# A Model-Constructing Satisfiability Calculus

VMCAI 2013

Leonardo de Moura
Microsoft Research

Dejan Jovanović
NYU

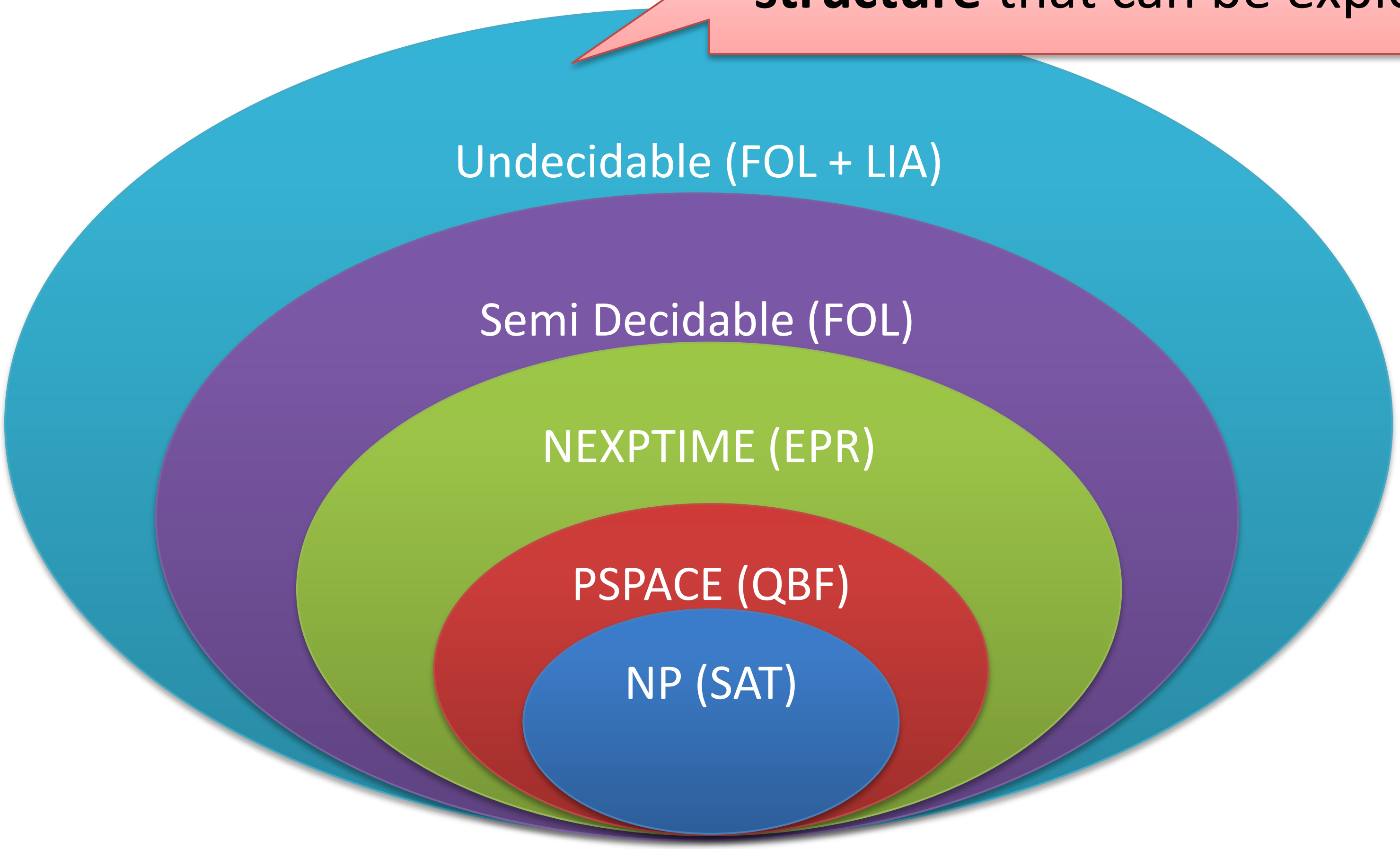
# Symbolic Reasoning

Software analysis/verification tools
need some form of symbolic reasoning

Logic is "The Calculus of Computer Science"
Zohar Manna

# Symbolic Reasoning

Practical problems often have **structure** that can be exploited.



# Logic Engines as a Service

FORMULA
Modeling Foundations.

BOOGIE

VeriFast

$Scala^{Z3}$

SAGE

ESBMC

Isabelle

Pex

Vcc

SLAM

SecGuru

Yogi

TERMINATOR

Spec Explorer

The Spec# Programming System

# Satisfiability

Solution/Model

$x^2 + y^2 < 1 \; and \; xy > 0.1$ → sat, $x = \frac{1}{8}, y = \frac{7}{8}$

$x^2 + y^2 < 1 \; and \; xy > 1$ → unsat, Proof

Is execution path *P* feasible?

SAGE

Is assertion *X* violated?

VCC

WITNESS

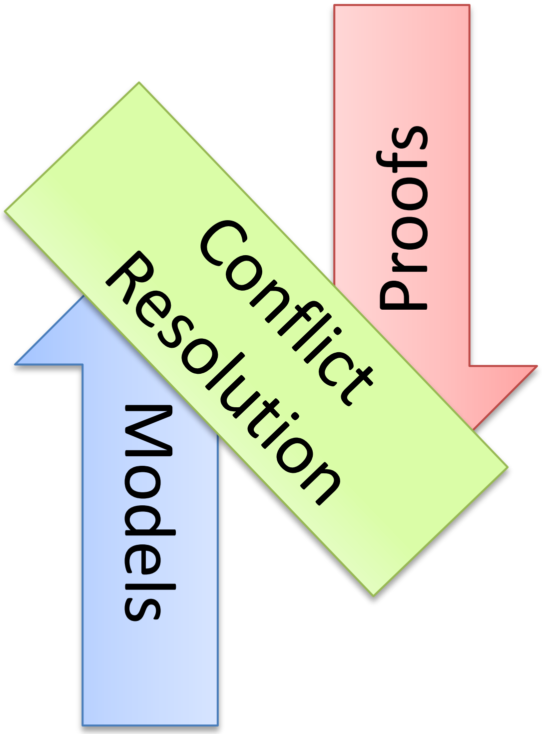
Is Formula *F* Satisfiable?

# The RISE of Model-Based Techniques in SMT

# Saturation x Search

Proof-finding

Model-finding



# SAT

$$p_1 \lor \neg p_2, \qquad \neg p_1 \lor p_2 \lor p_3, \qquad p_3$$

$$p_1 = true, \qquad p_2 = true, \qquad p_3 = true$$

CNF is a set (conjunction) set of clauses
Clause is a disjunction of literals
Literal is an atom or the negation of an atom

# Two procedures

| Resolution | DPLL |
|---|---|
| Proof-finder | Model-finder |
| Saturation | Search |

# Resolution

$$C \vee l,\quad D \vee \neg l \quad\Rightarrow\quad C \vee D$$

$$l, \neg l \quad\Rightarrow\quad \textbf{unsat}$$

Improvements

Delete tautologies $l \vee \neg l \vee C$

Ordered Resolution

Subsumption (delete redundant clauses)

$C$ *subsumes* $C \vee D$

…

# Resolution: Example

$\neg p \vee \neg q \vee r,\ \neg p \vee q,\ p \vee r,\ \neg r$

# Resolution: Example

$\neg p \vee \neg q \vee r,\ \neg p \vee q,\ p \vee r,\ \neg r \quad \Rightarrow$

$\neg p \vee \neg q \vee r,\ \neg p \vee q,\ p \vee r,\ \neg r,\ \neg q \vee r$

# Resolution: Example

$$\neg p \vee \neg q \vee r,\ \neg p \vee q,\ p \vee r,\ \neg r \quad \Rightarrow$$

$$\neg p \vee \neg q \vee r,\ {\color{red}\neg p} \vee q,\ {\color{red}p} \vee r,\ \neg r,\ \neg q \vee r \quad \Rightarrow$$

$$\neg p \vee \neg q \vee r,\ \neg p \vee q,\ p \vee r,\ \neg r,\ \neg q \vee r,\ {\color{red}q \vee r}$$

# Resolution: Example

$$\neg p \vee \neg q \vee r,\ \neg p \vee q,\ p \vee r,\ \neg r \qquad \Rightarrow$$

$$\neg p \vee \neg q \vee r,\ \neg p \vee q,\ p \vee r,\ \neg r,\ \neg q \vee r \qquad \Rightarrow$$

$$\neg p \vee \neg q \vee r,\ \neg p \vee q,\ p \vee r,\ \neg r,\ \color{red}{\neg q} \vee r,\ \color{red}{q} \vee r \qquad \Rightarrow$$

$$\neg p \vee \neg q \vee r,\ \neg p \vee q,\ p \vee r,\ \neg r,\ \neg q \vee r,\ q \vee r,\ \color{red}{r}$$

# Resolution: Example

$$\neg p \vee \neg q \vee r,\ \neg p \vee q,\ p \vee r,\ \neg r \qquad \Rightarrow$$

$$\neg p \vee \neg q \vee r,\ \neg p \vee q,\ p \vee r,\ \neg r,\ \neg q \vee r \qquad \Rightarrow$$

$$\neg p \vee \neg q \vee r,\ \neg p \vee q,\ p \vee r,\ \neg r,\ \neg q \vee r,\ q \vee r \qquad \Rightarrow$$

$$\neg p \vee \neg q \vee r,\ \neg p \vee q,\ p \vee r,\ \neg r,\ \neg q \vee r,\ q \vee r,\ r \qquad \Rightarrow$$

**unsat**

# Resolution: Problem

Exponential time and space

# Unit Resolution

$$C \lor l,\ \neg l \quad \Rightarrow \quad C$$

$C$

subsumes

$C \lor l$

# DPLL

Split rule

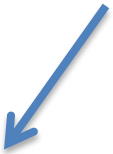
$S$

$S, p$ $\quad$ $S, \neg p$

DPLL = Unit Resolution + Split rule

# DPLL

$x \vee y, \quad \neg x \vee y, \quad x \vee \neg y, \quad \neg x \vee \neg y$



$x \vee y,$

$\neg x \vee y,$

$x \vee \neg y,$

$\neg x \vee \neg y,$

$x$

# DPLL

$x \vee y$, $\neg x \vee y$, $x \vee \neg y$, $\neg x \vee \neg y$

$x \vee y$,
$\neg x \vee y$,
$x \vee \neg y$,
$\neg x \vee \neg y$,
$x$

# DPLL

$x \vee y, \quad \neg x \vee y, \quad x \vee \neg y, \quad \neg x \vee \neg y$

$y,$
$\neg y,$
$x$

# DPLL

$x \vee y, \quad \neg x \vee y, \quad x \vee \neg y, \quad \neg x \vee \neg y$

$y,$

$\neg y,$

$x,$

$unsat$

# DPLL

$x \vee y, \quad \neg x \vee y, \quad x \vee \neg y, \quad \neg x \vee \neg y$

$y,$
$\neg y,$
$x,$
$unsat$

$x \vee y,$
$\neg x \vee y,$
$x \vee \neg y,$
$\neg x \vee \neg y,$
$\neg x$

# DPLL

$x \vee y, \quad \neg x \vee y, \quad x \vee \neg y, \quad \neg x \vee \neg y$

$y,$

$\neg y,$

$x,$

$unsat$

$x \vee y,$

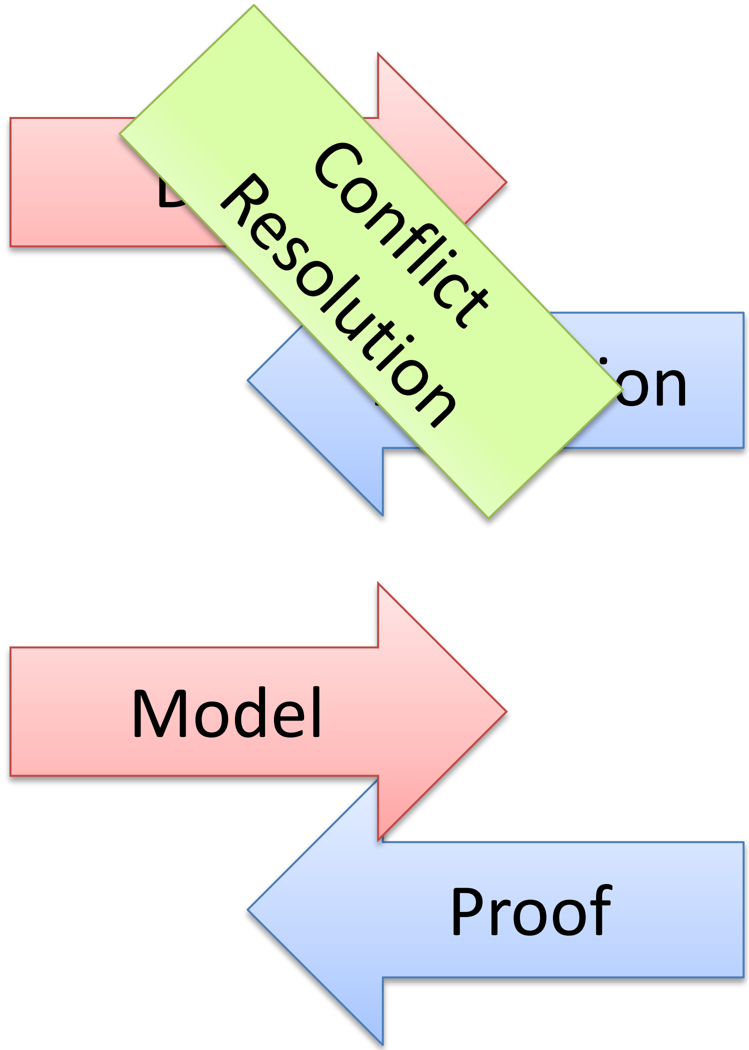
$\neg x \vee y,$

$x \vee \neg y,$

$\neg x \vee \neg y,$

$\neg x$

# CDCL: Conflict Driven Clause Learning



# Linear Arithmetic

| Fourier-Motzkin | Simplex |
|---|---|
| Proof-finder | Model-finder |
| Saturation | Search |

# Fourier-Motzkin

$$t_1 \leq ax, \qquad bx \leq t_2$$

$$\Downarrow$$

$$bt_1 \leq abx, \qquad abx \leq at_2$$

$$\Downarrow$$

$$bt_1 \leq at_2$$

Very similar to Resolution

Exponential time and space

# Simplex-based procedure

$$x \geq 0, \qquad \underbrace{x + y}_{s_1} \leq 2, \qquad \underbrace{x + 2y}_{s_2} > 4$$

$$s_1 = x + y$$
$$s_2 = x + 2y$$
$$x \geq 0,$$
$$s_1 \leq 2,$$
$$s_2 > 4$$

$s_1, s_2$ are basic (dependent)

$x, y$ are non-basic

# Simplex-based procedure: Pivoting

$$s_1 = x + y$$
$$s_2 = x + 2y$$
$$x \geq 0,$$
$$s_1 \leq 2,$$
$$s_2 > 4$$

$$s_1 = x + y$$
$$x = s_2 - 2y$$
$$x \geq 0,$$
$$s_1 \leq 2,$$
$$s_2 > 4$$

$$s_1 = s_2 - y$$
$$x = s_2 - 2y$$
$$x \geq 0,$$
$$s_1 \leq 2,$$
$$s_2 > 4$$

Example:
M(x) = 1
M(y) = 1
M($s_1$) = 2
M($s_2$) = 3

Key Property:
If an assignment satisfies the equations before a pivoting step, then it will also satisfy them after!

# Simplex: Repairing Models

If the assignment of a non-basic variable does not satisfy a bound, then fix it and propagate the change to all dependent variables.

| | | |
|---|---|---|
| $a = c - d$ | | $a = c - d$ |
| $b = c + d$ | | $b = c + d$ |
| $M(a) = 0$ | → | $M(a) = 1$ |
| $M(b) = 0$ | | $M(b) = 1$ |
| $M(c) = 0$ | | $M(c) = 1$ |
| $M(d) = 0$ | | $M(d) = 0$ |
| $1 \leq c$ | | $1 \leq c$ |

# Simplex: Repairing Models

If the assignment of a basic variable does not satisfy a bound, then pivot it, fix it, and propagate the change to its new dependent variables.

| | | |
|---|---|---|
| $a = c - d$ | $c = a + d$ | $c = a + d$ |
| $b = c + d$ | $b = a + 2d$ | $b = a + 2d$ |
| $M(a) = 0$ | $M(a) = 0$ | $M(a) = 1$ |
| $M(b) = 0$ | $M(b) = 0$ | $M(b) = 1$ |
| $M(c) = 0$ | $M(c) = 0$ | $M(c) = 1$ |
| $M(d) = 0$ | $M(d) = 0$ | $M(d) = 0$ |
| $1 \leq a$ | $1 \leq a$ | $1 \leq a$ |

# Polynomial Constraints

AKA
Existential Theory of the Reals
∃R

$$x^2 - 4x + y^2 - y + 8 < 1$$
$$xy - 2x - 2y + 4 > 1$$

# CAD "Big Picture"

1. Project/Saturate set of polynomials
2. Lift/Search: Incrementally build assignment $v$: $x_k \rightarrow \alpha_k$

   Isolate roots of polynomials $f_i(\boldsymbol{\alpha}, x)$

   Select a feasible cell $C$, and assign $x_k$ some $\alpha_k \in C$

   If there is no feasible cell, then backtrack

# CAD "Big Picture"

$x^2 + y^2 - 1 < 0$
$x\,y - 1 > 0$

1. Saturate

$x^4 - x^2 + 1$
$x^2 - 1$
$x$

2. Search

| | $(-\infty, -1)$ | $-1$ | $(-1, 0)$ | $0$ | $(0, 1)$ | $1$ | $(1, \infty)$ |
|---|---|---|---|---|---|---|---|
| $x^4 - x^2 + 1$ | + | + | + | + | + | + | + |
| $x^2 - 1$ | + | 0 | - | - | - | 0 | + |
| $x$ | - | - | - | 0 | + | + | + |

# CAD "Big Picture"

$x^2 + y^2 - 1 < 0$

$x\,y - 1 > 0$

1. Saturate

$x^4 - x^2 + 1$

$x^2 - 1$

$x$

| | $(-\infty, -\frac{1}{2})$ | $-\frac{1}{2}$ | $(-\frac{1}{2}, \infty)$ |
|---|---|---|---|
| $4 + y^2 - 1$ | + | + | + |
| $-2y - 1$ | + | 0 | - |

$x \to -2$

2. Search

| | $(-\infty, -1)$ | $-1$ | $(-1, 0)$ | $0$ | $(0, 1)$ | $1$ | $(1, \infty)$ |
|---|---|---|---|---|---|---|---|
| $x^4 - x^2 + 1$ | + | + | + | + | + | + | + |
| $x^2 - 1$ | + | 0 | - | - | - | 0 | + |
| $x$ | - | - | - | 0 | + | + | + |

# CAD "Big Picture"

$\boldsymbol{x^2 + y^2 - 1 < 0}$

$x\,y - 1 > 0$

1. Saturate

$x^4 - x^2 + 1$

$x^2 - 1$

$x$

| | $(-\infty, -\frac{1}{2})$ | $-\frac{1}{2}$ | $(-\frac{1}{2}, \infty)$ |
|---|---|---|---|
| $\boldsymbol{4 + y^2 - 1}$ | **+** | **+** | **+** |
| $-2y - 1$ | + | 0 | - |

CONFLICT

$\boldsymbol{x \to -2}$

2. Search

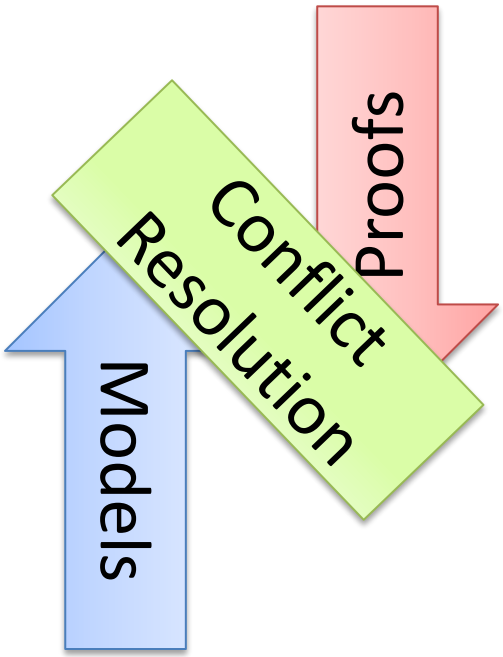
| | $(-\infty, -1)$ | $-1$ | $(-1, 0)$ | $0$ | $(0, 1)$ | $1$ | $(1, \infty)$ |
|---|---|---|---|---|---|---|---|
| $x^4 - x^2 + 1$ | + | + | + | + | + | + | + |
| $x^2 - 1$ | + | 0 | - | - | - | 0 | + |
| $x$ | - | - | - | 0 | + | + | + |

# NLSAT: Model-Based Search

Static x Dynamic

Optimistic approach

Key ideas




Start the Search before Saturate/Project

We saturate on demand

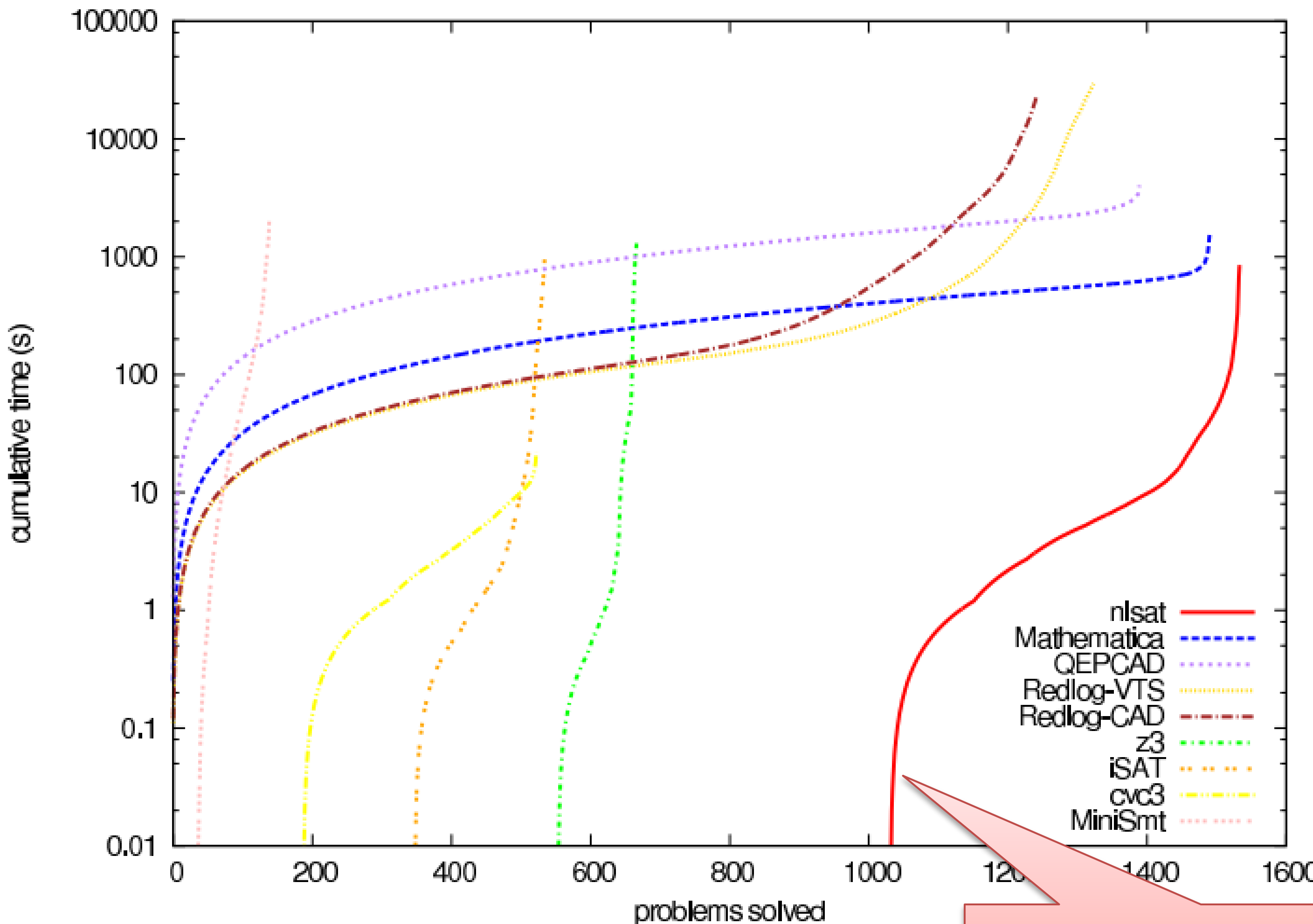
Model guides the saturation

# Experimental Results (1)

OUR NEW ENGINE

| solver | meti-tarski (1006) | | keymaera (421) | | zankl (166) | | hong (20) | | kissing (45) | | all (1658) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | solved | time (s) | solved | time (s) | solved | time (s) | solved | time (s) | solved | time (s) | solved | time (s) |
| nlsat | 1002 | 343 | **420** | **5** | **89** | **234** | 10 | 170 | 13 | 95 | **1534** | **849** |
| Mathematica | **1006** | **796** | 420 | 171 | 50 | 366 | 9 | 208 | 6 | 29 | 1491 | 1572 |
| QEPCAD | 991 | 2616 | 368 | 1331 | 21 | 38 | 6 | 43 | 4 | 5 | 1390 | 4036 |
| Redlog-VTS | 847 | 28640 | 419 | 78 | 42 | 490 | 6 | 3 | 10 | 275 | 1324 | 29488 |
| Redlog-CAD | 848 | 21706 | 363 | 730 | 21 | 173 | 6 | 2 | 4 | 0 | 1242 | 22613 |
| z3 | 266 | 83 | 379 | 1216 | 21 | 0 | 1 | 0 | 0 | 0 | 667 | 1299 |
| iSAT | 203 | 122 | 291 | 16 | 21 | 24 | **20** | **822** | 0 | 0 | 535 | 986 |
| cvc3 | 150 | 13 | 361 | 5 | 12 | 3 | 0 | 0 | 0 | 0 | 523 | 22 |
| MiniSmt | 40 | 697 | 35 | 0 | 46 | 1370 | 0 | 0 | **18** | **44** | 139 | 2112 |

# Experimental Results (2)




OUR NEW ENGINE

# Other examples

Delayed
Theory Combination
[Bruttomesso et al 2006]

X

Model-Based
Theory Combination

# Other examples

Array Theory by Axiom Instantiation

X

Lemmas on Demand For Theory of Array [Brummayer-Biere 2009]

$$\forall a, i, v: \quad a[i := v][i] = v$$
$$\forall a, i, j, v: \; i = j \lor a[i := v][j] = a[j]$$

# Other examples

(for linear arithmetic)

Fourier-Motzkin **X** Generalizing DPLL to richer logics

[McMillan et al 2009]

Conflict Resolution

[Korovin et al 2009]

# Saturation: successful instances

Polynomial time procedures

Gaussian Elimination

Congruence Closure

# SAT + Theory Solvers

**Basic Idea**

$x \geq 0,\ y = x + 1,\ (y > 2 \vee y < 1)$

$p_1,\ \ p_2,\ (p_3 \vee p_4)$

$p_1 \equiv (x \geq 0),\ p_2 \equiv (y = x + 1),$
$p_3 \equiv (y > 2),\ p_4 \equiv (y < 1)$

[Audemard et al - 2002], [Barrett et al - 2002], [de Moura et al - 2002]

# SAT + Theory Solvers

**Basic Idea**

$x \geq 0, y = x + 1, (y > 2 \vee y < 1)$

$p_1, \; p_2, (p_3 \vee p_4)$

$p_1 \equiv (x \geq 0), p_2 \equiv (y = x + 1),$

$p_3 \equiv (y > 2), p_4 \equiv (y < 1)$

SAT
Solver

# SAT + Theory Solvers

**Basic Idea**

$x \geq 0,\ y = x + 1,\ (y > 2 \vee y < 1)$

$p_1,\ p_2,\ (p_3 \vee p_4)$

$p_1 \equiv (x \geq 0),\ p_2 \equiv (y = x + 1),$

$p_3 \equiv (y > 2),\ p_4 \equiv (y < 1)$

SAT Solver

Assignment

$p_1,\ p_2,\ \neg p_3,\ p_4$

# SAT + Theory Solvers

**Basic Idea**

$x \geq 0,\ y = x + 1,\ (y > 2 \vee y < 1)$

$p_1,\ p_2,\ (p_3 \vee p_4)$

$p_1 \equiv (x \geq 0),\ p_2 \equiv (y = x + 1),$

$p_3 \equiv (y > 2),\ p_4 \equiv (y < 1)$

SAT Solver

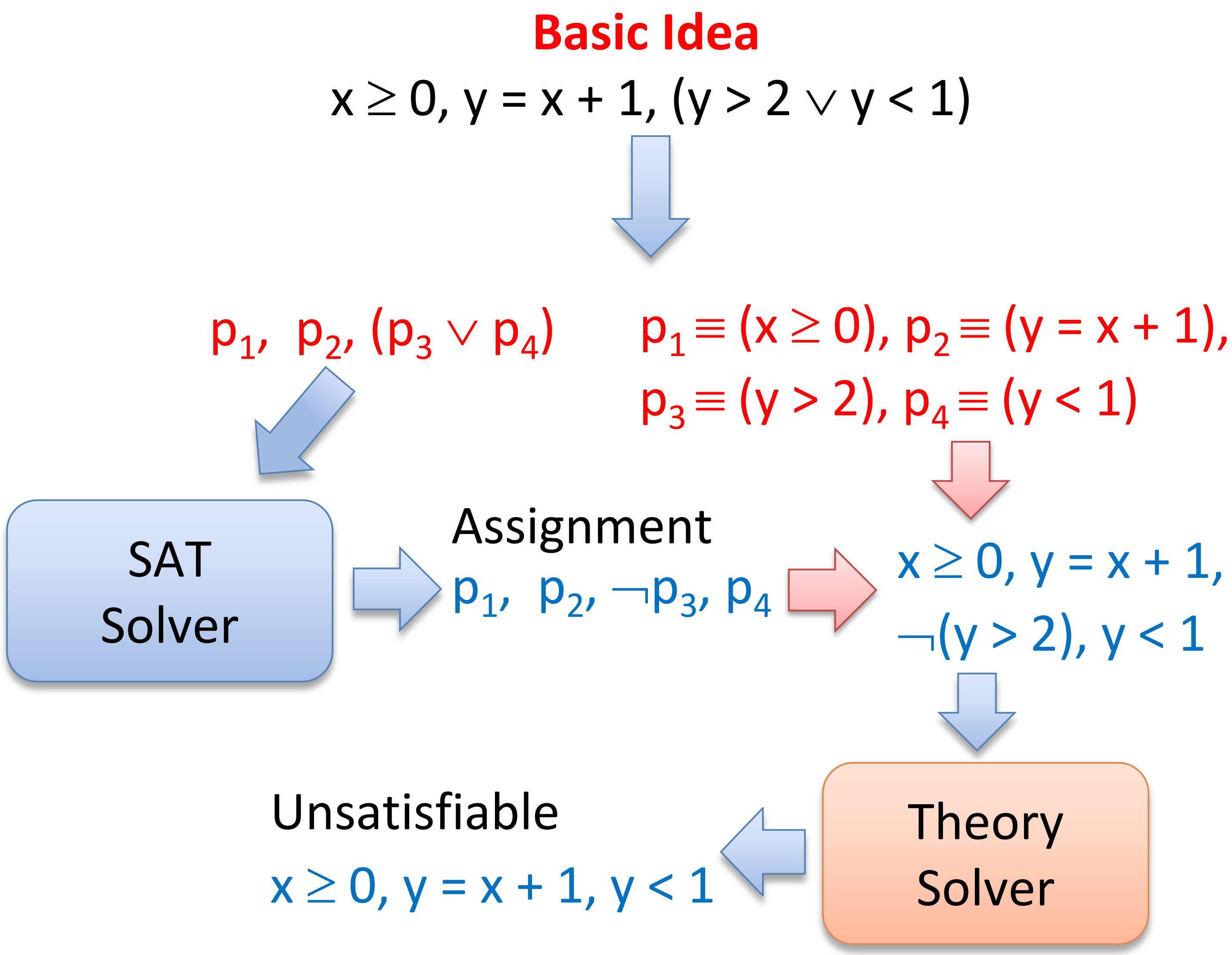
Assignment

$p_1,\ p_2,\ \neg p_3,\ p_4$

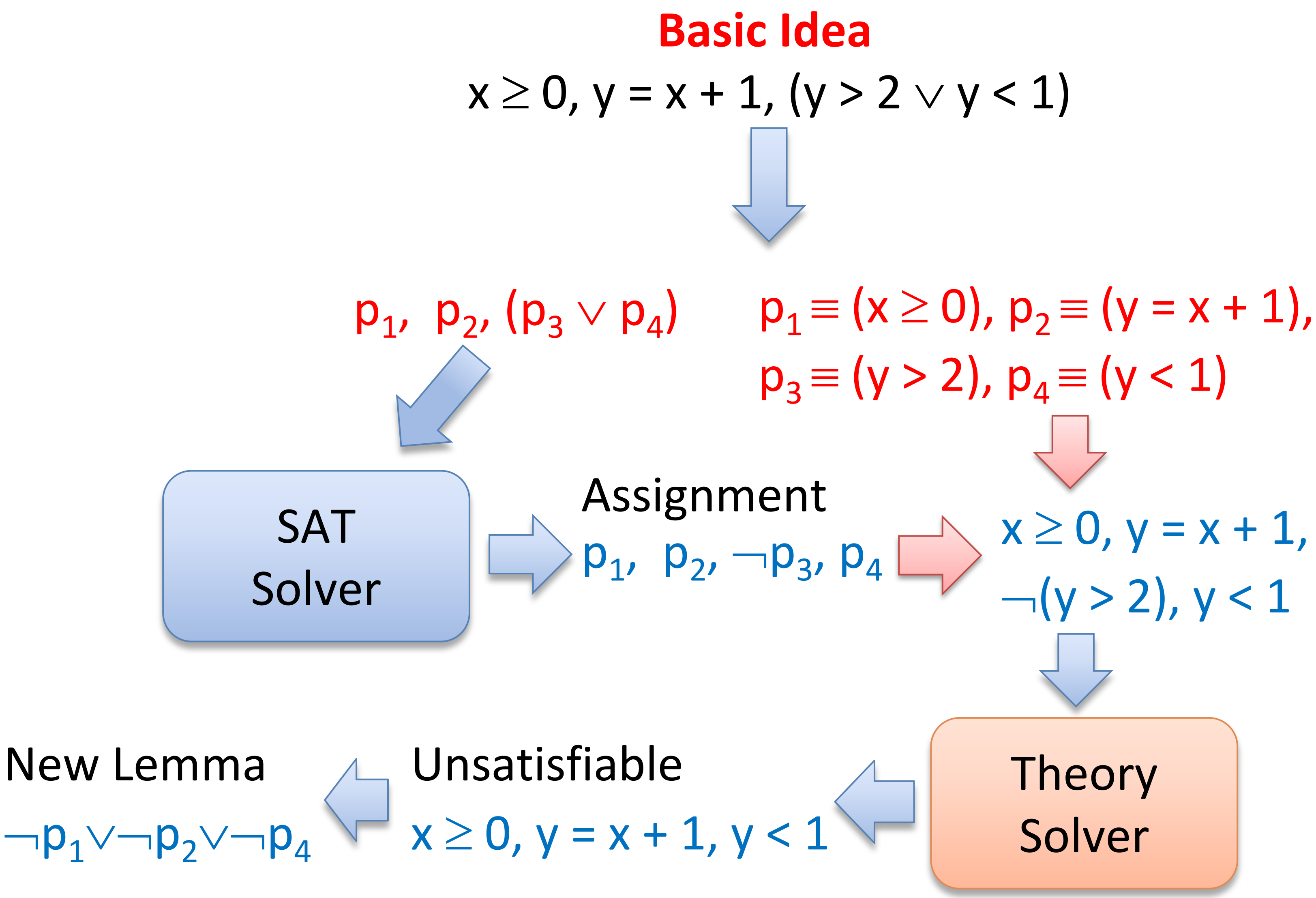
$x \geq 0,\ y = x + 1,$

$\neg(y > 2),\ y < 1$

# SAT + Theory Solvers



# SAT + Theory Solvers



# SAT + Theory Solvers: refinements

Incrementality

Efficient Backtracking

Efficient Lemma Generation

Theory propagation  [Ganzinger et all – 2004]

# MCSat

$x \geq 2, \qquad (\neg x \geq 1 \vee y \geq 1), \qquad (x^2 + y^2 \leq 1 \vee xy > 1)$

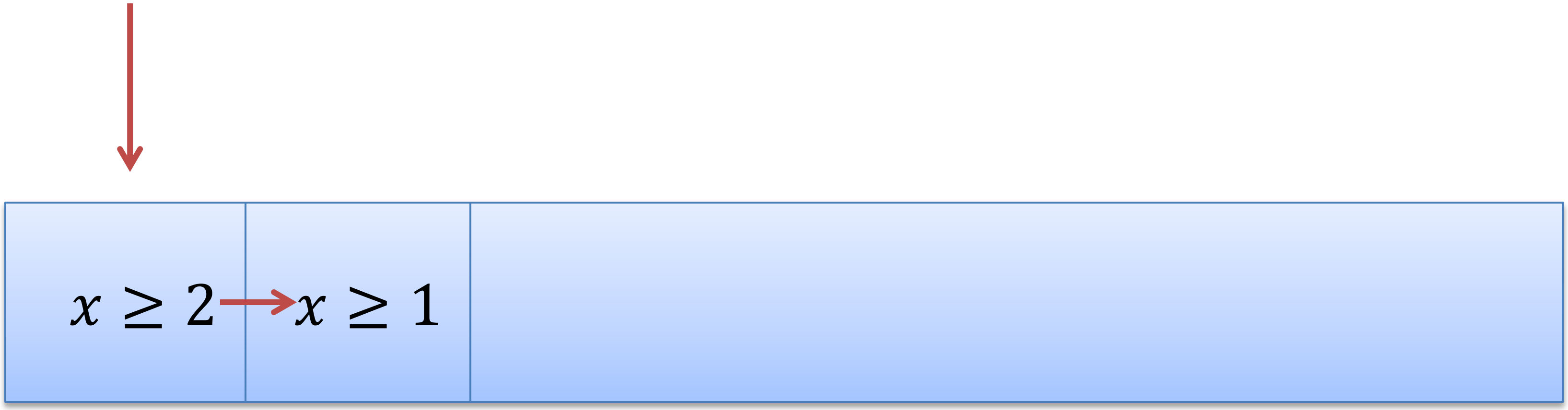
# MCSat

$x \geq 2, \quad (\neg x \geq 1 \vee y \geq 1), \quad (x^2 + y^2 \leq 1 \vee xy > 1)$


Propagations

# MCSat

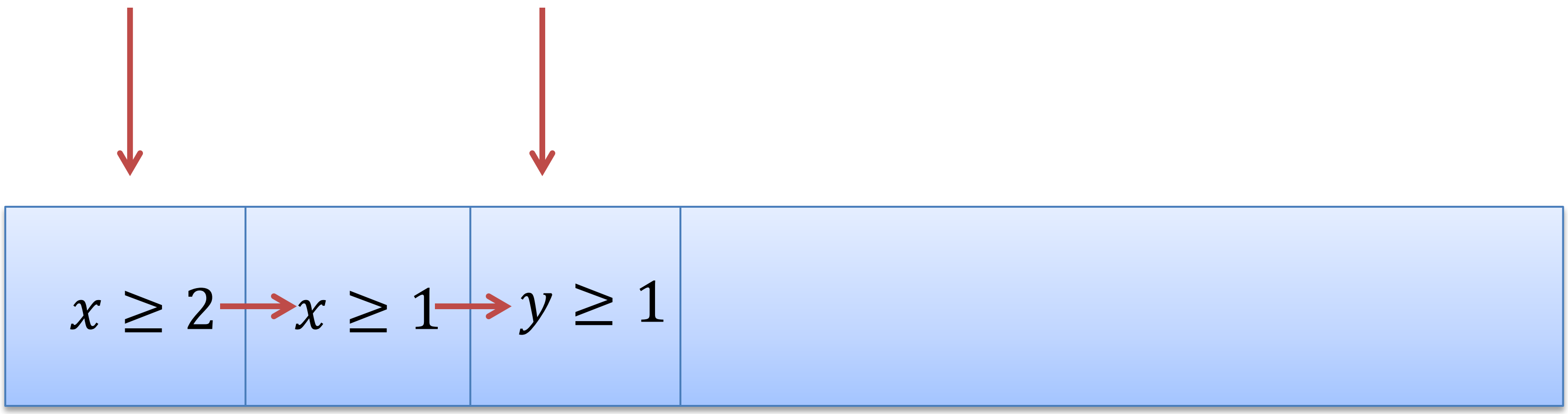
$$x \geq 2, \qquad (\neg x \geq 1 \lor y \geq 1), \qquad (x^2 + y^2 \leq 1 \lor xy > 1)$$




Propagations

# MCSat

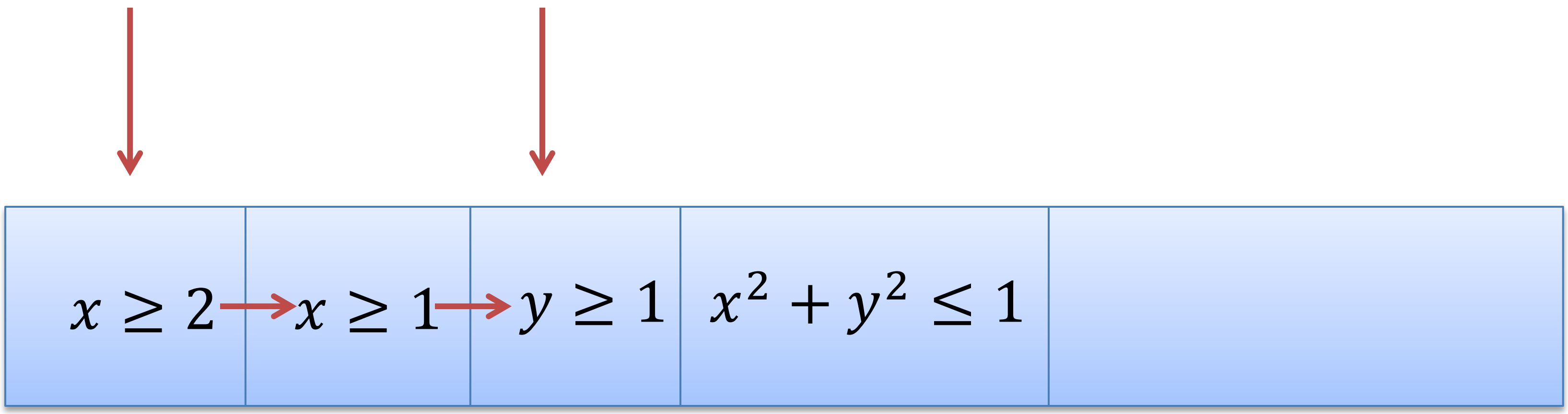
$x \geq 2, \quad (\neg x \geq 1 \vee y \geq 1), \quad (x^2 + y^2 \leq 1 \vee xy > 1)$




Propagations

# MCSat

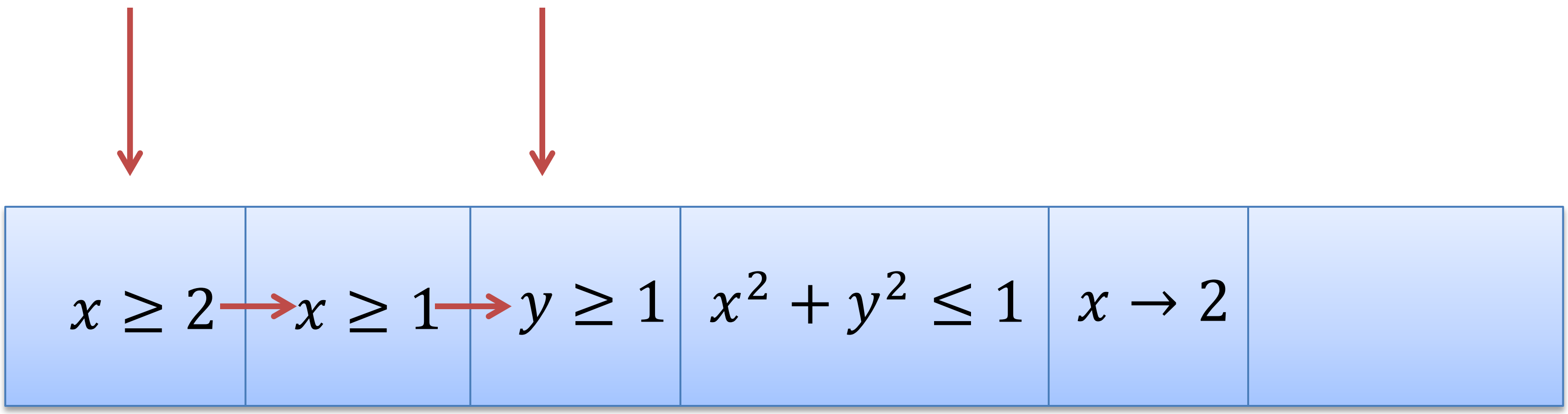
$x \geq 2$, $(\neg x \geq 1 \vee y \geq 1)$, $(x^2 + y^2 \leq 1 \vee xy > 1)$




Decisions

# MCSat

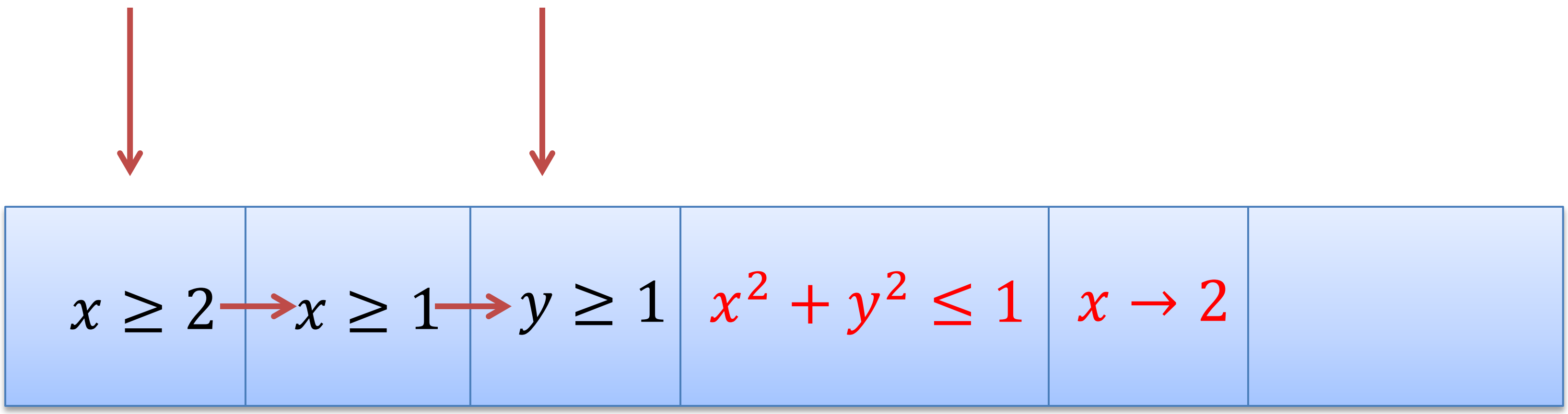
$$x \geq 2, \quad (\neg x \geq 1 \vee y \geq 1), \quad (x^2 + y^2 \leq 1 \vee xy > 1)$$




Model Assignments

# MCSat

$x \geq 2, \quad (\neg x \geq 1 \vee y \geq 1), \quad (x^2 + y^2 \leq 1 \vee xy > 1)$

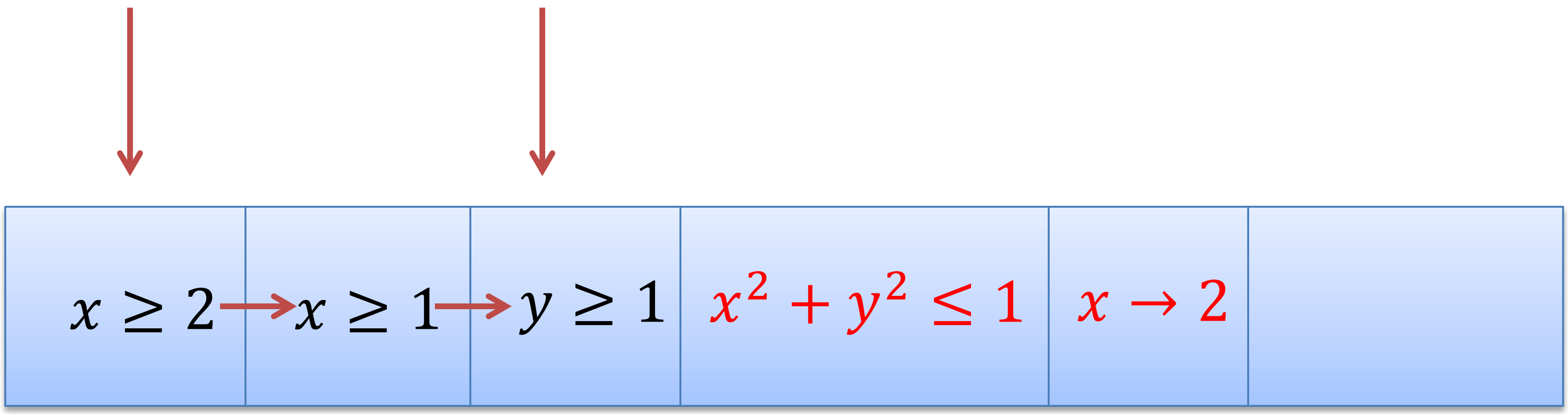



Conflict

We can't find a value of $y$
s.t. $4 + y^2 \leq 1$

# MCSat

$x \geq 2, \quad (\neg x \geq 1 \lor y \geq 1), \quad (x^2 + y^2 \leq 1 \lor xy > 1)$



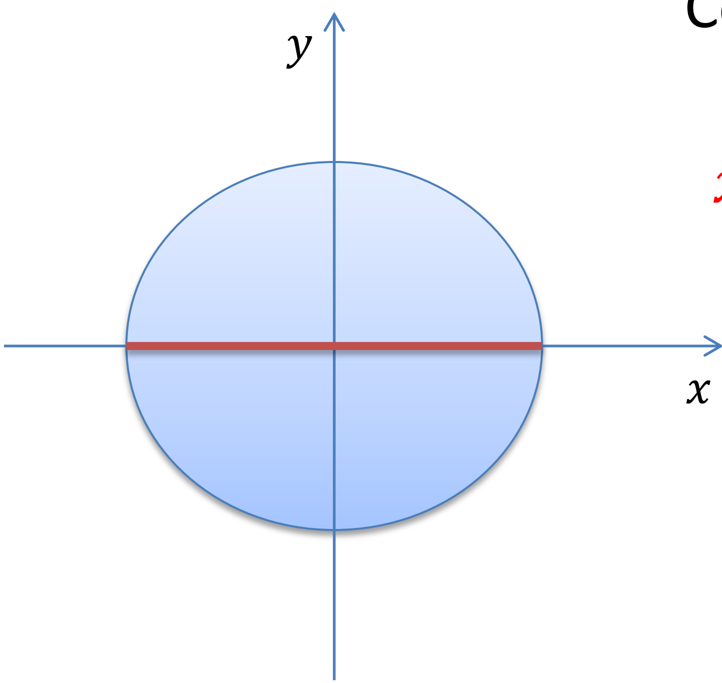

Conflict

We can't find a value of $y$ s.t. $4 + y^2 \leq 1$

Learning that $\neg(x^2 + y^2 \leq 1) \lor \neg(x = 2)$ is not productive

$x \geq 2, \quad (\neg x \geq 1 \vee y \geq 1), \quad (x^2 + y^2 \leq 1 \vee xy > 1)$

$x \geq 2 \rightarrow x \geq 1 \rightarrow y \geq 1$ | $x^2 + y^2 \leq 1$ | $x \rightarrow 2$




Conflict

$x^2 + y^2 \leq 1$

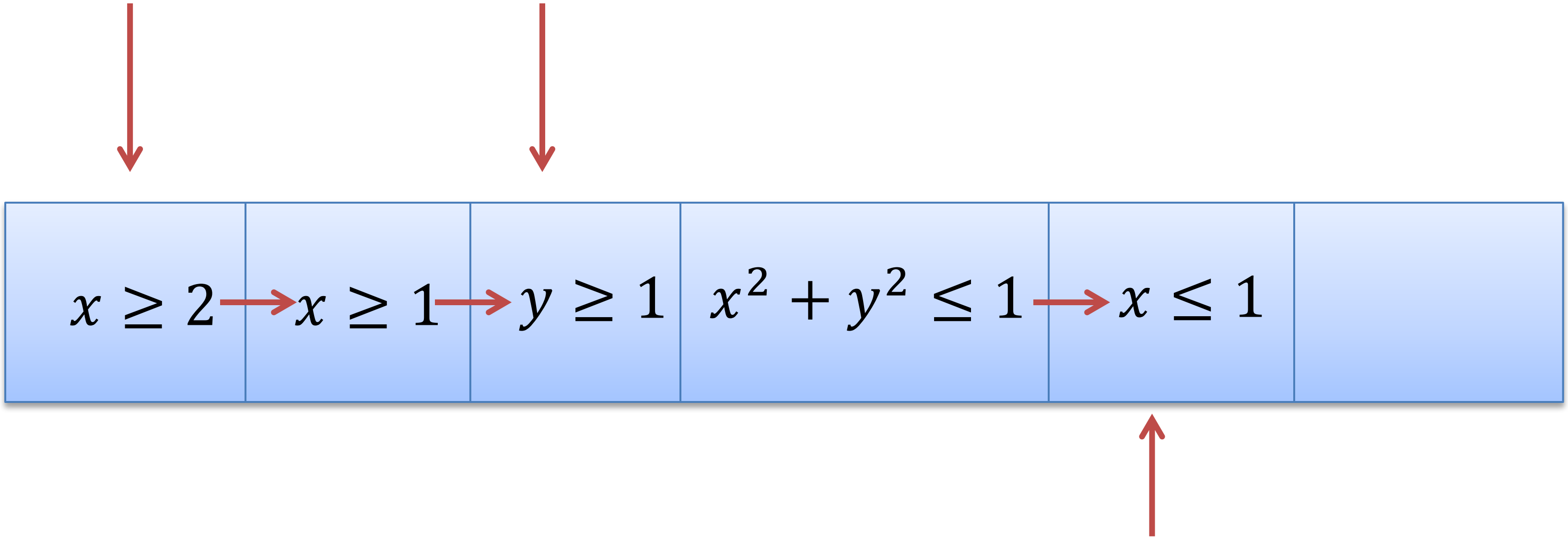
$x \rightarrow 2$

$-1 \leq x, x \leq 1$

$\neg(x^2 + y^2 \leq 1) \vee x \leq 1$

# MCSat

$x \geq 2, \quad (\neg x \geq 1 \vee y \geq 1), \quad (x^2 + y^2 \leq 1 \vee xy > 1)$




$\neg(x^2 + y^2 \leq 1) \vee x \leq 1$

# MCSat

$x \geq 2, \quad (\neg x \geq 1 \vee y \geq 1), \quad (x^2 + y^2 \leq 1 \vee xy > 1)$

$x \geq 2 \rightarrow x \geq 1 \rightarrow y \geq 1 \quad x^2 + y^2 \leq 1 \rightarrow x \leq 1$

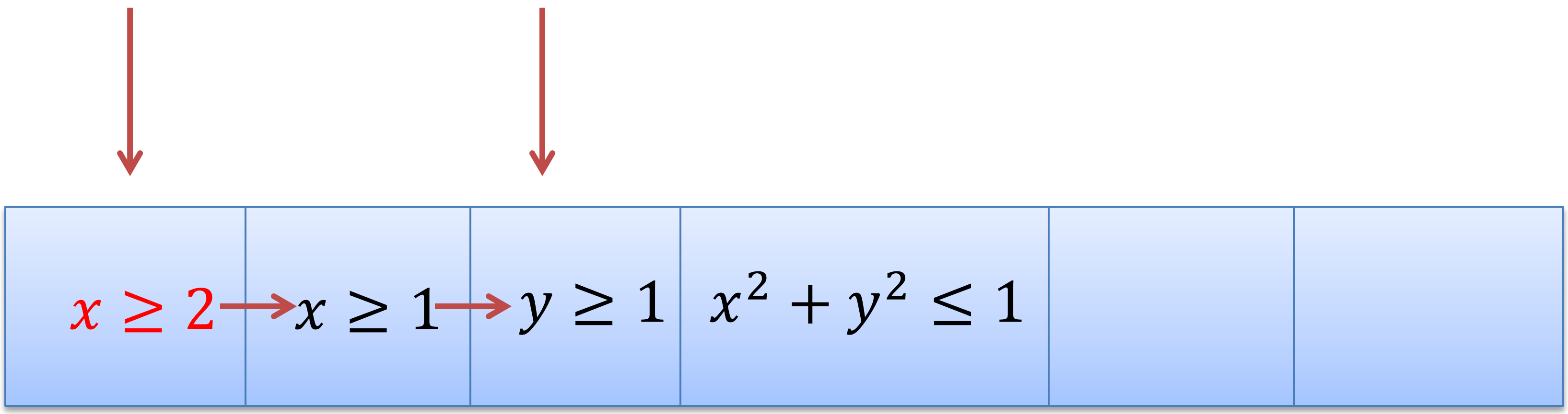
$\neg(x^2 + y^2 \leq 1) \vee x \leq 1$

Conflict

$\neg(x \geq 2) \vee \neg(x \leq 1)$

# MCSat

$x \geq 2, \qquad (\neg x \geq 1 \vee y \geq 1), \qquad (x^2 + y^2 \leq 1 \vee xy > 1)$



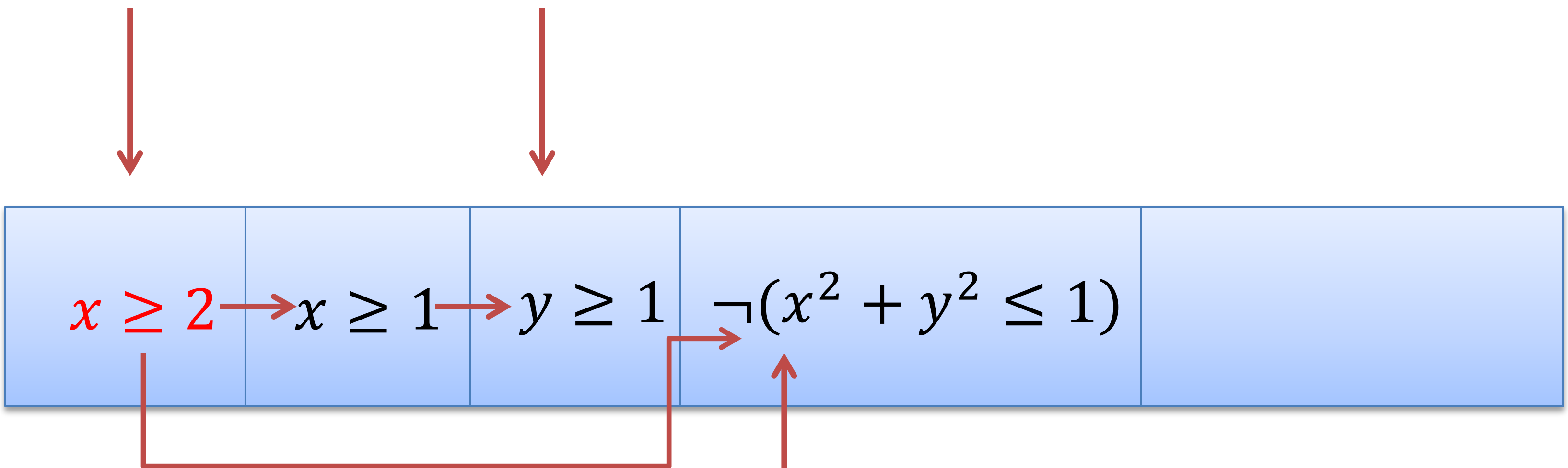

$\neg(x^2 + y^2 \leq 1) \vee x \leq 1$

Learned by resolution

$\neg(x \geq 2) \vee \neg(x^2 + y^2 \leq 1)$

# MCSat

$x \geq 2$, $(\neg x \geq 1 \vee y \geq 1)$, $(x^2 + y^2 \leq 1 \vee xy > 1)$




$\neg(x \geq 2) \vee \neg(x^2 + y^2 \leq 1)$ $\neg(x^2 + y^2 \leq 1) \vee x \leq 1$

# MCSat – Finite Basis

Every theory that admits quantifier elimination has a finite basis (given a fixed assignment order)

$$F[x_1, \dots, x_n, y_1, \dots, y_m]$$

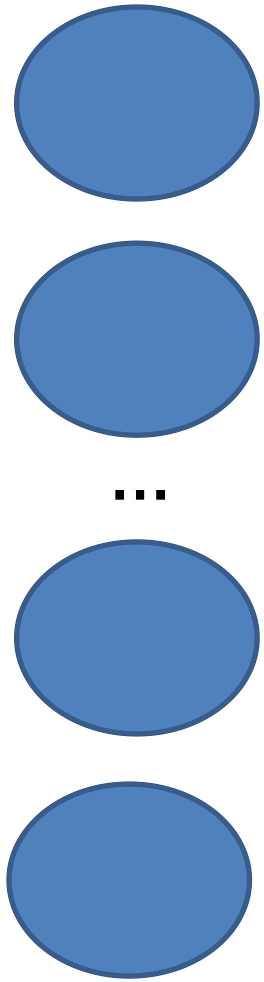
$$y_1 \to \alpha_1, \dots, y_m \to \alpha_m$$

$$\exists x_1, \dots, x_n : F[x_1, \dots, x_n, y]$$

$$C_1[y_1, \dots, y_m] \wedge \cdots \wedge C_k[y_1, \dots, y_m]$$

$$\neg F[x_1, \dots, x_n, y_1, \dots, y_m] \vee C_k[y_1, \dots, y_m]$$

# MCSat – Finite Basis



$F_n[x_{1,}x_2, \ldots, x_{n-1}, x_n]$

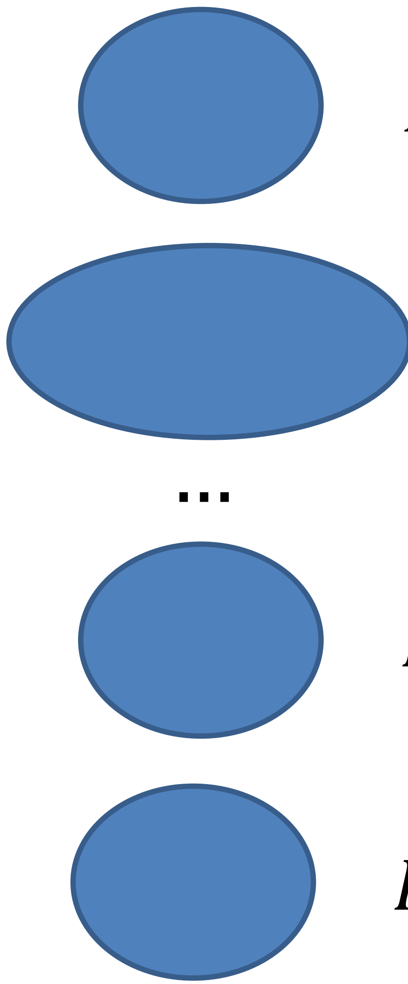
$F_{n-1}[x_{1,}x_2, \ldots, x_{n-1}]$

…

$F_2[x_{1,}x_2]$

$F_1[x_1]$

# MCSat – Finite Basis



$F_n[x_1, x_2, \ldots, x_{n-1}, x_n]$

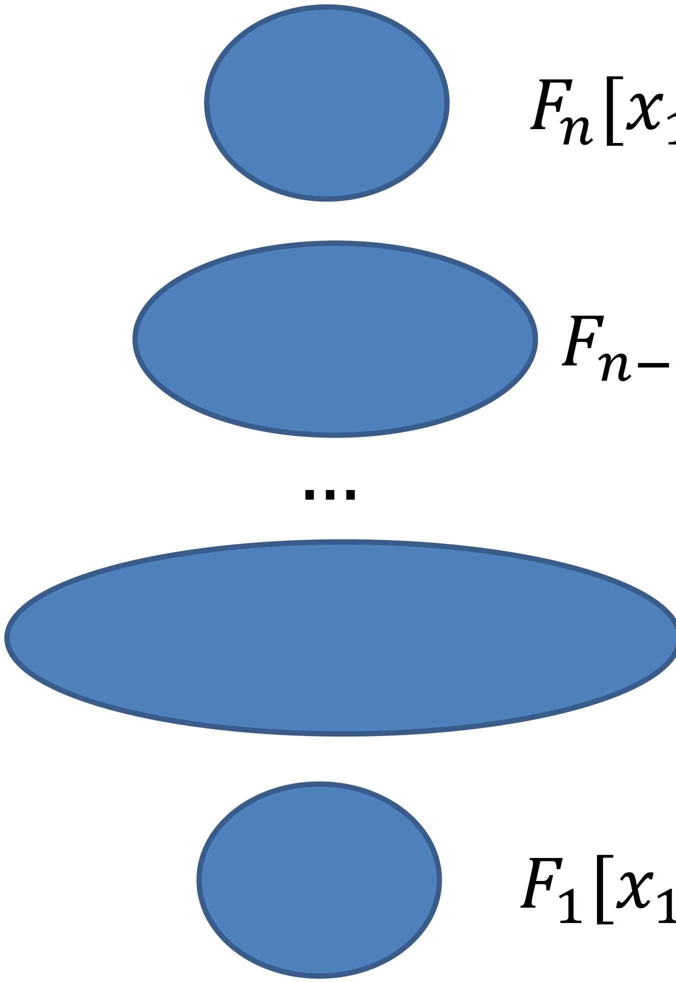
$F_{n-1}[x_1, x_2, \ldots, x_{n-1}]$

…

$F_2[x_1, x_2]$

$F_1[x_1]$

# MCSat – Finite Basis



$F_n[x_1, x_2, \ldots, x_{n-1}, x_n]$

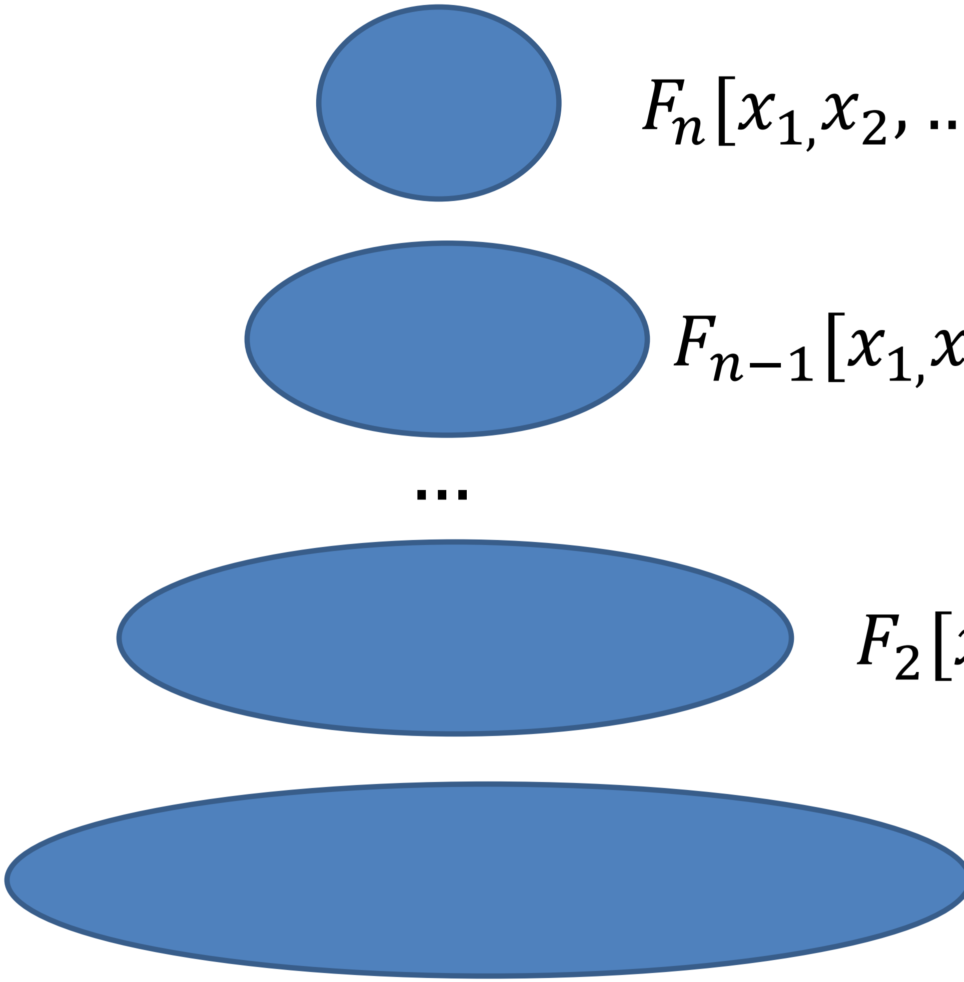
$F_{n-1}[x_1, x_2, \ldots, x_{n-1}]$

…

$F_2[x_1, x_2]$

$F_1[x_1]$

# MCSat – Finite Basis



$F_n[x_{1,}x_2, \ldots, x_{n-1}, x_n]$

$F_{n-1}[x_{1,}x_2, \ldots, x_{n-1}]$

$F_2[x_{1,}x_2]$

$F_1[x_1]$

# MCSat – Finite Basis

Every "finite" theory has a finite basis

$F[x_1, \dots, x_n, y_1, \dots, y_m]$ $y_1 \to \alpha_1, \dots, y_m \to \alpha_m$

$$y_1 = \alpha_1, \dots, y_m = \alpha_m$$

# MCSat – Finite Basis

Theory of uninterpreted functions has a finite basis

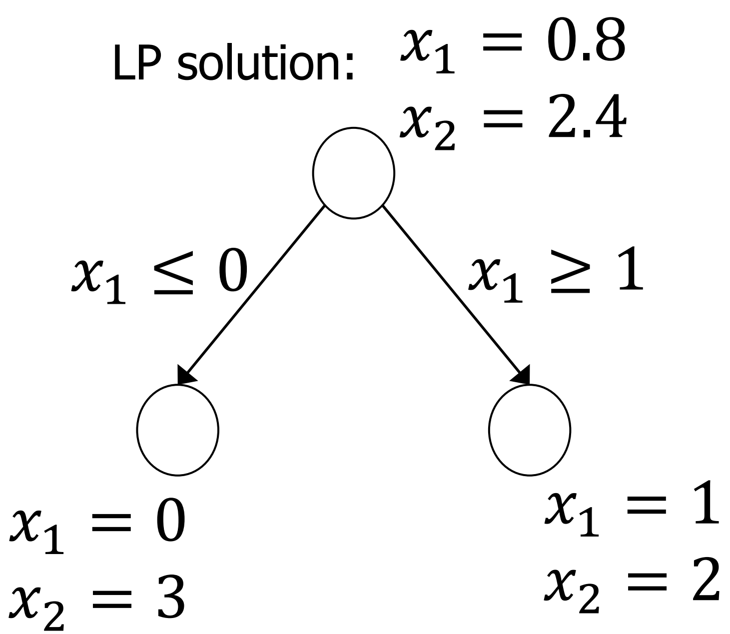
Theory of arrays has a finite basis [Brummayer- Biere 2009]

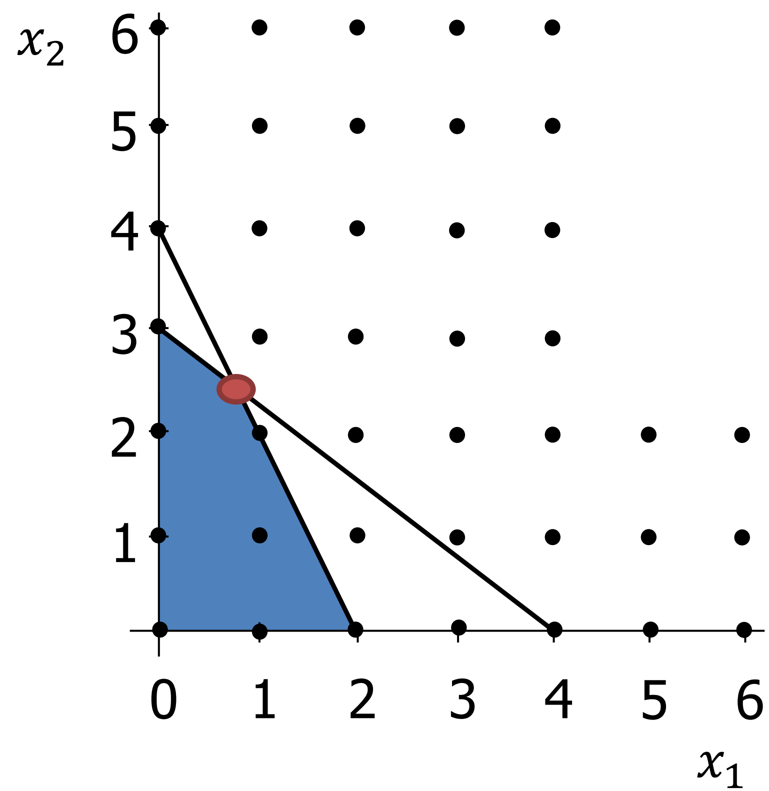
In both cases the Finite Basis is essentially composed of equalities between existing terms.

# MCSat – Finite Basis

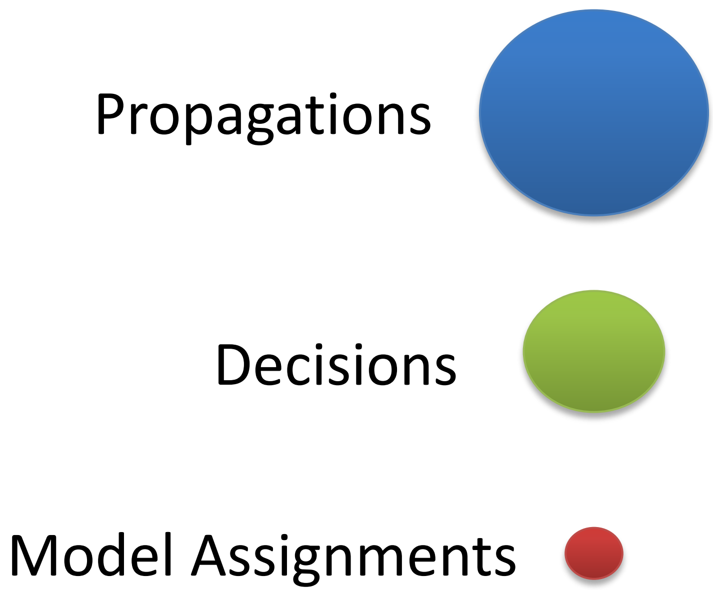
We can also use literals from the finite basis in decisions.

Application: simulate branch&bound for bounded linear integer arithmetic

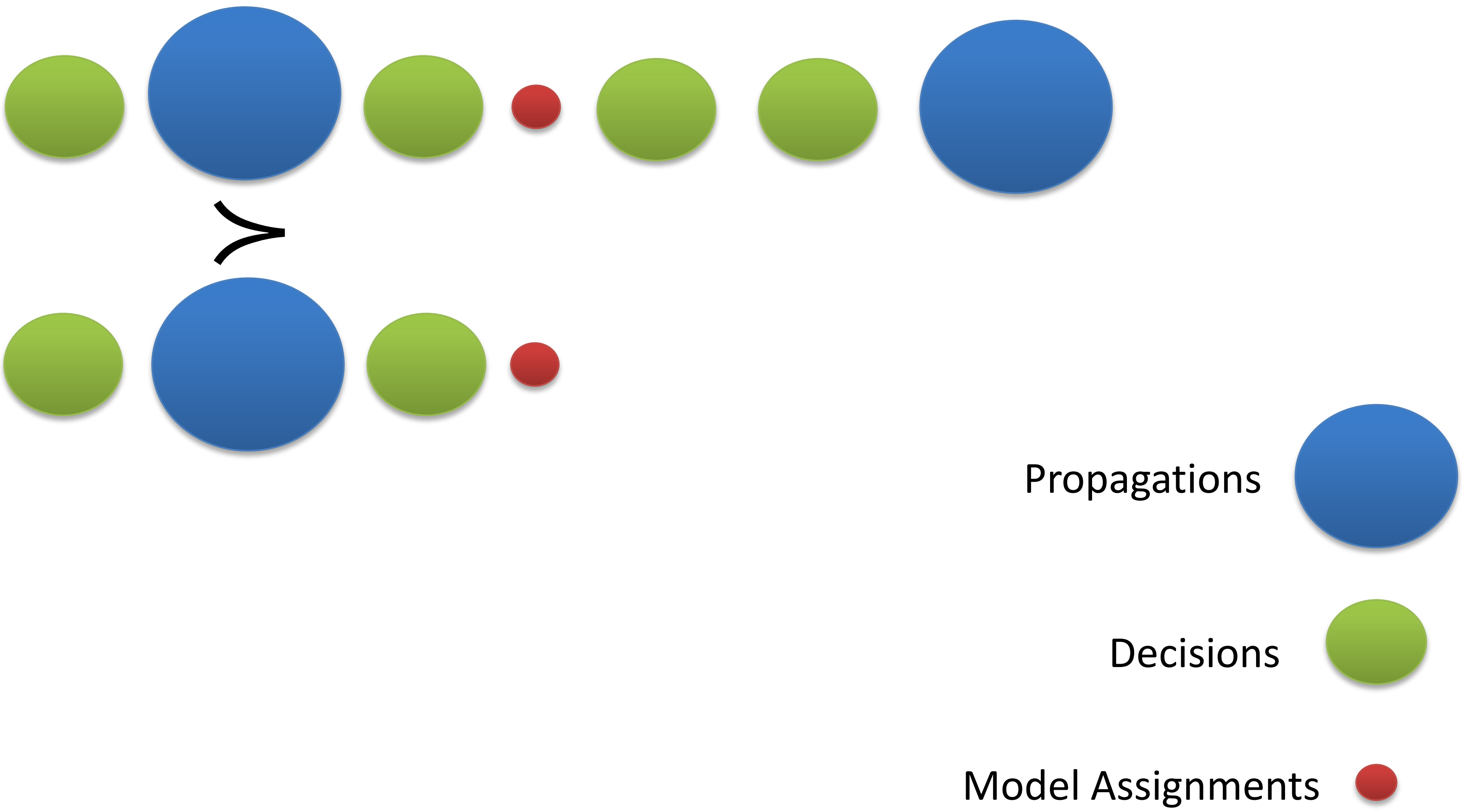


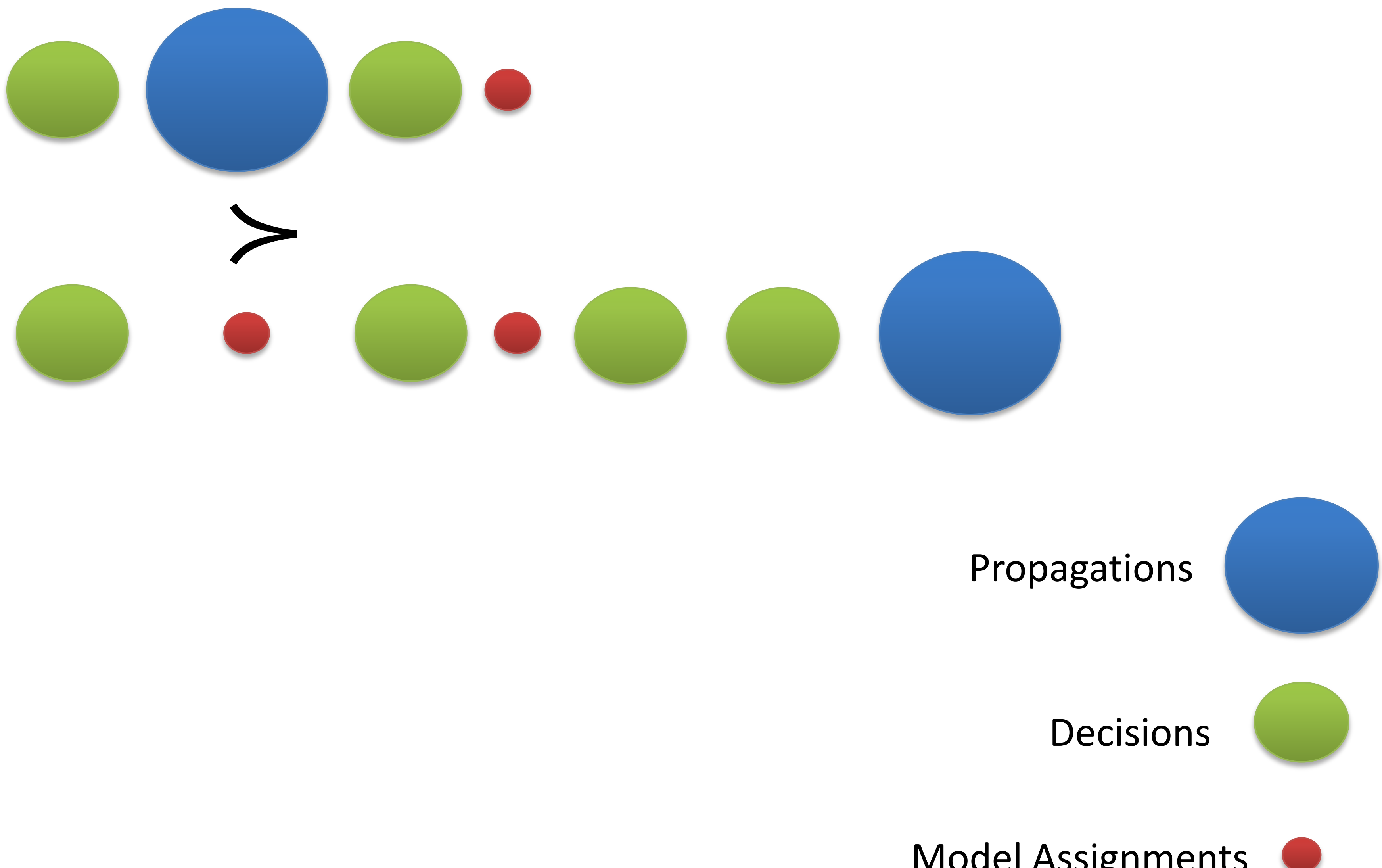
# MCSat: Termination



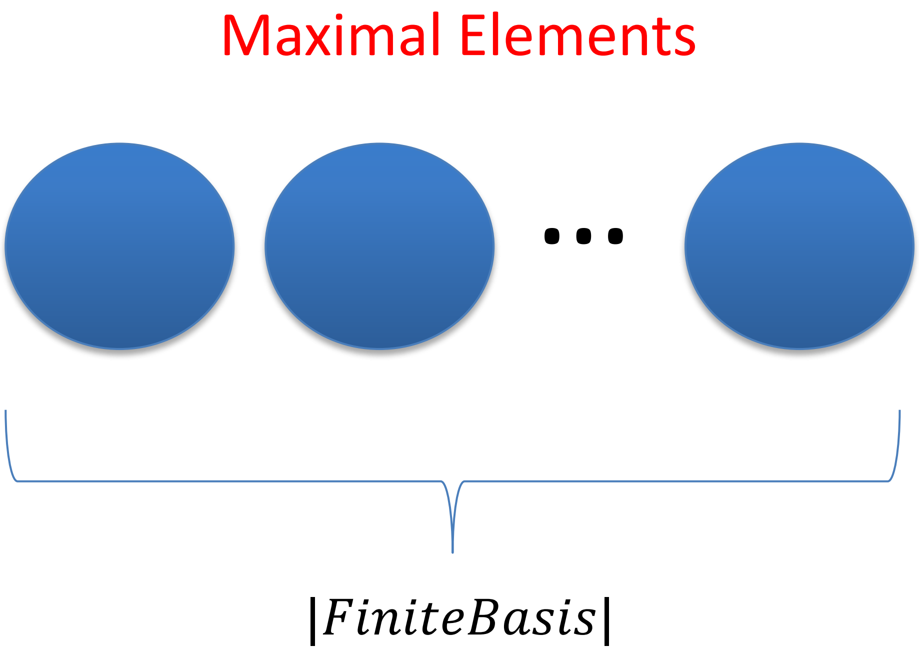
# MCSat



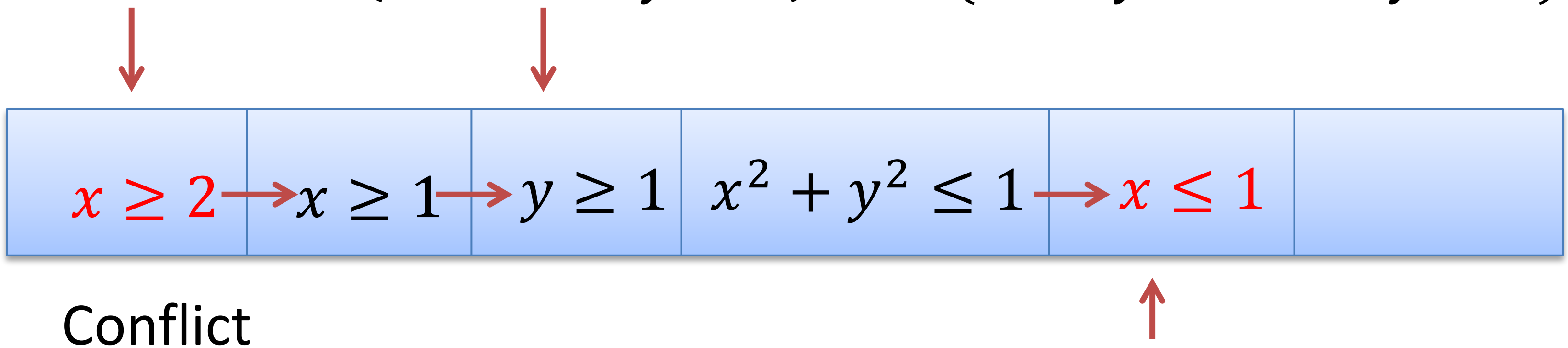
# MCSat



# MCSat



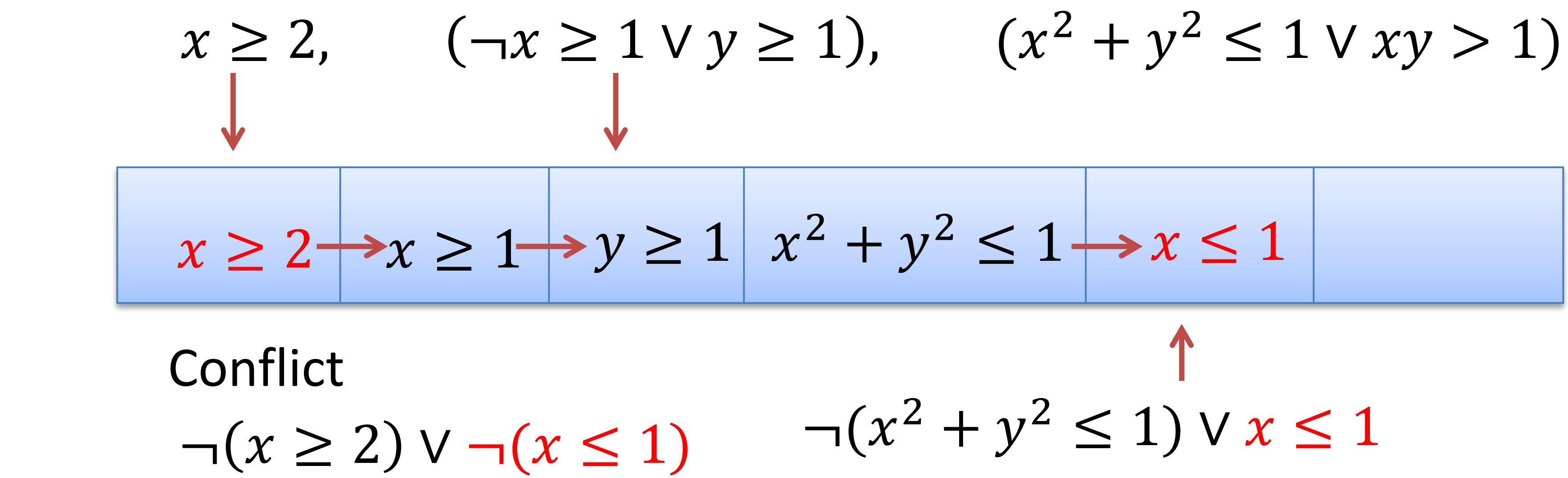
$x \geq 2$, $(\neg x \geq 1 \vee y \geq 1)$, $(x^2 + y^2 \leq 1 \vee xy > 1)$

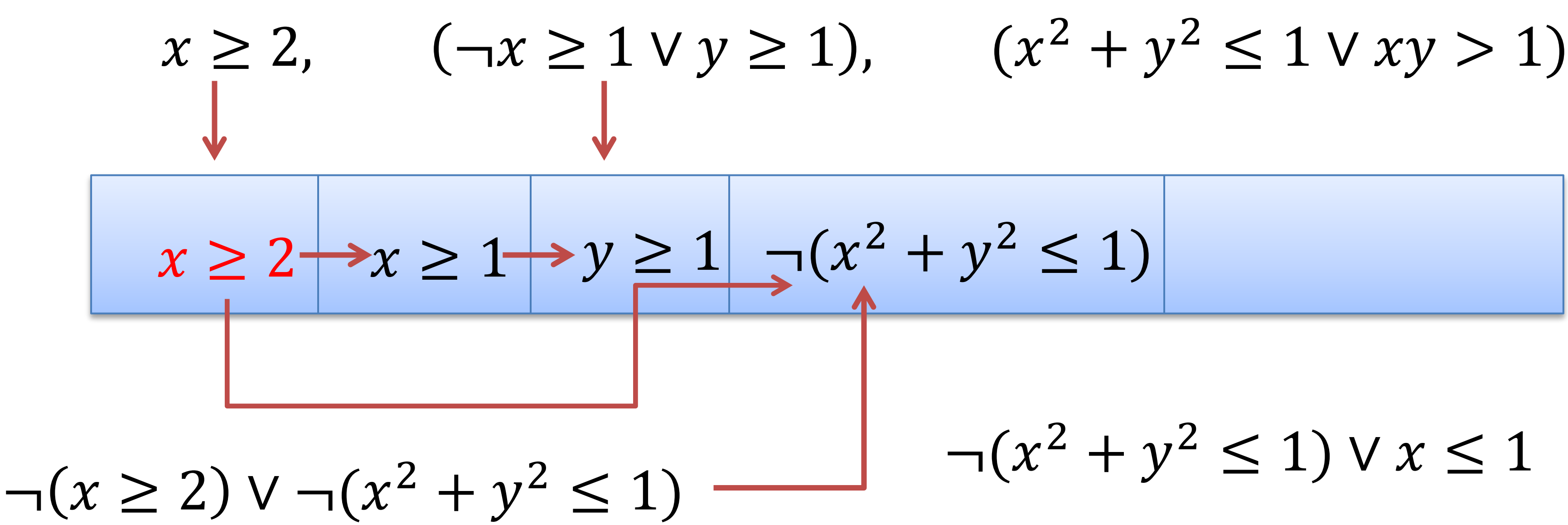


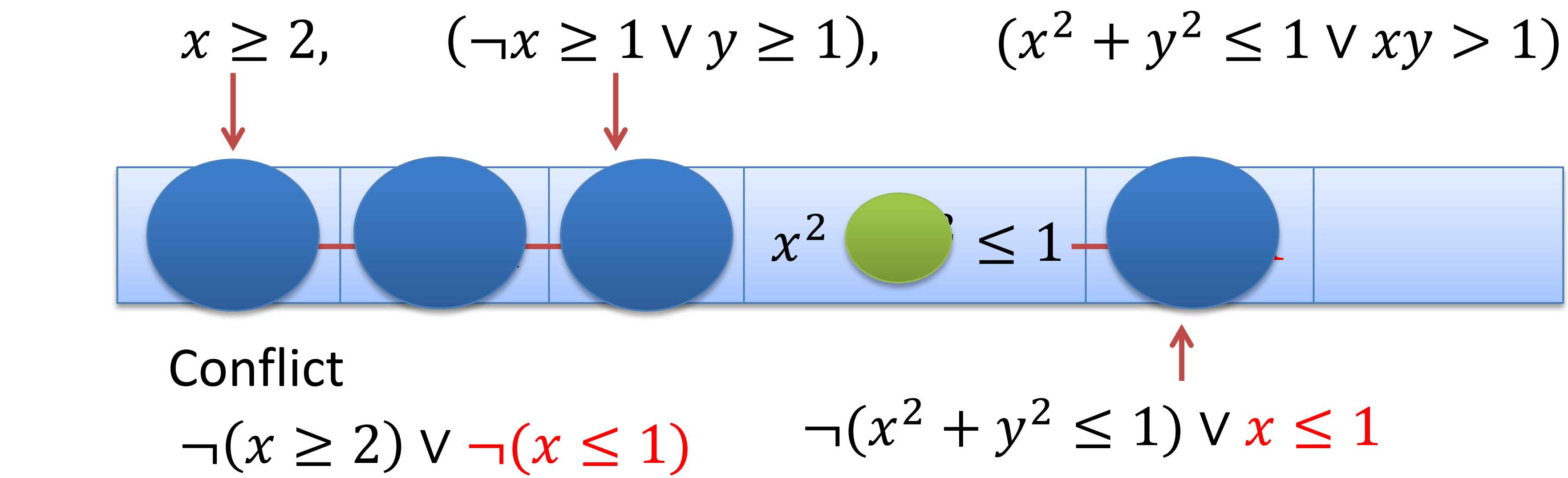

Conflict

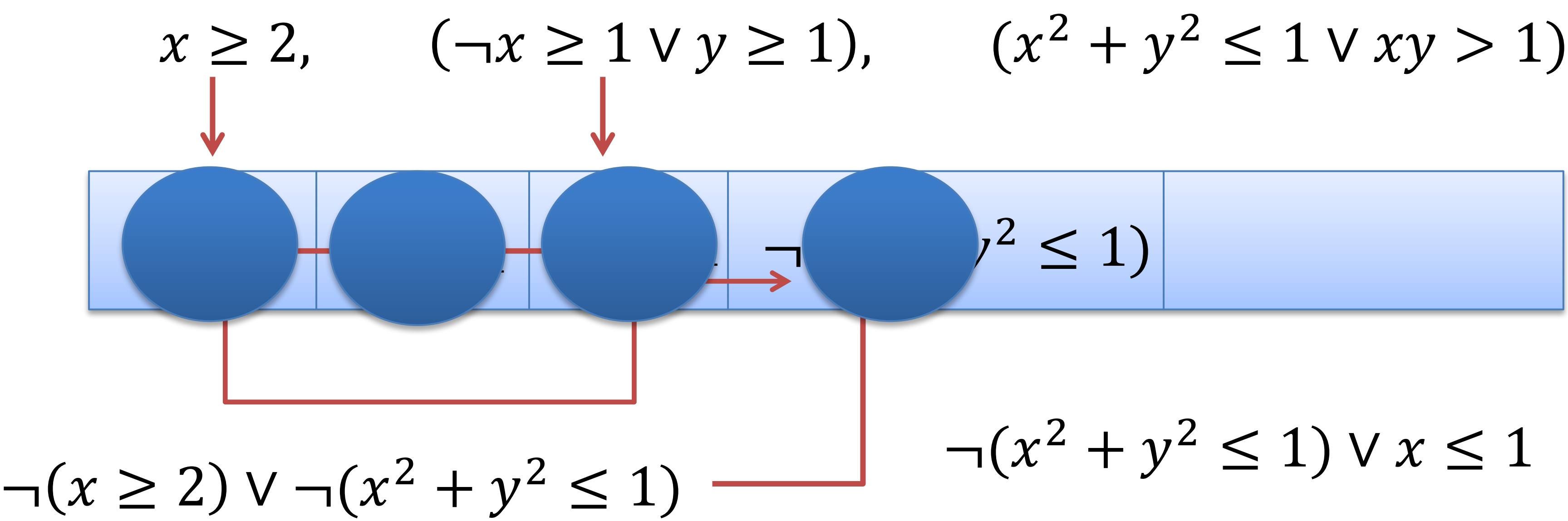
$\neg(x \geq 2) \vee \neg(x \leq 1)$

$\neg(x^2 + y^2 \leq 1) \vee x \leq 1$


x ≥ 2,
(¬x ≥ 1 ∨ y ≥ 1),
(x² + y² ≤ 1 ∨ xy > 1)
x ≥ 2
x ≥ 1
y ≥ 1
x² + y² ≤ 1
x ≤ 1
Conflict
¬(x ≥ 2) ∨ ¬(x ≤ 1)
¬(x² + y² ≤ 1) ∨ x ≤ 1


x ≥ 2,
(¬x ≥ 1 ∨ y ≥ 1),
(x² + y² ≤ 1 ∨ xy > 1)
x ≥ 2
x ≥ 1
y ≥ 1
¬(x² + y² ≤ 1)
¬(x ≥ 2) ∨ ¬(x² + y² ≤ 1)
¬(x² + y² ≤ 1) ∨ x ≤ 1


$x \geq 2,$
$(\neg x \geq 1 \vee y \geq 1),$
$(x^2 + y^2 \leq 1 \vee xy > 1)$
$\leq 1$
Conflict
$\neg(x \geq 2) \vee \neg(x \leq 1)$
$\neg(x^2 + y^2 \leq 1) \vee x \leq 1$


$x \geq 2,$
$(\neg x \geq 1 \vee y \geq 1),$
$(x^2 + y^2 \leq 1 \vee xy > 1)$
$\leq 1)$
$\neg(x^2 + y^2 \leq 1) \vee x \leq 1$
$\neg(x \geq 2) \vee \neg(x^2 + y^2 \leq 1)$

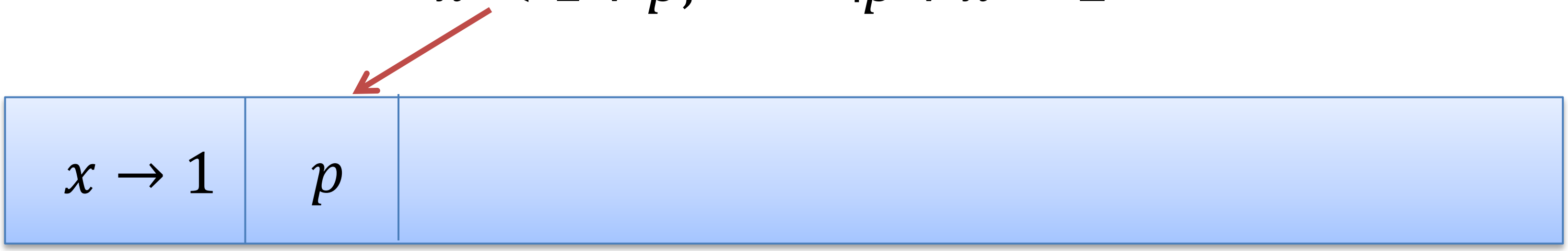
# MCSat

$$x < 1 \vee p, \qquad \neg p \vee x = 2$$

| | |
|---|---|
| $x \to 1$ | |

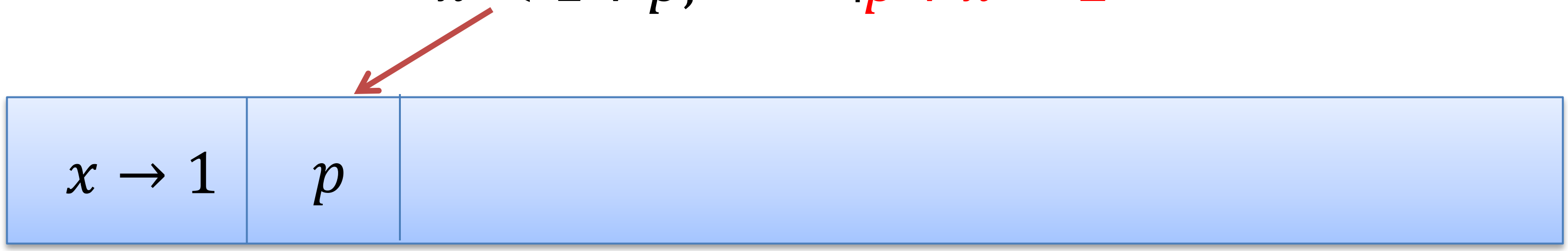
# MCSat

$x < 1 \vee p, \qquad \neg p \vee x = 2$



# MCSat

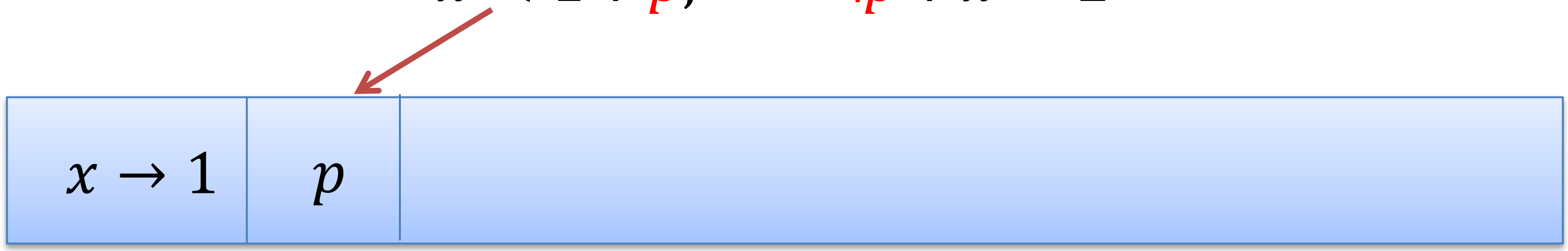
$x < 1 \vee p, \qquad \neg p \vee x = 2$




Conflict (evaluates to false)

# MCSat

$x < 1 \vee p, \qquad \neg p \vee x = 2$




New clause

$x < 1 \vee x = 2$

# MCSat

$x < 1 \vee p$, $\neg p \vee x = 2$

| | |
|---|---|
| $x \to 1$ | $p$ |

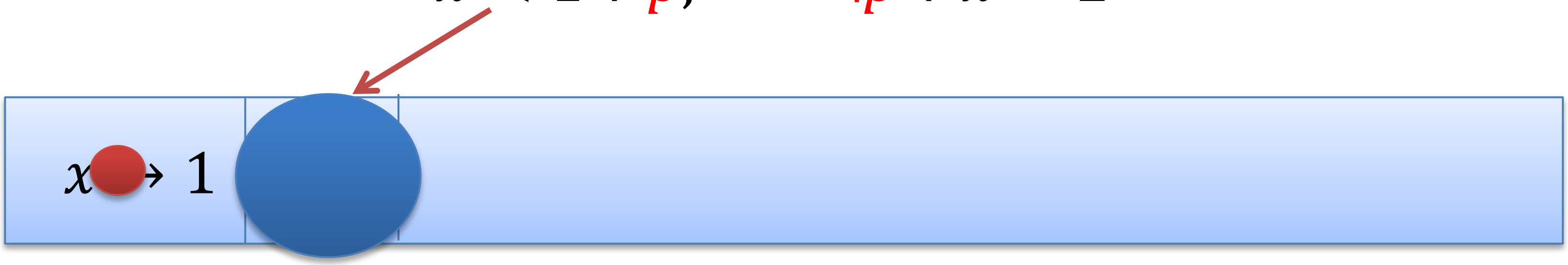
New clause

$x < 1 \vee x = 2$

| |
|---|
| $x < 1$ |

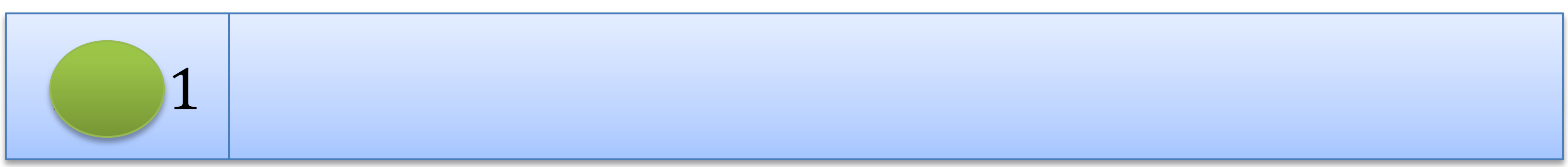
# MCSat

$x < 1 \vee p, \qquad \neg p \vee x = 2$

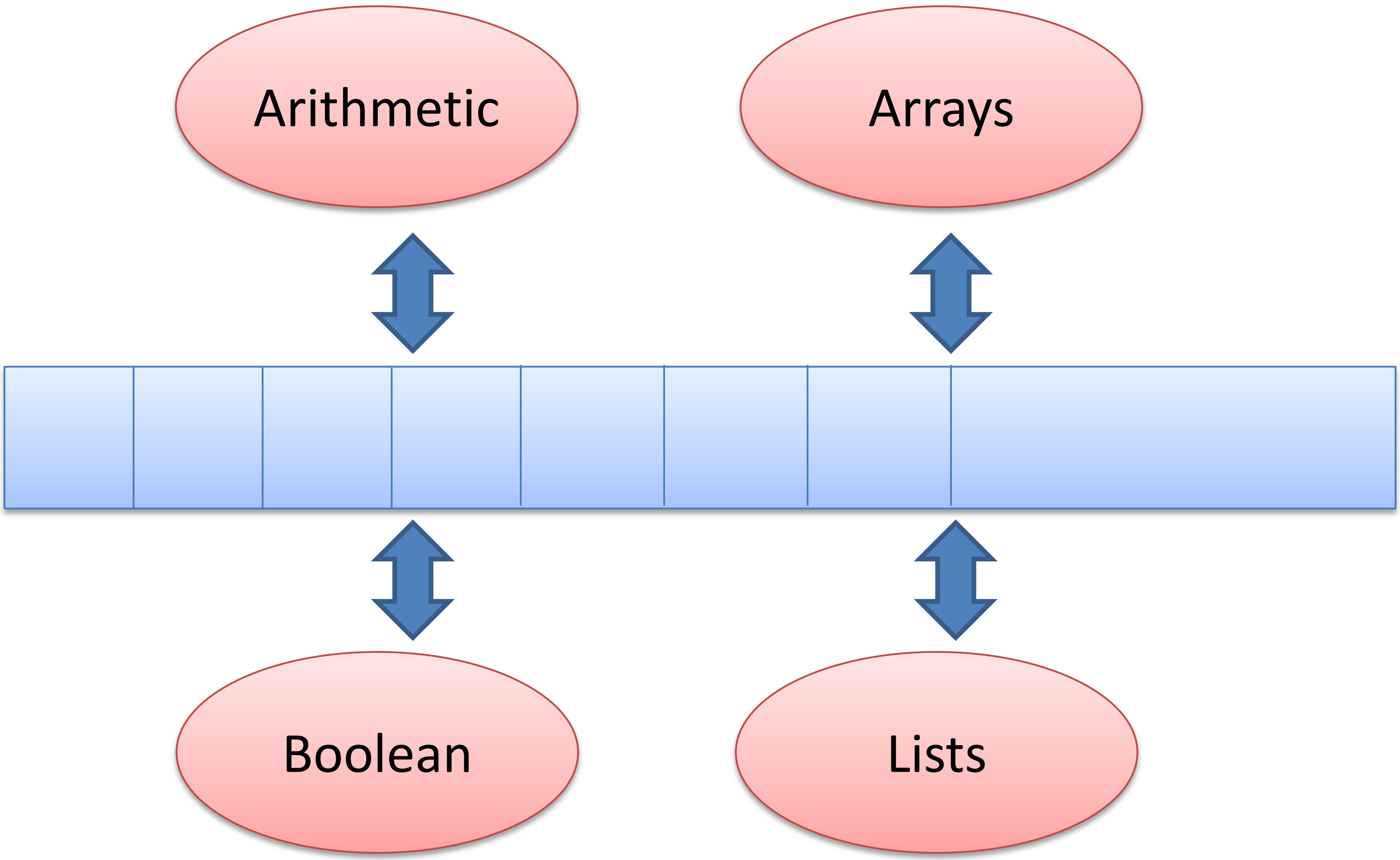



New clause

$x < 1 \vee x = 2$



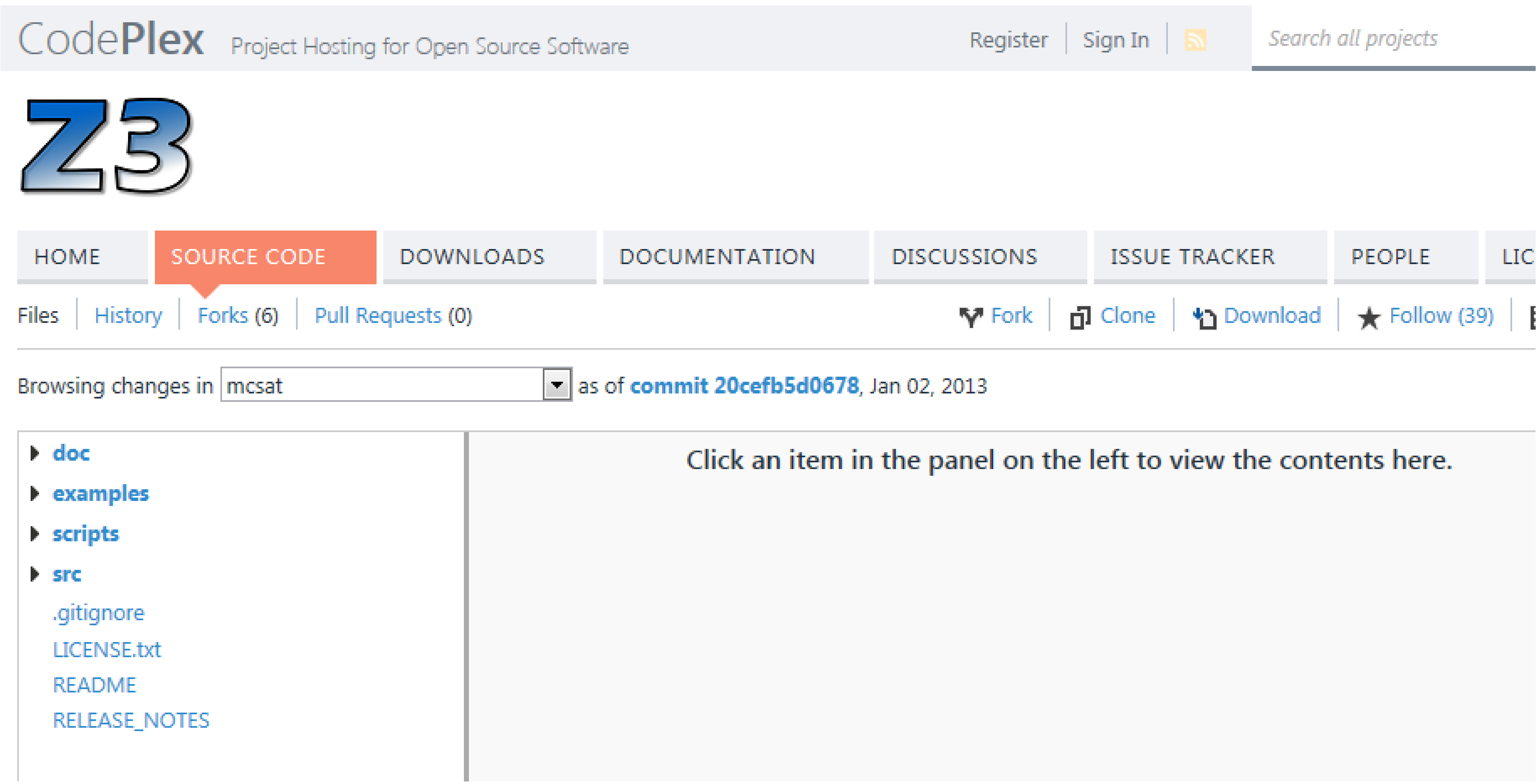
# MCSat: Architecture



# MCSat: development

# News: Z3 source code is available

http://z3.codeplex.com



# Conclusion

Logic as a Service

Model-Based techniques are very promising

MCSat

http://z3.codeplex.com

http://rise4fun.com/z3py