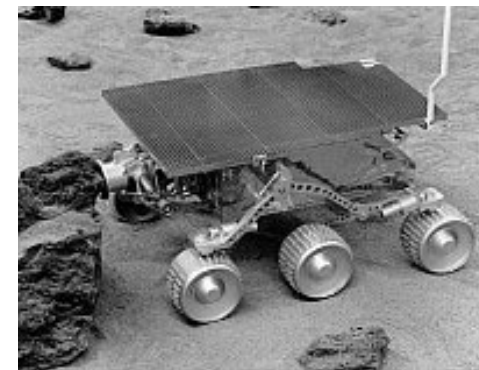


High-Level Optimizations for Low-Level Software

**John Regehr
University of Utah**

Embedded Systems

- ◆ Most new microprocessors are embedded
 - Consumer electronics
 - Vehicle control systems
 - Medical equipment
 - Sensor networks

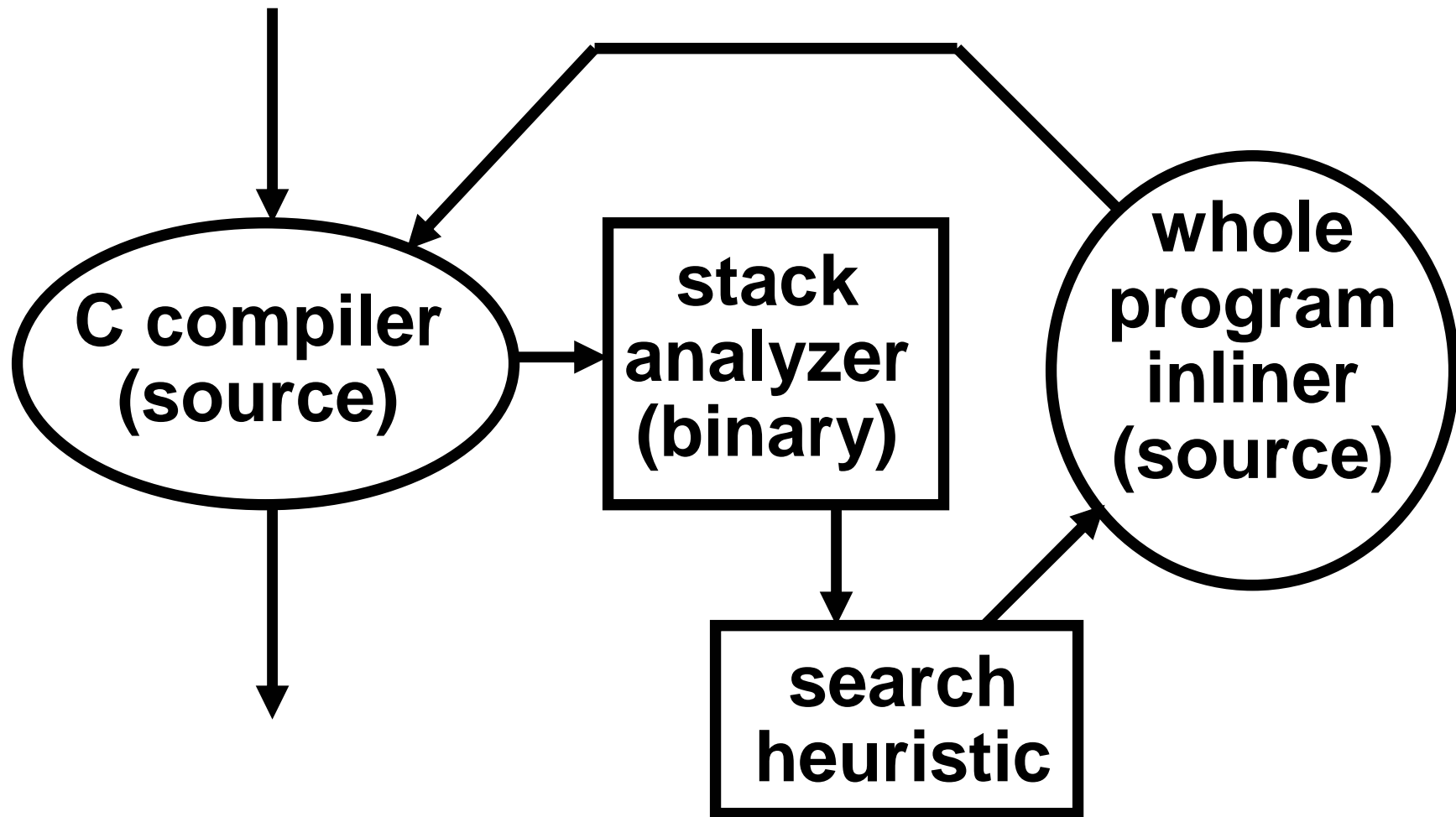


- ◆ **Compilers can make stupid code run fast**
 - **To a limited extent!**
- ◆ **This work: Help stupid embedded code run fast and use less memory**
 - **Coarse-grain program transformations**
 - **Explicit support for tradeoffs**

Strategy

- ◆ **Integrate analysis and transformation tools**
 - **Not hacking the compiler**
- ◆ **Common analysis result formats**
 - **Callgraph**
 - **Task/thread decomposition**
 - **Exclusive modes**

Reducing Stack Depth [EMSOFT 2003]



Result

- ◆ **Averaged over a bunch of TinyOS kernels...**
 - **60% reduction in stack requirements compared to no inlining**
 - **32% reduction compared to whole-program inlining not aimed at reducing stack depth**

Research Challenges

- ◆ **Maintaining invariants**
 - **Transformations will invalidate some analysis results**
- ◆ **Avoiding bloat in the trusted computing base**
 - **Embedded developers have a hard time trusting just the compiler**

More info here:

<http://www.cs.utah.edu/~regehr/>