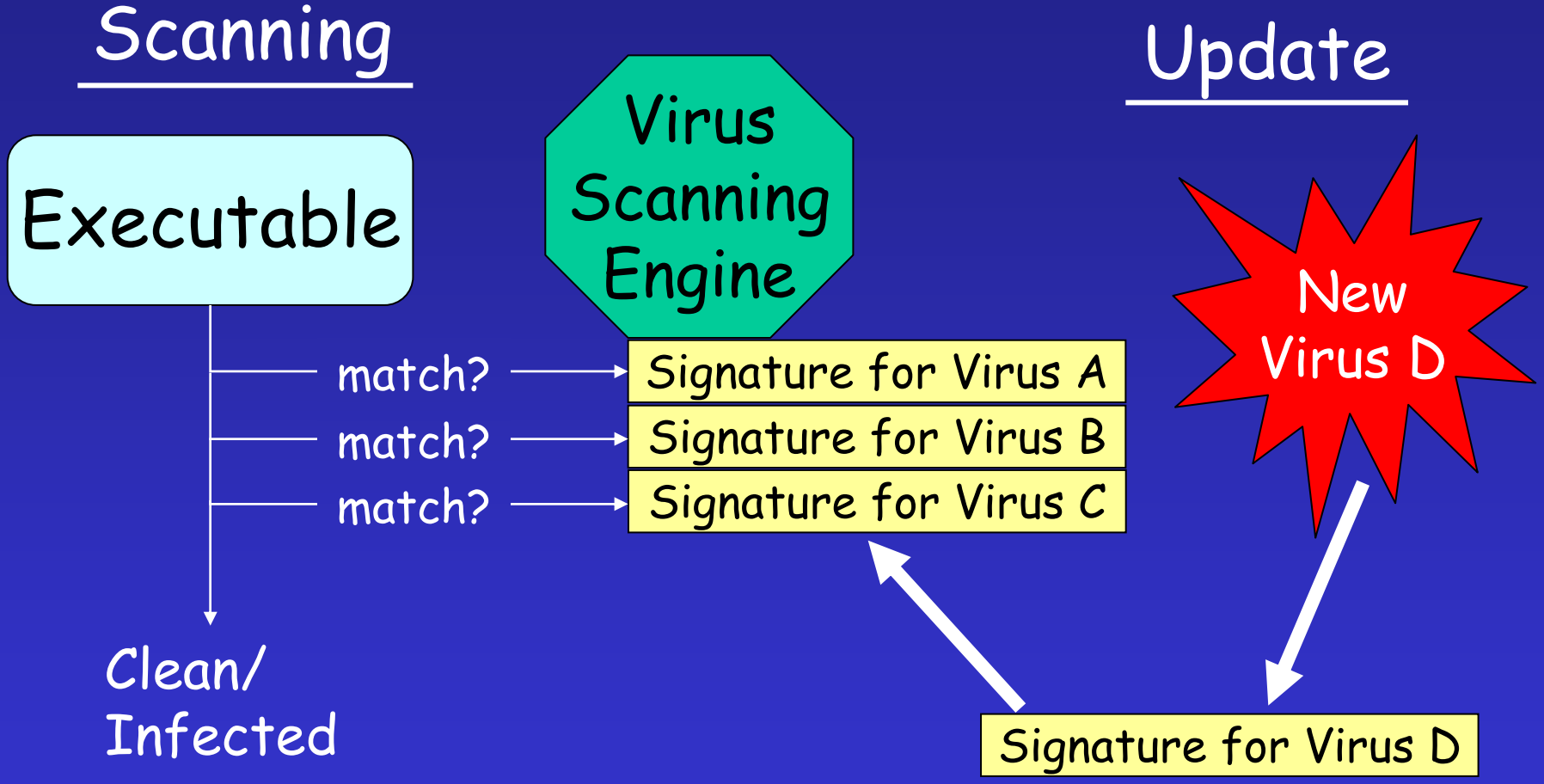
Model Checking for Binaries

Mihai Christodorescu
University of Wisconsin

Virus Scanning

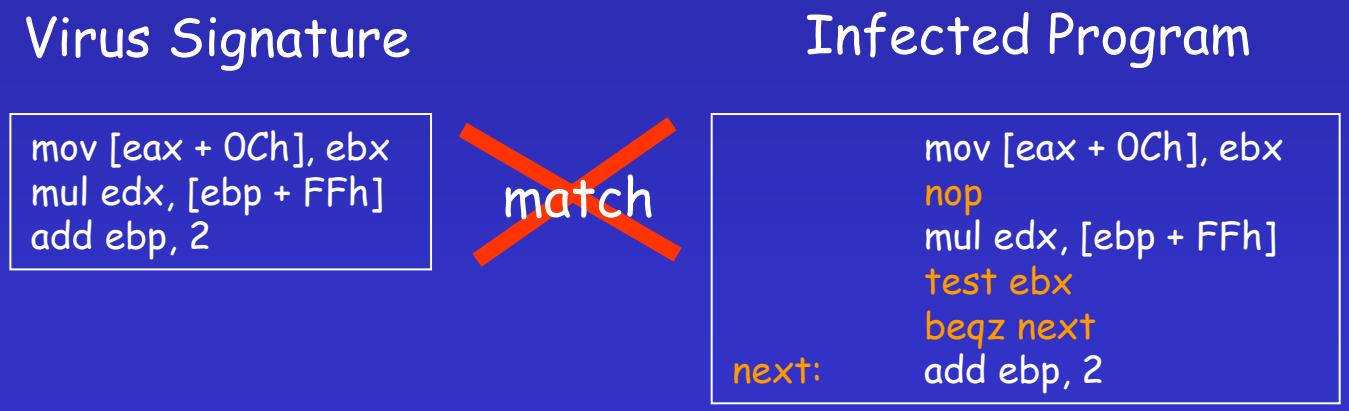
- Viruses
  - Likely to become a bigger problem in the future
- Virus scanners
  - Check binaries for each type of virus
  - Virus signatures

Virus Scanners



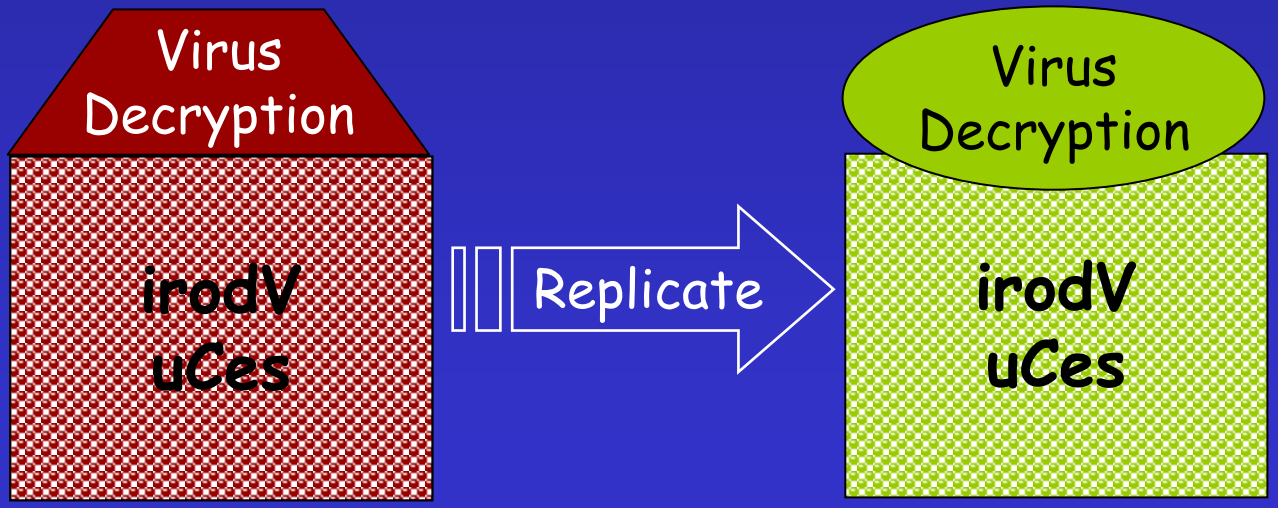
Virus Scanner Limitations

- Signatures are sequences of instructions
  - Small changes to infected program do not trigger a match



Is It Really A Problem?

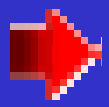
- Yes!
  - Virus writers are getting smarter
  - Viruses are better at hiding themselves



What to do?

- Build a better mouse trap!
- Better detection tool
  - Use static analysis information
  - More flexible

=> Smart Virus Scanner

 Demo 1  Demo 2  Demo 3

Smart Virus Scanner

1. Build a model of the virus code
   - Blueprint of virus behavior

2. Verify that program does not match the blueprint

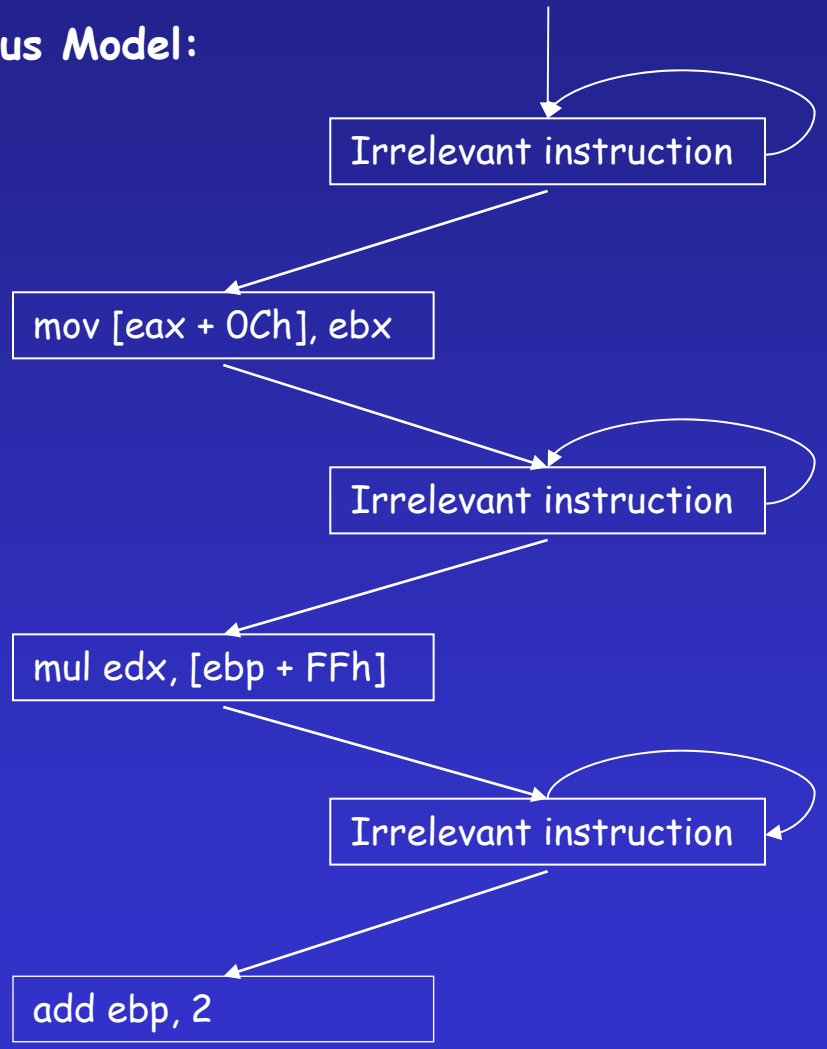
- More powerful

Smart Virus Scanner Example

Virus Signature:

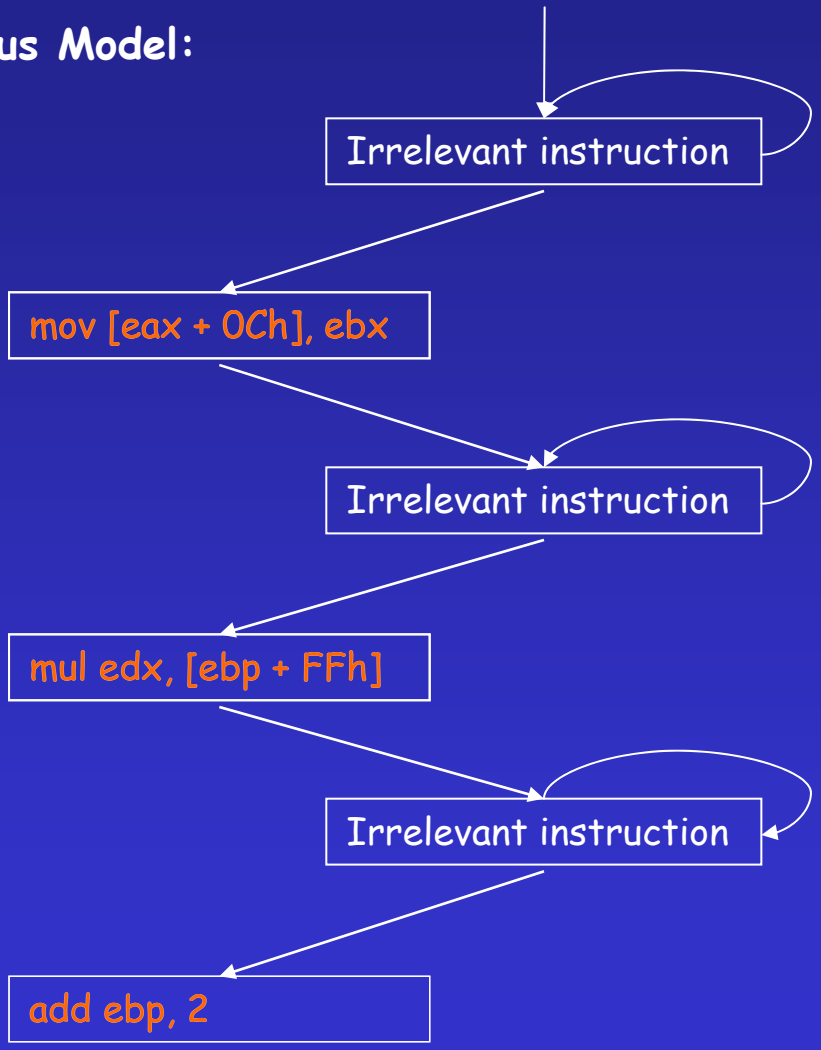
```
mov [eax + 0Ch], ebx
mul edx, [ebp + FFh]
add ebp, 2
```

Virus Model:



Smart Virus Scanner Example

Virus Model:



Program to be checked:

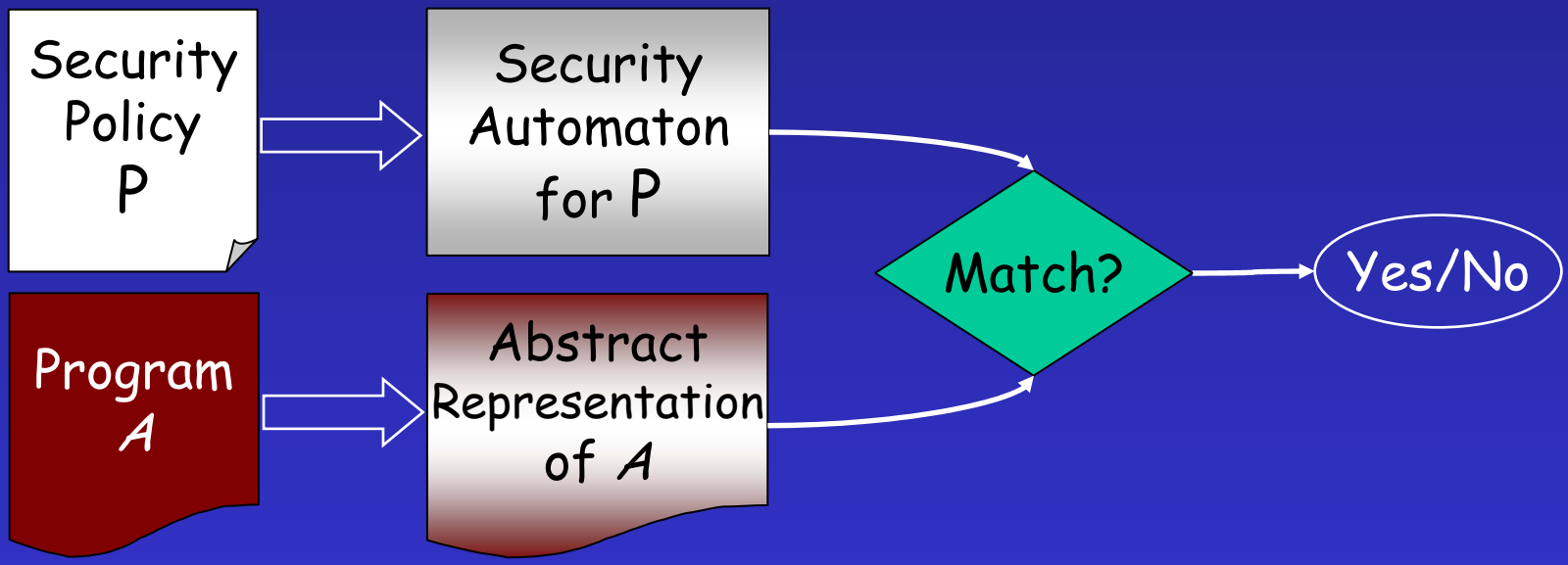
```
            mov [eax + 0Ch], ebx
            nop
            mul edx, [ebp + FFh]
            test ebx
            beqz next
next:       add ebp, 2
```

Model Checking

- Technique for checking program properties

- Build a "model program" that obeys the property
- Compare the program against the model

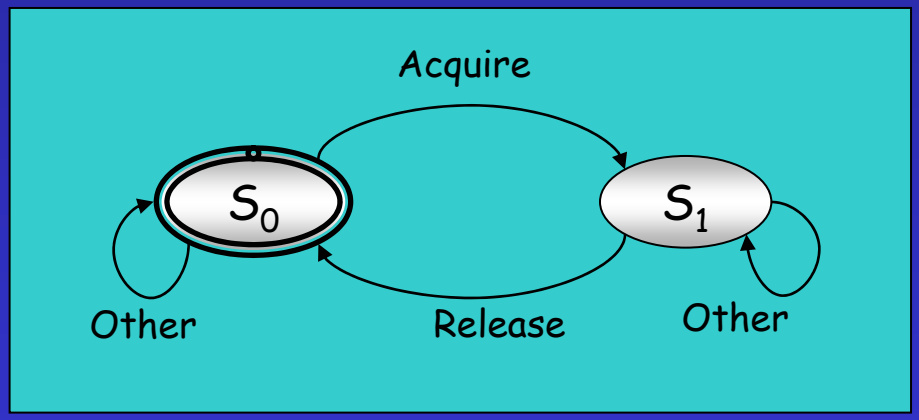
Model Checking



Security Automaton proposed by [Schneider 1999].

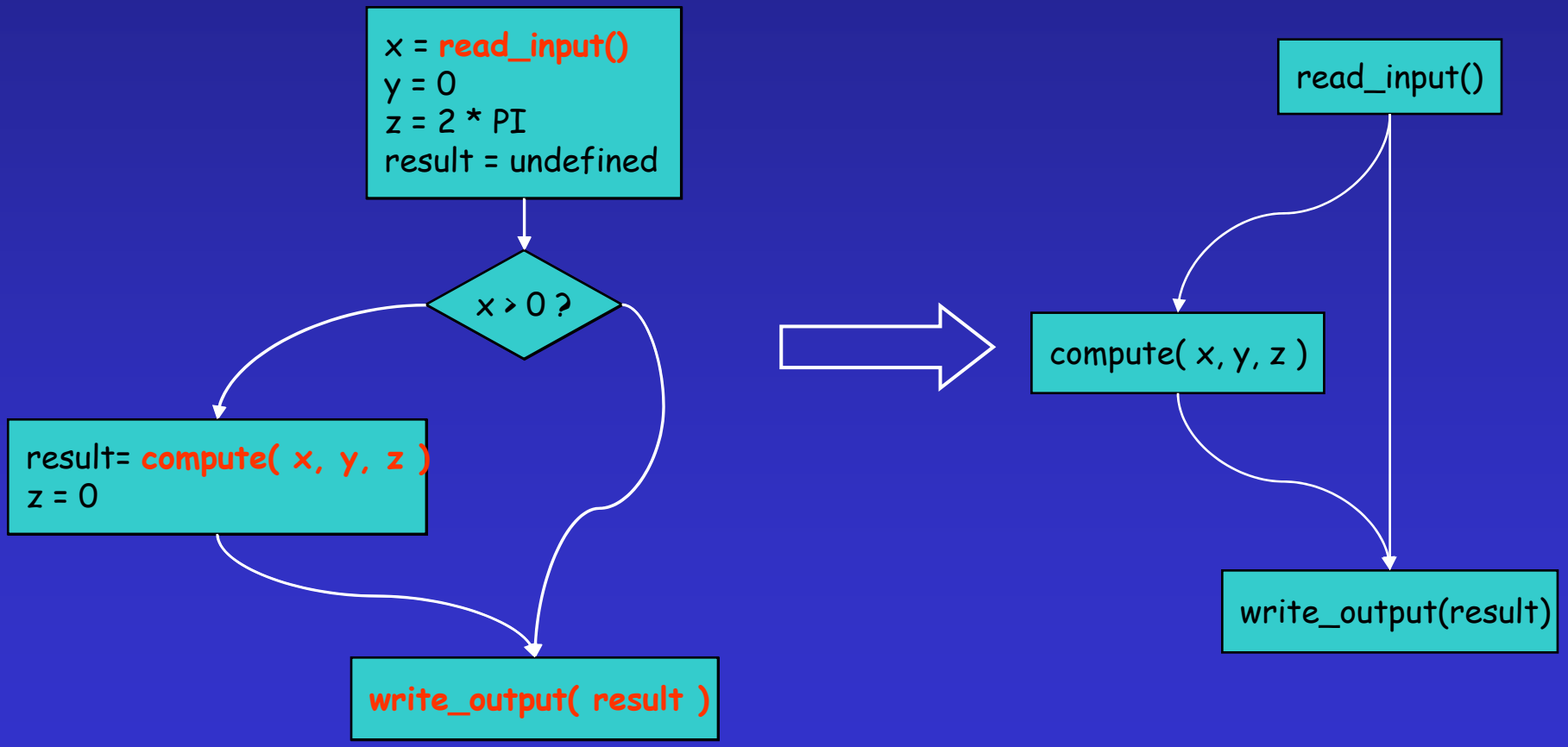
Model Checking

- Security Automaton
  - For policy "Always release a resource after acquiring it."

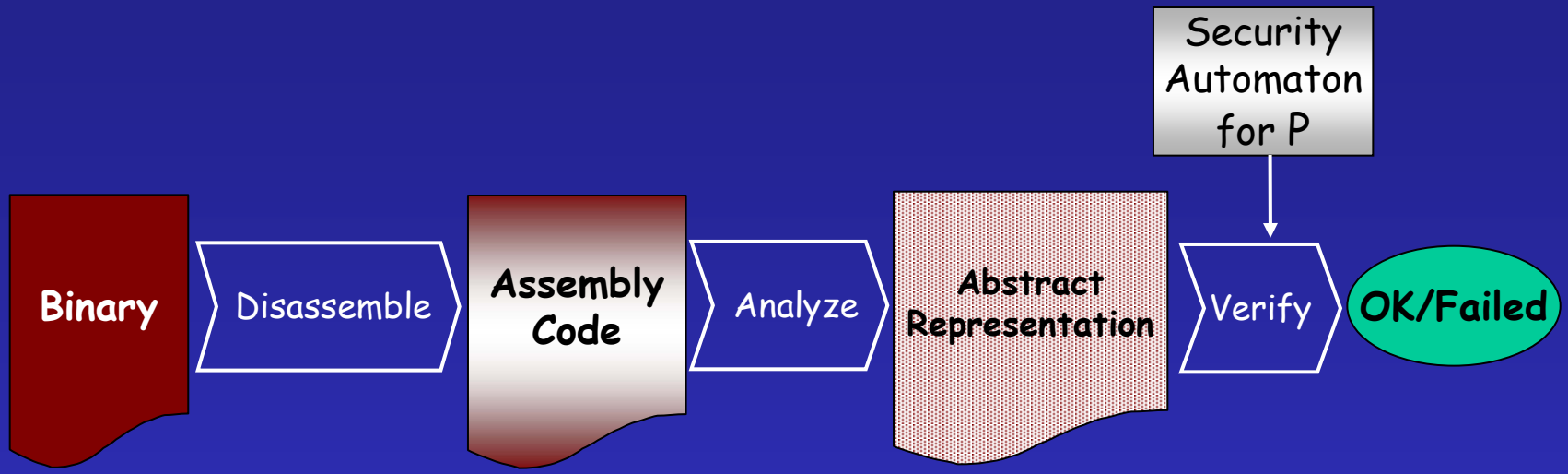


Model Checking

- Abstract Representation



Model Checking for Binaries



IDA Pro

(+ SDK plugins)

Transducer

(on top of CodeSurfer static analysis engine)

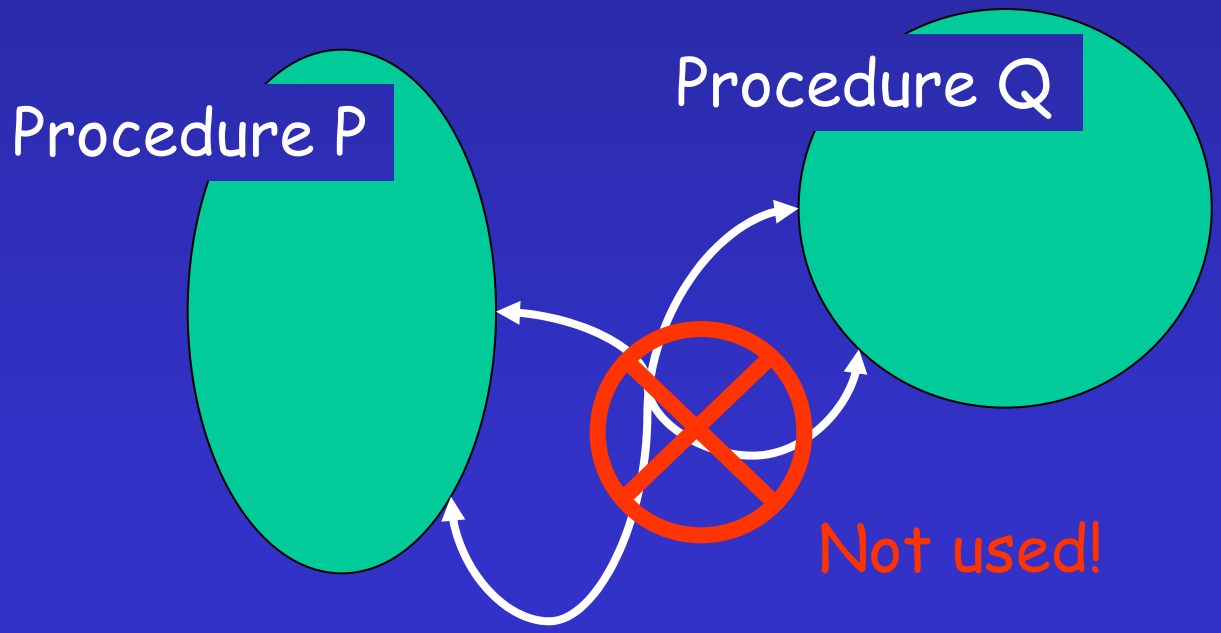
Model Checker

Current Status

- Transducer
  - Works on sequences of instructions (basic blocks) to build the abstract representation
  - Uses only the static analysis information provided by CodeSurfer

Current Status

- Model Checker
  - Intra-procedural only



Static Analysis and Model Checking

- More (precise) information from static analysis can enhance model checking

- Smart Virus Scanner
  - "irrelevant" instructions - better identified by alias/points-to analysis

Future Directions

- Context-sensitive model checking
- Better static analyses
- Model checking for component systems

References

- Schneider, F.B. *Enforceable Security Policies*. TR99-1759, July 27, 1999.

- Dawson Engler, Benjamin Chelf, Andy Chou, and Seth Hallem. *Checking System Rules Using System Specific, Programmer-Written Compiler Extensions.* In Proceedings of the Fourth Symposium on Operating Systems Design and Implementation, San Diego, CA, October 2000. http://citeseer.nj.nec.com/engler00checking.html