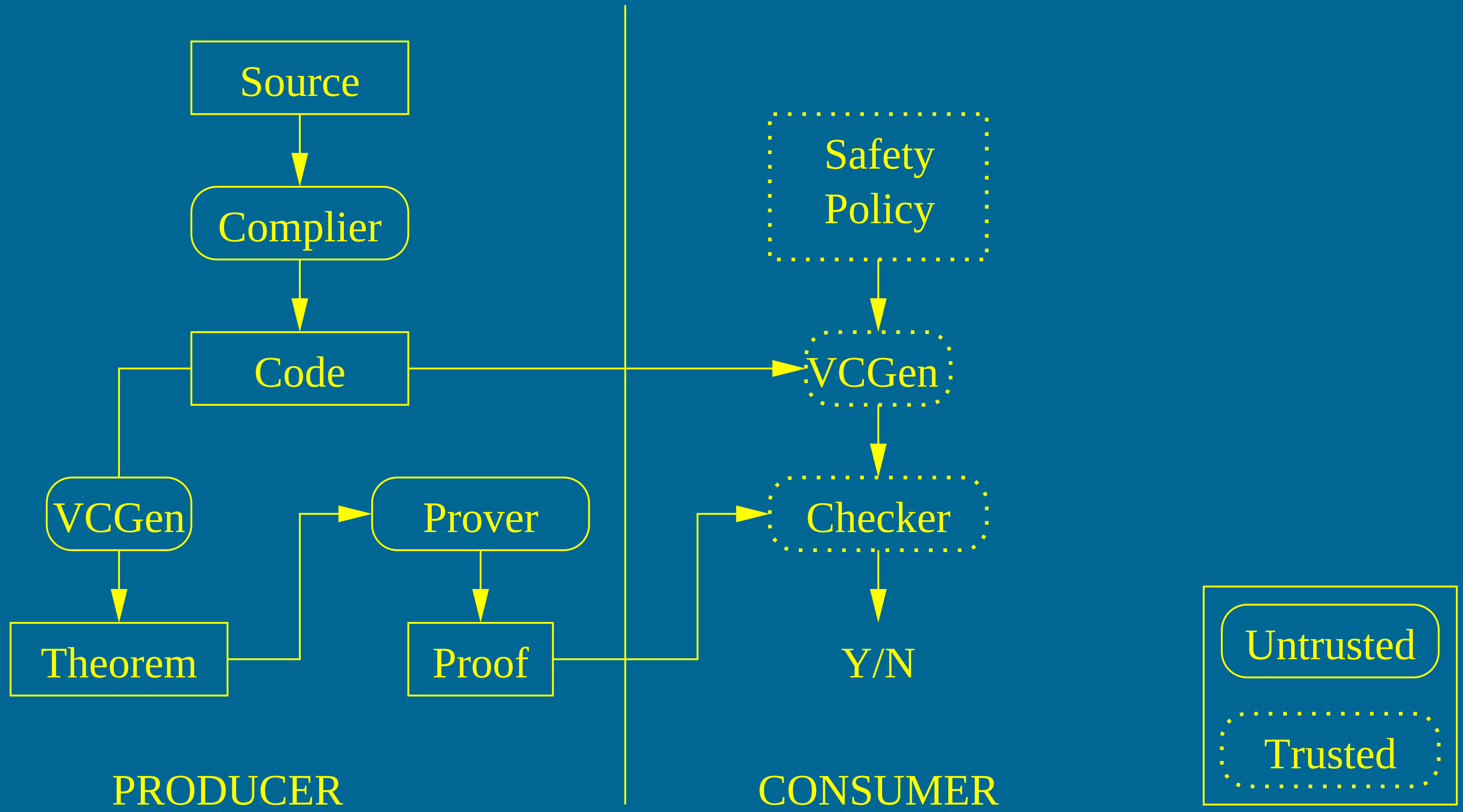
Proof-Carrying Code

COS 598E

Kedar Swadi

Proof-Carrying Code Framework



An Example Language

Instr	::=	`add r1, r2, r3` $\mid$ `load r1 r2 c` $\mid$ `mov r1 r2`
		$\mid$ `store r1 r2 c` $\mid$ `beq r1, c, L` $\mid$ `jump L`
		$\mid$ `halt`
Types τ	::=	`int` $\mid$ `array (int)`
Inv σ	::=	$\cdot \mid \{V : \tau\} \mid \sigma_1 \wedge \sigma_2 \mid V$ op $c \mid \sigma_1 \supset \sigma_2$
Value V	::=	$r_i \mid r_i + c \mid m[V]$
op	::=	$< \mid > \mid \leq \mid \geq \mid = \mid \neq$

Symbolic Evaluator(SE)

Precondition	Postcondition
$\sigma_1 = SE(I_1, \sigma_2)$	σ_1
I_1	I_1
$\sigma_2 = SE(I_2, \sigma_3)$	$\sigma_2 = SE'(I_1, \sigma_1)$
$\vdots$	$\vdots$
$\sigma_{n-1} = SE(I_n, \sigma_n)$	$\sigma_{n-1} = SE'(I_{n-1}, \sigma_{n-2})$
I_n	I_n
σ_n	$\sigma_n = SE'(I_n, \sigma_{n-1})$

- SE can either compute the *weakest* precondition, or the *strongest* postcondition
- Either approach is equivalent

Symbolic Evaluator(SE)

Instr I	$SE(I)$
add r1 r2 r3	$\lambda\sigma.[r2 + r3/r1]\sigma$
load r1 r2 c	$\lambda\sigma.[m[r2 + c]/r1]\sigma$
store r1 r2 c	$\lambda\sigma.[r2/m[r1 + c]]\sigma$
mov r1 r2	$\lambda\sigma.[r2/r1]\sigma$
beq r1 c L	$\lambda\sigma.r1 = c \supset \sigma_L \wedge r1 \neq c \supset \sigma$
jump L	$\lambda\sigma.\sigma_L$
halt	$\lambda\sigma. \cdot$

Theorem of Program Safety

- Safe to execute next instruction
- Safe to jump to any branch target, T, from any of its sources, S_T
- Postconditions of jumps at least as strong as preconditions of jump targets.

Theorem of Program Safety II

F: I_F

$\vdots$

jump G

G: I_G

$\vdots$

jump H

H: I_H

$\vdots$

halt

$$\forall I_{f \in \{F,G,H\}}.\ I_f \supset \sigma_f$$

Producing Theorems From Programs

`L1:` $\{r1 : array(int), r2 : int\}$

$\{m[r1 + r2 + 0] : int\}$

```
add r3 r2 r1
```

$\{m[r3 + 0] : int\}$

```
load r2 r3 0
```

$\{r2 : int\}$

```
jump L2
```

$\vdots$

`L2:` $\{r2 : int\}$

Safety Theorem = $r1 : array(int) \wedge r2 : int \supset m[r1 + r2 + 0] : int$

$\bigwedge \quad \ldots$

Producing Proofs For Theorems

Inference Rules	Typing Rules
$\dfrac{\Gamma \vdash P_1 \;\wedge\; P_2}{\Gamma \vdash P1}\ \wedge E1$	$\dfrac{\Gamma \vdash ra : int}{\Gamma \vdash ra + c : int}\ C\_ADD$
$\dfrac{\Gamma \vdash P_1 \;\wedge\; P_2}{\Gamma \vdash P2}\ \wedge E2$	$\dfrac{\Gamma \vdash V : int \quad \Gamma \vdash ra : int}{\Gamma \vdash ra + V : int}\ V\_ADD$
$\dfrac{P1, \Gamma \vdash P2}{\Gamma \vdash P_1 \;\supset\; P_2}\ \supset I$	$\dfrac{\Gamma \vdash ra : array(int) \quad \Gamma \vdash V : int}{\Gamma \vdash ra + V : array(int)}\ V\_ARRAY$
$\dfrac{}{P1, \Gamma \vdash P_1}\ CTX$ $\vdots$	$\dfrac{\Gamma \vdash V : array(int)}{\Gamma \vdash m[V] : int}\ M\_ARRAY$

Producing Proofs For Theorems II

$$I_1 = r1 : array(int) \wedge r2 : int$$

Safety Theorem = $I_1 \supset m[r1 + r2 + 0] : int$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{I_1 \vdash I_1}\,CTX}{I_1 \vdash r1 : array(int)}\,\wedge E1 \qquad \dfrac{\dfrac{\dfrac{}{I_1 \vdash I_1}\,CTX}{I_1 \vdash r2 : int}\,\wedge E2}{I_1 \vdash r2 + 0 : int}\,C\_ADD}{I_1 \vdash r1 + r2 + 0 : array(int)}\,V\_ARRAY}{I_1 \vdash m[r1 + r2 + 0] : int}\,M\_ARRAY}{I_1 \supset m[r1 + r2 + 0] : int}\,\supset I$$

Producing Proofs For Theorems II

$$I_1 = r1 : array(int) \wedge r2 : array(int)$$

Safety Theorem = $I_1 \supset m[r1 + r2 + 0] : int$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{I_1 \vdash I_1}\,CTX}{I_1 \vdash r1 : array(int)}\,\wedge E1 \qquad \dfrac{\dfrac{???}{I_1 \vdash r2 : int}}{I_1 \vdash r2 + 0 : int}\,C\_ADD}{I_1 \vdash r1 + r2 + 0 : array(int)}\,V\_ARRAY}{I_1 \vdash m[r1 + r2 + 0] : int}\,M\_ARRAY}{I_1 \supset m[r1 + r2 + 0] : int}\,\supset I$$

Defining Types as Predicates

- $\tau \Rightarrow \lambda v.\ \tau(v)$
- $\mathsf{char} \Rightarrow \lambda v.\ (v \geq 0 \wedge v < 256)$
- $\mathsf{ref}(\tau) \Rightarrow \lambda v.\ (v \geq 255 \wedge m[v] : \tau)$
- $\tau_1 \cap \tau_2 \Rightarrow \lambda v.\ (\tau_1(v) \wedge \tau_2(v))$
- $v : \tau \Rightarrow pf(\tau(v))$

$$\frac{v : \tau_1 \quad v : \tau_2}{v : \tau_1 \cap \tau_2} \cap I \quad \Rightarrow \quad \begin{array}{l} \cap I : pf(\tau_1(v)) \rightarrow pf(\tau_2(v)) \\ \quad \rightarrow pf((\tau_1 \cap \tau_2)(v)). \end{array}$$