus skipping the arduous task of writing a hard-coded compiler for the new language. A disadvantage of meta-compilers is that they tend to generate slow executables, so they are usually only employed for rapid prototyping of a new language. The main aim of this thesis is to create a meta-compiler which does not suffer from the disadvantage of inefficiency. It presents a meta-compiler called "Metacasanova", which eases the development cost of a compiler while simultaneously generating efficient executable code.

The thesis starts by analysing the recurring patterns of implementing a compiler, to define a series of requirements for Metacasanova. It then explains the architecture of the meta-compiler and provides examples of its usage by implementing a small imperative language called C--, followed by the reimplementing of a particular, existing domain-specific language, namely Casanova, which has been created for use in game development. The thesis presents a novel way to optimize the performance of generated code by means of functors; it demonstrates the effect of this optimization by comparing the efficiency of Casanova code generated with and without it. Finally, the thesis demonstrates the advantages of having a meta-compiler like Metacasanova, by using Metacasanova to extend the semantics of Casanova to allow the definition of multiplayer online games.

¹ Il titolo deve essere quello definitivo, uguale a quello che risulta stampato sulla copertina dell'elaborato consegnato.

Italiano:

I linguaggi di programmazione sono alla base dell'informatica, dal momento che forniscono astrazioni che consentono di esprimere la logica di un programma in modo indipendente dall'architettura hardware sottostante. In particolari scenari può essere conveniente utilizzare Domain-Specific Languages (Linguaggi di dominio specifico), che sono in grado di fornire un livello di astrazione maggiore per risolvere problemi comuni a domini specifici. Esempi di questi domini sono programmazione di database, elaborazione di testi, grafica tridimensionale, e sviluppo di videogiochi. L'uso di un linguaggio di dominio per lo sviluppo di particolari classi di software può drasticamente aumentare la velocità di sviluppo e la mantenibilità del codice in confronto all'uso di linguaggi di programmazione generici. Sebbene l'idea di avere linguaggi per un particolare dominio possa essere interessante, implementare tali linguaggi tende a richiedere un costo elevato: come è di norma per tutti i linguaggi di programmazione, i linguaggi di dominio specifico richiedono un compilatore che traduca i loro programmi in codice eseguibile. Implementare un compilatore tende ad essere un compito esoso in termini economici e di tempo, cosa che potrebbe essere un onere che nasconde i vantaggi di avere un linguaggio di dominio.

Per semplificare il processo di sviluppo di compilatori, è stata creata una categoria speciale di compilatori chiamati "meta-compilatori". I meta-compilatori hanno il vantaggio di richiedere solo la definizione del linguaggio per generare codice eseguibile per un programma scritto in tale linguaggio, in modo tale da evitare l'arduo compito di scrivere un compilatore "hard-coded" per il nuovo linguaggio. Uno svantaggio dei meta-compilatori è che tendono a generare codice eseguibile lento, quindi vengono generalmente utilizzati per prototipizzazione rapida di un nuovo linguaggio. Lo scopo principale di questa tesi è di creare un meta-compilatore che non soffra degli svantaggi di tale inefficienza. La tesi presenta un meta-compilatore chiamato "Metacasanova" che riduce il costo di sviluppo di un compilatore, generando allo stesso tempo codice eseguibile efficiente.

La tesi inizia ad analizzare gli schemi ricorrenti dell'implementazione di un compilatore, per definire una serie di requisiti per Metacasanova. In seguito spiega l'architettura del meta-compilatore e fornisce degli esempi d'uso implementando un piccolo linguaggio imperativo chiamato C--, seguito dalla reimplementazione di un particolare ed esistente linguaggio di dominio specifico, chiamato Casanova, che è stato creato per essere utilizzato nell'ambito dello sviluppo di videogiochi. La tesi presenta un modo innovativo di ottimizzare le prestazioni del codice generato per mezzo di funtori e dimostra l'effetto di questa ottimizzazione comparando l'efficienza del codice generato da Casanova con e senza questa ottimizzazione. Infine, la tesi mostra i vantaggi di avere un meta-compilatore come Metacasanova utilizzandolo per estendere la semantica di Casanova in modo da poter definire giochi multigiocatore in rete.

Firma dello studente
