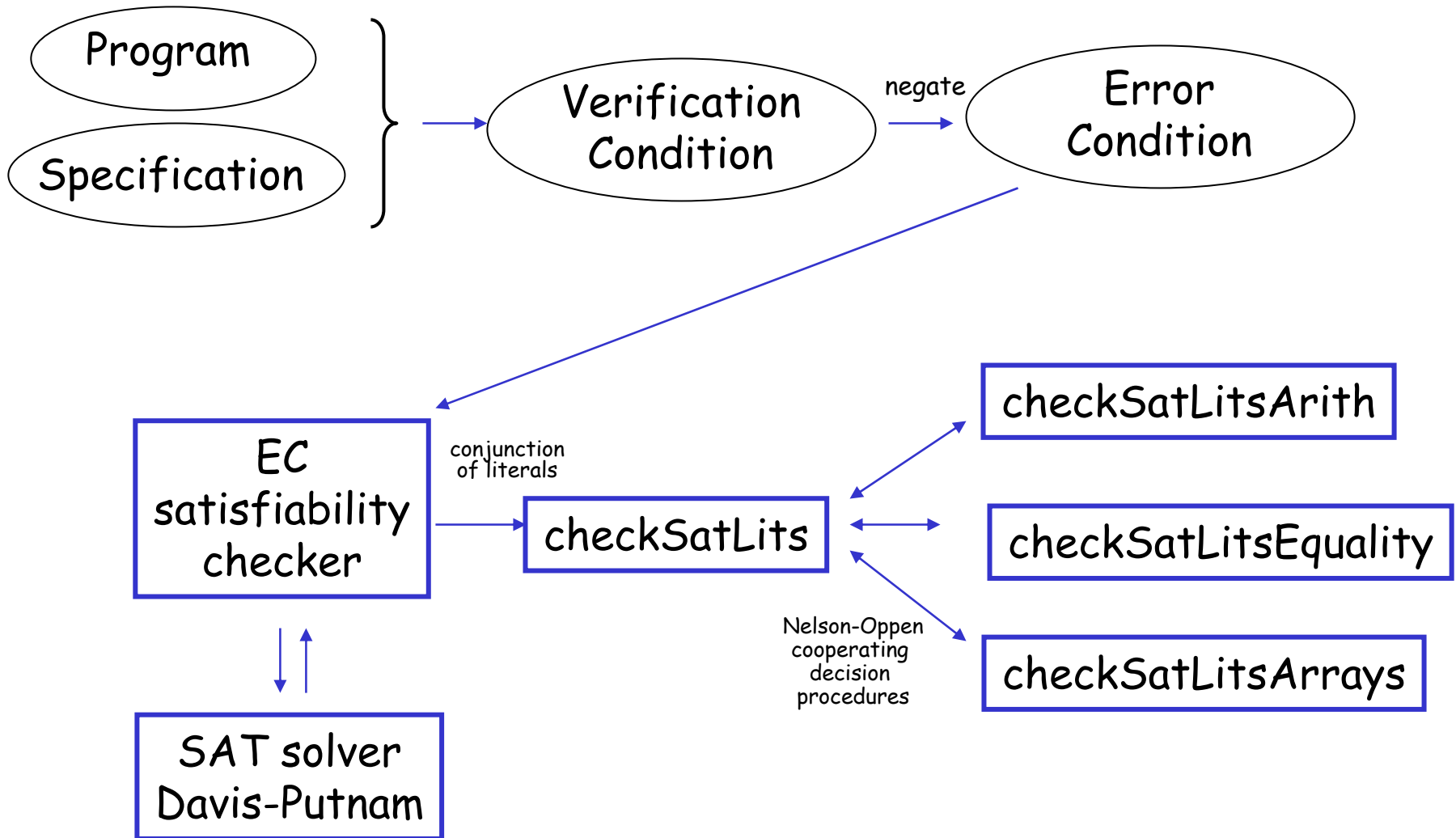


Satisfying Error Conditions 3

Lecture 15

Overview of Verification Architecture



EC Satisfiability Checker

$\{a=b\}$	$\{f(a)=f(b)\}$	$\{a=c\}$	$\{f(a)=f(c)\}$	Sat?
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	X 0
1	0	0	1	0
1	0	1	0	X 0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

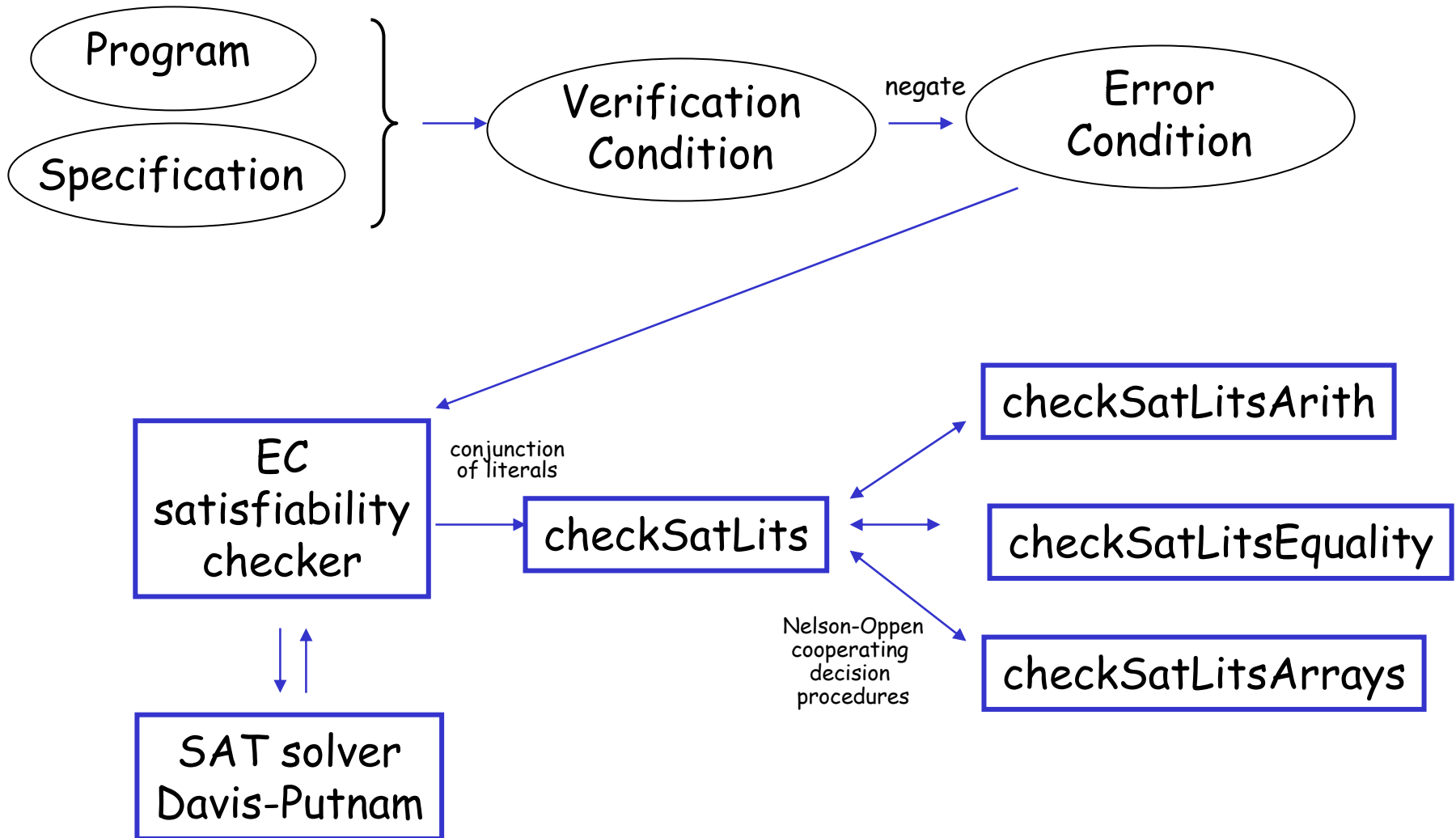
$$\begin{aligned}
 & a=b \\
 & \wedge f(a) \neq f(b) \\
 & \wedge b \neq c \\
 & \wedge f(a) \neq f(c)
 \end{aligned}$$

$$(a=b \Rightarrow f(a)=f(b))$$

$$\{a=b\} \Rightarrow \{f(a)=f(b)\}$$

Explicated tautology
removes many other
truth assignments

Overview of Verification Architecture

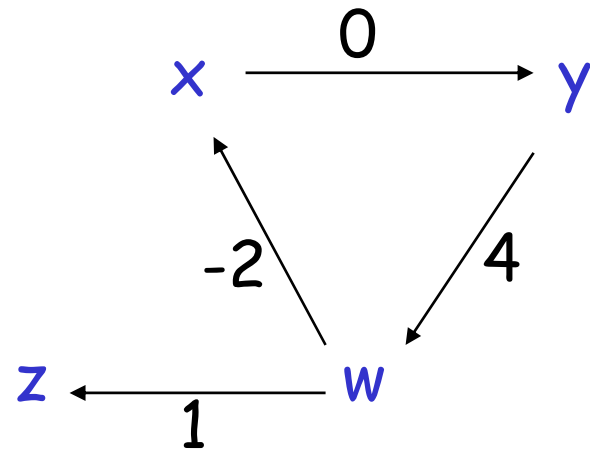


checkSatLitsArith: Difference Constraints

- A special case of linear arithmetic
- All constraints of the form:
 $x + c \leq y$
- c is a constant
- Special variable z representing 0

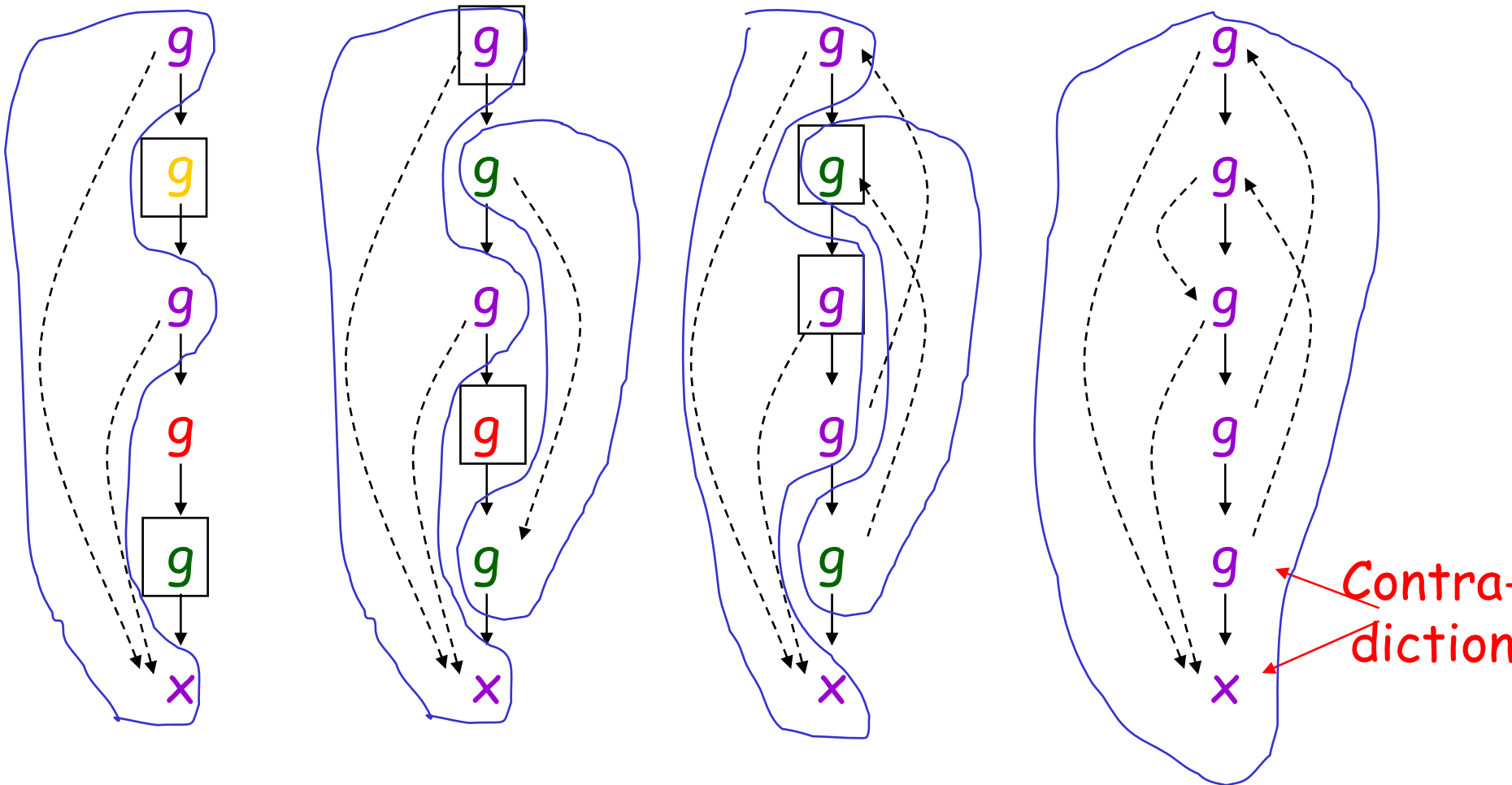
- Example

- $x \leq y$
- $y + 4 \leq w$
- $w - 2 \leq x$
- $w + 1 \leq z$

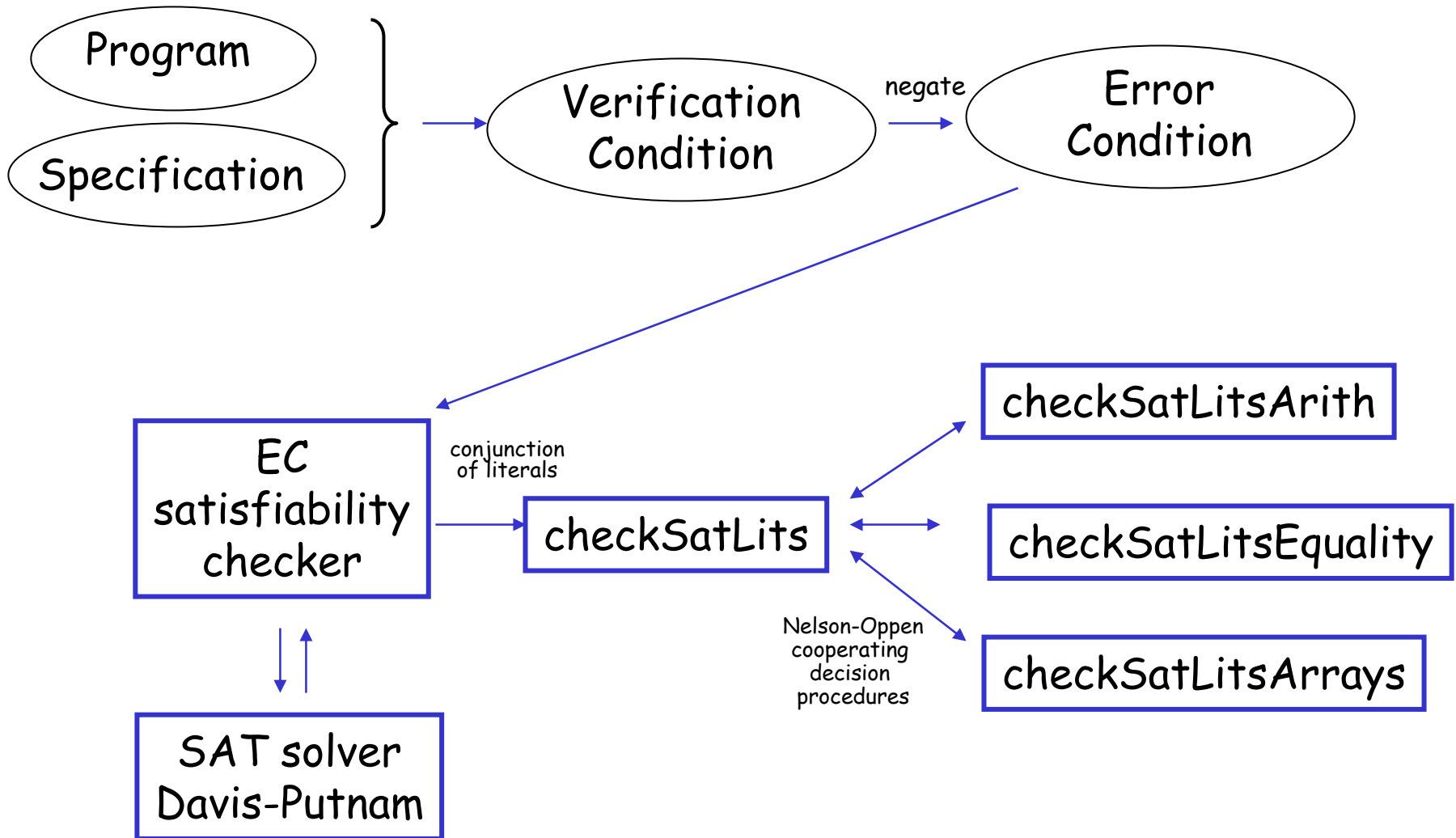


checkSatLitsEquality

- Consider: $g(g(g(x)) = x \wedge g(g(g(g(g(x)))) = x \wedge g(x) \neq x$

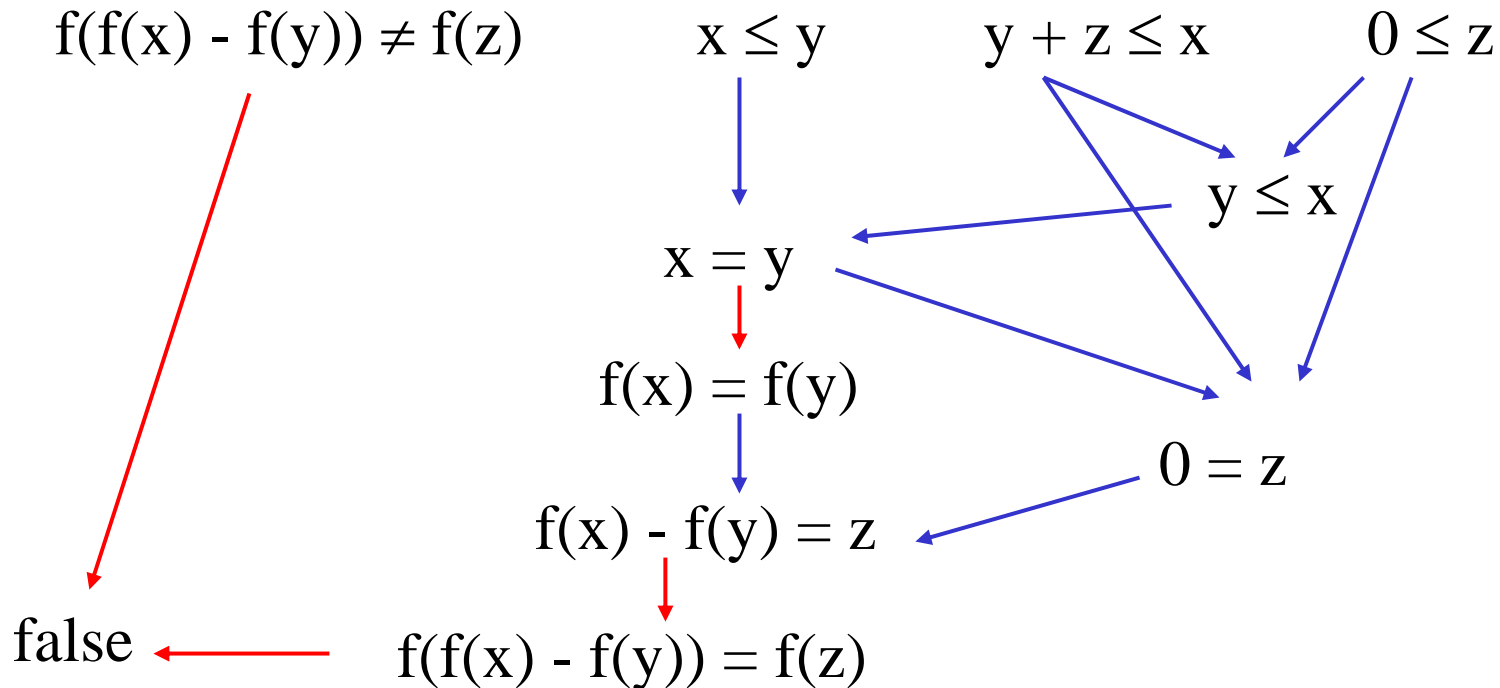


Overview of Verification Architecture



Cooperating Satisfiability Procedures

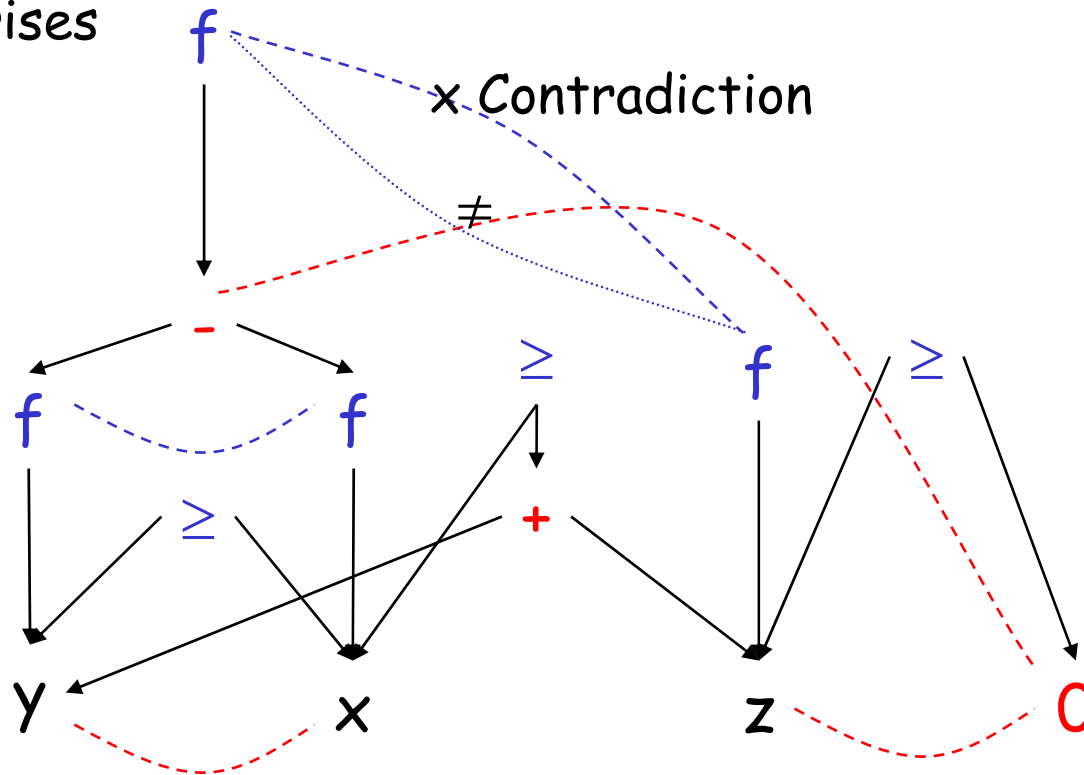
- Consider **equality** and **arithmetic**



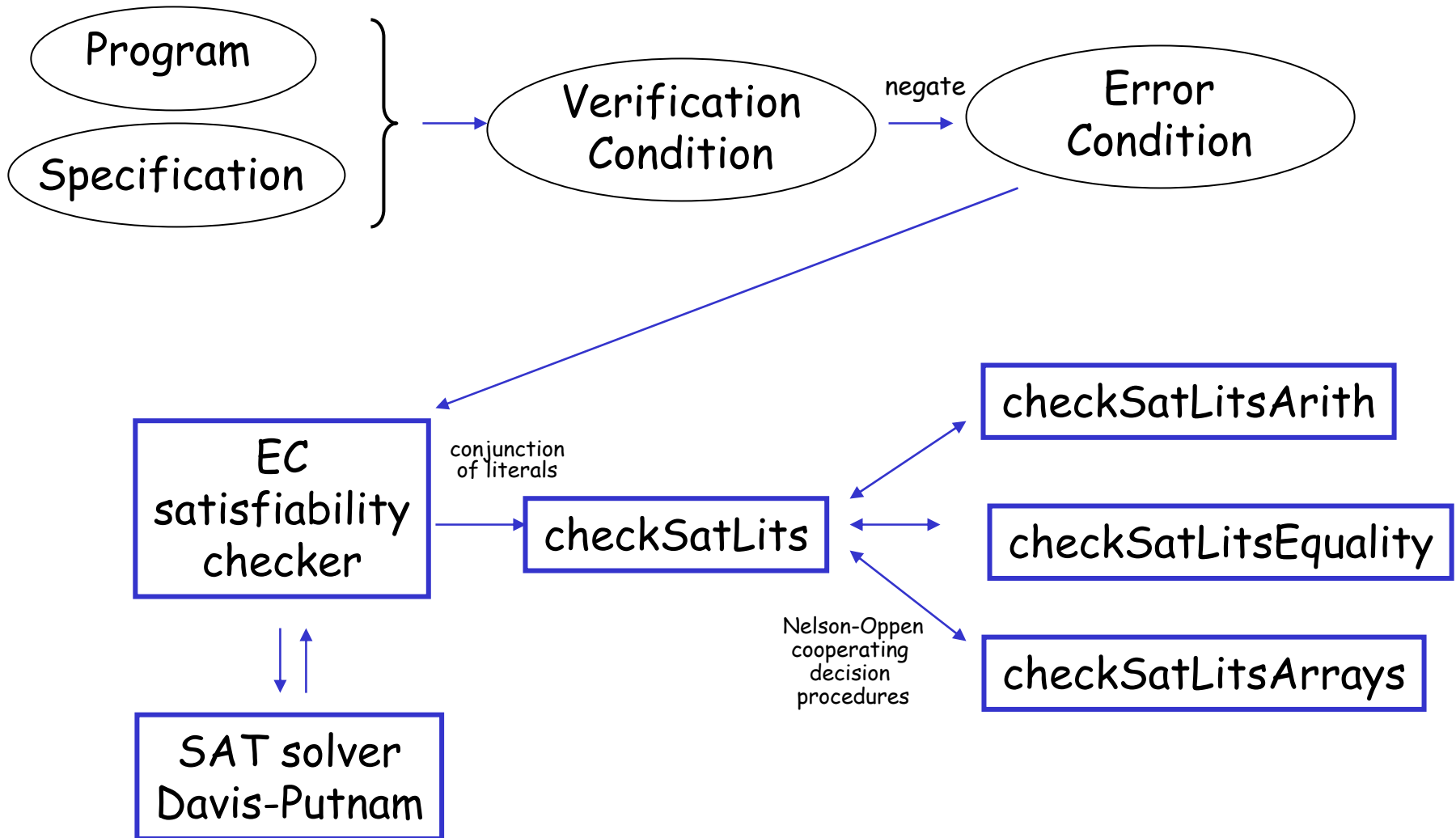
Nelson-Oppen Method (3)

3. Broadcast all discovered equalities and re-run sat. procedures

- Until no more equalities are discovered or a contradiction arises



Overview of Verification Architecture



Theory of Arrays

- Syntax and informal semantics:
 - If E denotes an address and μ a heap state then:
 - $sel(\mu, E)$ denotes the contents of memory cell
 - $upd(\mu, E, V)$ denotes a new heap state obtained from μ by writing V at address E
- Decision procedure implements following rule
 - $x = y \Rightarrow sel(upd(\mu, x, v), y) = v$
 - $x \neq y \Rightarrow sel(upd(\mu, x, v), y) = sel(\mu, y)$
 - what if $x = y$ is unknown?

Theory of Arrays

- Syntax and informal semantics:
 - If E denotes an address and μ a heap state then:
 - $\text{sel}(\mu, E)$ denotes the contents of memory cell
 - $\text{upd}(\mu, E, V)$ denotes a new heap state obtained from μ by writing V at address E

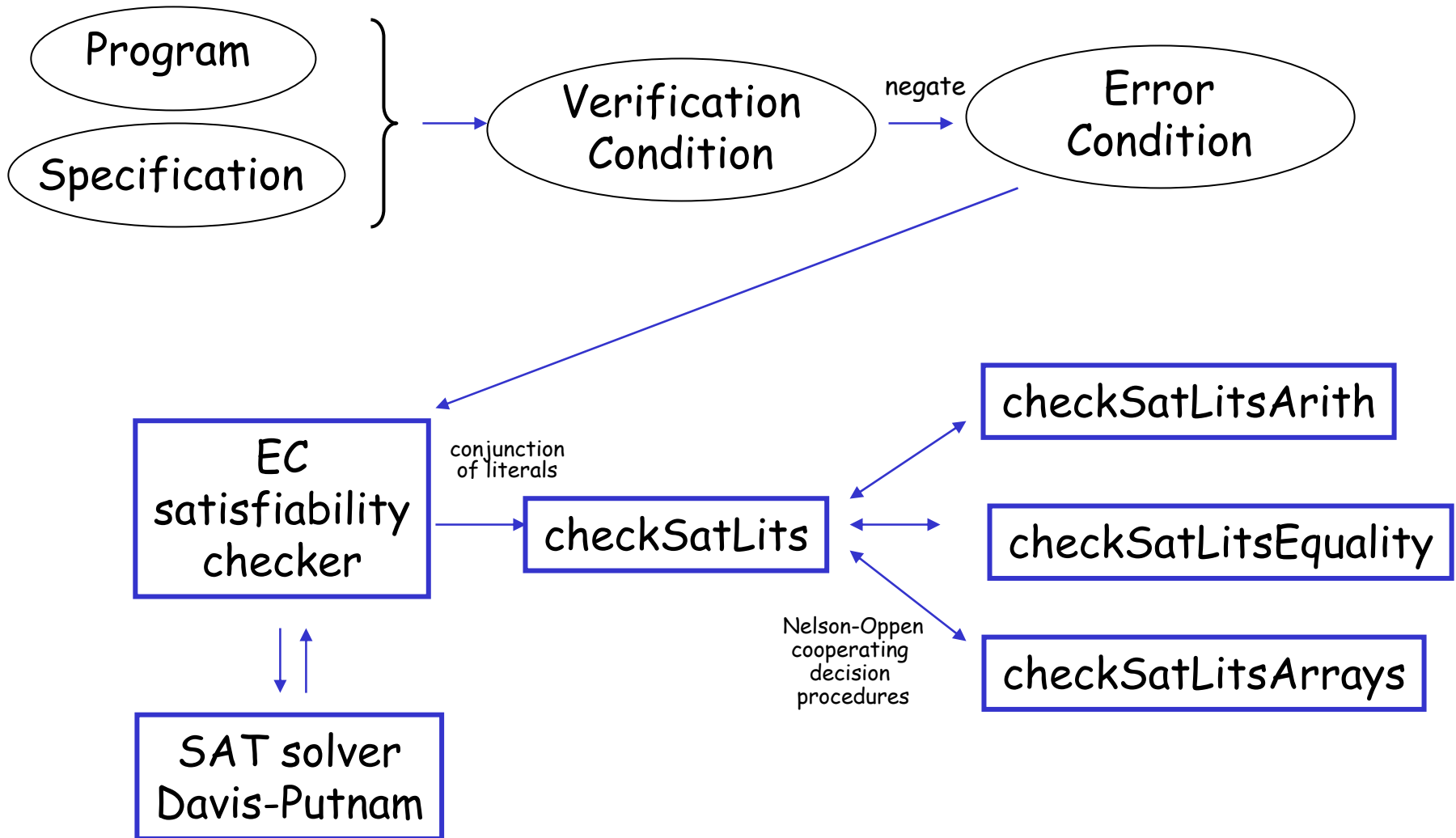
- Decision procedure implements following rule

$$x \neq y \vee \text{sel}(\text{upd}(\mu, x, v), y) = v$$

$$x = y \vee \text{sel}(\text{upd}(\mu, x, v), y) = \text{sel}(\mu, y)$$

- what if $x = y$ is unknown?
- *non-convex theory*: input facts entail disjunction of equalities, but do not entail any individual equality
- can add SAT literal $\{x=y\}$

Overview of Verification Architecture



Example

```
//@ requires x != y
//@ ensures *x < *y+1

void sort2(int *x, int *y) {
    if (*x > *y) {
        int t = *x;
        *x = *y;
        *y = t;
    }
}
```

- **Class Challenge:** Use theorem proving techniques to cooperatively verify the correctness of sort2