Determining the Integrity of Remote System Call Streams

Jonathon Giffin
University of Wisconsin

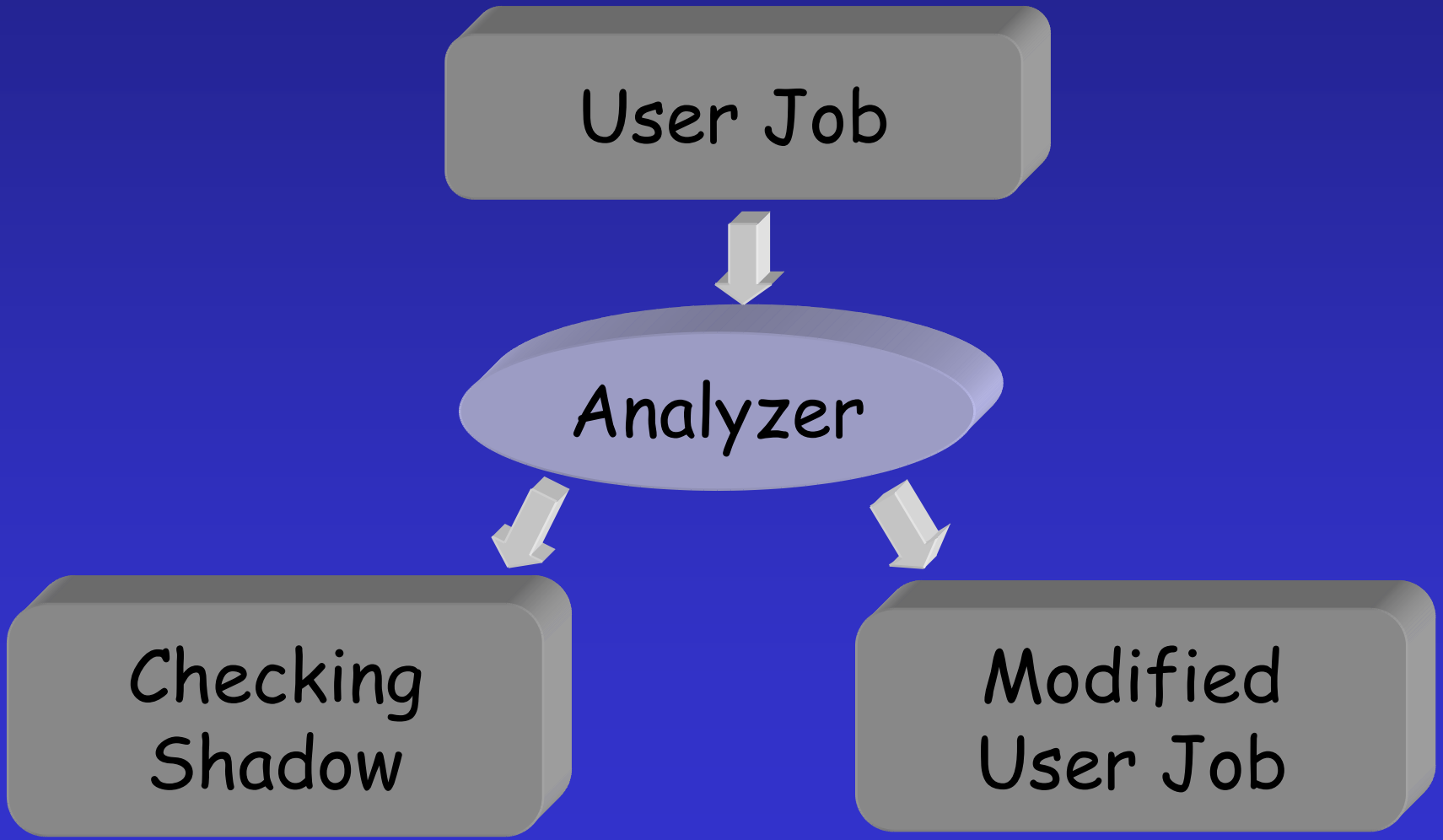
Overview

- Runtime Monitoring
- Model Construction
- Binary Rewriting
- Model Precision

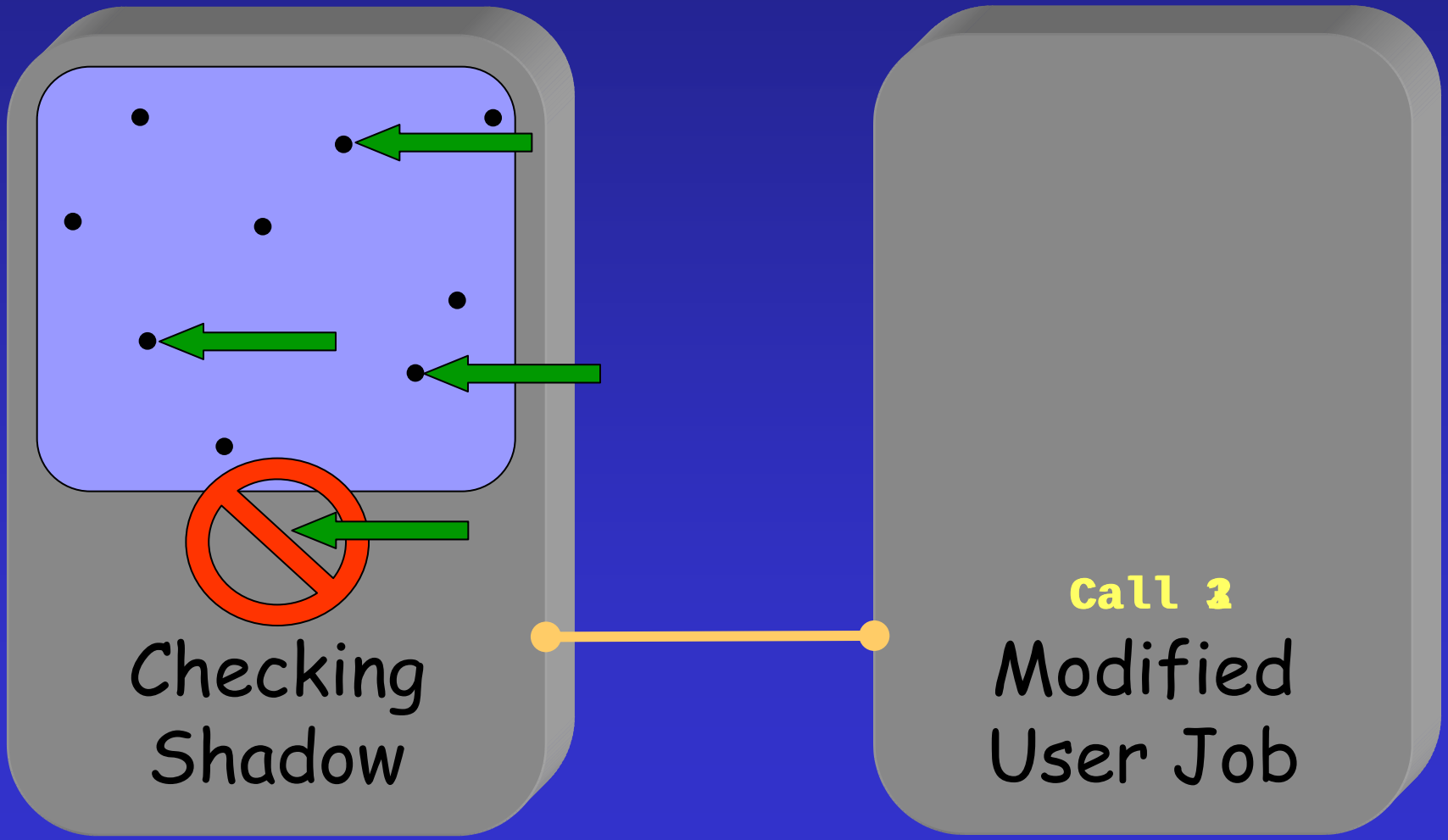
Countering Remote Attacks

- Goal: Even if an intruder can see, examine, and fully control the remote job, no harm can come to the local machine.
- Key technology: Static analysis of binary code.

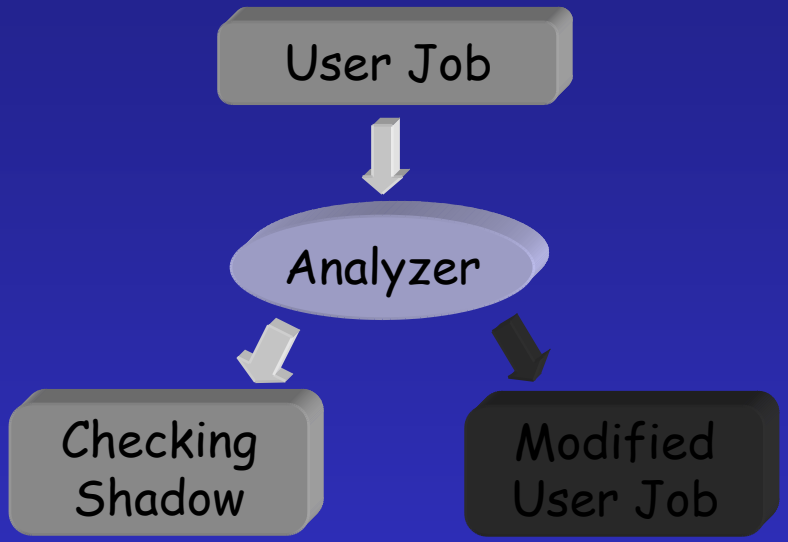
Countering Remote Attacks



Runtime Monitoring



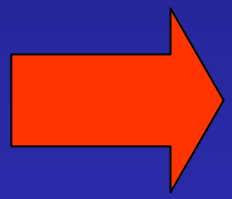
Model Construction

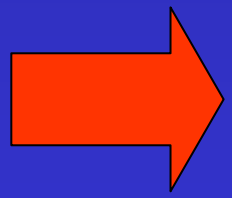


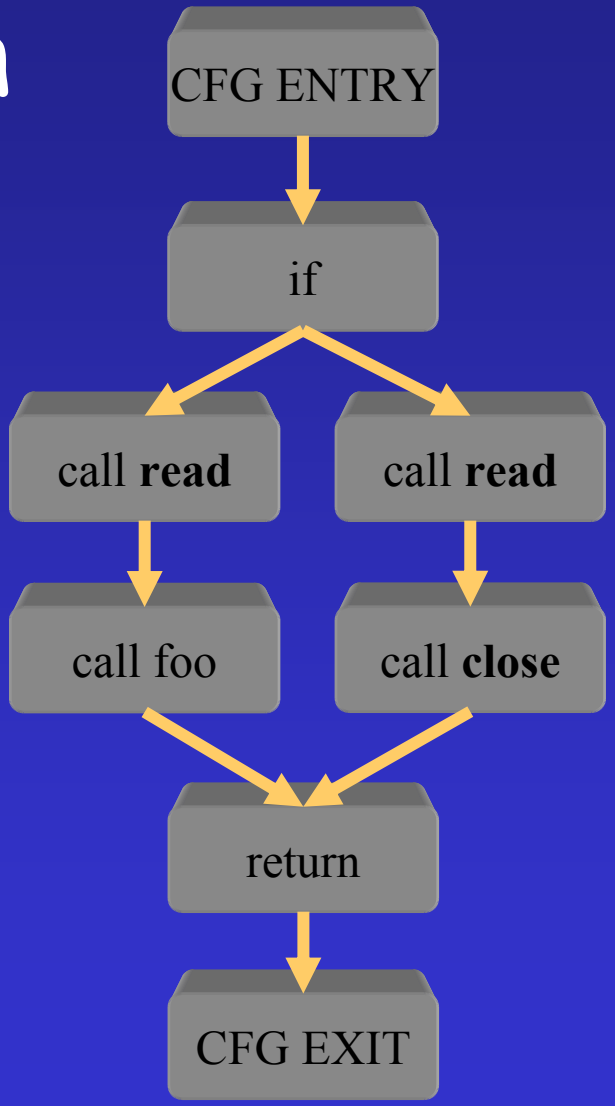
Binary Program → Control Flow Graphs → Local Automata → Global Automata

Control Flow Graph Generation

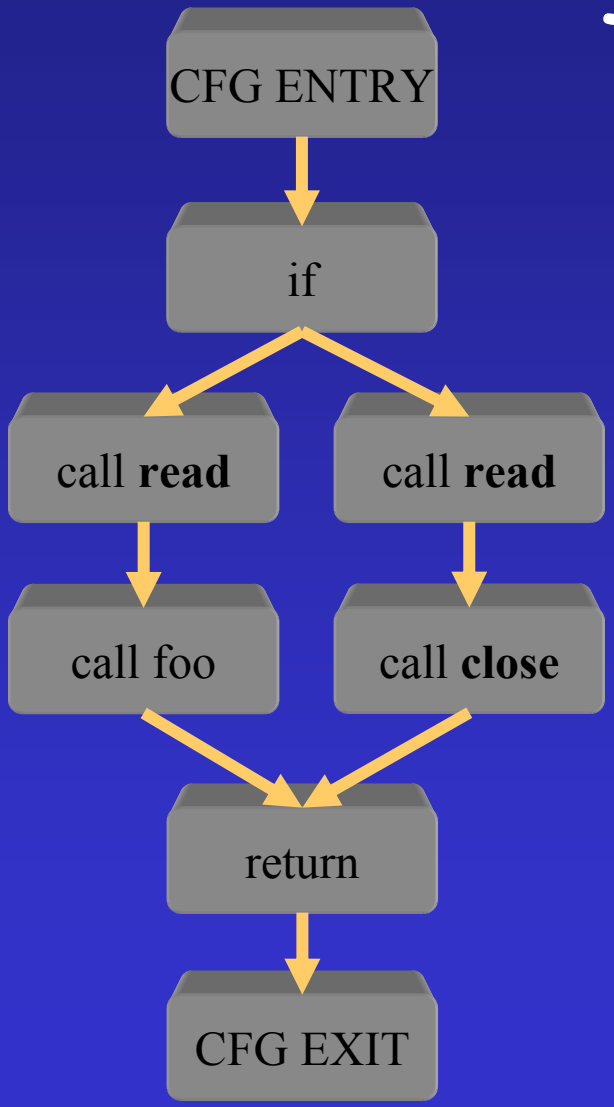
```
function( int a ) {
   if( a < 0 ) {
        read( 0, 15 );
        foo();
   } else {
        read( a, 15 );
        close( a );
   }
}
```

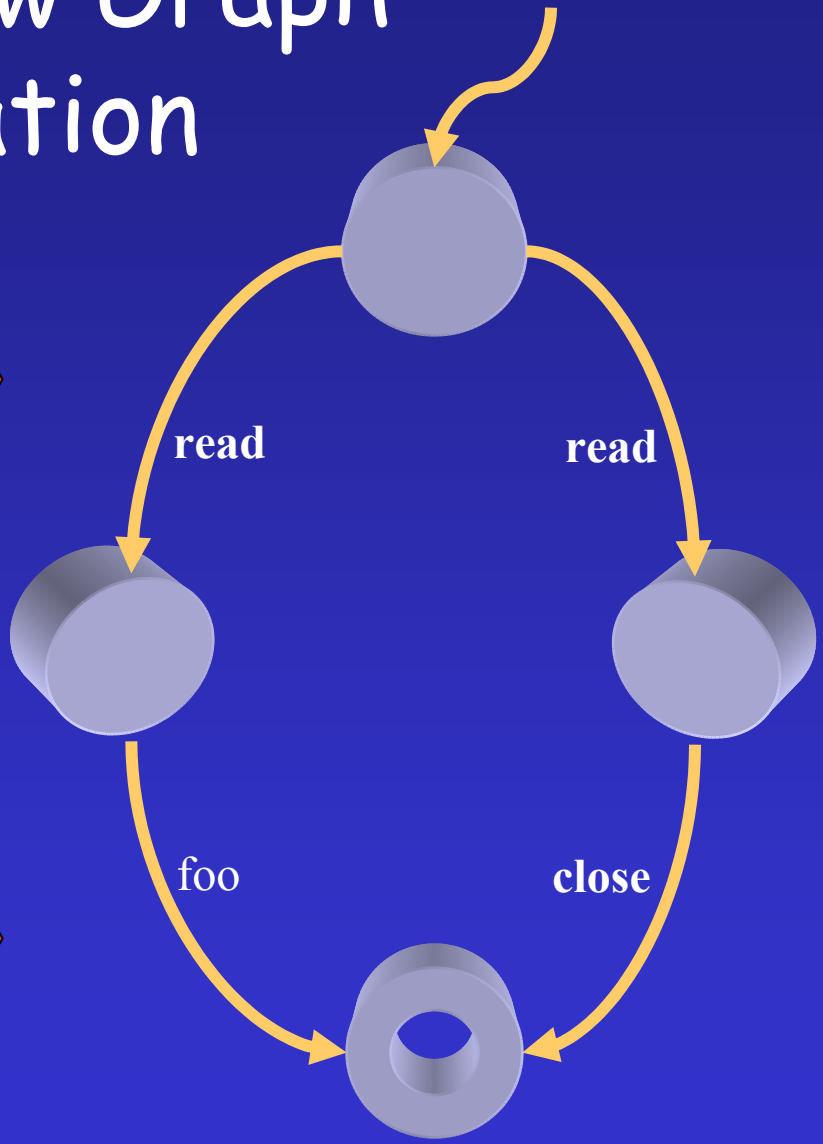






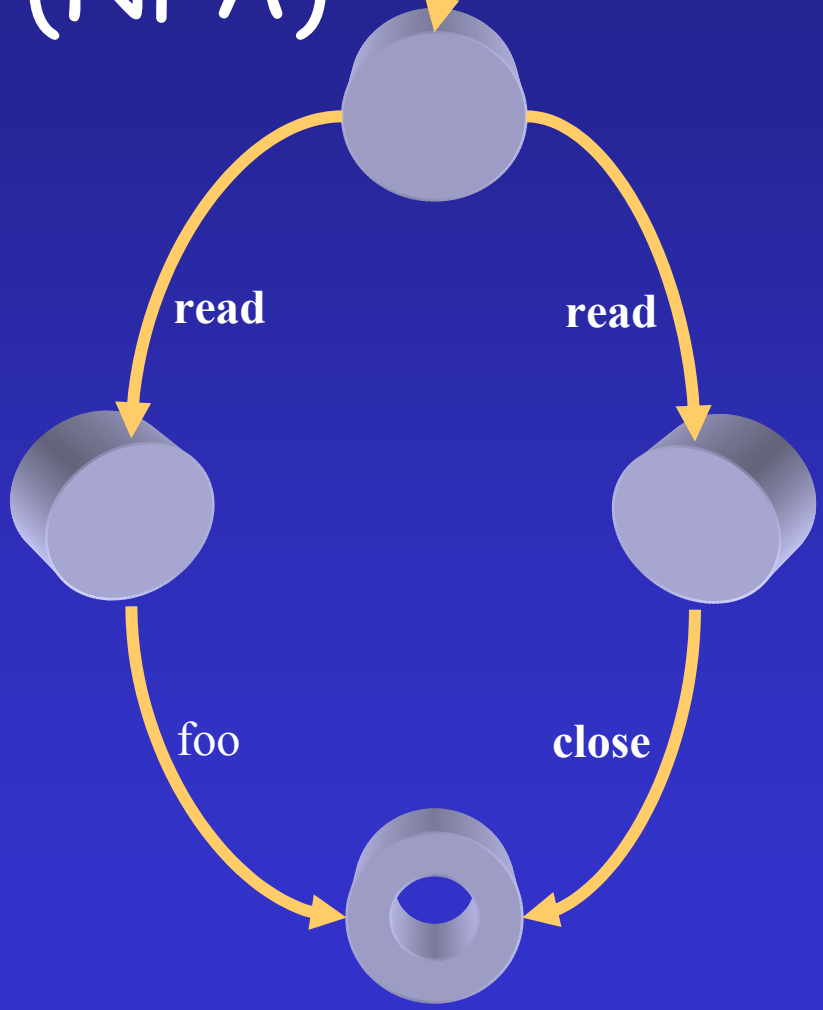
Control Flow Graph Translation



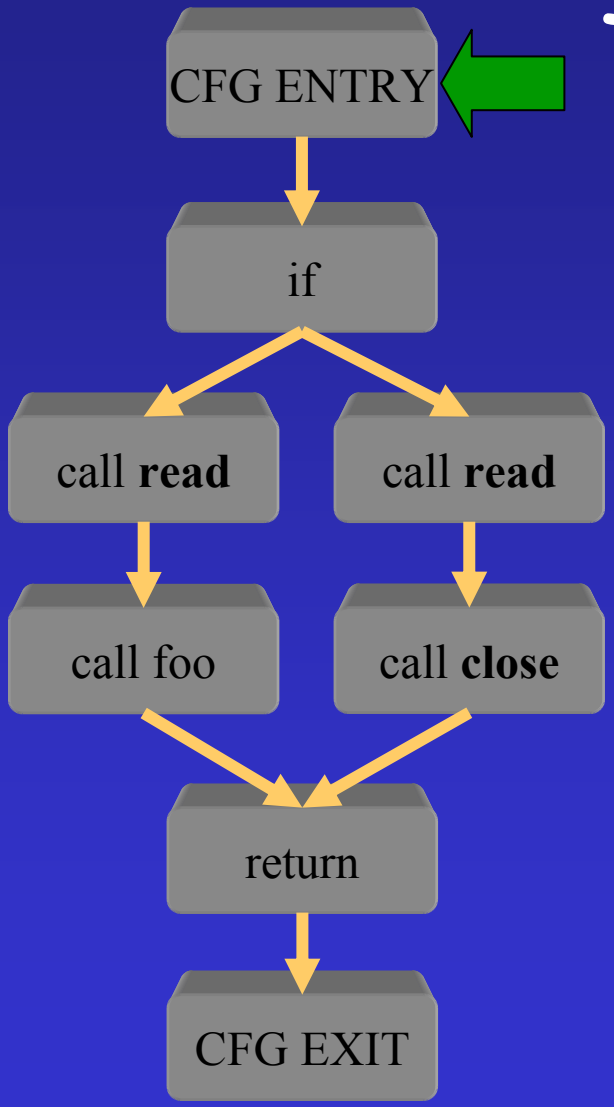


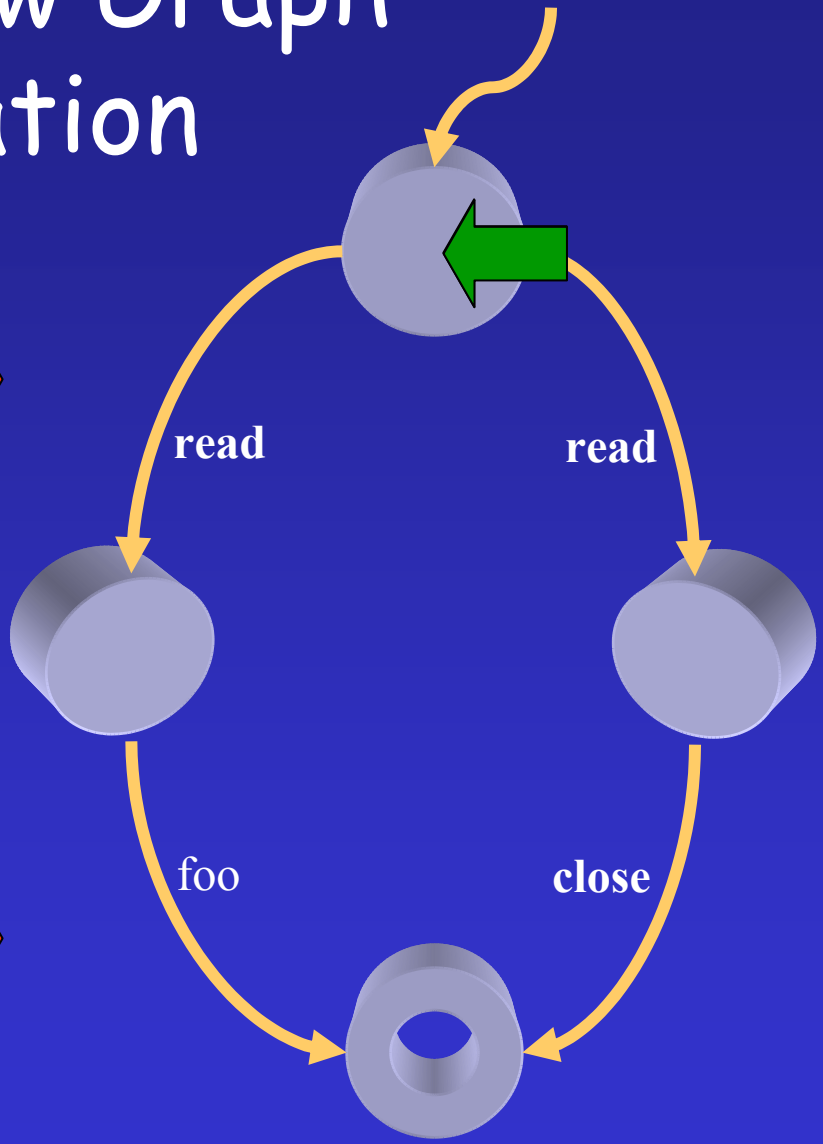
Non-deterministic Finite Automata (NFA)

- Structure
  - States
  - Labeled edges between states
- Edge labels are input symbols – call names
- Path to any accepting state defines valid sequence of calls

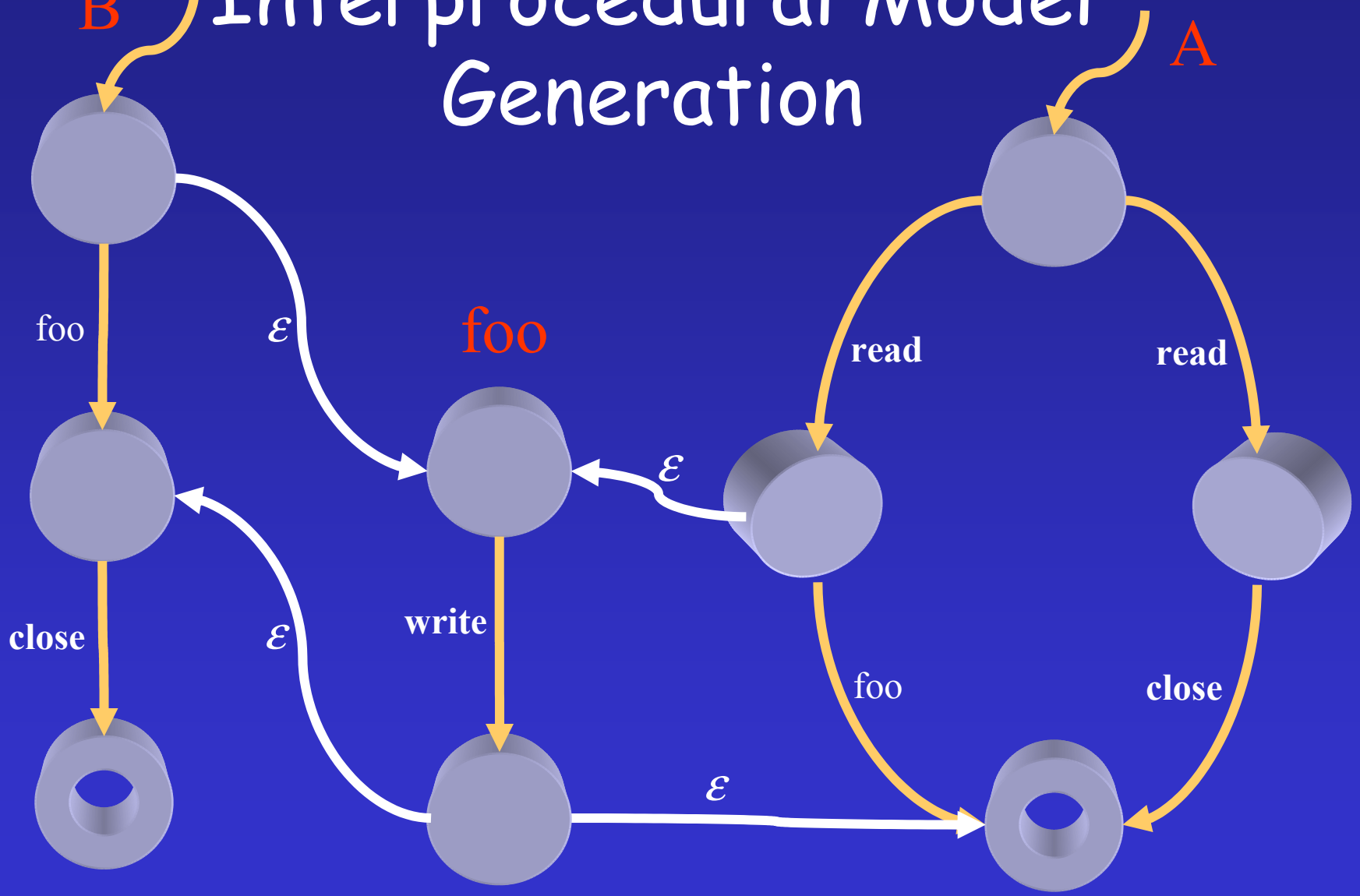


Control Flow Graph Translation

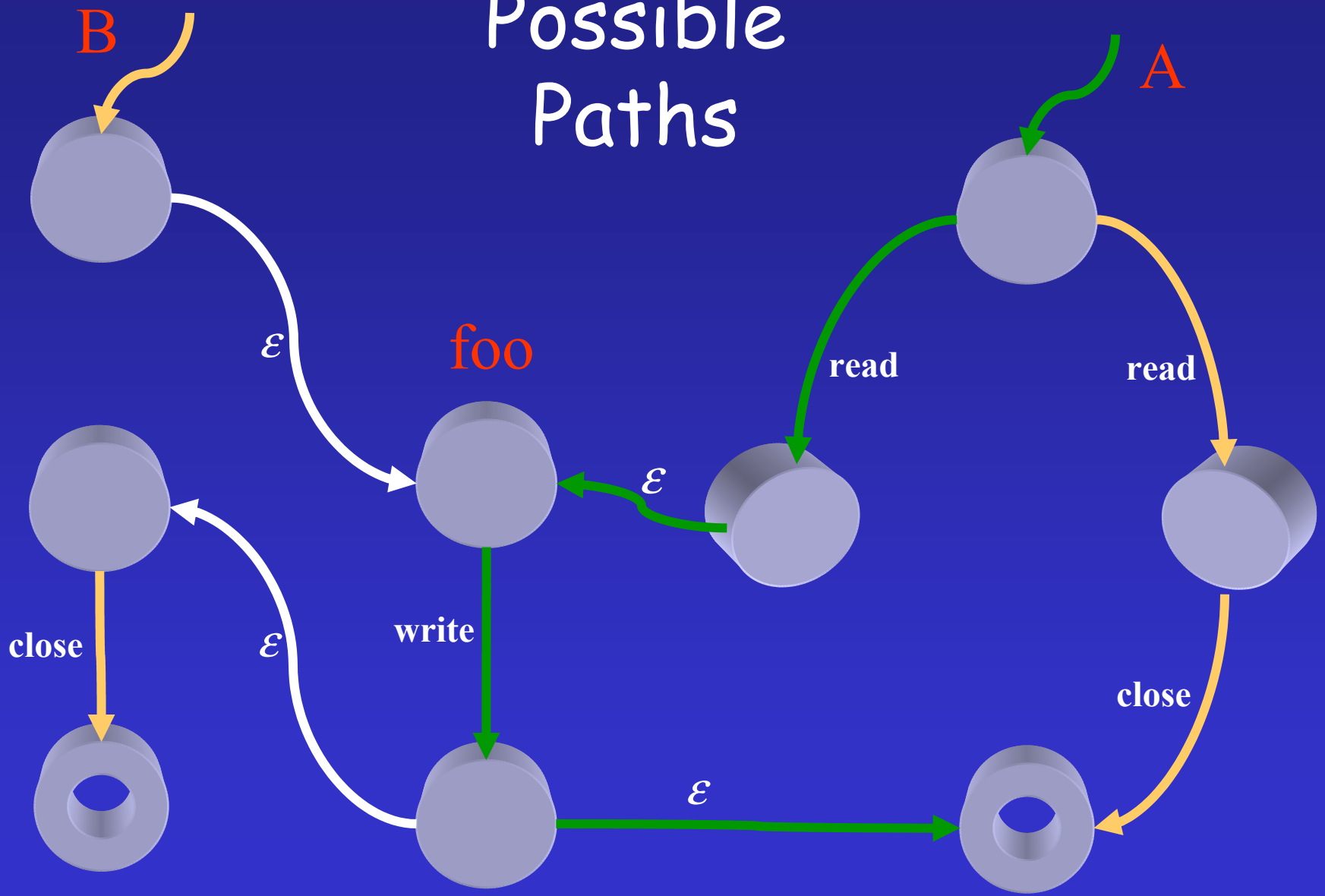




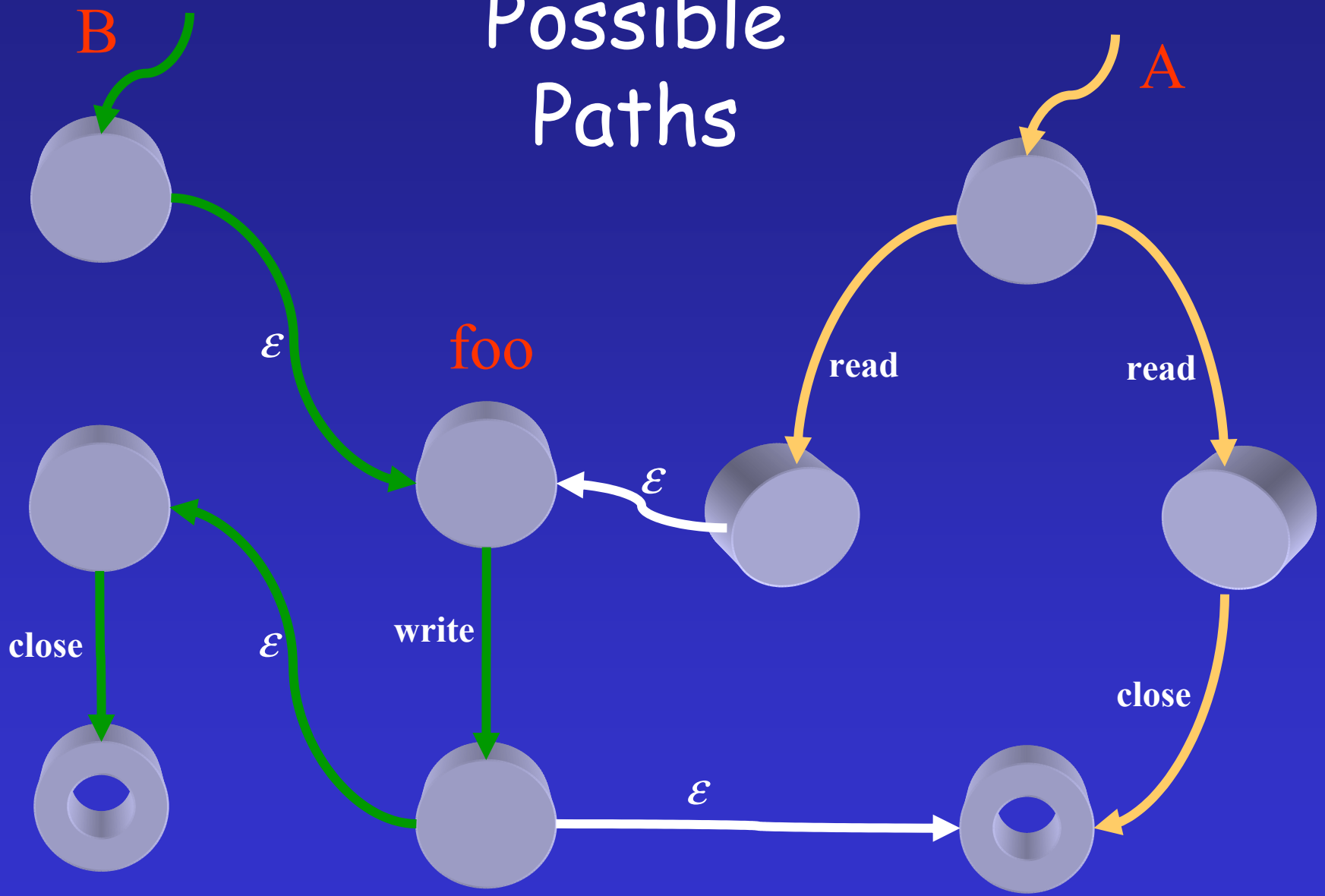
Interprocedural Model Generation



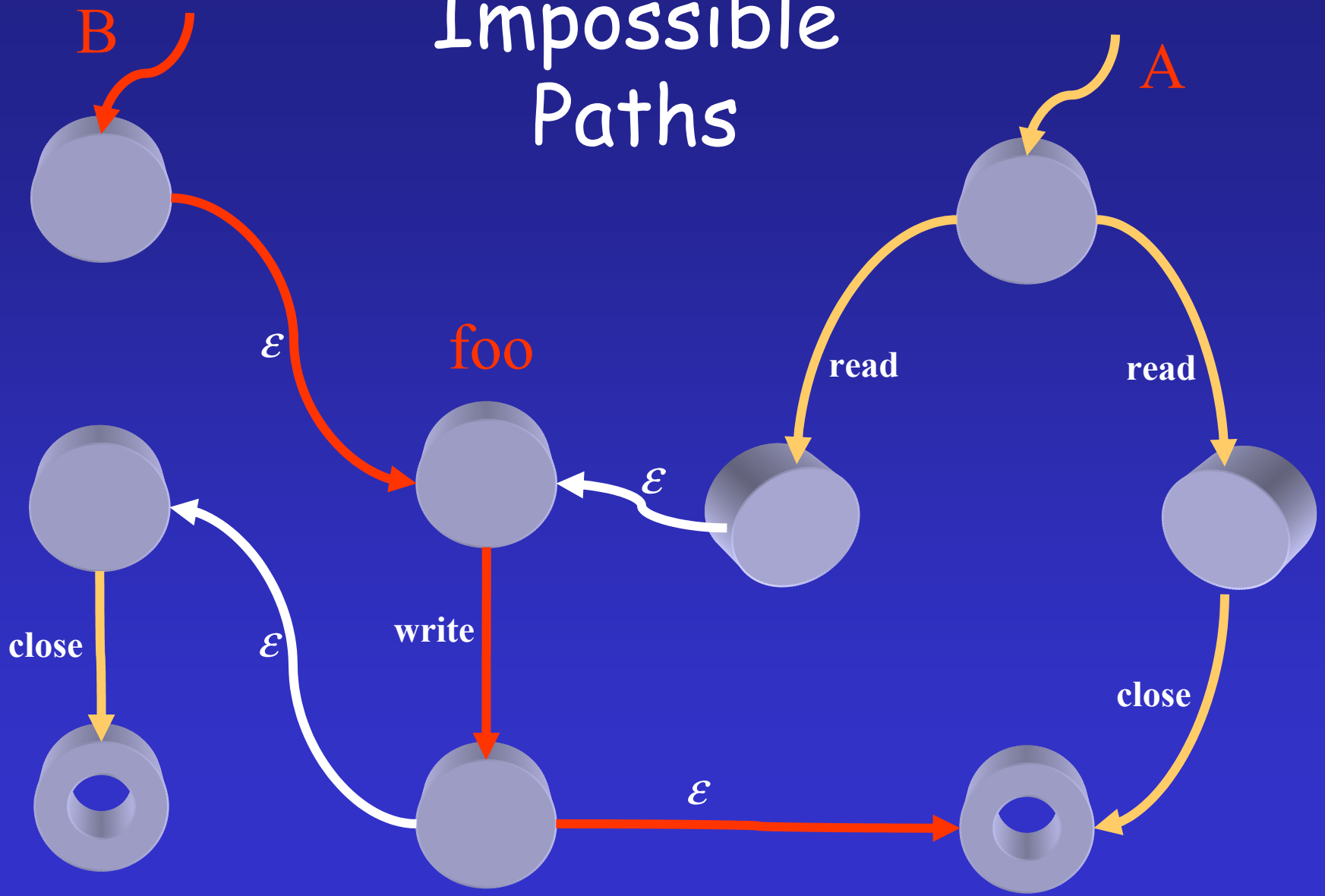
Possible Paths



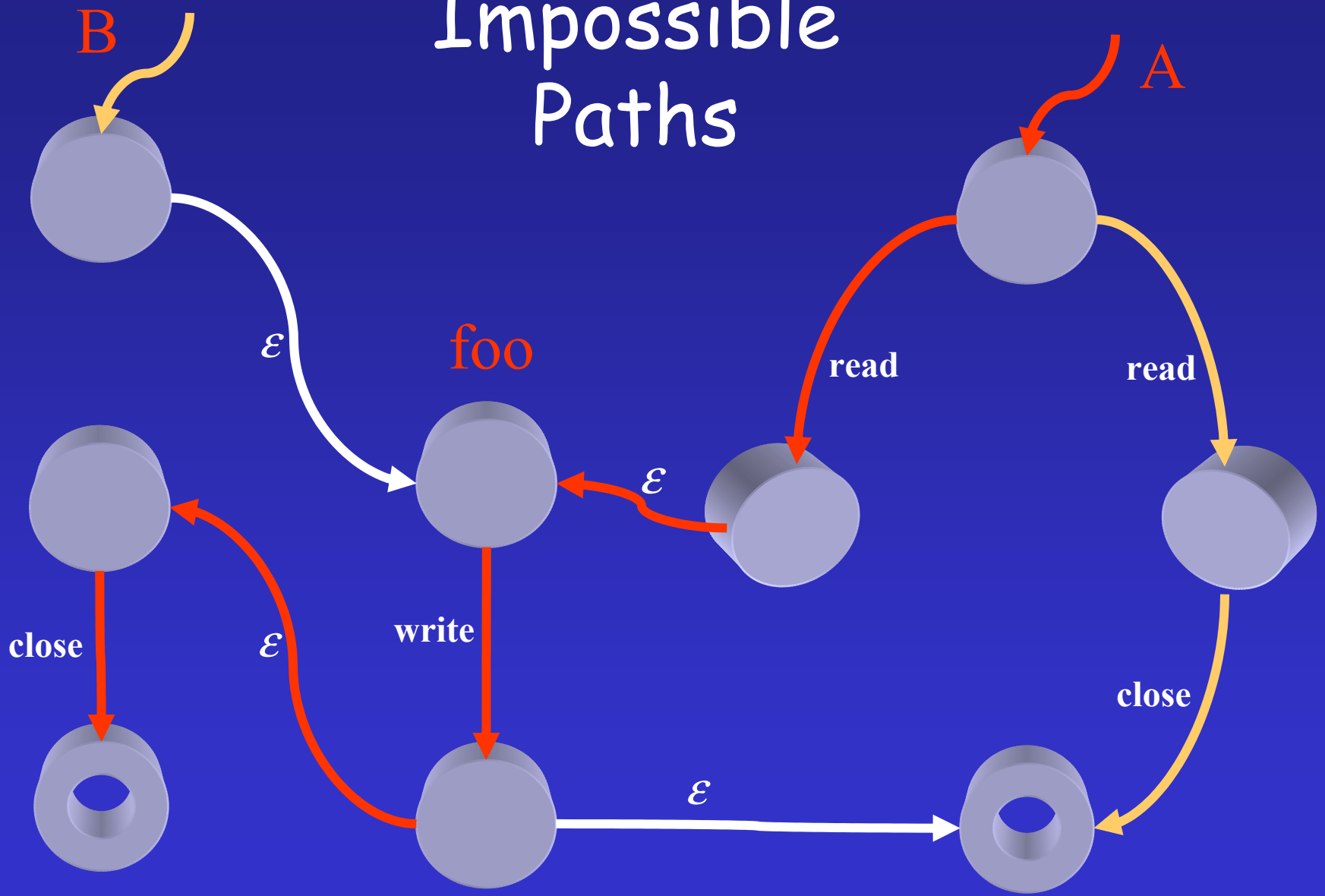
Possible Paths



Impossible Paths



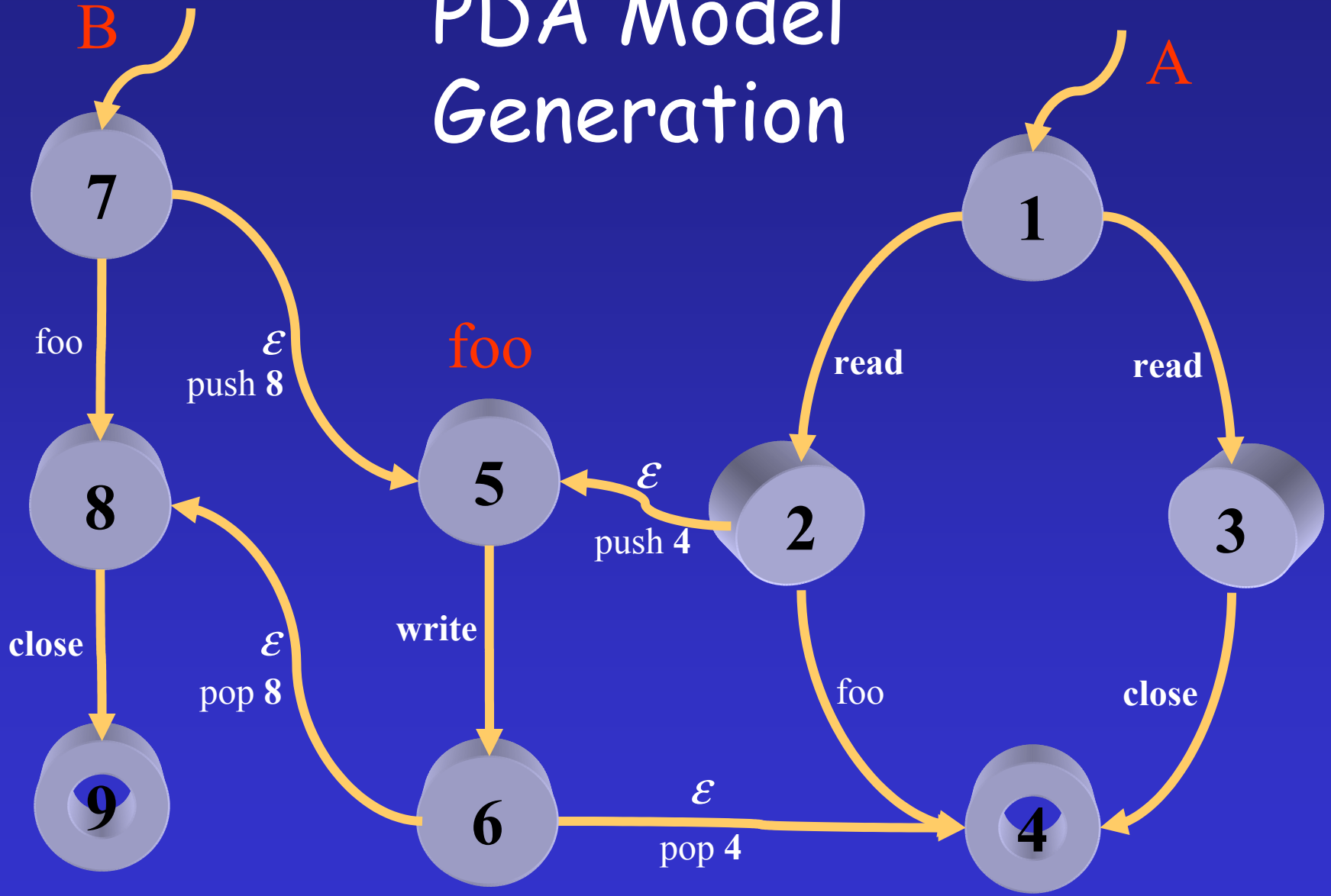
Impossible Paths



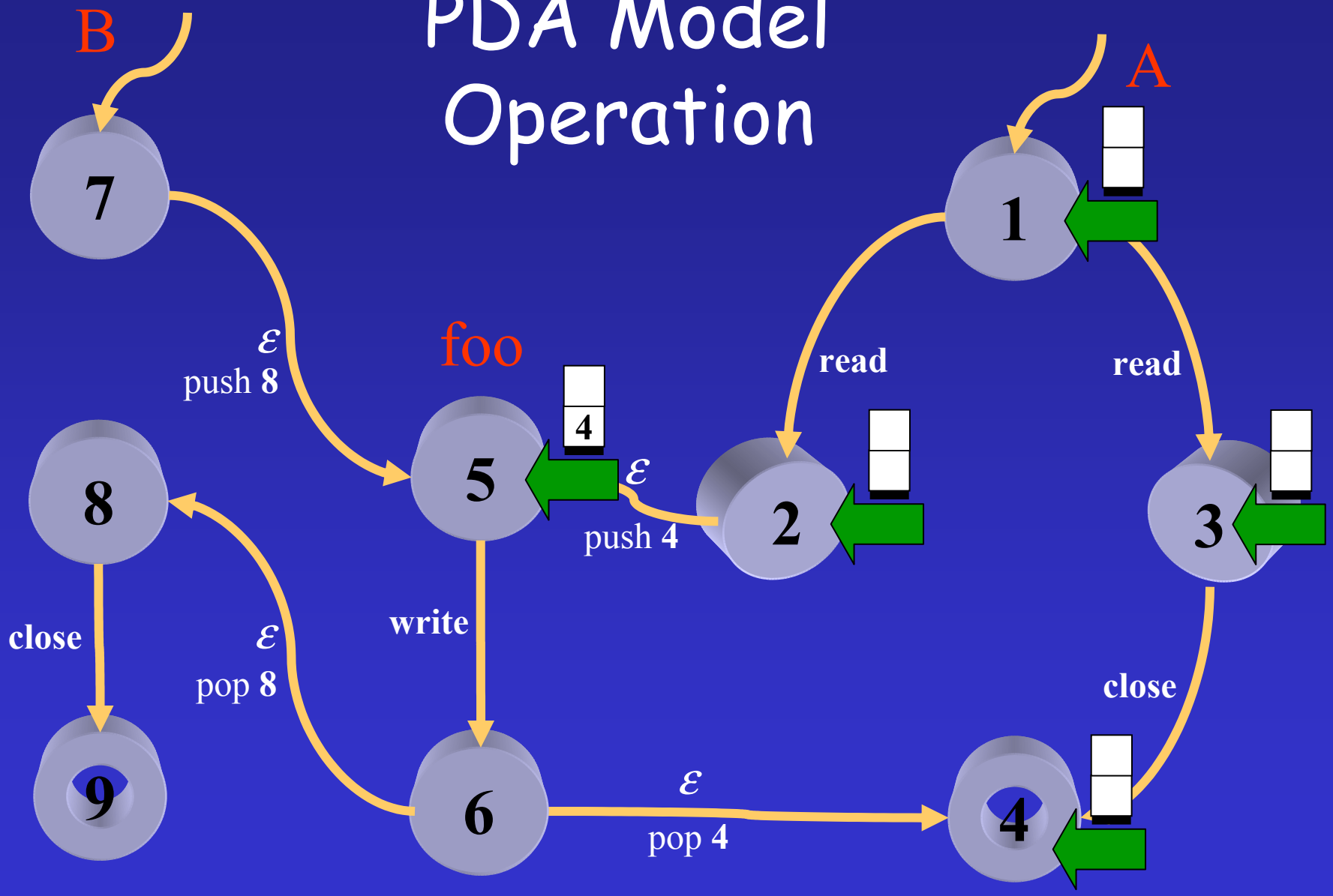
Push-down Automata (PDA)

- NFA has interprocedural imprecision
  - Does not model state of call stack
- Language of application is context-free
- Solution:
  - Splice automata using PDA edges
  - Introduces new challenges

PDA Model Generation

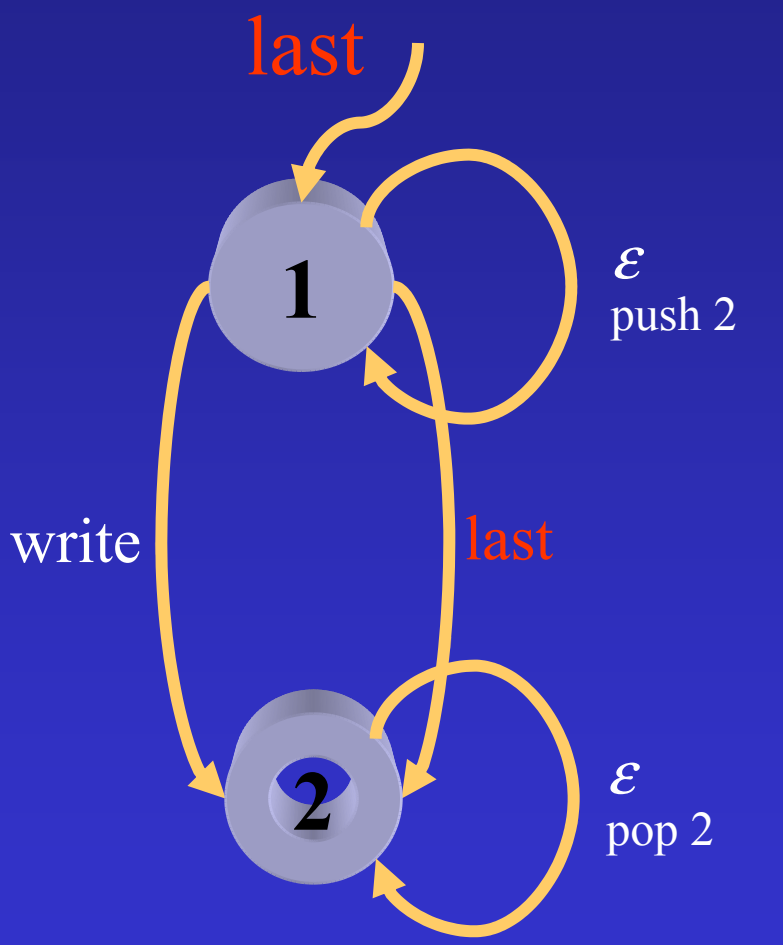


PDA Model Operation



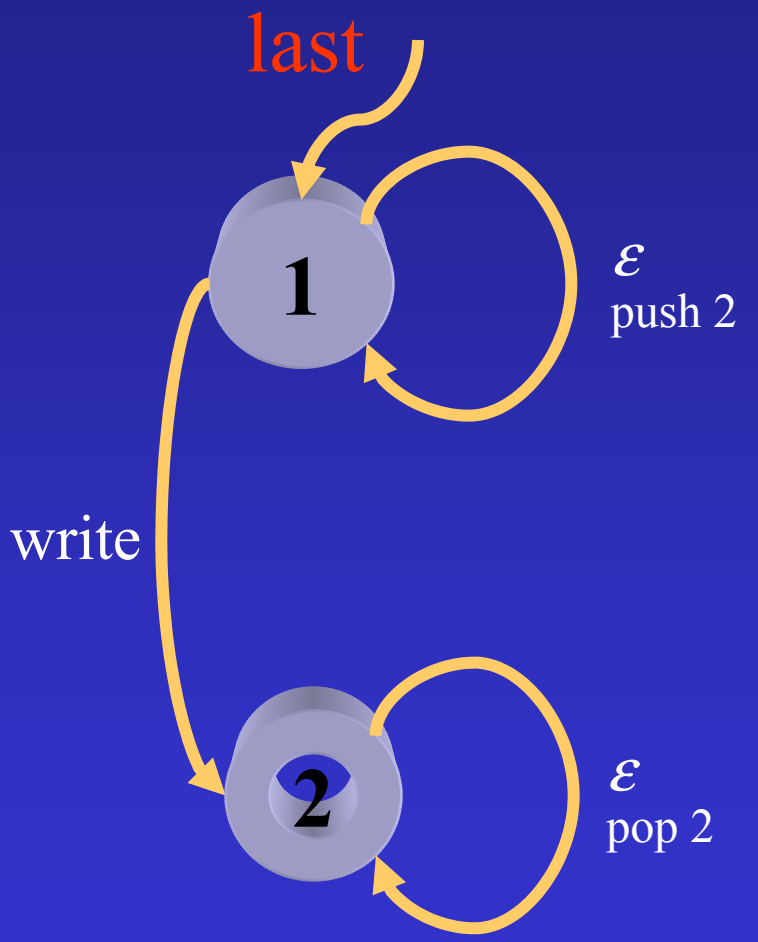
Left Recursion Challenge

```
function last( Node n ) {
   if( n.next == null )
        write( fd, n.name );
   else
        last( n.next );
}
```



Left Recursion Challenge

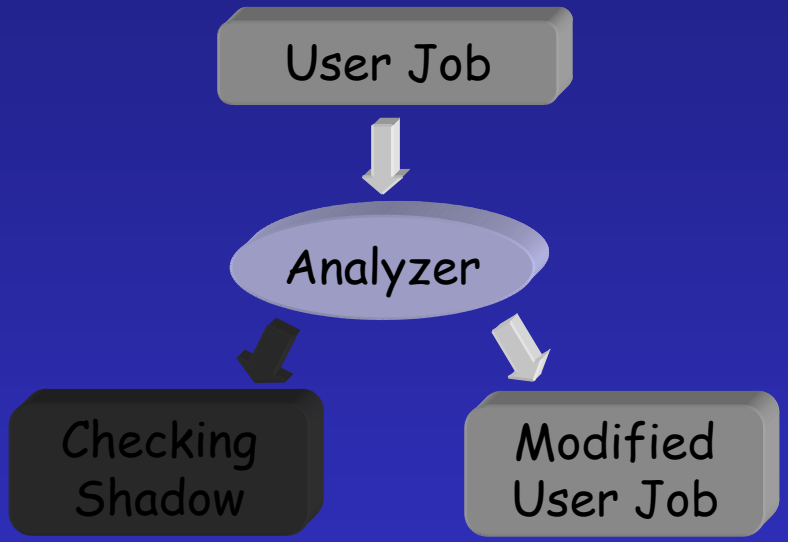
- Possible solutions
  - Top down parsing
  - Esparza, Hansel, Rossmanith, Schwoon Algorithm
  - Bounded Stack PDA
  - PDA / NFA Hybrid



Hard Control Flow Issues

- Indirect calls
  - Solution: slice entire program
- `Longjmps`
  - Solution: assume all `setjmps` possible
- Indirect jumps
  - Solution: recover jump tables; slice program
- Signals
  - Solution: out-of-band data

Rewriting User Job



Binary Program → Rewritten Binary

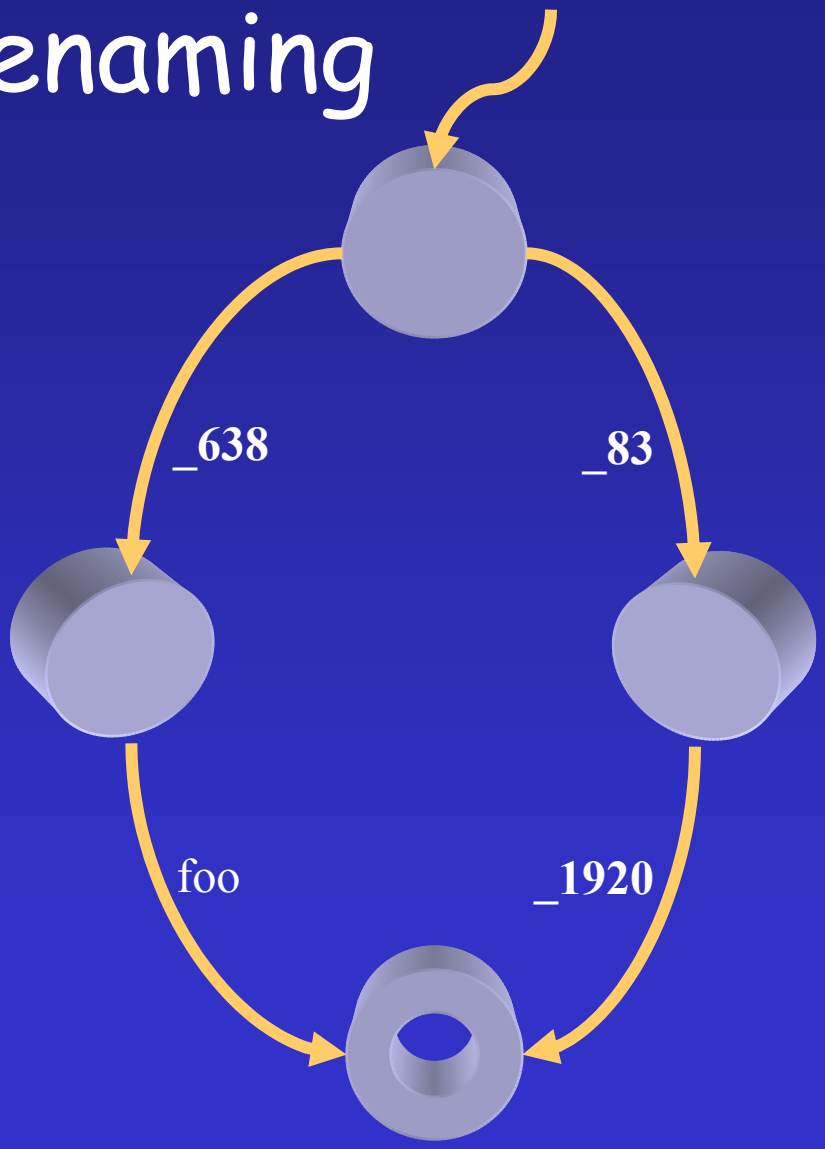
Call Site Renaming

- Give each monitored call site a unique name
- Captures arguments
- Obfuscation
- Limits attack call set
- Reduces nondeterminism

```
function( int a ) {
  if( a < 0 ) {
      read( 0, 15 );
      foo();
  } else {
      read( a, 15 );
      close( a );
  }
}
```

Call Site Renaming

- Give each monitored call site a unique name
- Captures arguments
- Obfuscation
- Limits attack call set
- Reduces nondeterminism

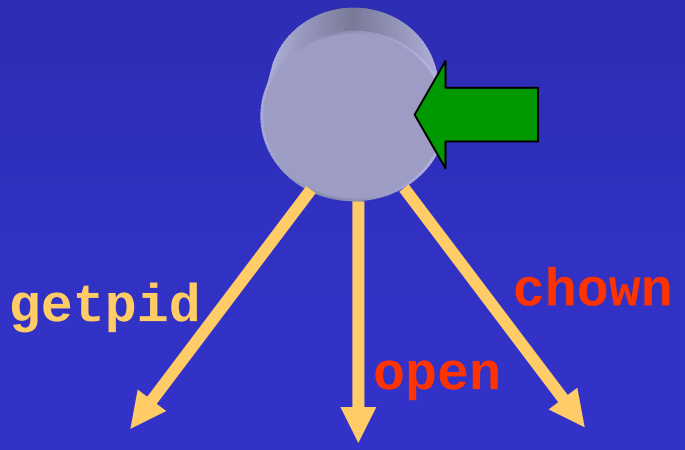


Prototype Implementation

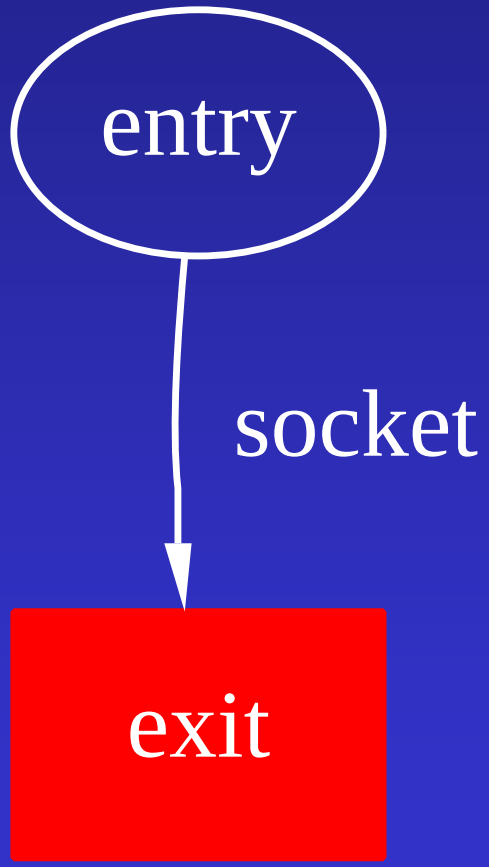
- Simulates remote execution environment
- Models supported:
  - NFA, PDA, Bounded PDA, Hybrid
- Optimizations:
  - Epsilon reduction, minimization, dependency calculation, automata inlining, dead automata removal

Analysis Metric

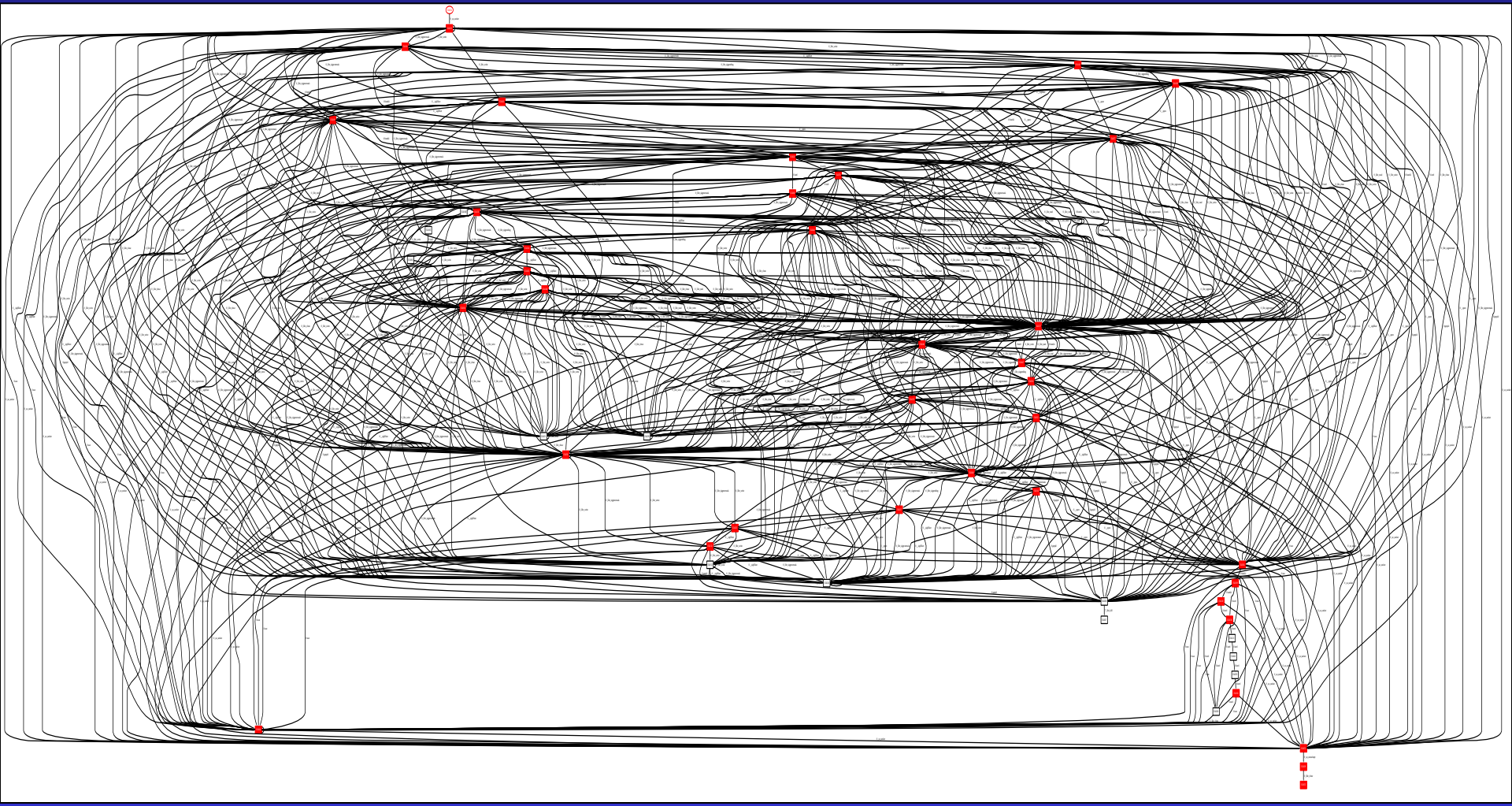
- Experiments evaluate model precision
- Average branching factor metric



Linux glibc socket model



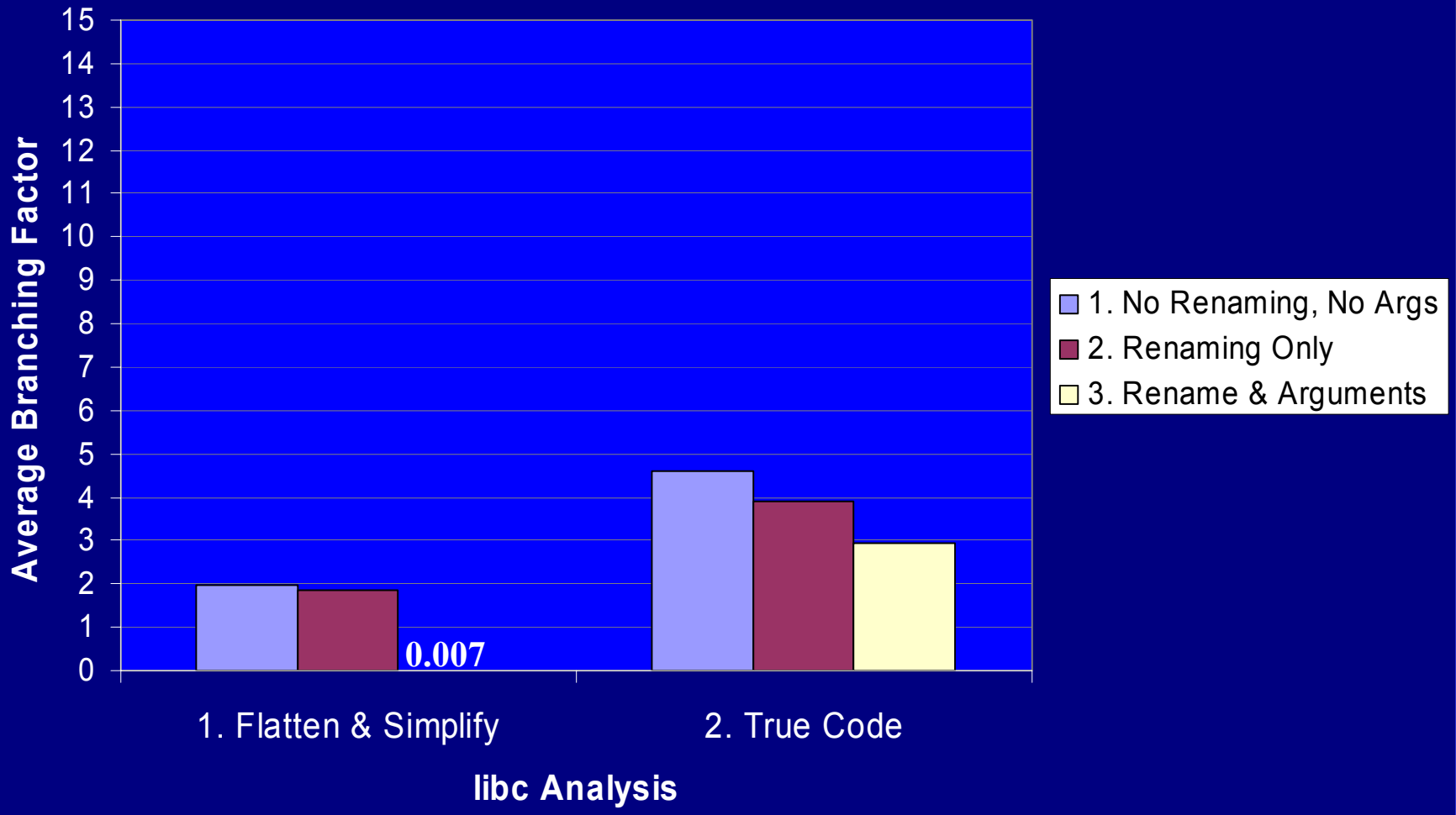
Solaris 8 libc socket model



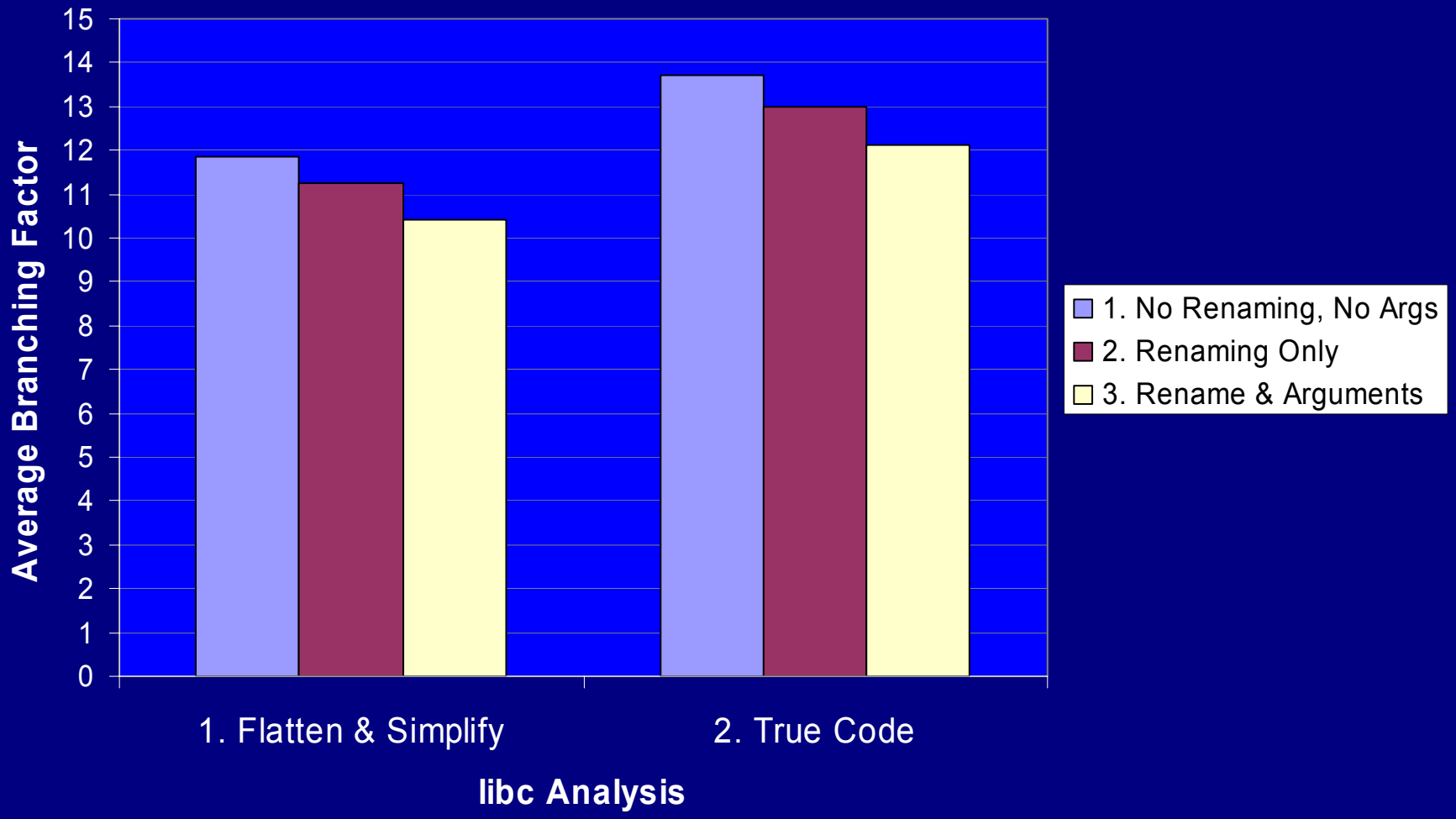
Analysis Levels

- Two levels of Solaris 8 libc analysis
  - 1: Calling structure flattened & simplified
  - 2: True code
- Three levels of program analysis
  - 1: No renaming; no argument capture
  - 2: Renaming only
  - 3: Renaming & argument capture

Analysis - GNU Finger



Analysis - Procmail



Contributions

- Binary modeling
- Model optimizations
- Obfuscation

Determining the Integrity of Remote System Call Streams

Jonathon Giffin
University of Wisconsin