

Distributed SMT Solving Based on Dynamic Variable-level Partitioning

Mengyu Zhao, Shaowei Cai*, and Yuhang Qian

Key Laboratory of System Software (Chinese Academy of Sciences)
and State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences, Beijing, China

`{zhaomy, caisw, qianyh}@ios.ac.cn`

Outline

- **Introduction**
- Dynamic Variable-level Partitioning
- Experiments & Summary

Propositional Satisfiability (SAT)

Propositional Satisfiability (SAT): Given a propositional formula φ , test whether there is an assignment to the variables that makes φ true.

e.g., a CNF formula:

$$\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (x_2 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_3 \vee x_4)$$

- The first NP-Complete problem [Cook, STOC'71]
- A core problem in computer science and a basic problem in logic

Solve a Math Problem with Arithmetic Constraints

$$x > 1$$

$$x < 4$$

$$xy > 4$$

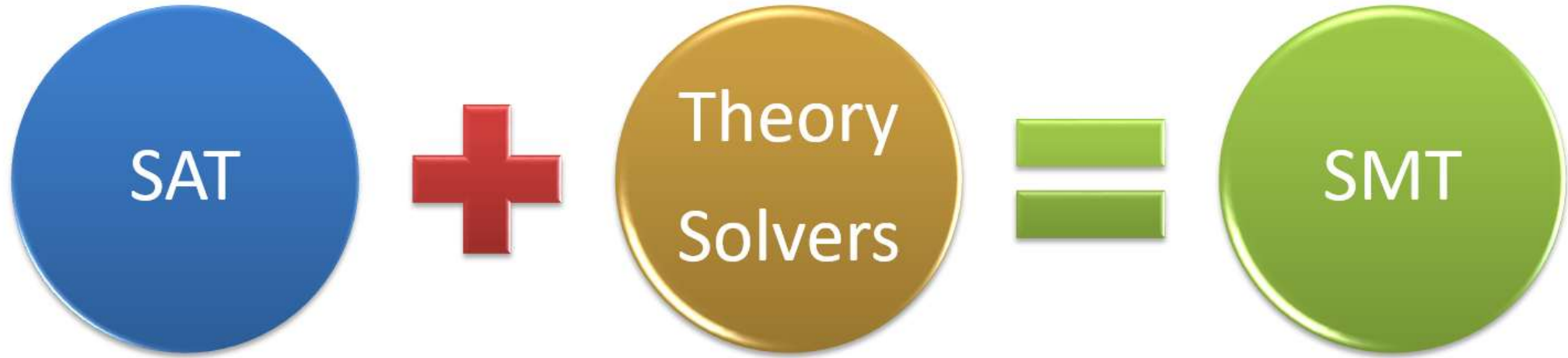
$$yz^2 \leq 4$$

$$2xz + y^2 < -20$$



- Linear system
 - Simplex
 - Branch and Bound
- Non-linear system:
 - Cylindrical Algebraic Decomposition
 - **Interval Constraint Propagation**

Satisfiability Module Theories (SMT)



Propositional Satisfiability

$(\neg a \vee b)$
 $\wedge (c \vee d)$
 $\wedge (a \vee e \vee f)$

- Equality + UF
- Arithmetic
- Bit-vectors
- ...

$(\neg a \vee x < -2)$
 $\wedge (y > 0 \vee x^2 z + y = 3)$
 $\wedge (a \vee x^2 \geq 4 \vee y > 5)$

Boolean Skeleton of SMT Formulas

SMT:

$$(\neg a \vee x < -2) \wedge (y > 0 \vee x^2 z + y = 3) \wedge (a \vee x^2 \geq 4 \vee y > 5)$$

Boolean Skeleton:

$$(\neg a \vee b) \wedge (c \vee d) \wedge (a \vee e \vee f)$$

Theory Level:

$$b: x < -2 \qquad c: y > 0$$

$$d: x^2 z + y = 3$$

$$e: x^2 \geq 4 \qquad f: y > 5$$

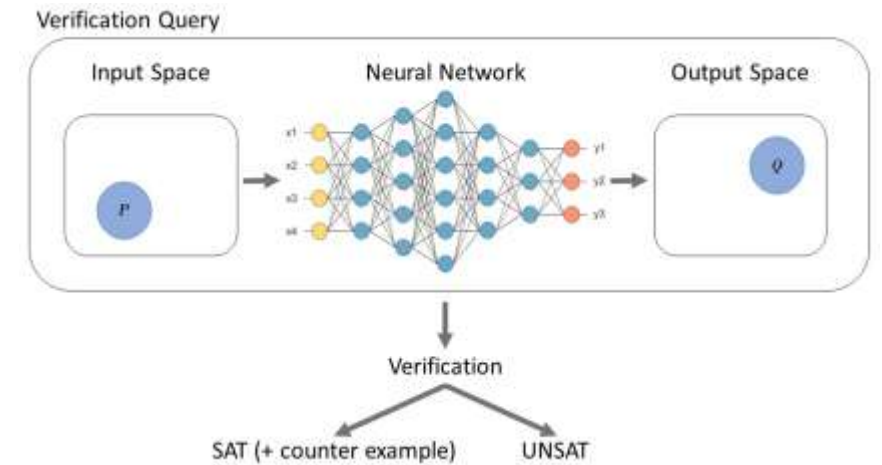
Satisfiability Module Theories (SMT)



Program Verification



Security Applications



Neural Network
Verification

SMT Solvers and Solving Paradigms

SMT Solvers:

Z3 Z3++

CVC5

OpenSMT2

Yices2

...

Solving Paradigms:

- **CDCL(T)**
- **MCSAT**
- **Bit-blasting**
- **Local Search**

...

Distributed SMT Solving

- Portfolio
 - Diversification
 - Clause sharing
- **Partitioning**
 - Cube and conquer
 - Scattering

[Wintersteiger, CAV'09]

[Heule, HVC'12]

OpenSMT2 Team:

[Hyvärinen, SAT'16, FMCAD'21]

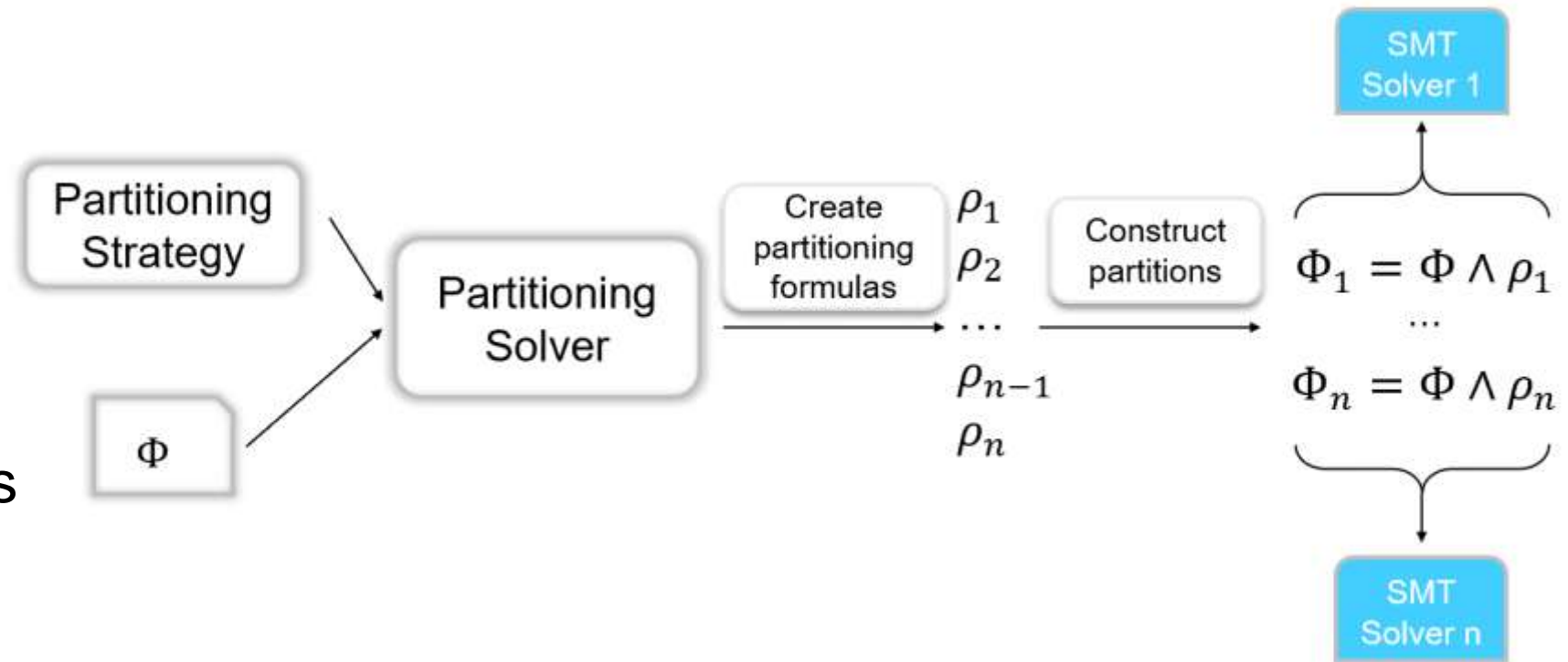
[Marescotti, LPAR'22]

CVC5 Team:

[Wilson, FMCAD'23]

Distributed SMT Solving

In partitioning strategies provided by CVC5 [Wilson, FMCAD'23], the original problem is divided into sub-problems before solving.



OpenSMT2 implements a dynamic partitioning method. [Marescotti, LPAR'22]

- partitions the instance dynamically on-demand
- shares learnt clauses

Outline

- Introduction
- **Dynamic Variable-level Partitioning**
- Experiments & Summary

Motivation of Dynamic Variable-level Partitioning

- Term-level partitioning doesn't work all the time. Pure-conjunction formulas can not be partitioned at term-level.

$$(x^2 + y^2 \leq 5) \wedge (2xy > 3) \wedge (x > -2) \wedge (y < 9)$$

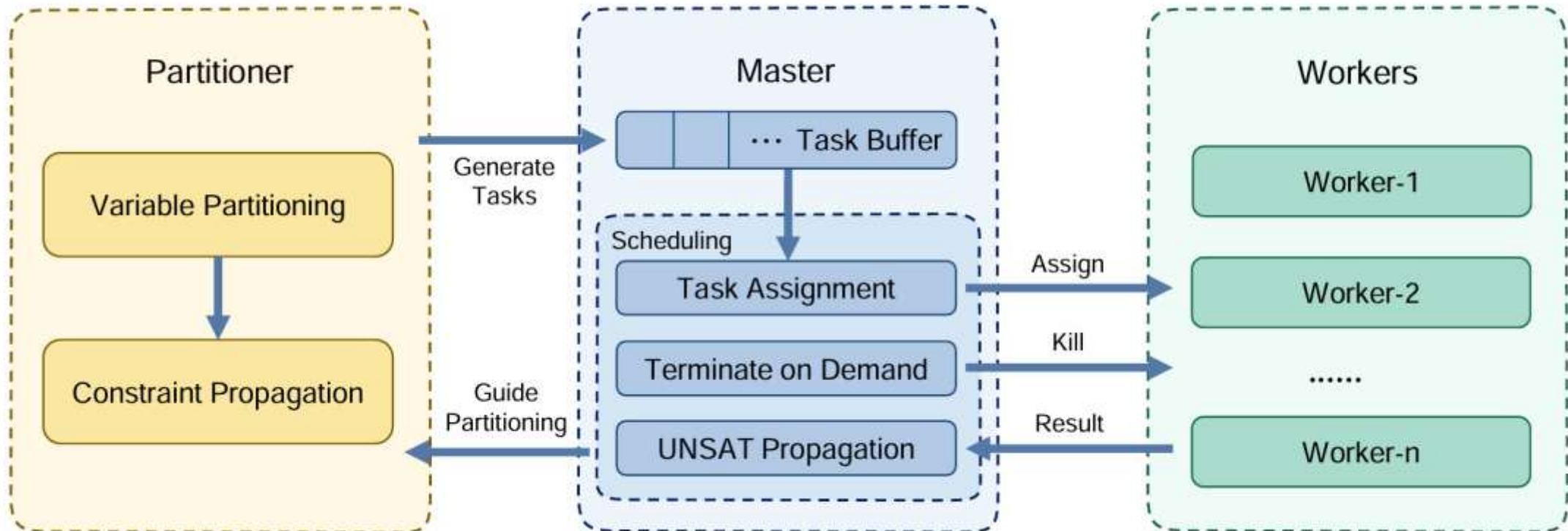
- Many deep simplification technologies has not been well integrated with SMT to accelerate solving.
- Need a more flexible dynamic partitioning strategy.

Dynamic Distributed Framework

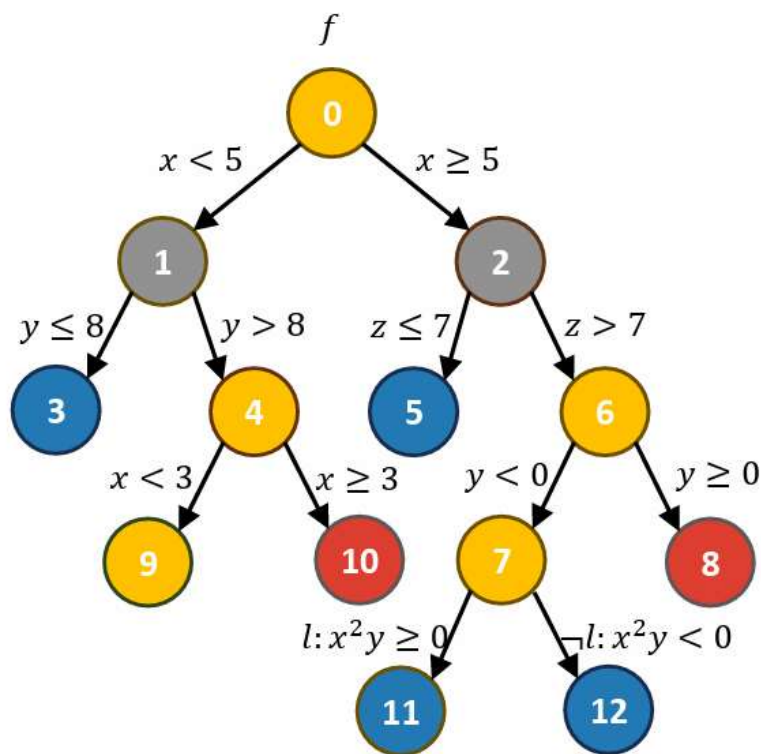
Partitioner: variable-level partitioning, sub-problems generation, and constraint propagation

Master: task assignments, on-demand terminations, and UNSAT propagations

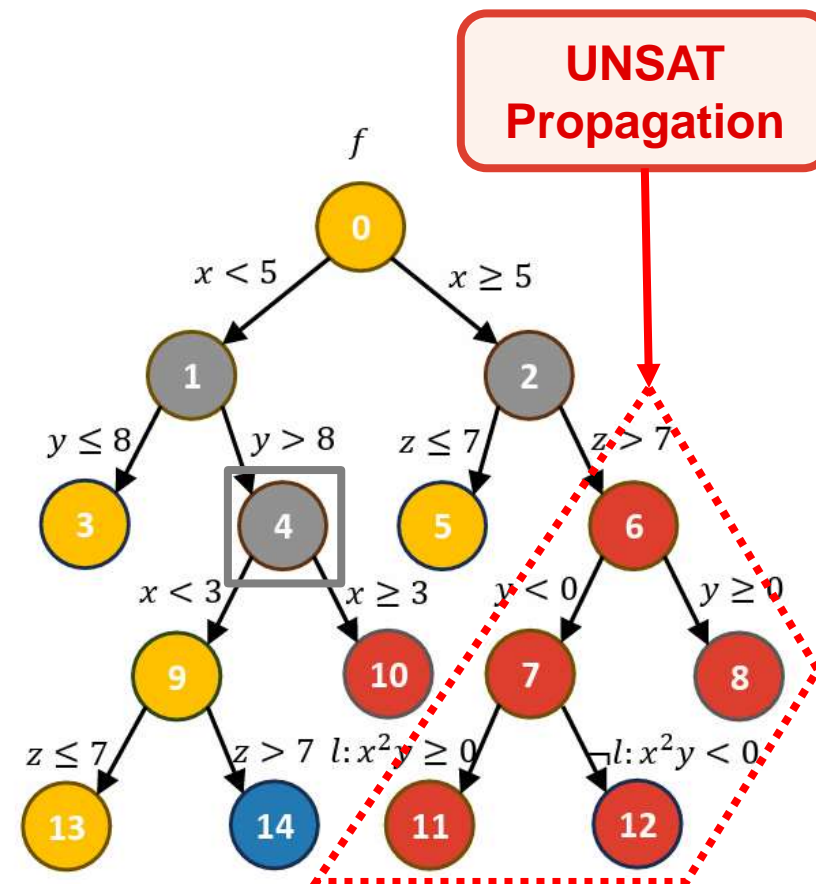
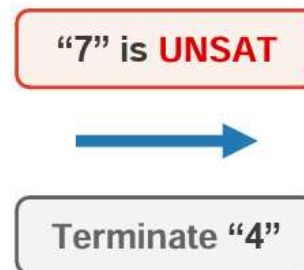
Workers: task solving and result notification



Dynamic Distributed Framework

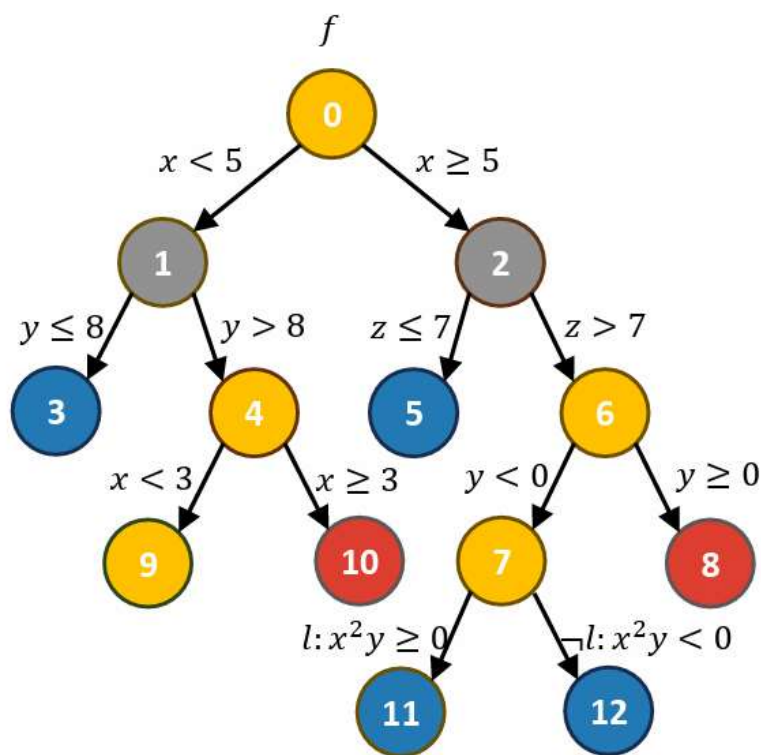


(a)

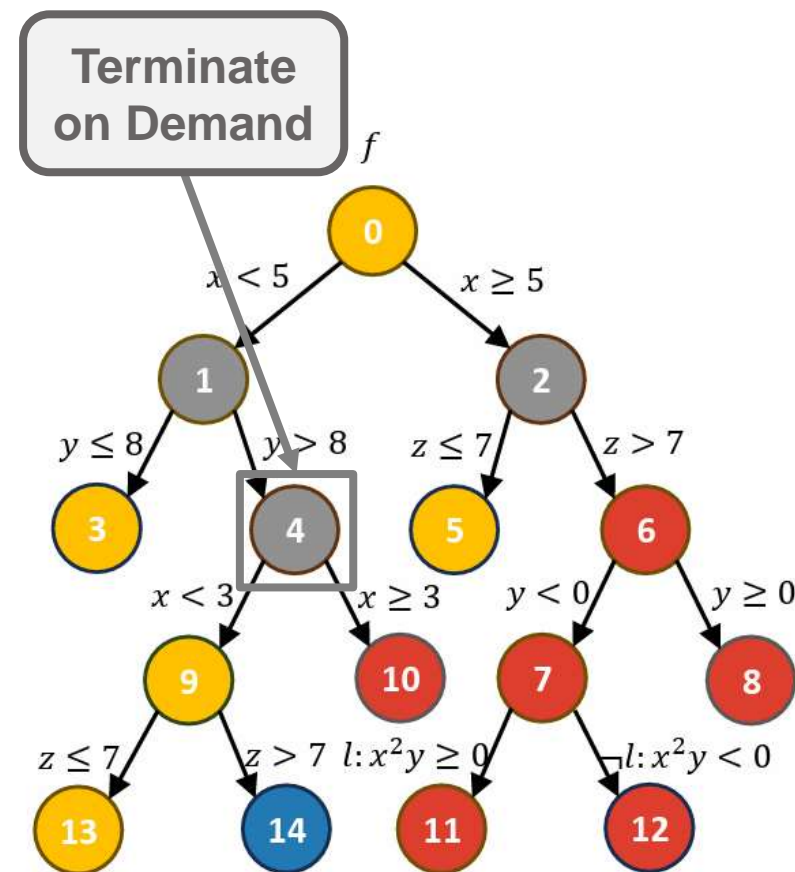
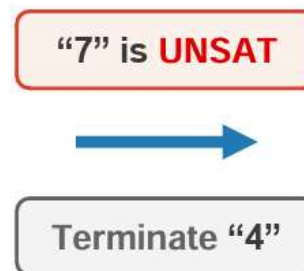


(b)

Dynamic Distributed Framework

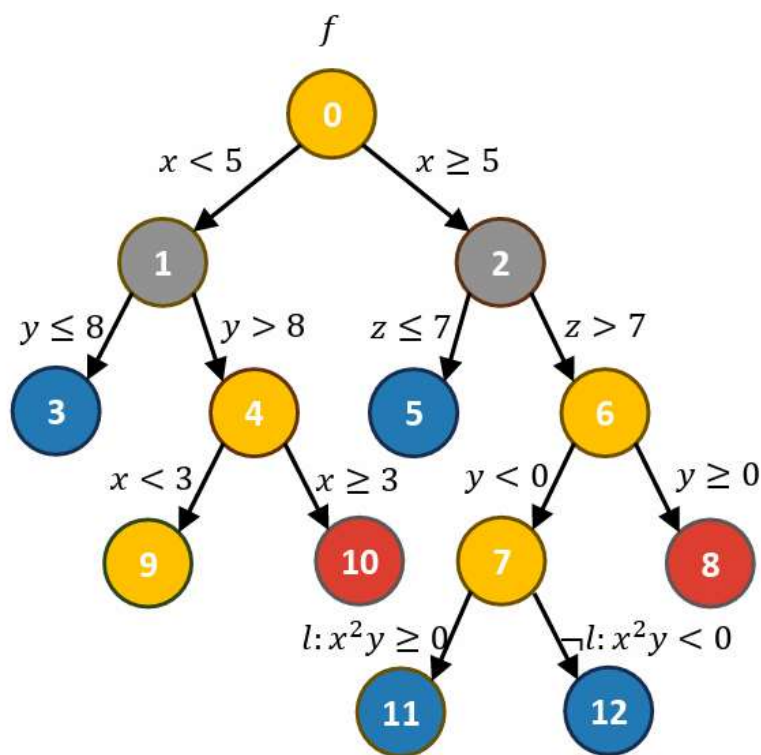


(a)



(b)

Dynamic Distributed Framework

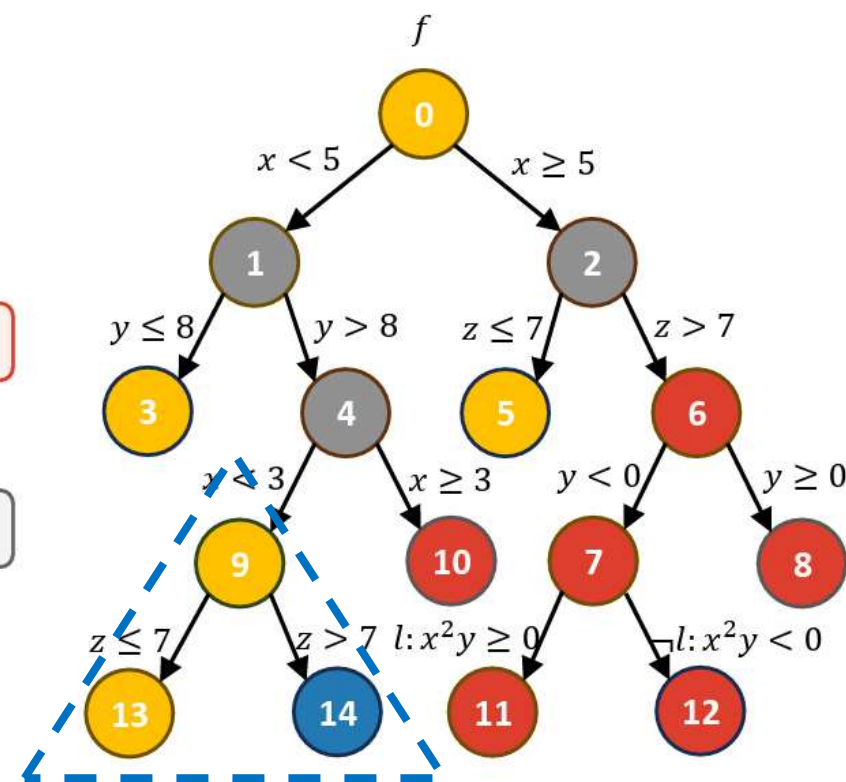


(a)

"7" is **UNSAT**



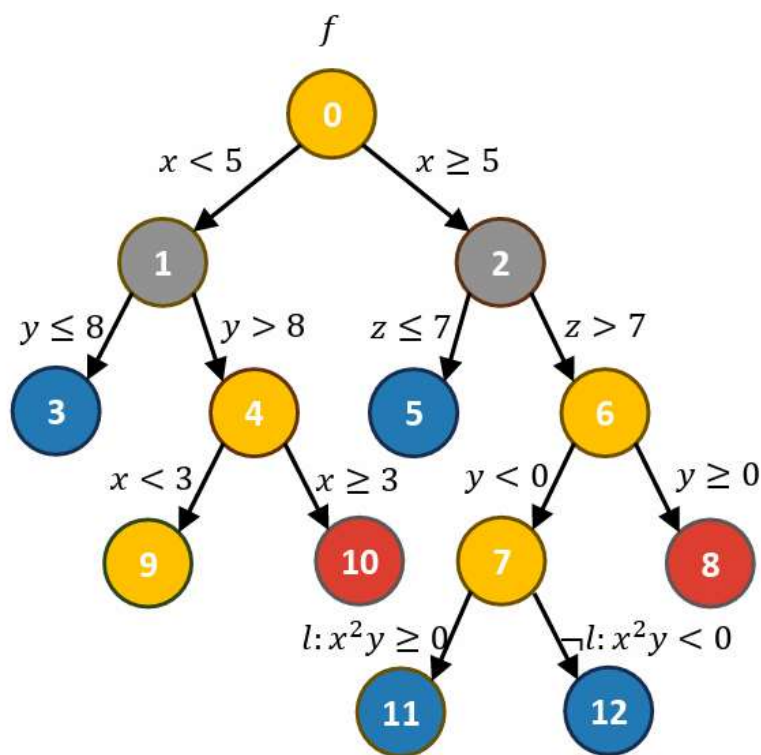
Terminate "4"



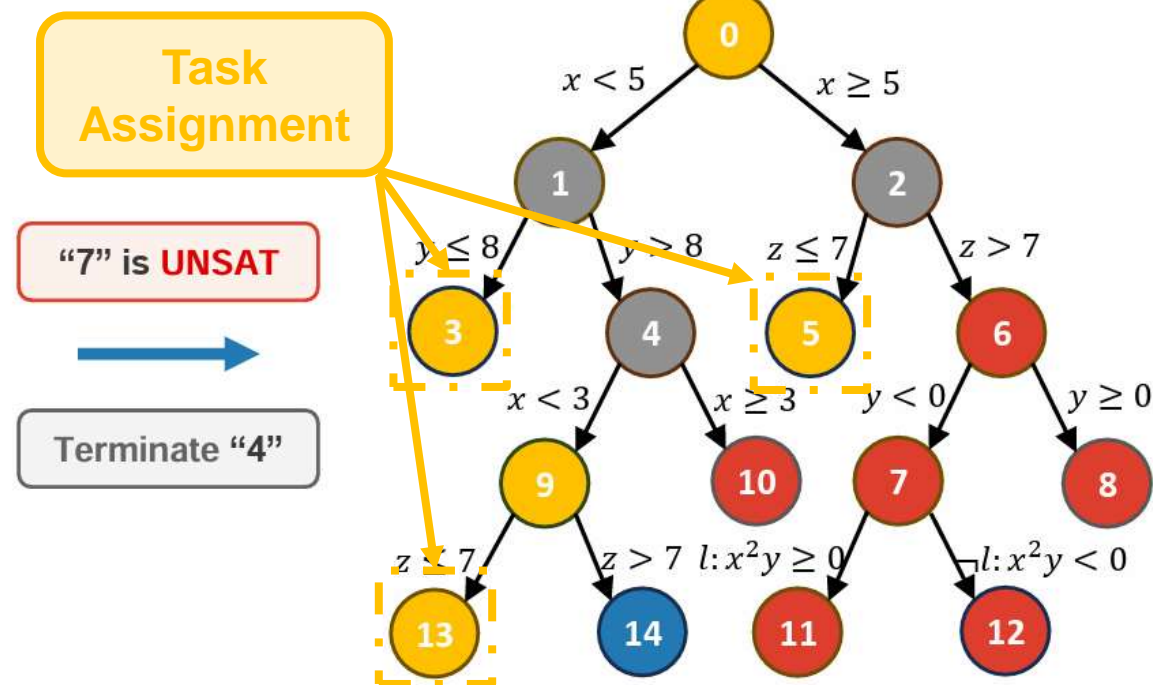
(b)

Variable-level Partitioning
and Enhanced Simplification

Dynamic Distributed Framework

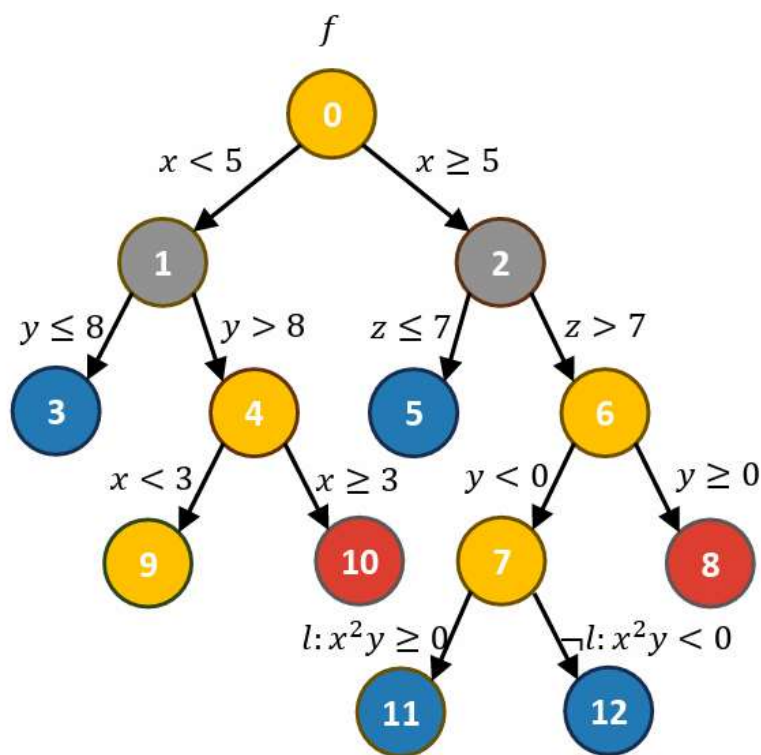


(a)

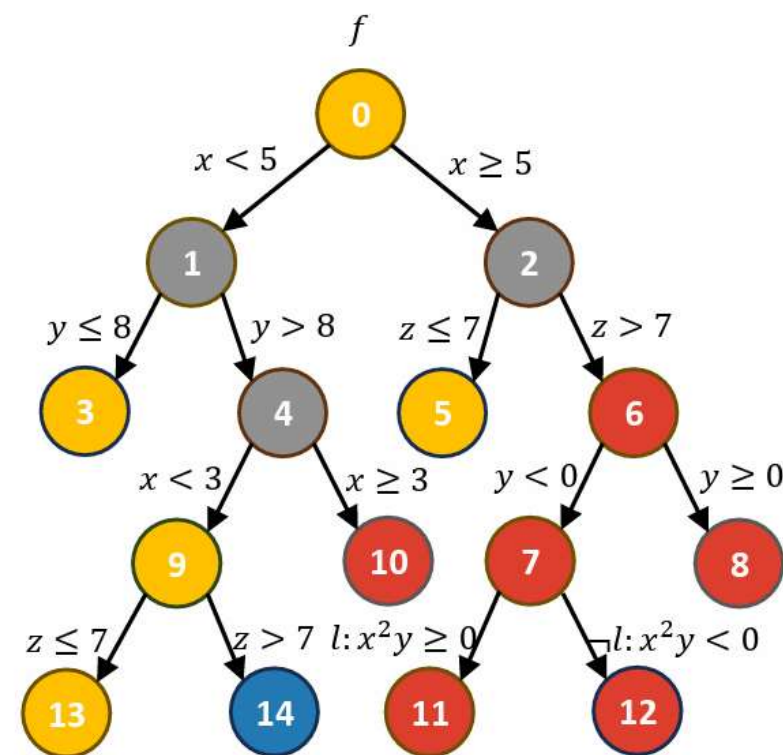
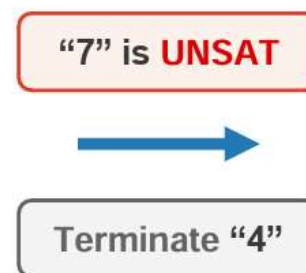


(b)

Dynamic Distributed Framework



(a)

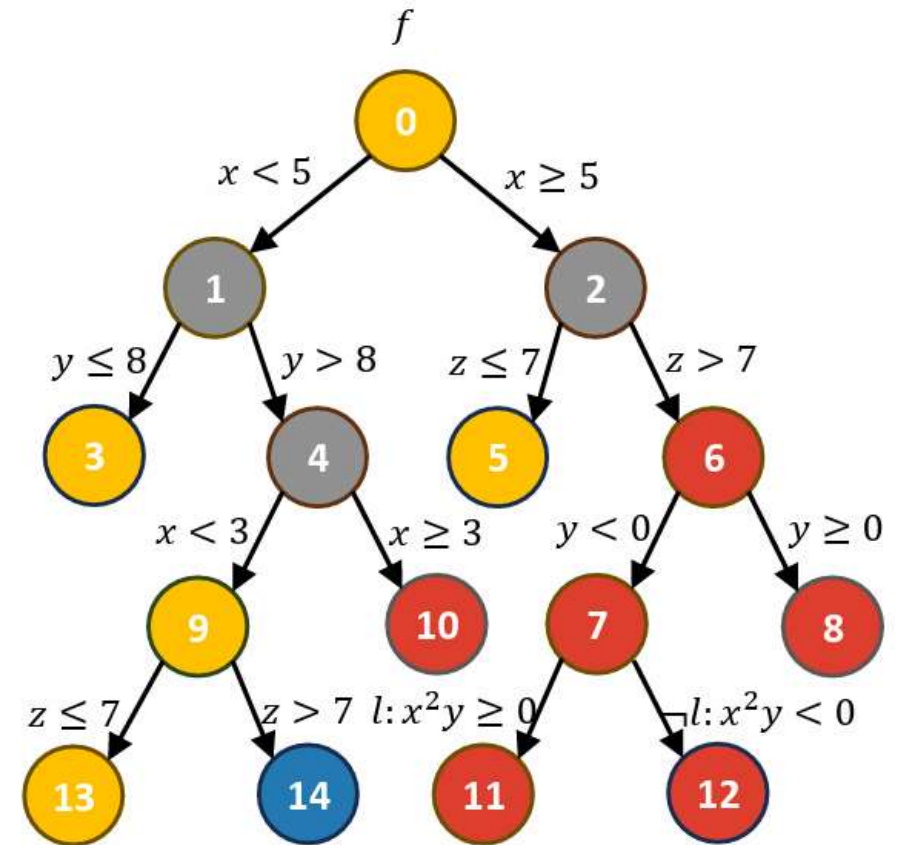


(b)

Dynamic Variable-level Partitioning

Key Ideas

- Variable-level partitioning
- Enhanced propagation and simplification
- Dynamic distributed framework

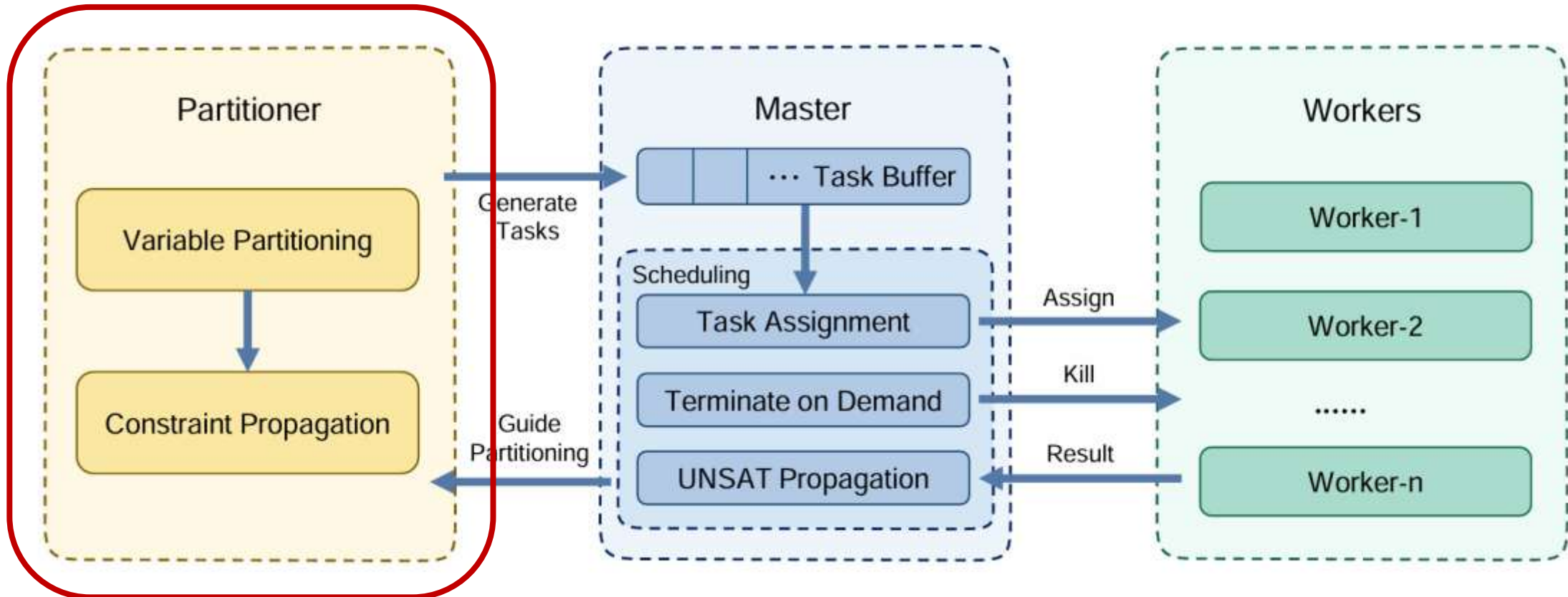


Dynamic Distributed Framework

Partitioner: variable-level partitioning, sub-problems generation, and constraint propagation

Master: task assignments, on-demand terminations, and UNSAT propagations

Worker: task solving and result notification



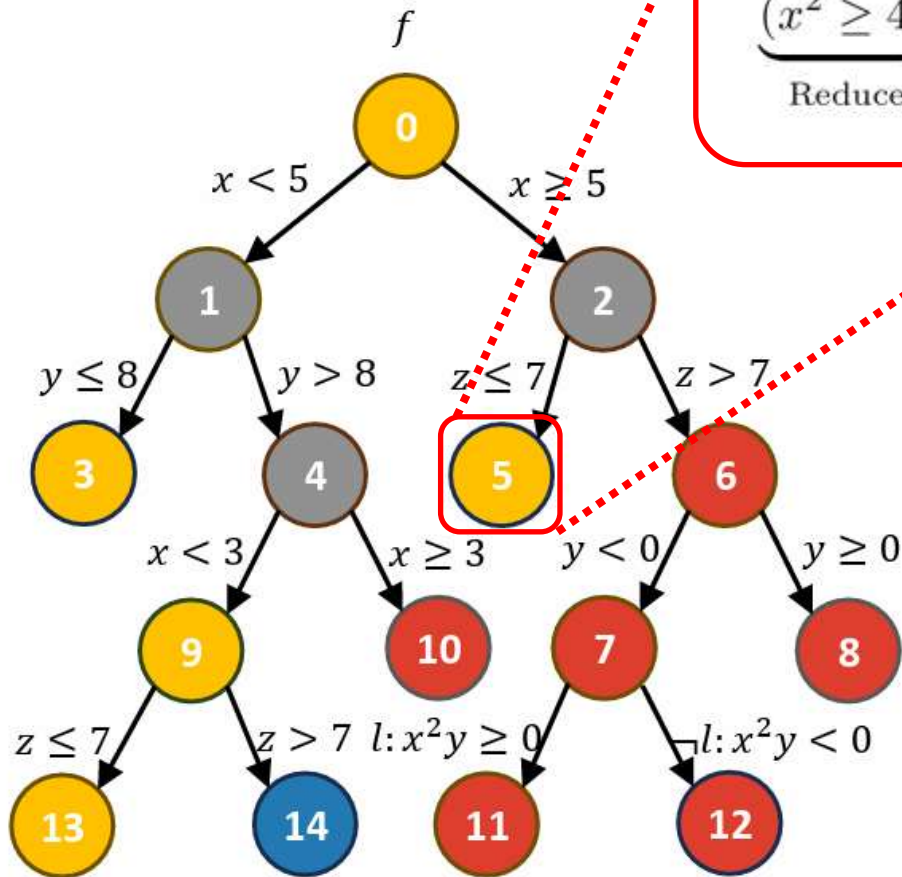
Based on the “**subpaving**” module of Z3

How does arithmetic variable-level partitioning work

Heuristic in Partition Node Selection

Selection of partition node 5, (**Minimum tree depth, most complex formula**)

$$\underbrace{(x^2 \geq 4 \vee y > 5)}_{\text{Reduced Clauses}} \wedge \underbrace{(\neg a \wedge x \in (1, 4) \wedge y \in (1, \infty) \wedge z \in (-2, 2))}_{\text{Feasible Domain of Variables}} \wedge \underbrace{(xy > 4 \wedge yz^2 \leq 4)}_{\text{Propagated Literals}}.$$



How does arithmetic variable-level partitioning work

Heuristic in Partition Variable Selection

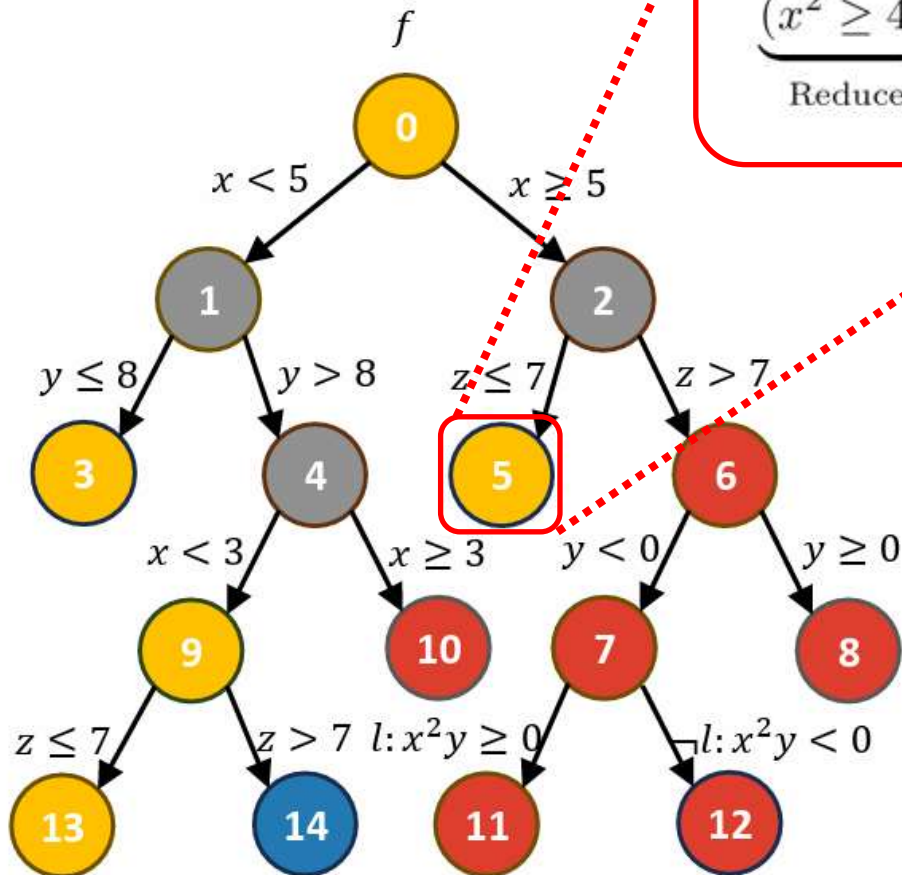
Selection of partition node 5, (Minimum tree depth, most complex formula)

$$\underbrace{(x^2 \geq 4 \vee y > 5)}_{\text{Reduced Clauses}} \wedge \underbrace{(\neg a \wedge x \in (1, 4) \wedge y \in (1, \infty) \wedge \boxed{z \in (-2, 2)})}_{\text{Feasible Domain of Variables}} \wedge \underbrace{(xy > 4 \wedge yz^2 \leq 4)}_{\text{Propagated Literals}}.$$

Selection of partition variable z :

$$z \in (-2, 2)$$

- Highest degree
- Highest occurrence
- Least partitioning times



How does arithmetic variable-level partitioning work

Heuristic in Partitioning at the Selected Variable

Selection of partition node 5, (Minimum tree depth, most complex formula)

$$\underbrace{(x^2 \geq 4 \vee y > 5)}_{\text{Reduced Clauses}} \wedge \underbrace{(\neg a \wedge x \in (1, 4) \wedge y \in (1, \infty) \wedge \boxed{z \in (-2, 2)})}_{\text{Feasible Domain of Variables}} \wedge \underbrace{(xy > 4 \wedge yz^2 \leq 4)}_{\text{Propagated Literals}}.$$

Selection of partition variable z :

$$z \in \boxed{(-2, 2)}$$

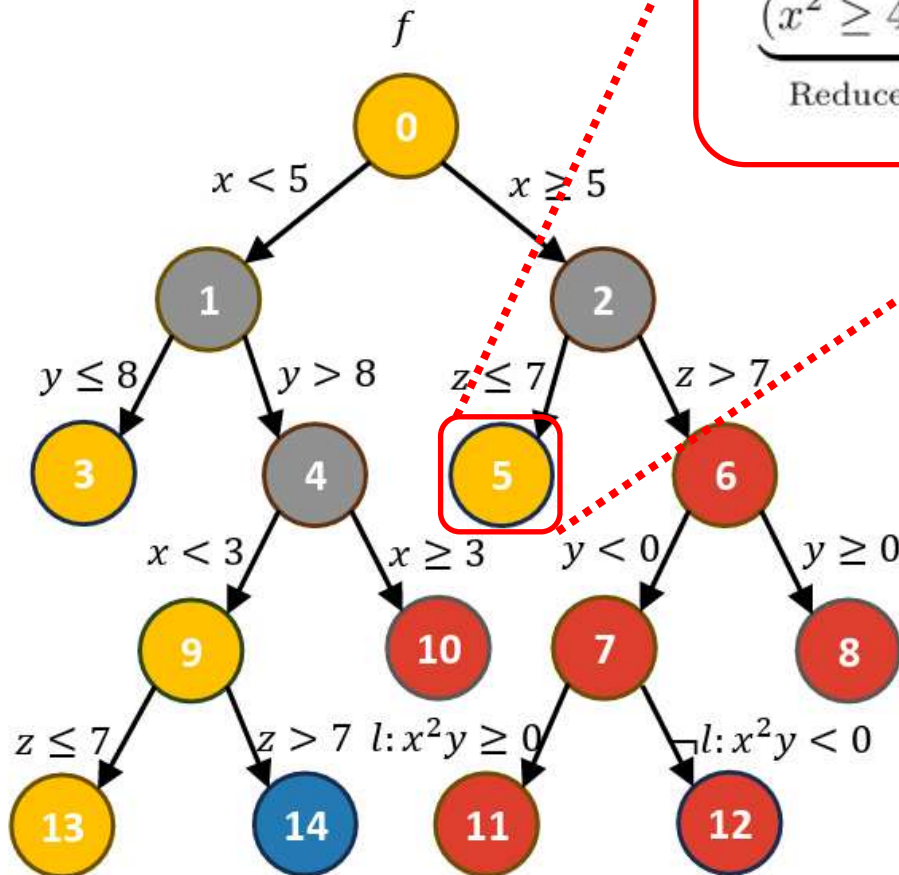
- Highest degree
- Highest occurrence
- Least partitioning times

Partitioning at the value 0:

generate two sub-tasks

$$\phi_5 \wedge (z < 0) \text{ and } \phi_5 \wedge (z \geq 0)$$

- Value 0
- Midpoint
- Heuristic value

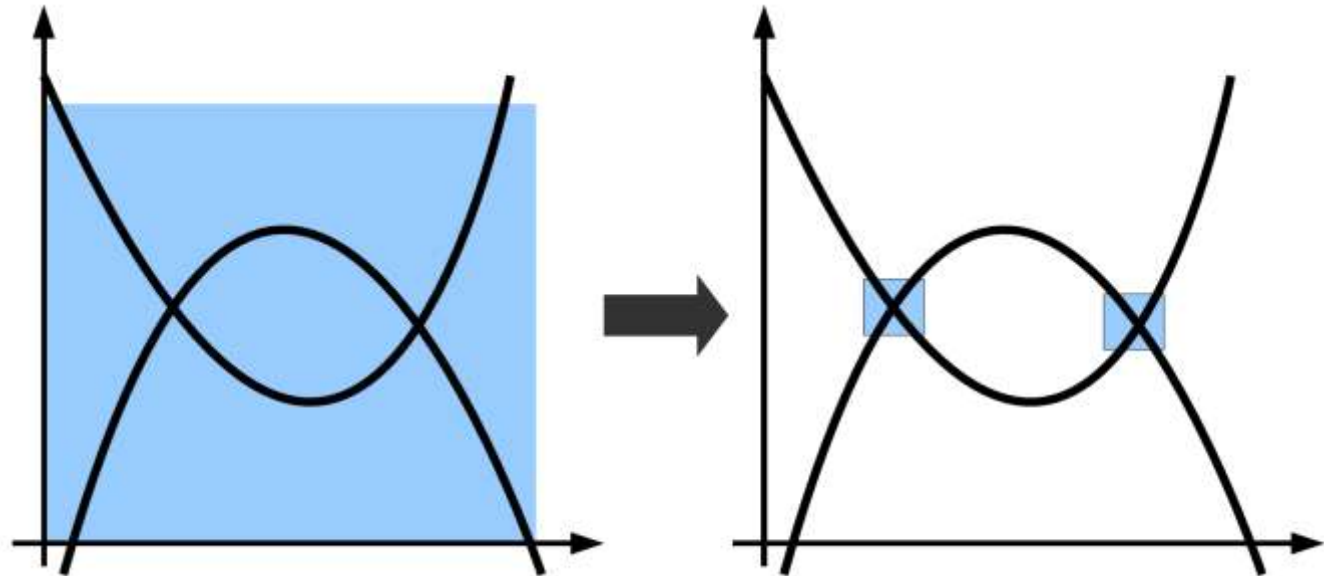


Simplify Formulas via Interval Constraint Propagation (ICP)

It maintains a **feasible interval** for each variable and shrinks these intervals using simple constraint propagation.

ICP has been successfully implemented in various SMT solvers such as dReal, HySAT, and SMT-RAT.

- Shrink **variables' bounds**
- Effectively **exclude** extensive portions of **the search space**
- Sometimes proving **unsatisfiability**



[Gao, FMCAD'10]
[Schupp, Thesis'13]

Simplify Formulas via Interval Constraint Propagation (ICP)

$$x > 1$$

$$x < 4$$

$$xy > 4$$

$$yz^2 \leq 4$$

$$2xz + y^2 < -20$$



\Rightarrow Derive $y \in (1, \infty)$ from $x \in (1, 4) \wedge xy > 4$

\Rightarrow Derive $z \in (-2, 2)$ from $y \in (1, \infty) \wedge yz^2 \leq 4$

Simplify Formulas via Interval Constraint Propagation (ICP)

$$x > 1$$

$$x < 4$$

$$xy > 4$$

$$yz^2 \leq 4$$

$$2xz + y^2 < -20$$



\Rightarrow Derive $y \in (1, \infty)$ from $x \in (1, 4) \wedge xy > 4$

\Rightarrow Derive $z \in (-2, 2)$ from $y \in (1, \infty) \wedge yz^2 \leq 4$

\Rightarrow By interval arithmetic, we can obtain:

$$2xz + y^2 = 2 \times (1, 4) \times (-2, 2) + (1, \infty)^2 \\ \in (-15, \infty)$$

$2xz + y^2 < -20$ cannot be satisfied, when the above 4 constraints are satisfied.

Unsatisfiable!

Enhance ICP with BCP

Interval Constraint Propagation (ICP)

$$x > 1$$

$$x < 4$$

$$\neg a \vee x < -2$$

$$a \vee xy > 4$$

$$a \vee y > 5$$

$$2xz + y^2 < -20$$

We cannot
simplify this
formula by ICP.

Enhance ICP with BCP

Boolean Constraint Propagation (BCP) the unassigned literal in **unit clause** can only be assigned to single value to satisfy the clause.

Interval Constraint Propagation (ICP)

$$x > 1$$

$$x < 4$$

$$\neg a \vee x < -2$$

$$a \vee xy > 4$$

$$a \vee y > 5$$

$$2xz + y^2 < -20$$

We cannot
simplify this
formula by ICP.

Enhanced with BCP

$$x > 1$$

$$x < 4$$

$$\neg a \vee x < -2$$

$$a \vee xy > 4$$

$$a \vee y > 5$$

$$2xz + y^2 < -20$$

Simplify this
formula a lot,
and prove
unsatisfiable
directly by ICP.

Combining ICP with BCP

BICP and Formula Simplification

\Rightarrow Derive $x \in (1, 4)$, $y \in (1, \infty)$, $z \in (-2, 2)$ by ICP

\Rightarrow Infer $(x < -2) \mapsto \text{False}$ and propagate $\neg a$ by BCP

$$x > 1$$

$$x < 4$$

$$xy > 4$$

$$yz^2 \leq 4$$

$$\neg a \vee x < -2$$

$$y > 0 \vee x^2z + y = 3$$

$$a \vee x^2 \geq 4 \vee y > 5$$

BICP and Formula Simplification

$$x > 1$$

$$x < 4$$

$$xy > 4$$

$$yz^2 \leq 4$$

$$\neg a \vee x < -2$$

$$y > 0 \vee x^2z + y = 3$$

$$a \vee x^2 \geq 4 \vee y > 5$$

\Rightarrow Derive $x \in (1, 4)$, $y \in (1, \infty)$, $z \in (-2, 2)$ by ICP

\Rightarrow Infer $(x < -2) \mapsto \text{False}$ and propagate $\neg a$ by BCP

\Rightarrow Check the status of literals in the given formulas

$$(xy > 4 \wedge yz^2 \leq 4) \mapsto (\text{True} \wedge \text{True}),$$

$$(\neg a \vee x < -2) \mapsto (\text{True} \vee \text{False}),$$

$$(y > 0 \vee x^2z + y = 3) \mapsto (\text{True} \vee \text{Unknown}),$$

$$(a \vee x^2 \geq 4 \vee y > 5) \mapsto (\text{False} \vee \text{Unknown} \vee \text{Unknown}).$$

BICP and Formula Simplification

$$x > 1$$

$$x < 4$$

$$xy > 4$$

$$yz^2 \leq 4$$

$$\neg a \vee x < -2$$

$$y > 0 \vee x^2z + y = 3$$

$$a \vee x^2 \geq 4 \vee y > 5$$

\Rightarrow Derive $x \in (1, 4)$, $y \in (1, \infty)$, $z \in (-2, 2)$ by ICP

\Rightarrow Infer $(x < -2) \mapsto \text{False}$ and propagate $\neg a$ by BCP

\Rightarrow Check the status of literals in the given formulas

$$(xy > 4 \wedge yz^2 \leq 4) \mapsto (\text{True} \wedge \text{True}),$$

$$(\neg a \vee x < -2) \mapsto (\text{True} \vee \text{False}),$$

$$(y > 0 \vee x^2z + y = 3) \mapsto (\text{True} \vee \text{Unknown}),$$

$$(a \vee x^2 \geq 4 \vee y > 5) \mapsto (\text{False} \vee \text{Unknown} \vee \text{Unknown}).$$

So, the formula after simplification is:

$$\underbrace{(x^2 \geq 4 \vee y > 5)}_{\text{Reduced Clauses}} \wedge \underbrace{(\neg a \wedge x \in (1, 4) \wedge y \in (1, \infty) \wedge z \in (-2, 2))}_{\text{Feasible Domain of Variables}} \wedge \underbrace{(xy > 4 \wedge yz^2 \leq 4)}_{\text{Propagated Literals}}.$$

Outline

- Introduction
- Dynamic Variable-level Partitioning
- **Experiments & Summary**

Evaluation

Comparison to Sequential Solving in Arithmetic Theories Benchmarks

	QF_LRA(1753)					QF_LIA(13226)				
	SAT	UNSAT	Failed	PAR-2	Improve	SAT	UNSAT	Failed	PAR-2	Improve
CVC5(S)	958	685	110	354714	0%	7046	3212	2968	7562277	0%
CVC5(AP-p8)	980	689	84	287256	19.02%	7321	3252	2653	6791509	10.19%
CVC5(AP-p16)	980	689	84	281524	20.63%	7350	3274	2602	6678936	11.68%
CVC5(AP-p32)	982	690	81	275957	22.20%	7365	3285	2576	6603235	12.68%
OpenSMT2(S)	991	700	62	173971	0%	7985	4645	596	1994585	0%
OpenSMT2(AP-p8)	1008	701	44	132925	23.59%	8116	4660	450	1629696	18.29%
OpenSMT2(AP-p16)	1008	701	44	133043	23.53%	8138	4663	425	1555190	22.03%
OpenSMT2(AP-p32)	1009	701	43	127489	26.72%	8160	4665	401	1489780	25.31%
Z3(S)	966	680	107	316097	0%	7862	3903	1461	4025347	0%
Z3(AP-p8)	995	683	75	235645	25.45%	8055	4152	1019	3031732	24.68%
Z3(AP-p16)	996	683	74	231738	26.69%	8066	4157	1003	2983526	25.88%
Z3(AP-p32)	998	684	71	225268	28.73%	8076	4160	990	2945091	26.84%

	QF_NRA(12134)					QF_NIA(25358)				
	SAT	UNSAT	Failed	PAR-2	Improve	SAT	UNSAT	Failed	PAR-2	Improve
CVC5(S)	5485	5811	838	2100561	0%	9460	4803	11095	27485835	0%
CVC5(AP-p8)	5709	5864	561	1425236	32.15%	13030	5504	6824	17250199	37.24%
CVC5(AP-p16)	5731	5864	539	1372485	34.66%	13045	5513	6800	17186305	37.47%
CVC5(AP-p32)	5743	5864	527	1343006	36.06%	13691	5588	6079	15346228	44.17%
Z3(S)	5626	5375	1133	2770153	0%	13779	5836	5743	14636656	0%
Z3(AP-p8)	5744	5686	704	1741660	37.13%	14191	6785	4382	11225626	23.30%
Z3(AP-p16)	5766	5705	663	1637352	40.89%	14193	6789	4376	11206526	23.44%
Z3(AP-p32)	5789	5712	633	1561862	43.62%	14320	6884	4154	10610746	27.51%

Summary (with 8 Cores) Based on PAR-2 score

Theory	#Instance	Speed Up
QF_LRA	1753	22.4%
QF_LIA	13226	15.7%
QF_NRA	12134	35.0%
QF_NIA	25358	32.4%

Our method with 8 cores **solves 1211 additional instances (out of 6247 previously unsolved)** that any single solver could not solve without our partitioner.

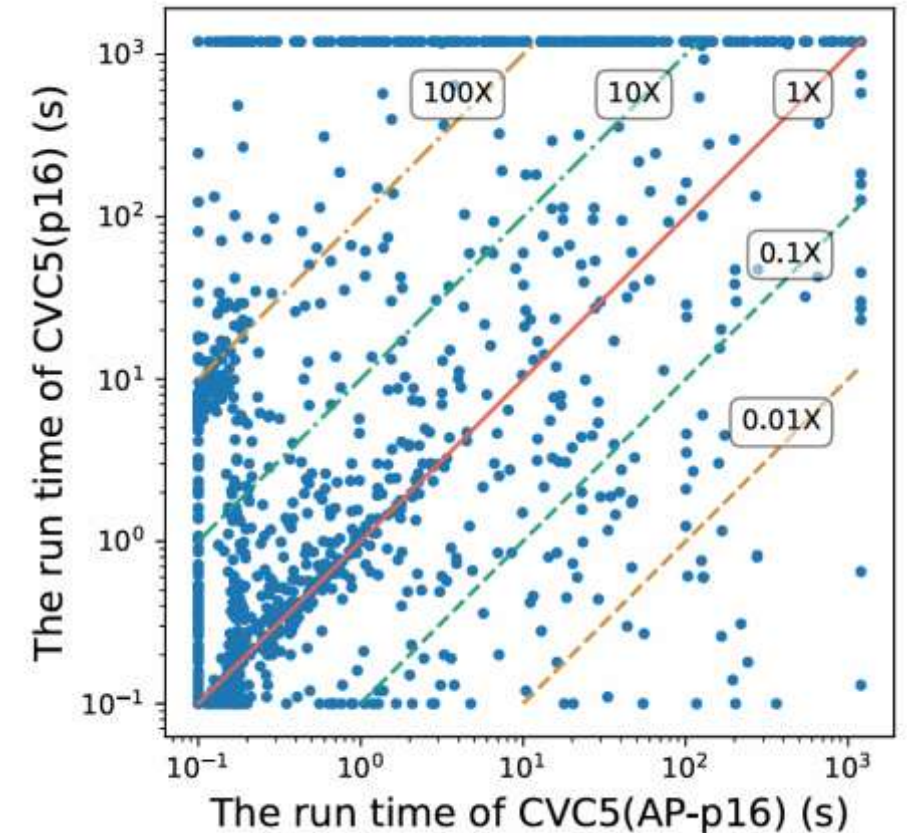
Evaluation

Comparison to state-of-the-art partitioning strategy.

	QF_LRA(1753)				QF_LIA(13226)			
	SAT	UNSAT	Solved	PAR-2	SAT	UNSAT	Solved	PAR-2
CVC5(p8)	964	677	1641	347087	7288	3199	10487	7004216
CVC5(AP-p8)	980	689	1669	287256	7321	3252	10573	6791509
OpenSMT2(p8)	998	701	1699	147360	8169	4698	12867	1384618
OpenSMT2(AP-p8)	1008	701	1709	132925	8116	4660	12776	1629696
CVC5(p16)	965	676	1641	346435	7316	3225	10541	6895797
CVC5(AP-p16)	980	689	1669	281524	7350	3274	10624	6678936
OpenSMT2(p16)	997	698	1695	157708	8097	4623	12720	1778020
OpenSMT2(AP-p16)	1008	701	1709	133043	8138	4663	12801	1555190

	QF_NRA(12134)				QF_NIA(25358)			
	SAT	UNSAT	Solved	PAR-2	SAT	UNSAT	Solved	PAR-2
CVC5(p8)	5559	5798	11357	1948280	12503	4480	16983	20716252
CVC5(AP-p8)	5709	5864	11573	1425236	13030	5504	18534	17250199
CVC5(p16)	5575	5796	11371	1920929	12821	4405	17226	20123111
CVC5(AP-p16)	5731	5864	11595	1372485	13045	5513	18558	17186305

**Comparison of the solving ability of the SOTA
parallel strategy with CVC5 and OpenSMT2**



**Run time comparison in
pure-conjunction instances**

Summary

AriParti in GitHub

Parallel Version: <https://github.com/shaowei-cai-group/AriParti>

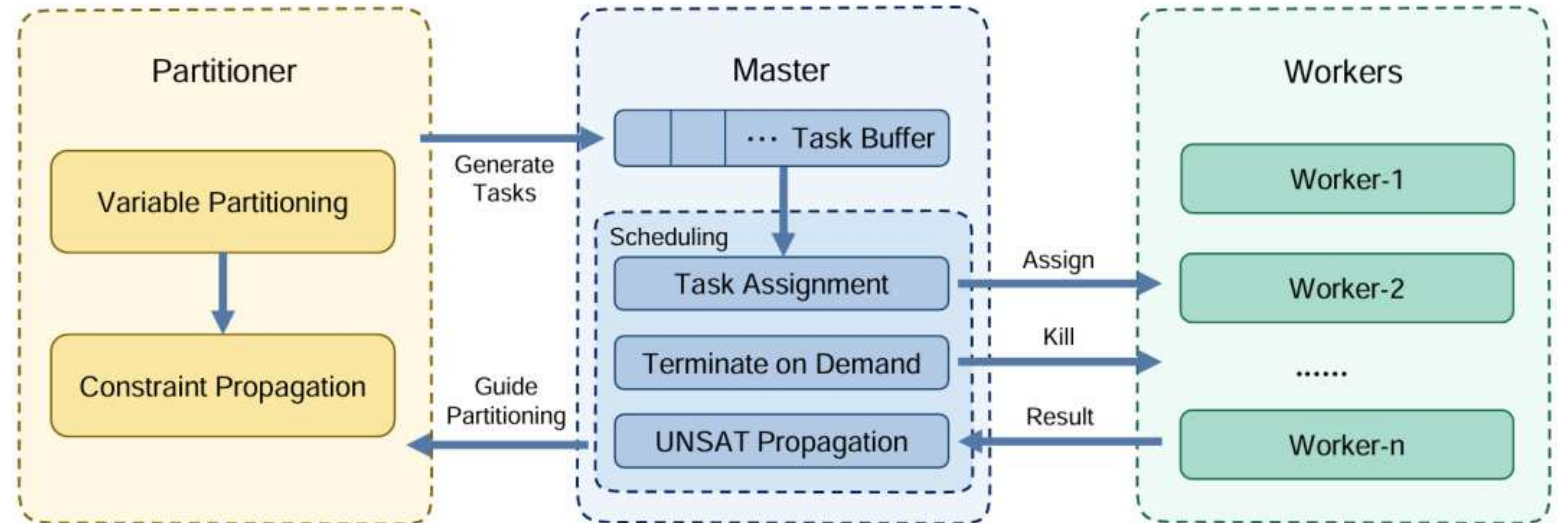
Distributed Version in SMT-COMP 2024: ~/Z3-Parti-Z3pp-at-SMT-COMP-2024

Key ideas:

- Variable-level partition
- BICP for simplification
- Flexible dynamic

Experimental Results:

- After being applied to the cutting-edge solvers, we **solved 3495 more instances** on average, and the solving **speed improved by about 30%**.
- Compared with the SOTA partitioning strategies, it has **significantly improved in nonlinear theories and pure-conjunction type instances**.



Distributed SMT Solving Based on Dynamic Variable-level Partitioning

Mengyu Zhao, Shaowei Cai*, and Yuhang Qian

Key Laboratory of System Software (Chinese Academy of Sciences)
and State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences, Beijing, China

`{zhaomy, caisw, qianyh}@ios.ac.cn`