

# CATpiler

## A Compiler for the LOLCode Language

Stephanie Stroka



25. Januar 2010

# Besondere Eigenschaften von LOLCode

- Esoterische Programmiersprache
- Keywords können aus mehreren Wörtern bestehen
- Prefix statt Infix-Operatoren
- Typenangabe ist optional
- Operationsergebnisse werden in globaler Variable **IT** gespeichert

# Einschränkungen von LOLCode Spec1.2

- Keine einzelnen Character
- Keine Arrays
- Keine Records

# CATpiler - Grundlegendes

- Erster verfügbarer Compiler für LOLCode
- Geschrieben in Java
- Entwickelt für MIPS Architekturen (RISC)
- Generiert MIPS Assembler Instruktionen in Text-Form

# Besonderheiten des CATpilers

- Verzögerte Typ-Erkennung
- Lazy Loading
- Überspringen von Verzweigungen und Schleifen, die nie erreicht werden
- I/O Operationen durch syscalls
- Separate Compilation
- Compiletime Garbage-Collection

# Separate Compilation

- Rekursives Parsen nach imports
- MIPS Code wird in .cat files gelegt
- Header der .cat files gibt Auskunft über benötigte Imports und über implementierte Funktionen
- Linker kopiert alle .cat files in ein .asm file

# Completetime Garbage-Collection

- Heap wird initial in gleich große Blöcke geteilt
- Anfangsadressen werden als globale Variablen gehalten
- Es können ein oder mehrere Blöcke zugeteilt werden
- Bei der Zuteilung wird ein Referenzcounter erhöht
- Parser erhöht beim Übergabe an Funktionen den Referenzcounter
- Parser dekrementiert beim Verlassen von Funktionen den Referenzcounter
- Sobald Referenzcounter = 0 ist, werden die Blöcke als *unused* markiert

# Zusätzliche Informationen

- 166 Tests (3310 Code-Zeilen)
- Github Repository:  
`http://github.com/StexX/catpiler`
- Informationen über LOLcode:  
`http://lolcode.com/`