

# Proof-Carrying Code

Language-based Security

Spring 2008

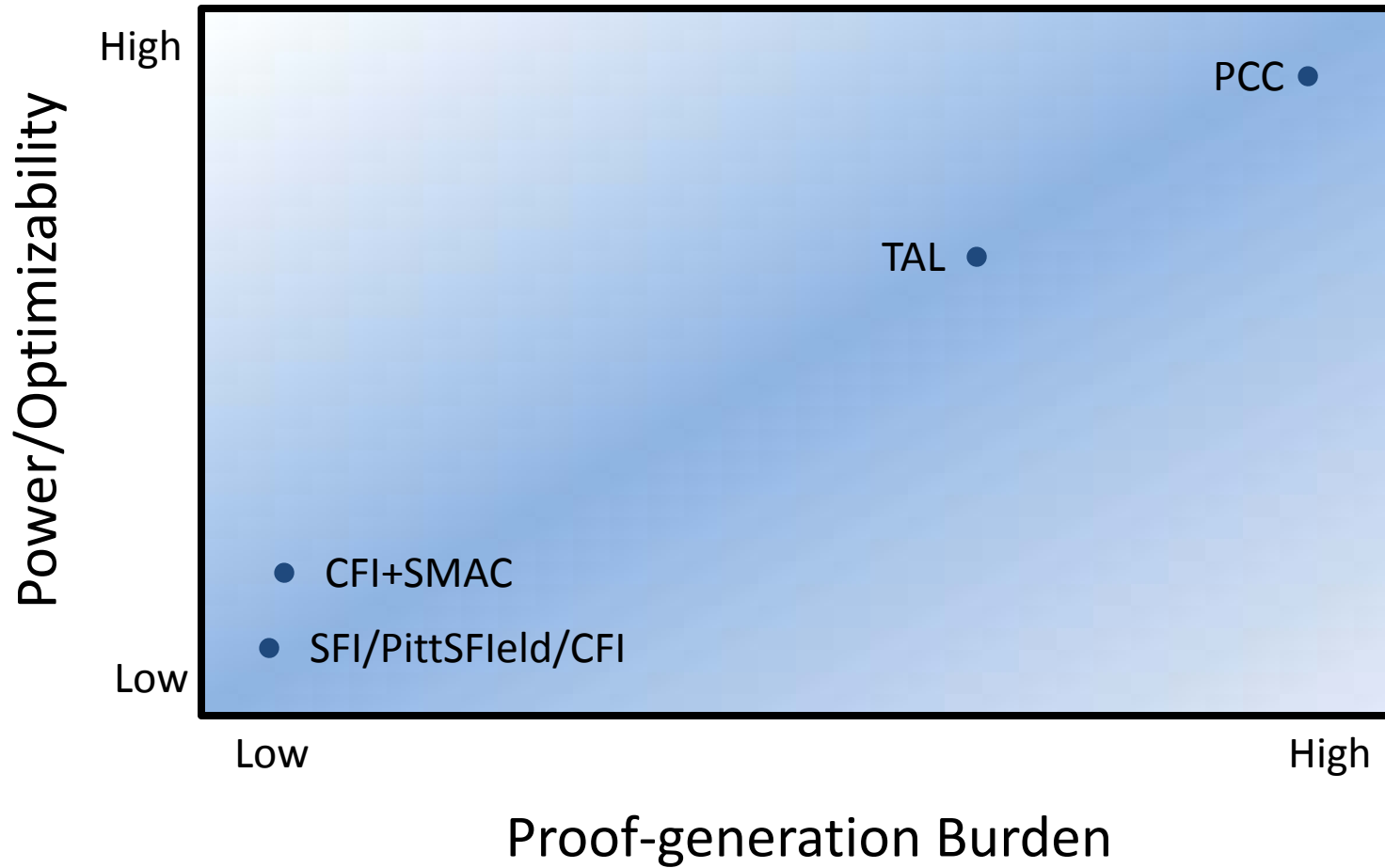
# SFI, PittSFIeld, CFI

- Security Policies Enforced
  - no memory writes to restricted addresses
    - SFI/PittSFIeld: no writes outside fault domain
    - SMAC: application-specific memory-write policy
  - no jumps to restricted addresses
    - SFI: no jumps outside fault domain
    - PittSFIeld: restrict jumps to chunk boundaries
    - CFI: jumps must follow CFG edges
- Verification
  - check for specific guard instructions before every jump and memory-write instruction
  - PittSFIeld: jumps must appear at chunk-ends, guards must use a mask that matches the chunk size, etc.
  - CFI: verify that ID's are unique, verify that the guard for each jump checks for the correct ID, etc.

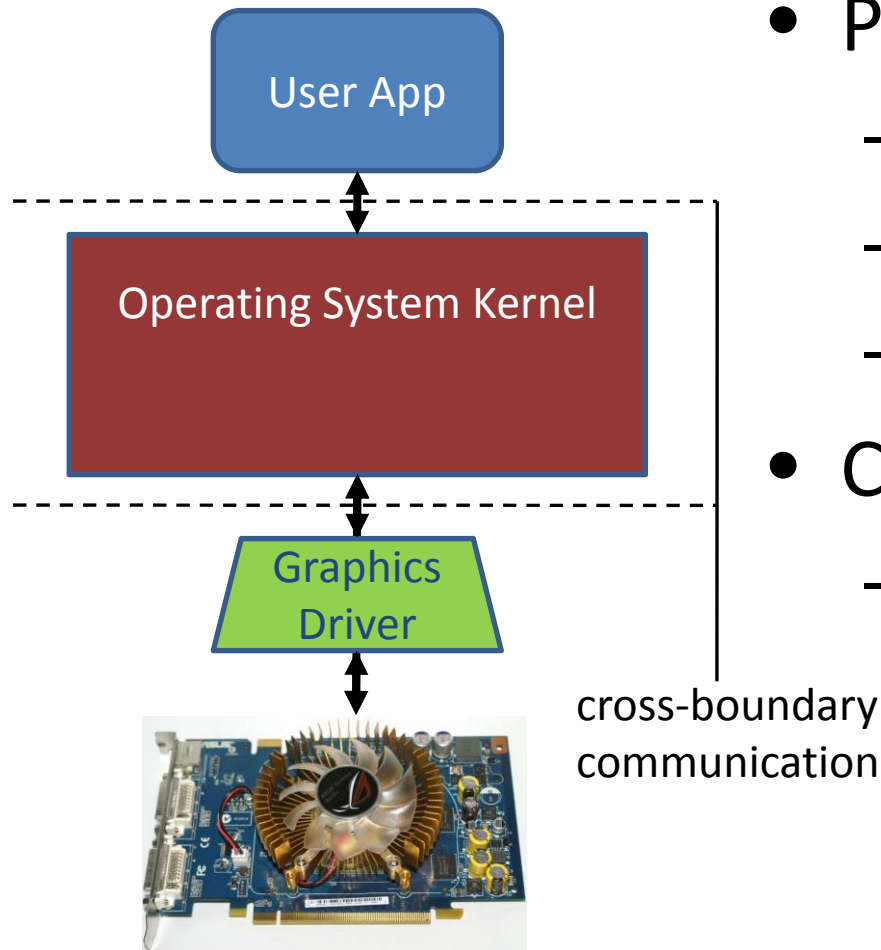
# Shortcomings

- Limited class of security policies
  - Policy: Email app should not create an “.exe” file
  - Policy: Do not leak my credit card number
  - Policy: Thread must sleep after at most 1000 instructions
- Limited optimizing opportunities
  - verifier checks *exact* rewriting strategy
  - Example: cannot merge two chunks
- Tradeoffs
  - enforcement power
  - optimizability
  - proof-generation burden

# Power vs. Proof

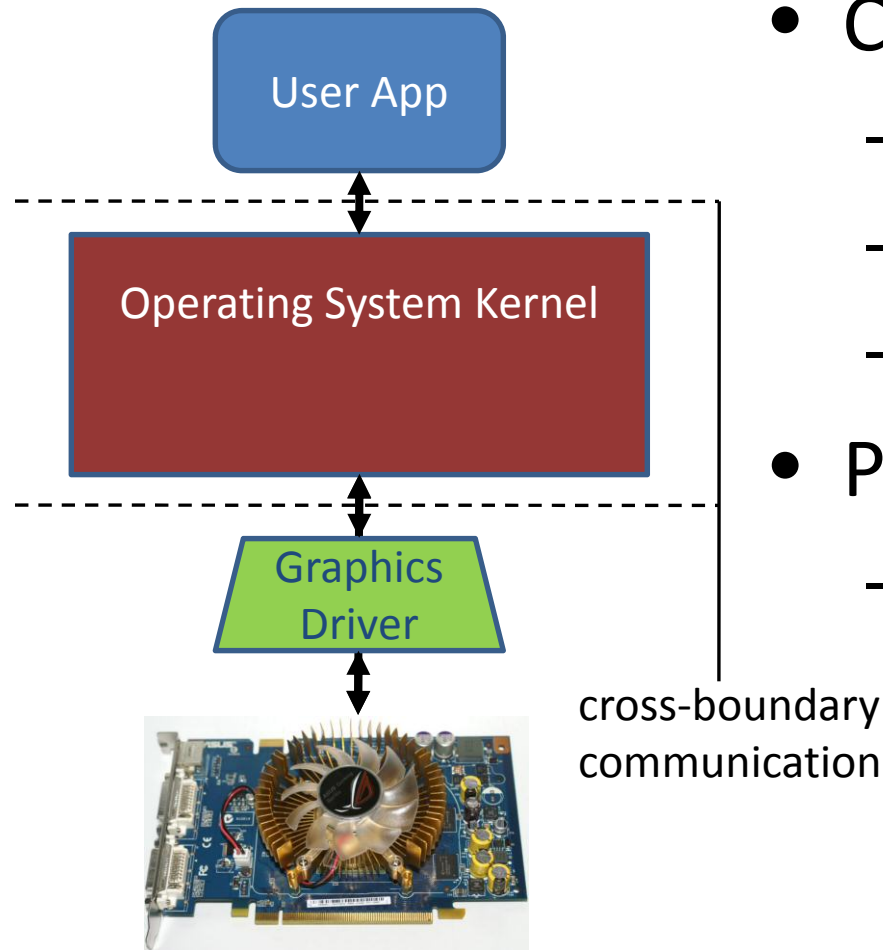


# Kernel Extensions: Unix Approach



- Pros
  - fault-tolerant
  - modular design
  - Unix people happy
- Cons
  - slow

# Kernel Extensions: Windows Approach



- Cons
  - not fault-tolerant
  - non-modular design
  - Unix people jeer
- Pros
  - fast

# Kernel Extensions

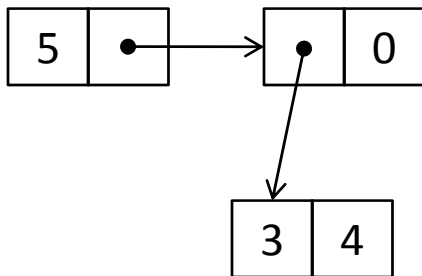
- Relevant Properties
  - Speed-critical
  - Mission-critical (crashes kill kernel)
  - Small code size (written in C / Assembly / Forth)
- Goals
  - Certify safety *prior* to deployment
  - Allow *arbitrary* (safe) programs
  - High proof-generation burden acceptable

# Procedure

- Producer & consumer agree upon a policy
  - precondition & postcondition for untrusted code
  - set of “dangerous” instructions (e.g. Mem[...])
  - policy predicate associated with each dangerous instruction
- Producer
  - annotates code (e.g., with certifying compiler)
  - generates verification condition
  - generates proof of verification condition (e.g., theorem prover)
- Consumer
  - generates verification condition (using code & annotations)
  - checks proof of verification condition
  - executes or rejects, as appropriate

# Sample Kernel Extension

- Kernel passes list to extension
- Extension uses list and returns
- Each list element is...
  - an integer, or
  - a pointer to pair of integers
- Kernel's list representation:
  - pointers are even-valued
  - null pointer is 0
  - int x represented as  $2x+1$
- Example: [2, (3,4)]



```
1 sum:
2 Loop:
3     if  $r_x \neq 0$  jump LCons
4      $r_R := r_{acc}$ 
5     return
6 LCons:  $r_t := \text{Mem}[r_x]$ 
7     if even( $r_t$ ) jump LPair
8      $r_t := r_t \text{ div } 2$ 
9      $r_{acc} := r_{acc} + r_t$ 
10    jump LTail
11 LPair:  $r_s := \text{Mem}[r_t]$ 
12     $r_{acc} := r_{acc} + r_s$ 
13     $r_t := \text{Mem}[r_t + 4]$ 
14     $r_{acc} := r_{acc} + r_t$ 
15 LTail:  $r_x := \text{Mem}[r_x + 4]$ 
16    jump Loop
```

# Security Policy

- Notation
  - $sel(M, a)$  = contents of address  $a$  in memory state  $M$
  - $readok(M, a)$  = addresses  $a..a+3$  are readable in memory state  $M$
- Policy: No invalid memory-reads
  - every address  $a$  that is read must satisfy  $readok(M, a)$  where  $M$  is current memory state)
- Entrypoint Precondition:
  - dictated by code-consumer (kernel)
  - register  $r_x$  holds a valid list on entry:  $mp\_list(M, X)$

$$pair(M, a) = readok(M, a) \wedge readok(M, a + 4)$$

$$ptr(a) = (even(a) \wedge (a \neq 0))$$

$$mp\_list(M, a) = ptr(a) \implies (pair(M, a) \wedge (ptr(sel(M, a)) \implies pair(M, sel(M, a))) \wedge mp\_list(M, a + 4))$$

# VCGen: Symbolic Evaluation

$\{ \text{mpl}(M, X) \} \{ r_x = X, r_R = R, r_{\text{acc}} = A, r_t = T, r_s = S, r_M = M \}$

```
1 sum:
2 Loop:
3     if  $r_x \neq 0$  jump LCons
4      $r_R := r_{\text{acc}}$ 
5     return
6 LCons:  $r_t := \text{Mem}[r_x]$ 
7     if  $\text{even}(r_t)$  jump LPair
8      $r_t := r_t \text{ div } 2$ 
9      $r_{\text{acc}} := r_{\text{acc}} + r_t$ 
10    jump LTail
11 LPair:  $r_s := \text{Mem}[r_t]$ 
12     $r_{\text{acc}} := r_{\text{acc}} + r_s$ 
13     $r_t := \text{Mem}[r_t + 4]$ 
14     $r_{\text{acc}} := r_{\text{acc}} + r_t$ 
15 LTail:  $r_x := \text{Mem}[r_x + 4]$ 
16    jump Loop
```


# VCGen: Symbolic Evaluation

$\{ \text{mpl}(M, X) \} \{ r_x = X, r_R = R, r_{\text{acc}} = A, r_t = T, r_s = S, r_M = M \}$

$\{ \dots, X=0 \} \{ r_x = X, r_R = R, r_{\text{acc}} = A, r_t = T, r_s = S, r_M = M \}$

$\{ \dots, X \neq 0 \} \{ r_x = X, r_R = R, r_{\text{acc}} = A, r_t = T, r_s = S, r_M = M \}$

```
1 sum:
2 Loop:
3   if  $r_x \neq 0$  jump LCons
4    $r_R := r_{\text{acc}}$ 
5   return
6 LCons:  $r_t := \text{Mem}[r_x]$ 
7   if  $\text{even}(r_t)$  jump LPair
8    $r_t := r_t \text{ div } 2$ 
9    $r_{\text{acc}} := r_{\text{acc}} + r_t$ 
10  jump LTail
11 LPair:  $r_s := \text{Mem}[r_t]$ 
12    $r_{\text{acc}} := r_{\text{acc}} + r_s$ 
13    $r_t := \text{Mem}[r_t + 4]$ 
14    $r_{\text{acc}} := r_{\text{acc}} + r_t$ 
15 LTail:  $r_x := \text{Mem}[r_x + 4]$ 
16   jump Loop
```



# VCGen: Symbolic Evaluation

$\{ \text{mpl}(M, X) \} \{ r_x = X, r_R = R, r_{\text{acc}} = A, r_t = T, r_s = S, r_M = M \}$

$\{ \dots, X=0 \} \{ r_x = X, r_R = R, r_{\text{acc}} = A, r_t = T, r_s = S, r_M = M \}$

$\{ \dots, X=0 \} \{ r_x = X, r_R = A, r_{\text{acc}} = A, r_t = T, r_s = S, r_M = M \}$

$\{ \dots, X \neq 0 \} \{ r_x = X, r_R = R, r_{\text{acc}} = A, r_t = T, r_s = S, r_M = M \}$

```
1 sum:
2 Loop:
3     if  $r_x \neq 0$  jump LCons
4          $r_R := r_{\text{acc}}$ 
5         return
6 LCons:  $r_t := \text{Mem}[r_x]$ 
7         if even( $r_t$ ) jump LPair
8          $r_t := r_t \text{ div } 2$ 
9          $r_{\text{acc}} := r_{\text{acc}} + r_t$ 
10        jump LTail
11 LPair:  $r_s := \text{Mem}[r_t]$ 
12         $r_{\text{acc}} := r_{\text{acc}} + r_s$ 
13         $r_t := \text{Mem}[r_t + 4]$ 
14         $r_{\text{acc}} := r_{\text{acc}} + r_t$ 
15 LTail:  $r_x := \text{Mem}[r_x + 4]$ 
16        jump Loop
```

# VCGen: Symbolic Evaluation

$\{ \text{mpl}(M,X) \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots, X=0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots, X=0 \} \{ r_x=X, r_R=A, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots, X \neq 0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots \} \{ r_t = \text{sel}(M,X), \dots \}$

```

1 sum:
2 Loop:
3     if  $r_x \neq 0$  jump LCons
4      $r_R := r_{\text{acc}}$ 
5     return
6 LCons:  $r_t := \text{Mem}[r_x]$ 
7     if  $\text{even}(r_t)$  jump LPair
8      $r_t := r_t \text{ div } 2$ 
9      $r_{\text{acc}} := r_{\text{acc}} + r_t$ 
10    jump LTail
11 LPair:  $r_s := \text{Mem}[r_t]$ 
12     $r_{\text{acc}} := r_{\text{acc}} + r_s$ 
13     $r_t := \text{Mem}[r_t + 4]$ 
14     $r_{\text{acc}} := r_{\text{acc}} + r_t$ 
15 LTail:  $r_x := \text{Mem}[r_x + 4]$ 
16    jump Loop
  
```

$(\text{mpl}(M,X) \wedge (X \neq 0) \Rightarrow \text{readok}(X))$

# VCGen: Symbolic Evaluation

$\{ \text{mpl}(M,X) \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots, X=0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots, X=0 \} \{ r_x=X, r_R=A, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots, X \neq 0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots \} \{ r_t=\text{sel}(M,X), \dots \}$

$\{ \dots, \sim\text{even}(\text{sel}(M,X)) \} \{ \dots \}$

$\{ \dots, \text{even}(\text{sel}(M,X)) \} \{ r_t=\text{sel}(M,X), \dots \}$

```

1 sum:
2 Loop:
3     if  $r_x \neq 0$  jump LCons
4      $r_R := r_{\text{acc}}$ 
5     return
6 LCons:  $r_t := \text{Mem}[r_x]$ 
7     if  $\text{even}(r_t)$  jump LPair
8      $r_t := r_t \text{ div } 2$ 
9      $r_{\text{acc}} := r_{\text{acc}} + r_t$ 
10    jump LTail
11 LPair:  $r_s := \text{Mem}[r_t]$ 
12     $r_{\text{acc}} := r_{\text{acc}} + r_s$ 
13     $r_t := \text{Mem}[r_t + 4]$ 
14     $r_{\text{acc}} := r_{\text{acc}} + r_t$ 
15 LTail:  $r_x := \text{Mem}[r_x + 4]$ 
16    jump Loop
    
```



$(\text{mpl}(M,X) \wedge (X \neq 0) \Rightarrow \text{readok}(X))$

# VCGen: Symbolic Evaluation

$\{ \text{mpl}(M,X) \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	1	sum:
	2	Loop:
$\{ \dots, X=0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	3	if $r_x \neq 0$ jump LCons
$\{ \dots, X=0 \} \{ r_x=X, r_R=A, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	4	$r_R := r_{\text{acc}}$
$\{ \dots, X \neq 0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	5	return
$\{ \dots \} \{ r_t=\text{sel}(M,X), \dots \}$	6	LCons: $r_t := \text{Mem}[r_x]$
$\{ \dots, \sim\text{even}(\text{sel}(M,X)) \} \{ \dots \}$	7	if $\text{even}(r_t)$ jump LPair
$\{ \dots \} \{ r_t=\text{sel}(M,X)/2, \dots \}$	8	$r_t := r_t \text{ div } 2$
$\{ \dots \} \{ r_{\text{acc}}=A+\text{sel}(M,X)/2, \dots \}$	9	$r_{\text{acc}} := r_{\text{acc}} + r_t$
$\{ \dots, \text{even}(\text{sel}(M,X)) \} \{ r_t=\text{sel}(M,X), \dots \}$	10	jump LTail
	11	LPair: $r_s := \text{Mem}[r_t]$
	12	$r_{\text{acc}} := r_{\text{acc}} + r_s$
	13	$r_t := \text{Mem}[r_t + 4]$
	14	$r_{\text{acc}} := r_{\text{acc}} + r_t$
	15	LTail: $r_x := \text{Mem}[r_x + 4]$
	16	jump Loop

$(\text{mpl}(M,X) \wedge (X \neq 0)) \Rightarrow \text{readok}(X)$

# VCGen: Symbolic Evaluation

$\{ \text{mpl}(M,X) \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots, X=0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots, X=0 \} \{ r_x=X, r_R=A, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots, X \neq 0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots \} \{ r_t=\text{sel}(M,X), \dots \}$

$\{ \dots, \sim\text{even}(\text{sel}(M,X)) \} \{ \dots \}$

$\{ \dots \} \{ r_t=\text{sel}(M,X)/2, \dots \}$

$\{ \dots \} \{ r_{\text{acc}}=A+\text{sel}(M,X)/2, \dots \}$

$\{ \dots, \text{even}(\text{sel}(M,X)) \} \{ r_t=\text{sel}(M,X), \dots \}$

$\{ \dots \} \{ r_{\text{acc}}=A+\text{sel}(M,X)/2, \dots \}$

```

1 sum:
2 Loop:
3   if  $r_x \neq 0$  jump LCons
4    $r_R := r_{\text{acc}}$ 
5   return
6 LCons:  $r_t := \text{Mem}[r_x]$ 
7   if  $\text{even}(r_t)$  jump LPair
8    $r_t := r_t \text{ div } 2$ 
9    $r_{\text{acc}} := r_{\text{acc}} + r_t$ 
10  jump LTail
11 LPair:  $r_s := \text{Mem}[r_t]$ 
12    $r_{\text{acc}} := r_{\text{acc}} + r_s$ 
13    $r_t := \text{Mem}[r_t + 4]$ 
14    $r_{\text{acc}} := r_{\text{acc}} + r_t$ 
15 LTail:  $r_x := \text{Mem}[r_x + 4]$ 
16   jump Loop
  
```



$(\text{mpl}(M,X) \wedge (X \neq 0)) \Rightarrow \text{readok}(X)$

# VCGen: Symbolic Evaluation

$\{ \text{mpl}(M,X) \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	1	sum:
	2	Loop:
$\{ \dots, X=0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	3	if $r_x \neq 0$ jump LCons
$\{ \dots, X=0 \} \{ r_x=X, r_R=A, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	4	$r_R := r_{\text{acc}}$
$\{ \dots, X \neq 0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	5	return
$\{ \dots \} \{ r_t=\text{sel}(M,X), \dots \}$	6	LCons: $r_t := \text{Mem}[r_x]$
$\{ \dots, \sim\text{even}(\text{sel}(M,X)) \} \{ \dots \}$	7	if $\text{even}(r_t)$ jump LPair
$\{ \dots \} \{ r_t=\text{sel}(M,X)/2, \dots \}$	8	$r_t := r_t \text{ div } 2$
$\{ \dots \} \{ r_{\text{acc}}=A+\text{sel}(M,X)/2, \dots \}$	9	$r_{\text{acc}} := r_{\text{acc}} + r_t$
$\{ \dots, \text{even}(\text{sel}(M,X)) \} \{ r_t=\text{sel}(M,X), \dots \}$	10	jump LTail
$\{ \dots \} \{ r_s=\text{sel}(M,\text{sel}(M,X)), \dots \}$	11	LPair: $r_s := \text{Mem}[r_t]$
	12	$r_{\text{acc}} := r_{\text{acc}} + r_s$
	13	$r_t := \text{Mem}[r_t + 4]$
	14	$r_{\text{acc}} := r_{\text{acc}} + r_t$
$\{ \dots \} \{ r_{\text{acc}}=A+\text{sel}(M,X)/2, \dots \}$	15	LTail: $r_x := \text{Mem}[r_x + 4]$
	16	jump Loop

$(\text{mpl}(M,X) \wedge (X \neq 0) \Rightarrow \text{readok}(X)) \wedge (\text{mpl}(M,X) \wedge (X \neq 0) \wedge \text{even}(\text{sel}(M,X)) \Rightarrow \text{readok}(\text{sel}(M,X)))$

# VCGen: Symbolic Evaluation

$\{ \text{mpl}(M,X) \}$	$\{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	1	sum:
		2	Loop:
$\{ \dots, X=0 \}$	$\{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	3	if $r_x \neq 0$ jump LCons
$\{ \dots, X=0 \}$	$\{ r_x=X, r_R=A, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	4	$r_R := r_{\text{acc}}$
$\{ \dots, X \neq 0 \}$	$\{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	5	return
	$\{ \dots \} \{ r_t = \text{sel}(M,X), \dots \}$	6	LCons: $r_t := \text{Mem}[r_x]$
$\{ \dots, \sim \text{even}(\text{sel}(M,X)) \}$	$\{ \dots \}$	7	if $\text{even}(r_t)$ jump LPair
	$\{ \dots \} \{ r_t = \text{sel}(M,X)/2, \dots \}$	8	$r_t := r_t \text{ div } 2$
	$\{ \dots \} \{ r_{\text{acc}} = A + \text{sel}(M,X)/2, \dots \}$	9	$r_{\text{acc}} := r_{\text{acc}} + r_t$
$\{ \dots, \text{even}(\text{sel}(M,X)) \}$	$\{ r_t = \text{sel}(M,X), \dots \}$	10	jump LTail
	$\{ \dots \} \{ r_s = \text{sel}(M, \text{sel}(M,X)), \dots \}$	11	LPair: $r_s := \text{Mem}[r_t]$
	$\{ \dots \} \{ r_{\text{acc}} = A + \text{sel}(M, \text{sel}(M,X)), \dots \}$	12	$r_{\text{acc}} := r_{\text{acc}} + r_s$
	$\{ \dots \} \{ r_t = \text{sel}(M, \text{sel}(M,X) + 4), \dots \}$	13	$r_t := \text{Mem}[r_t + 4]$
	$\{ \dots \} \{ r_{\text{acc}} = A + \text{sel}(M,X)/2, \dots \}$	14	$r_{\text{acc}} := r_{\text{acc}} + r_t$
		15	LTail: $r_x := \text{Mem}[r_x + 4]$
		16	jump Loop

$(\text{mpl}(M,X) \wedge (X \neq 0) \Rightarrow \text{readok}(X)) \wedge (\text{mpl}(M,X) \wedge (X \neq 0) \wedge \text{even}(\text{sel}(M,X)) \Rightarrow \text{readok}(\text{sel}(M,X)))$   
 $\wedge (\text{mpl}(M,X) \wedge (X \neq 0) \wedge \text{even}(\text{sel}(M,X)) \Rightarrow \text{readok}(\text{sel}(M,X) + 4))$

# Problem: Join Points

{ mpl(M,X) } { r<sub>x</sub>=X, r<sub>R</sub>=R, r<sub>acc</sub>=A, r<sub>t</sub>=T, r<sub>s</sub>=S, r<sub>M</sub>=M }

{ ..., X=0 } { r<sub>x</sub>=X, r<sub>R</sub>=R, r<sub>acc</sub>=A, r<sub>t</sub>=T, r<sub>s</sub>=S, r<sub>M</sub>=M }

{ ..., X=0 } { r<sub>x</sub>=X, r<sub>R</sub>=A, r<sub>acc</sub>=A, r<sub>t</sub>=T, r<sub>s</sub>=S, r<sub>M</sub>=M }

{ ..., X≠0 } { r<sub>x</sub>=X, r<sub>R</sub>=R, r<sub>acc</sub>=A, r<sub>t</sub>=T, r<sub>s</sub>=S, r<sub>M</sub>=M }

{ ... } { r<sub>t</sub>=sel(M,X), ... }

{ ..., ~even(sel(M,X)) } { ... }

{ ... } { r<sub>t</sub>=sel(M,X)/2, ... }

{ ... } { r<sub>acc</sub>=A+sel(M,X)/2, ... }

{ ..., even(sel(M,X)) } { r<sub>t</sub>=sel(M,X), ... }

{ ... } { r<sub>s</sub>=sel(M,sel(M,X)), ... }

{ ... } { r<sub>acc</sub>=A+sel(M,sel(M,X)), ... }

{ ... } { r<sub>t</sub>=sel(M,sel(M,X)+4), ... }

{ ? } { ? }

1 sum:

2 Loop:

3 if r<sub>x</sub>≠0 jump LCons

4 r<sub>R</sub> := r<sub>acc</sub>

5 return

6 LCons: r<sub>t</sub> := Mem[r<sub>x</sub>]

7 if even(r<sub>t</sub>) jump LPair

8 r<sub>t</sub> := r<sub>t</sub> div 2

9 r<sub>acc</sub> := r<sub>acc</sub> + r<sub>t</sub>

10 jump LTail

11 LPair: r<sub>s</sub> := Mem[r<sub>t</sub>]

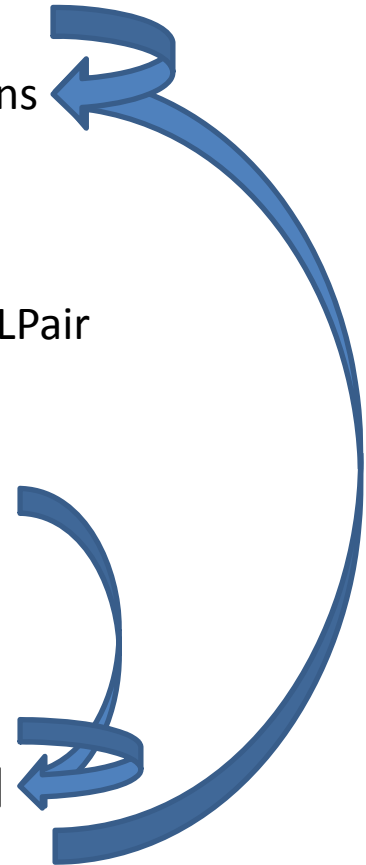
12 r<sub>acc</sub> := r<sub>acc</sub> + r<sub>s</sub>

13 r<sub>t</sub> := Mem[r<sub>t</sub> + 4]

14 r<sub>acc</sub> := r<sub>acc</sub> + r<sub>t</sub>

15 LTail: r<sub>x</sub> := Mem[r<sub>x</sub> + 4]

16 jump Loop



(mpl(M,X) ∧ (X≠0) ⇒ readok(X)) ∧ (mpl(M,X) ∧ (X≠0) ∧ even(sel(M,X)) ⇒ readok(sel(M,X)))  
 ∧ (mpl(M,X) ∧ (X≠0) ∧ even(sel(M,X)) ⇒ readok(sel(M,X)+4))

# Solution: Annotations

$\{ \text{mpl}(M,X) \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots, X=0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots, X=0 \} \{ r_x=X, r_R=A, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots, X \neq 0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots \} \{ r_t=\text{sel}(M,X), \dots \}$

$\{ \dots, \sim\text{even}(\text{sel}(M,X)) \} \{ \dots \}$

$\{ \dots \} \{ r_t=\text{sel}(M,X)/2, \dots \}$

$\{ \dots \} \{ r_{\text{acc}}=A+\text{sel}(M,X)/2, \dots \}$

$\{ \dots, \text{even}(\text{sel}(M,X)) \} \{ r_t=\text{sel}(M,X), \dots \}$

$\{ \dots \} \{ r_s=\text{sel}(M,\text{sel}(M,X)), \dots \}$

$\{ \dots \} \{ r_{\text{acc}}=A+\text{sel}(M,\text{sel}(M,X)), \dots \}$

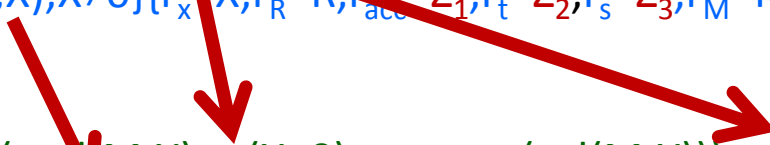
$\{ \dots \} \{ r_t=\text{sel}(M,\text{sel}(M,X)+4), \dots \}$

$\{ \text{mpl}(M,X), X \neq 0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=Z_1, r_t=Z_2, r_s=Z_3, r_M=M \}$

```

1 sum:
2 Loop:
3   if  $r_x \neq 0$  jump LCons
4    $r_R := r_{\text{acc}}$ 
5   return
6 LCons:  $r_t := \text{Mem}[r_x]$ 
7   if  $\text{even}(r_t)$  jump LPair
8    $r_t := r_t \text{ div } 2$ 
9    $r_{\text{acc}} := r_{\text{acc}} + r_t$ 
10  jump LTail
11 LPair:  $r_s := \text{Mem}[r_t]$ 
12    $r_{\text{acc}} := r_{\text{acc}} + r_s$ 
13    $r_t := \text{Mem}[r_t + 4]$ 
14    $r_{\text{acc}} := r_{\text{acc}} + r_t$ 
15 LTail:  $r_x := \text{Mem}[r_x + 4]$ 
16   jump Loop
  
```

$\dots \wedge (\text{mpl}(M,X) \wedge (X \neq 0) \wedge \sim\text{even}(\text{sel}(M,X))) \vee (\text{mpl}(M,X) \wedge (X \neq 0) \wedge \text{even}(\text{sel}(M,X)))$   
 $\Rightarrow (\text{mpl}(M,X) \wedge (X \neq 0))$



# Solution: Annotations

$\{ \text{mpl}(M,X) \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots, X=0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots, X=0 \} \{ r_x=X, r_R=A, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots, X \neq 0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$

$\{ \dots \} \{ r_t=\text{sel}(M,X), \dots \}$

$\{ \dots, \sim\text{even}(\text{sel}(M,X)) \} \{ \dots \}$

$\{ \dots \} \{ r_t=\text{sel}(M,X)/2, \dots \}$

$\{ \dots \} \{ r_{\text{acc}}=A+\text{sel}(M,X)/2, \dots \}$

$\{ \dots, \text{even}(\text{sel}(M,X)) \} \{ r_t=\text{sel}(M,X), \dots \}$

$\{ \dots \} \{ r_s=\text{sel}(M,\text{sel}(M,X)), \dots \}$

$\{ \dots \} \{ r_{\text{acc}}=A+\text{sel}(M,\text{sel}(M,X)), \dots \}$

$\{ \dots \} \{ r_t=\text{sel}(M,\text{sel}(M,X)+4), \dots \}$

$\{ \text{mpl}(M,X), X \neq 0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=Z_1, r_t=Z_2, r_s=Z_3, r_M=M \}$

$\{ \dots \} \{ r_x=\text{sel}(M,X+4), \dots \}$

1 sum:

2 Loop:

3 if  $r_x \neq 0$  jump LCons

4  $r_R := r_{\text{acc}}$

5 return

6 LCons:  $r_t := \text{Mem}[r_x]$

7 if  $\text{even}(r_t)$  jump LPair

8  $r_t := r_t \text{ div } 2$

9  $r_{\text{acc}} := r_{\text{acc}} + r_t$

10 jump LTail

11 LPair:  $r_s := \text{Mem}[r_t]$

12  $r_{\text{acc}} := r_{\text{acc}} + r_s$

13  $r_t := \text{Mem}[r_t + 4]$

14  $r_{\text{acc}} := r_{\text{acc}} + r_t$

15 LTail:  $r_x := \text{Mem}[r_x + 4]$

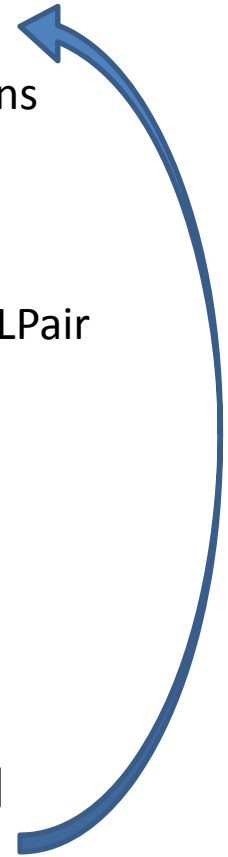
16 jump Loop

$\dots \wedge (\text{mpl}(M,X) \wedge (X \neq 0)) \Rightarrow \text{readok}(M,X+4)$

# Solution: Annotations

$\{ \text{mpl}(M,X) \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	1	sum:
$\{ \text{mpl}(M,Z_4) \} \{ r_x=Z_4, r_R=R, r_{\text{acc}}=Z_5, r_t=Z_6, r_s=Z_7, r_M=M \}$	2	Loop:
$\{ \dots, X=0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	3	if $r_x \neq 0$ jump LCons
$\{ \dots, X=0 \} \{ r_x=X, r_R=A, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	4	$r_R := r_{\text{acc}}$
$\{ \dots, X \neq 0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=A, r_t=T, r_s=S, r_M=M \}$	5	return
$\{ \dots \} \{ r_t = \text{sel}(M,X), \dots \}$	6	LCons: $r_t := \text{Mem}[r_x]$
$\{ \dots, \sim \text{even}(\text{sel}(M,X)) \} \{ \dots \}$	7	if $\text{even}(r_t)$ jump LPair
$\{ \dots \} \{ r_t = \text{sel}(M,X)/2, \dots \}$	8	$r_t := r_t \text{ div } 2$
$\{ \dots \} \{ r_{\text{acc}} = A + \text{sel}(M,X)/2, \dots \}$	9	$r_{\text{acc}} := r_{\text{acc}} + r_t$
$\{ \dots, \text{even}(\text{sel}(M,X)) \} \{ r_t = \text{sel}(M,X), \dots \}$	10	jump LTail
$\{ \dots \} \{ r_s = \text{sel}(M, \text{sel}(M,X)), \dots \}$	11	LPair: $r_s := \text{Mem}[r_t]$
$\{ \dots \} \{ r_{\text{acc}} = A + \text{sel}(M, \text{sel}(M,X)), \dots \}$	12	$r_{\text{acc}} := r_{\text{acc}} + r_s$
$\{ \dots \} \{ r_t = \text{sel}(M, \text{sel}(M,X) + 4), \dots \}$	13	$r_t := \text{Mem}[r_t + 4]$
$\{ \text{mpl}(M,X), X \neq 0 \} \{ r_x=X, r_R=R, r_{\text{acc}}=Z_1, r_t=Z_2, r_s=Z_3, r_M=M \}$	14	$r_{\text{acc}} := r_{\text{acc}} + r_t$
$\{ \dots \} \{ r_x = \text{sel}(M, X+4), \dots \}$	15	LTail: $r_x := \text{Mem}[r_x + 4]$
	16	jump Loop

$\dots \wedge (\text{mpl}(M,X) \wedge (X \neq 0)) \Rightarrow \text{readok}(M, X+4)$



# Proofs

- Producer & consumer have common set of proof rules
  - axioms of logic (e.g.,  $A \wedge B \Rightarrow A$ )
  - axioms of 32-bit arithmetic (e.g.,  $x+0=x$ )
  - memory axioms (e.g.,  $a1=a2 \Rightarrow \text{sel}(\text{upd}(M,a1,v),a2)=v$ )
- Proofs consist of a list of rule applications
  - let fact 1 be the precondition
  - apply rule 27 to fact 1 to conclude new fact 2
  - apply rule 19 to facts 1 and 2 to get new fact 3
  - ...
  - fact 19370 is the verification condition. QED.
- How do we generate these proofs? (theorem-provers)

# How Are Annotations Generated?

- Certifying Compiler (e.g., Touchstone)
  - high-level language is well-typed
  - types = predicates (e.g.,  $x:\text{list} \rightarrow \text{mp\_list}(M,x)$ )
  - compiler asserts a type-derived predicate with each security-relevant register/memory value
- More on this when we cover TAL

# What are the Limitations?

- What policies are enforced?
  - Short answer: Anything provable from axioms
- What optimizations are permitted?
  - Short answer: Anything provable from axioms
- Only real limitation is axioms
  - Are the set of axioms comprehensive?
  - Does there exist a set of axioms such that...
    - for every program  $M$ , and for every property  $P$  such that  $P$  is true of  $M$ , there exists a proof of  $P(M)$ ?
- Practical Limitations:
  - How big are proofs?
  - Is annotation-generation / proof-generation practical for real applications and architectures?
  - Is verification efficient?