Vignettes

David Notkin
Autumn 2008

Various compiler-related topics

- Decompiling (reverse engineering)
- Obfuscation
- Syntax-directed editing
- Tools
- IDLs (intermediate description languages)
- JIT
- Multicore
- Attribute grammars
- ...

Decompiling (reverse engineering)

Source Program  ➔  Compiler  ➔  Target Program

Source Program  ➔  Decompiler  ➔  Target Program

Why might you want to do this?

Y2K: Year 2000 Projects May Be Overlooking Millions of Lines of Missing Computer Code

Edge: Work-Group Computing Report, April 5, 1999

Another reason

How? (Kind of roughly sort of the reverse of compiling)

- Parse binary (often called loading)
  - Often can infer the architecture (if needed), the main entry point, etc.
  - Goal is to produce assembly language
- Disassembly
  - Convert assembly language to intermediate language
  - May exploit idioms aggressively
- Analysis
  - Program analysis: e.g., combine binary expressions into higher-level constructs
  - Type analysis: infer types of variables
  - Structure analysis: find control structures
- Generate high-level code
http://www.dmoz.org/Computers/Programming/Disassemblers/

1 December 2008

• AVATAR - A disassembler/patcher/code-explorer for PA-RISC based HP-UX systems, by Allegro Consultants, Inc.
  • the dc: Disassembler - an X11 disassembler. Written in C as a part of X, intended to support multiple CPUs, operating systems, and file formats. Source and binaries public.
  • The dc: Disassembler - its disassembler that can disassemble code on the JVM platform to an assembly language.
  • DSP5600x - A 5600x disassembler by Miloslaw Smyk.
  • FARGDIS - Fargo Disassembler for TR/90 DSS version, by John Grafton.
  • High Level Assembler and ToolKit (HLM) - System/38 assembler suitable for MVS and VM and VSE (HLSM) including disassembler, by IBM.
  • IDA Pro - The multi-platform, multi-OS, interactive disassembler, by DatelLiance.
  • McConroy's disassembler - a Mac application that disassembles the Macintosh ROM or any 68K or PowerPC code, by John Grafton.
  • MELPS7700 Disassembler - Sato H/Kashima.
  • Misosys Disassembler - Tim Mann's TRS-80 Page includes Misosys Disassembler, aka PRO-DUCE.
  • Open Source Code Engineering - An open community site offering a number of services including blogs, forums, download and reference libraries.
  • Reverse Engineering Compiler - Program that tries to make source code (C) from binary, multplatform. There are MIPS disassembler too, by Giampiero Caprino.
  • SST Global Decompilers - Decompilers for IBM midrange systems.

Reverse engineering

• A generalization of decompiling
• Accept a lower-level model as input, produce a higher-level model
  – Common example: inferring UML from source code
• Roughly same motivations, etc.
• Inference may be more difficult

Legality?

• So, is decompiling/reverse engineering legal, illegal, or somewhere in between?
• Why?

It depends

• Copyright law applies to most programs: the owner of the copyright generally has a set of exclusive rights, including making copies (including those in memory)
• Decompilation and reverse engineering usually requires making of copies, so it requires permission of the copyright owner
• However, if decompilation is needed to attain interoperability, US and European copyright laws permit it in some cases
  – One US example allowed a company to decompile to get around a software locking mechanism for a Sega game console

Europe: 1991 Software Directive

• Explicit right to decompile for interoperability only if:
  – The program must be properly licensed
  – Decompilation must be necessary and the burden is on the decompiler to show that manuals, API documents, etc. is insufficient
  – The process must be as confined as much as possible to the parts relevant to interoperability.
  – Decompiled information may only be used for the specific interoperability purpose and may not be shared

Obfuscation

• Making source (or intermediate) code very hard to read, usually intentionally
• Why?
  – Intellectual property protection
  – Malicious intent
  – Reduce security exposure
  – Minimization
• Why not?

"..."
Syntax-directed editing

- Why have programmers take their unambiguous ideas about a program, and then enter text that is then parsed using a potentially ambiguous process?
- Why not provide an editor that is knowledgeable about the abstract syntax (and some semantics) to ensure that programs are entered unambiguously and without syntactic error?
- Research around 1980’s: Gandalf, Cornell Program Synthesizer, Mentor, …

Didn’t work

- Syntax is not generally the problem for experienced programmers
- Integration with other tools (debugger, compiler, etc.) was much harder
- Some new problems appeared: for example, searching for unparsed text that didn’t appear in the AST
  - …
- But influence modern environments in several ways