Provably Pointless Propagation Calls

Chris Jefferson, Karen E. Petrie and Dave Cohen

Overview

- Solvers **propagate** constraints to remove domain values.
- If we can **avoid calling propagators** we can speed up solvers

Propagating SAT

$$a \vee b \vee c \vee d \vee e$$

- Each variable has domain contained in {*true*, *false*}
- Remove values which occur in **no solution**

Propagating SAT

$a \vee b \vee c \vee d \vee e$

One rule to get all propagation:

If all but one variable assigned *false*:
Assign other variable *true*.

Tradition CP Implementation

0/1	0/1	0/1	0/1
a	b	c	d

Variables Assigned False: 0

$a \vee b \vee c \vee d$

Tradition CP Implementation

0	0/1	0/1	0/1
a	b	c	d

Variables Assigned False: 1

$a \vee b \vee c \vee d$

Tradition CP Implementation

0	0/1	0	0/1
a	b	c	d

Variables Assigned False: 2

$a \vee b \vee c \vee d$

Tradition CP Implementation

0	0	0	0/1
a	b	c	d

Variables Assigned False: 3

$a \vee b \vee c \vee d$

Tradition CP Implementation

0	0	0	1
a	b	c	d

Variables Assigned False: 3

$a \vee b \vee c \vee d$

Propagation

- Can we reduce / change those requirements?
  - Need to trigger on all assignments?
  - Need to count assigned variables?

SAT Propagation

- Idea: If two variables are either unassigned or assigned true, no need to do anything.

- If can't find two, have to propagate or fail.

Propagation Example

0/1	0/1	??	??
a	b	c	d

$a \vee b \vee c \vee d$

Propagation Example

0/1	0/1	0/1	0/1
a	b	c	d

Triggers:

$a \vee b \vee c \vee d$

Propagation Example

0	0/1	0/1	0/1
a	b	c	d

Triggers:

- *a* assigned false.
- Update pointer.

Propagation Example

0	0/1	0/1	0/1
a	b	c	d

Triggers:

- *a* assigned false.
- Update pointer.

Propagation Example

0/1	0/1	0/1	0/1
a	b	c	d

Triggers:

- Backtrack. *a* unassigned.

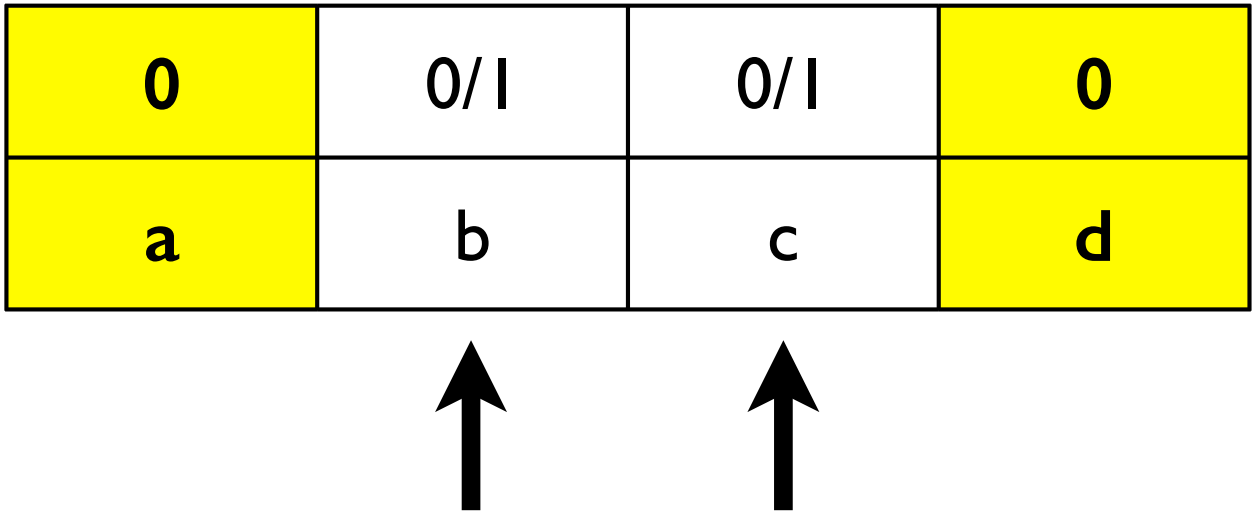
Propagation Example

0/1	**1**	0/1	0/1
a	**b**	c	d

Triggers:

- If *b* is assigned true,
  pointer doesn't move.

Propagation Example



Triggers:

- **If other variables assigned, nothing happens!**

Propagation Example

0	**0**	0/1	0
a	**b**	c	d

Triggers:

- If we cannot find something new to watch...

Propagation Example

0	0	1	0
a	b	c	d

Triggers:

- Assign other watch!

Propagation Example

0	0	1	0
a	b	c	d

Triggers:

- Leave triggers where they are!

Dynamic Support & Watched Literals

- One feature of **Watched Literals** is that they do not move during backtrack
- We ignore that feature in this paper
  - It means we need to consider the effect on backtracking, which is surprisingly difficult
  - The theory is hard enough already!

Dynamic Triggers

- Minion supports both static (non-moving) and dynamic triggers.
  - Dynamic triggers have up to **2x** overhead.
- But, when you only need a few, they provide a massive speedup!

Previous Papers

- SAT clause: 10x speedup
- Element: 3x speedup
- Table: 2x speedup
- Constraint Disjunction: 50x speedup
- Half-reification: 10x speedup

Previous Papers

- AllDifferent: (mostly) slower
- Global Cardinality constraint: slower
- GAC Lexicographic Ordering: small speedup

Support

- A support is a set of literals where:
  - If none of them are removed, propagator will do nothing

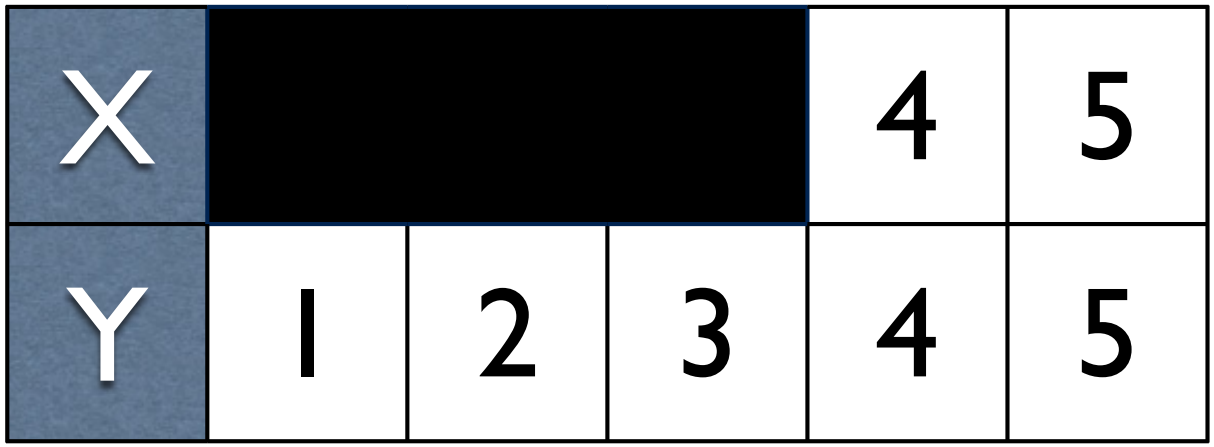
Questions

- Given a propagator and set of domains:
  - Are a given set of literals a support?
  - Is there a set of literals of size < n?

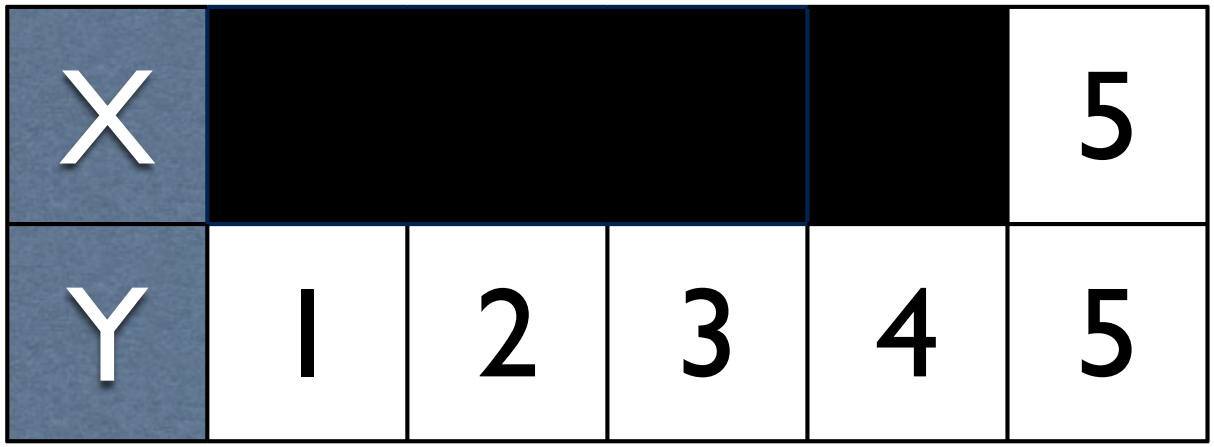
X≠Y

X	1	2	3	4	5
Y	1	2	3	4	5

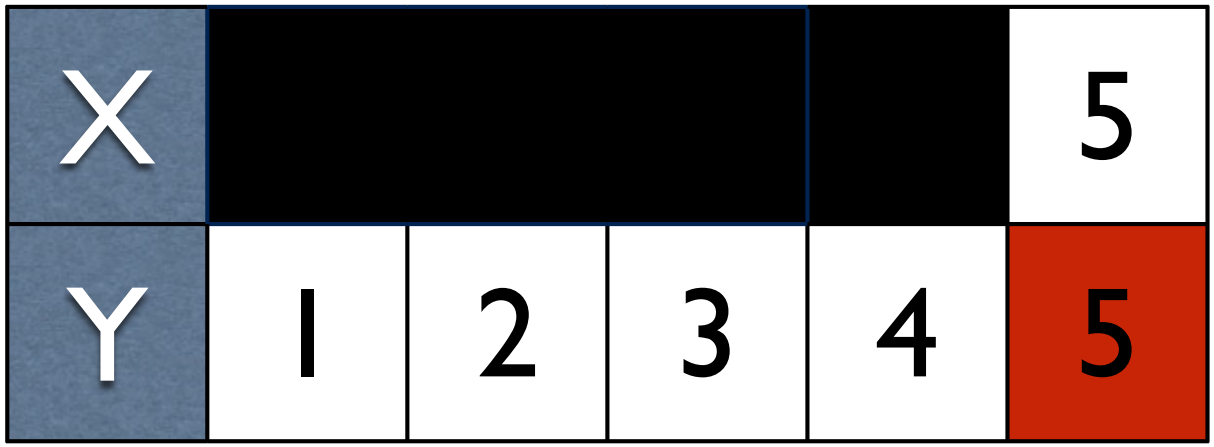
X≠Y



X≠Y



X≠Y



X≠Y

X	1	2	3	4	5
Y	1	2	3	4	5

X≠Y

X	1	2	3	4	5
Y	1	2	3	4	5

X=Y

X	1		3	4	5
Y	1	2	3	4	5

X=Y

X	1		3	4	5
Y	1	2	3	4	5

X=Y

X	1	2	3	4	5
Y	1	2	3	4	5

$A+B+C+D \geq 4$

A	0	1	2	3	4
B	0	1	2	3	4
C	0	1	2	3	4
D	0	1	2	3	4

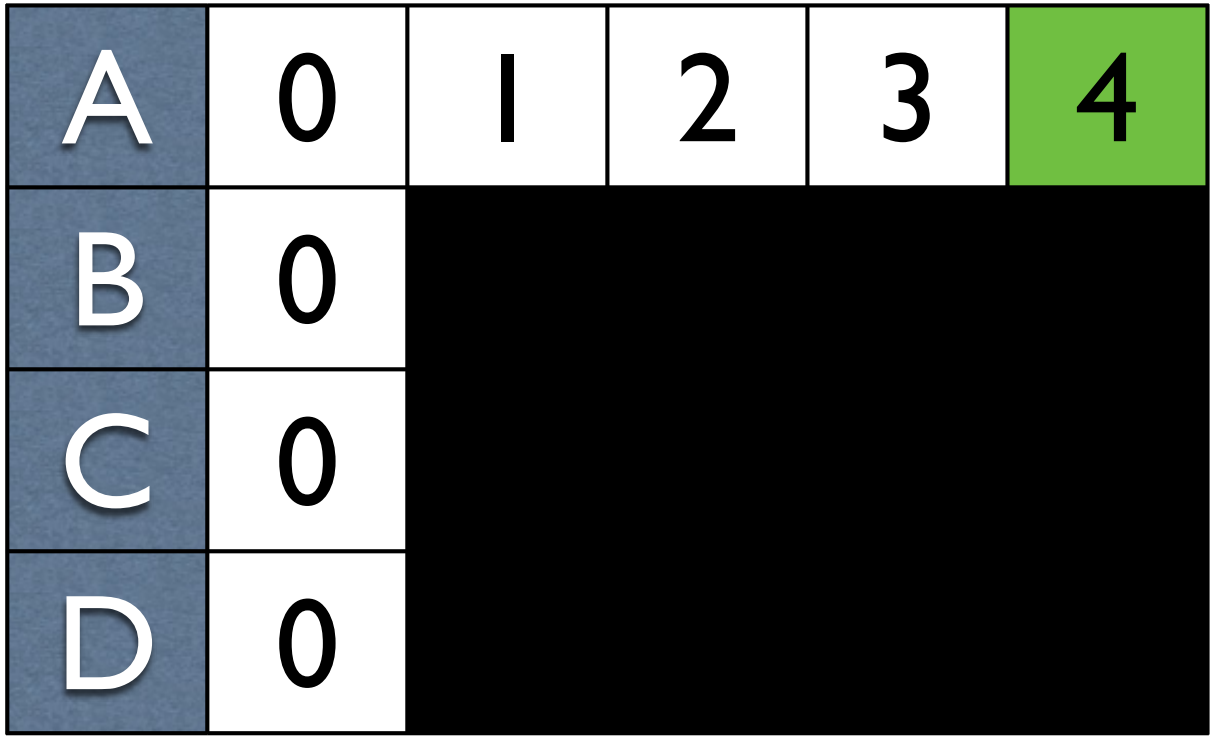
$A+B+C+D \geq 4$

A	0	1	2	3	4
B	0	1	2	3	4
C	0	1	2	3	4
D	0	1	2	3	4

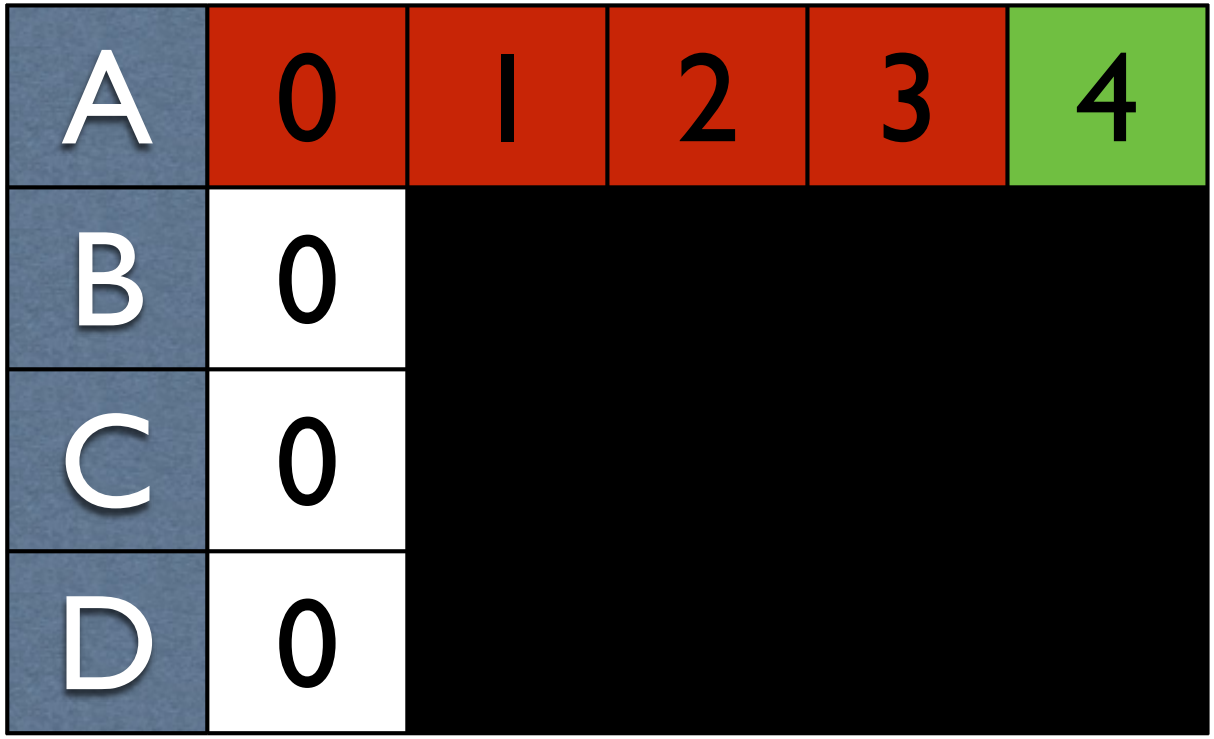
$A+B+C+D \geq 4$

A	0	1	2	3	4
B	0	1	2	3	4
C	0	1	2	3	4
D	0	1	2	3	4

A+B+C+D ≥ 4



$A+B+C+D \geq 4$



$A+B+C+D \geq 4$

A	0	1	2	3	4
B	0	1	2	3	4
C	0	1	2	3	4
D	0	1	2	3	4

$A+B+C+D \geq 4$

A	0	1	2	3	4
B	0	1	2	3	4
C	0	1	2	3	4
D	0	1	2	3	4

$M_{Index} = Result$

M_1	1	2	3	4
M_2	1	2	3	4
M_3	1	2	3	4

Index	1	2	3

Result	1	2	3	4

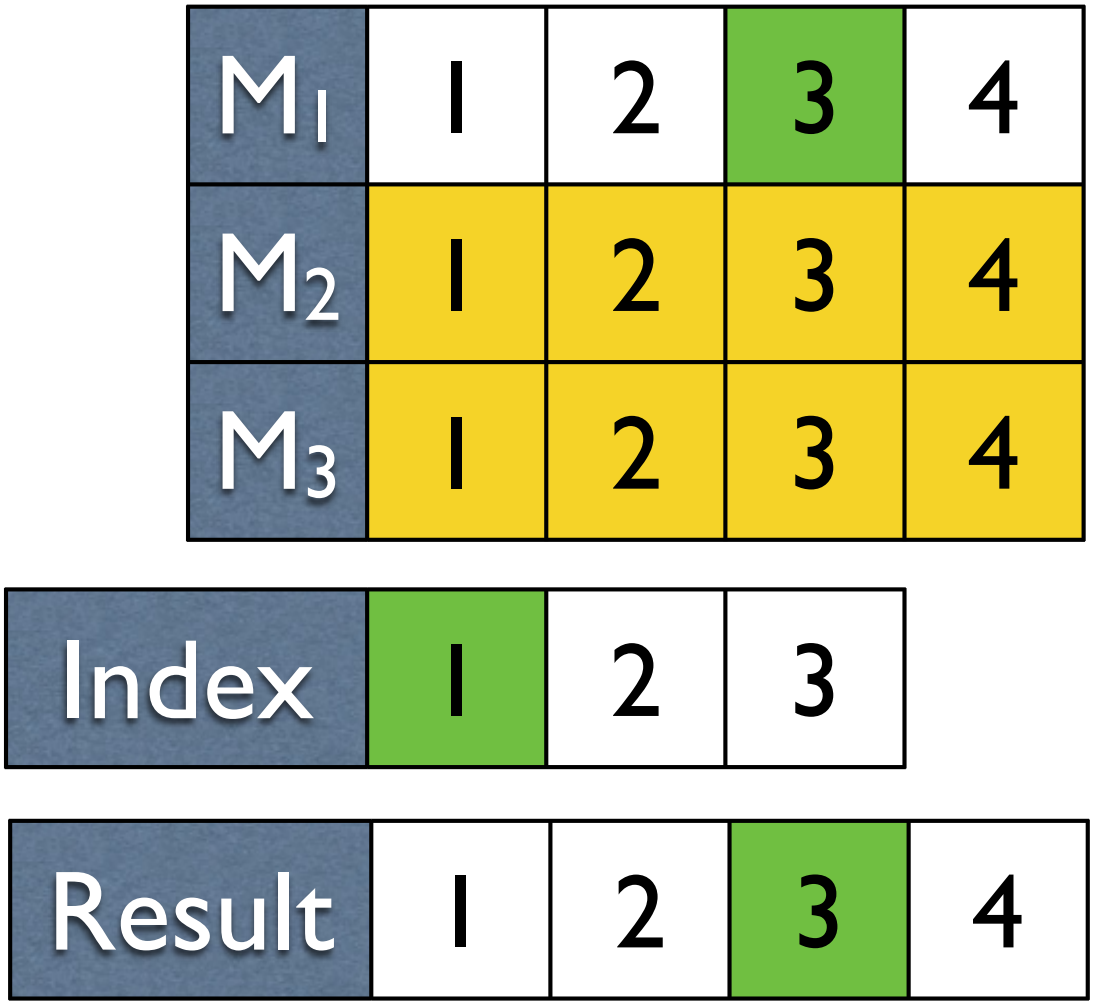
$M_{Index} = Result$

M_1	1	2	**3**	4
M_2	1	2	3	4
M_3	1	2	3	4

Index	**1**	2	3

Result	1	2	**3**	4

$M_{Index} = Result$



Support

- Full Supports are stitched together from sets of literals which support some subset.

Finding Supports

- How hard is it to check a support is valid?
  - Can be NP-hard, even for propagators which run in polynomial time.

$$(M[X,Y] = A) \wedge (A = B)$$

Different Propagator Levels

- What about weaker propagators?
  - Sometimes makes support, sometimes increases it!

Trivially Fixable

- Given a constraint C
- Any non-satisfying assignment A
- Change any assignment to any variable in A, get a satisfying assignment.

Trivially Fixable

$$A+B+C+D \neq E$$

$$a \vee b \vee c \vee d$$

Parity

Why Trivially Fixable

Forwarding Checking == GAC
if and only if
a constraint is Trivially Fixable

Supports for Forward Checking

- Watch two different values in two different variables.
  - If only one (or no) variables or unassigned, run FC propagator!

Static Triggering

- What if we are forced to place all triggers on at the start of search?
  - We show which literals you have to watch (usually, all of them).
  - … if you can only watch literals

Other Trigger Types

- Solvers usually have other types of triggers
  - Maximum / Minimum Domain Value
  - Variable Assigned
  - Variable Changed

Conclusions

- Dynamic supports are useful sometimes..
- But not always!

- Further investigation needed!