

Hard Examples for Common Variable Decision Heuristics

Marc Vinyals

Technion
Haifa, Israel

AAAI 2020

DPLL

$$y \vee z \quad y \vee \bar{z} \quad x \vee \bar{y} \vee z \quad x \vee \bar{y} \vee \bar{z} \quad \bar{x} \vee \bar{y}$$

Algorithm 1: DPLL

while *not solved* **do**

if *conflict* **then** backtrack()

else if *unit* **then** propagate()

else branch()

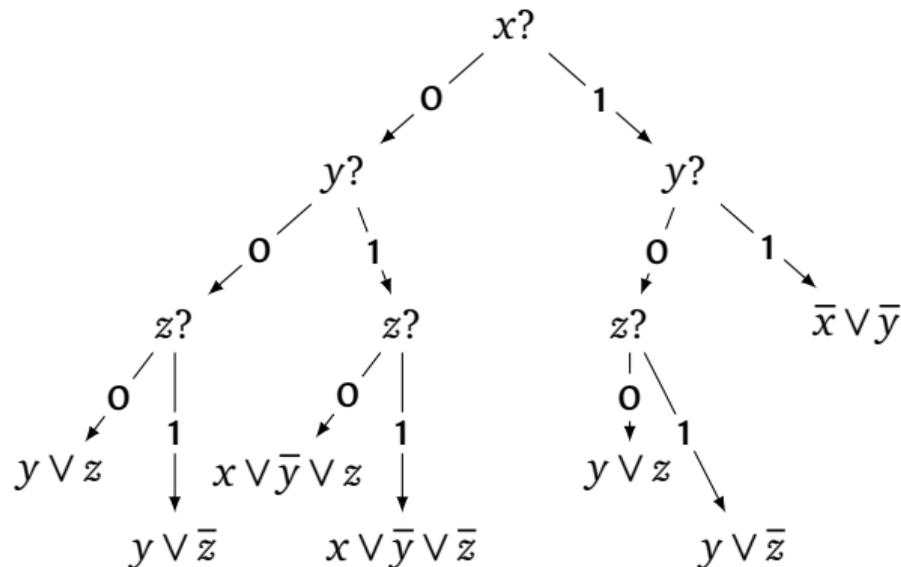
State: partial assignment

DPLL

$$y \vee z \quad y \vee \bar{z} \quad x \vee \bar{y} \vee z \quad x \vee \bar{y} \vee \bar{z} \quad \bar{x} \vee \bar{y}$$

Algorithm 1: DPLL**while** *not solved* **do** **if** *conflict* **then** backtrack() **else if** *unit* **then** propagate() **else** branch()

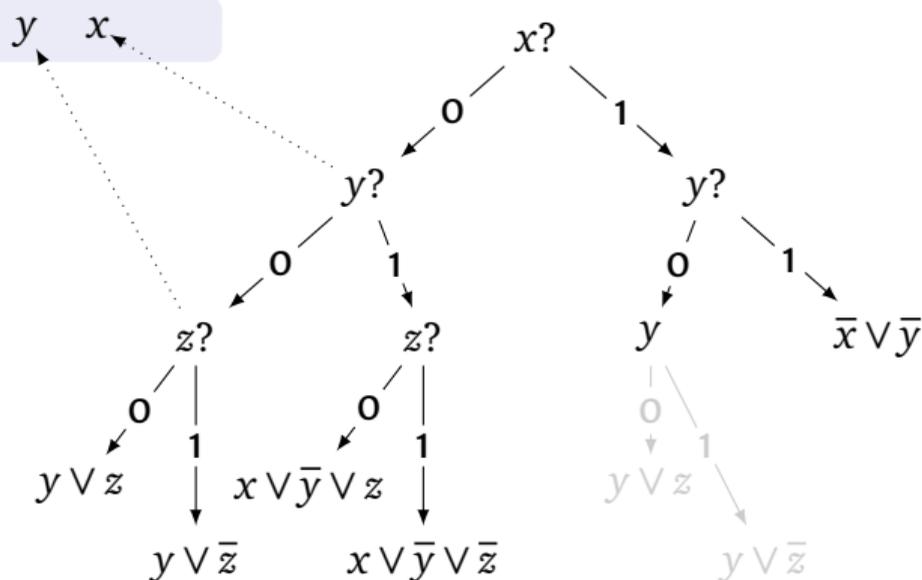
State: partial assignment



CDCL

Algorithm 2: CDCL**while** *not solved* **do** **if** *conflict* **then** **learn**() **else if** *unit* **then** propagate() **else** branch()State: partial assignment
& learned clauses

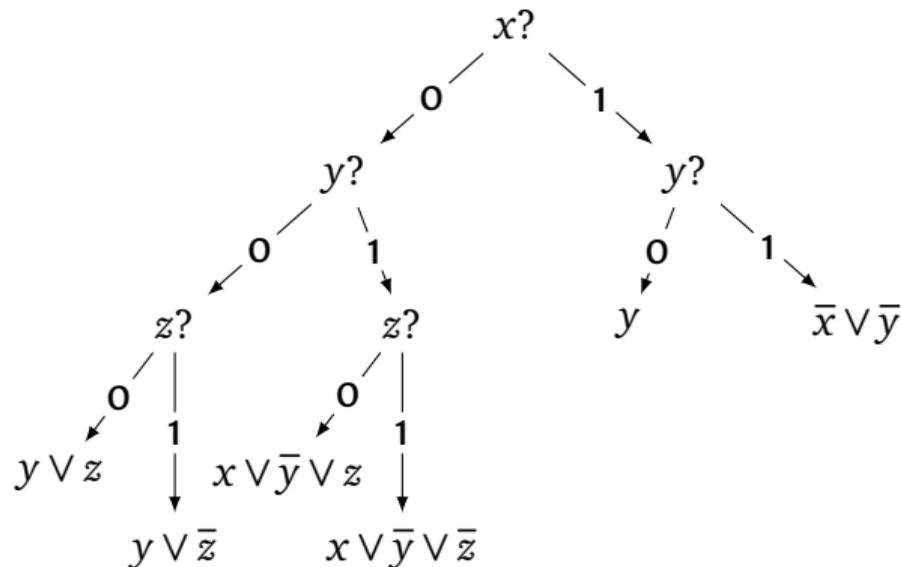
$$y \vee z \quad y \vee \bar{z} \quad x \vee \bar{y} \vee z \quad x \vee \bar{y} \vee \bar{z} \quad \bar{x} \vee \bar{y}$$



Resolution

- Analyse CDCL using proof complexity tools

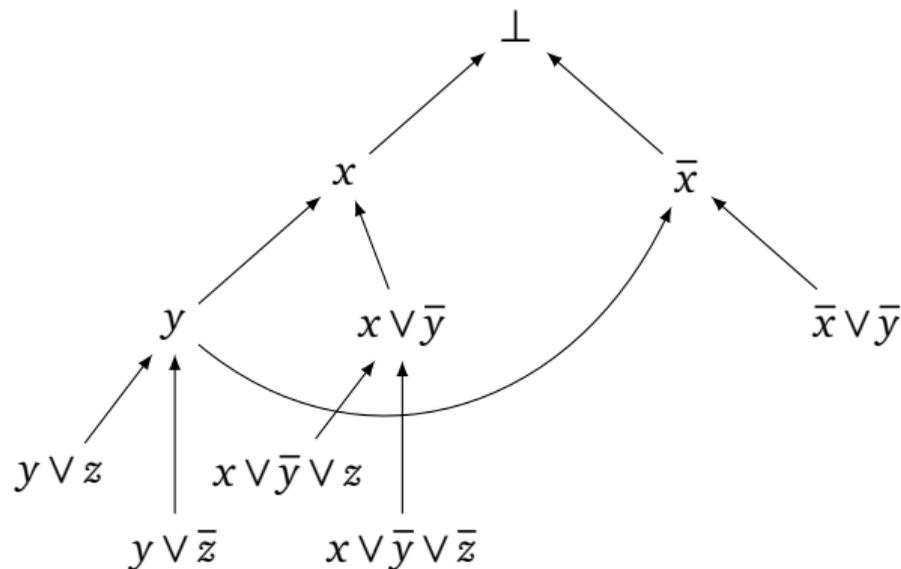
$$\frac{C \vee v \quad D \vee \bar{v}}{C \vee D}$$



Resolution

- Analyse CDCL using proof complexity tools

$$\frac{C \vee v \quad D \vee \bar{v}}{C \vee D}$$



CDCL equivalent to Resolution

Observation

[Beame, Kautz, Sabharwal '04]

CDCL produces resolution proofs

CDCL equivalent to Resolution

Observation

[Beame, Kautz, Sabharwal '04]

CDCL produces resolution proofs

Theorem

[Pipatsrisawat, Darwiche '09]

CDCL can efficiently find resolution proofs

CDCL equivalent to Resolution

Observation

[Beame, Kautz, Sabharwal '04]

CDCL produces resolution proofs

Theorem

[Pipatsrisawat, Darwiche '09]

CDCL can efficiently find resolution proofs

Wait a minute...

Theorem

[Atserias, Müller '19]

If a deterministic algorithm efficiently finds resolution proofs then $P = NP$

CDCL equivalent to Resolution

Observation

[Beame, Kautz, Sabharwal '04]

CDCL produces resolution proofs

Theorem

[Pipatsrisawat, Darwiche '09]

CDCL can efficiently find resolution proofs

Wait a minute...

Theorem

[Atserias, Müller '19]

If a deterministic algorithm efficiently finds resolution proofs then $P = NP$

with non-deterministic variable decisions

Separation of CDCL vs Resolution

Theorem

There are formulas such that

- ▶ Resolution refutations of polynomial length
- ▶ Exponential time in CDCL with common variable decision heuristics

Separation of CDCL vs Resolution

Theorem

There are formulas such that

- ▶ Resolution refutations of polynomial length
- ▶ Exponential time in CDCL with **common variable decision heuristics**

Variable Decision Heuristics

Which literal do we pick next?

- ▶ Will lead to a conflict quickly.
- ▶ Was involved in conflicts recently.

Algorithm 2: CDCL

while *not solved* **do**

if *conflict* **then** learn()

else if *unit* **then** propagate()

else branch()

Variable Decision Heuristics

Which literal do we pick next?

- ▶ Will lead to a conflict quickly.
- ▶ Was involved in conflicts recently.

VSIDS

- ▶ Give a score $q(x)$ to variable x .
- ▶ At each conflict
 - ▶ Bump $q' = q + 1$ if x involved.
 - ▶ Decay $q' = 0.95 \cdot q$ all variables.
- ▶ Pick variable with largest score

Algorithm 2: CDCL

while *not solved* **do**

if *conflict* **then** learn()

else if *unit* **then** propagate()

else branch()

Variable Decision Heuristics

Which literal do we pick next?

- ▶ Will lead to a conflict quickly.
- ▶ Was involved in conflicts recently.

VSIDS

- ▶ Give a score $q(x)$ to variable x .
- ▶ At each conflict
 - ▶ Bump $q' = q + 1$ if x involved.
 - ▶ Decay $q' = 0.95 \cdot q$ all variables.
- ▶ Pick variable with largest score

Sign

- ▶ Last assigned.

Algorithm 2: CDCL

while *not solved* **do**

if *conflict* **then** learn()

else if *unit* **then** propagate()

else branch()

Properties of VSIDS

- ▶ Each conflict
 - ▶ Bump $q' = q + 1$ if x involved.
 - ▶ Decay $q' = 0.95 \cdot q$ all variables.

Observation

A variable that has been in a conflict is picked before a variable that never has.

Properties of VSIDS

- ▶ Each conflict
 - ▶ Bump $q' = q + 1$ if x involved.
 - ▶ Decay $q' = 0.95 \cdot q$ all variables.

Observation

A variable that has been in a conflict is picked before a variable that never has.

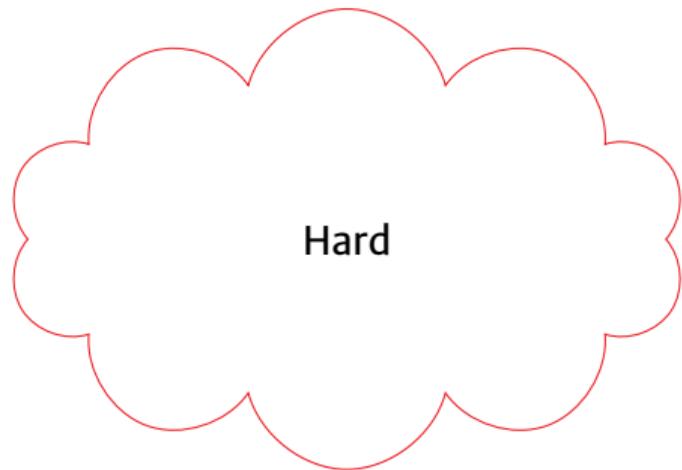
Fine Print

Not true if finite precision.

Does hold if stable priority queue.

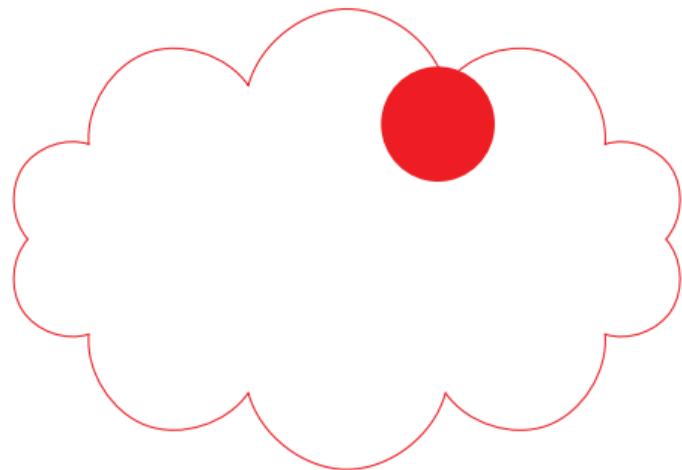
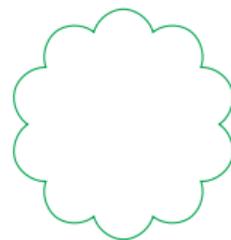
Intuition

- ▶ **Easy** part + **Hard** part.



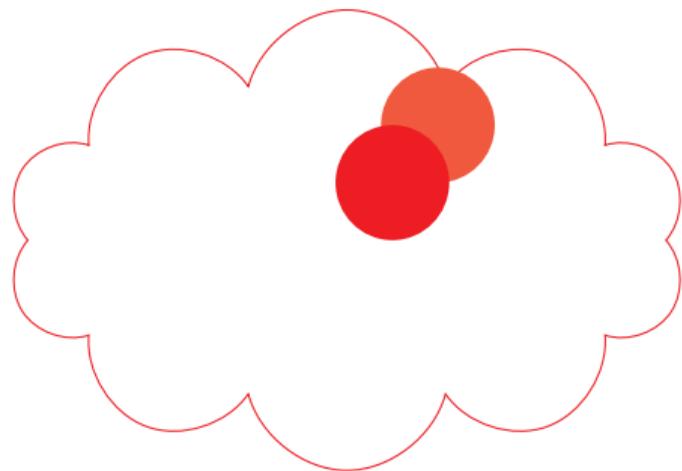
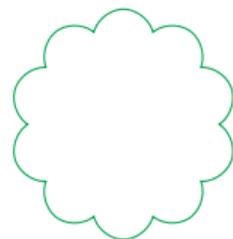
Intuition

- ▶ **Easy** part + **Hard** part.
- ▶ Conflict in hard part \implies More conflicts in hard part.



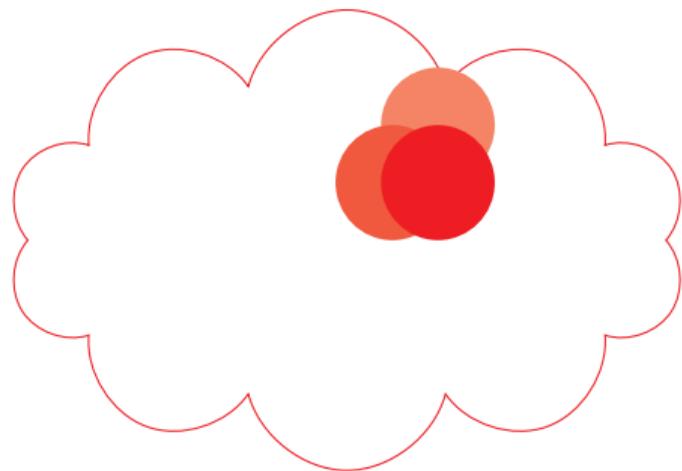
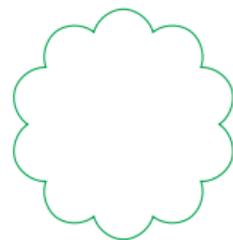
Intuition

- ▶ **Easy** part + **Hard** part.
- ▶ Conflict in hard part \implies More conflicts in hard part.



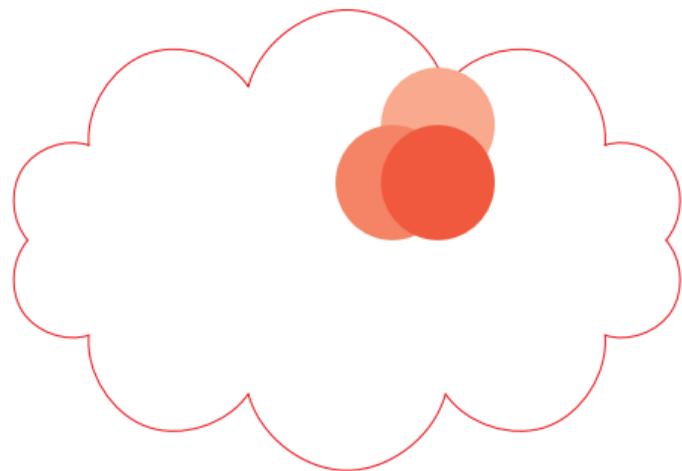
Intuition

- ▶ **Easy** part + **Hard** part.
- ▶ Conflict in hard part \implies More conflicts in hard part.



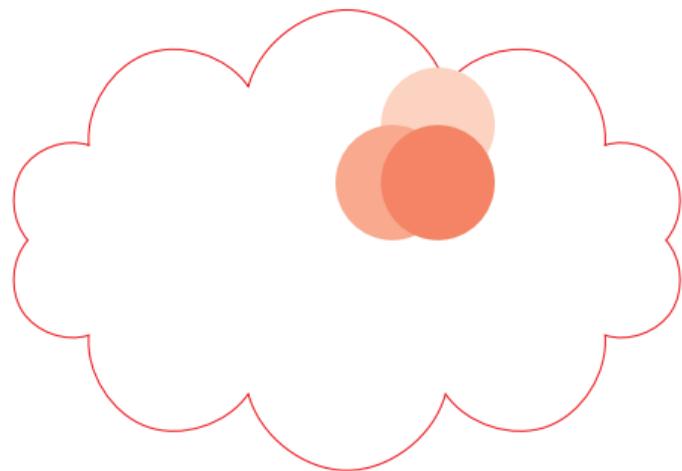
Intuition

- ▶ **Easy** part + **Hard** part.
- ▶ Conflict in hard part \implies More conflicts in hard part.
- ▶ But hard formulas are global.
- ▶ Eventually stabilize.
- ▶ Then chance to hit easy formula.



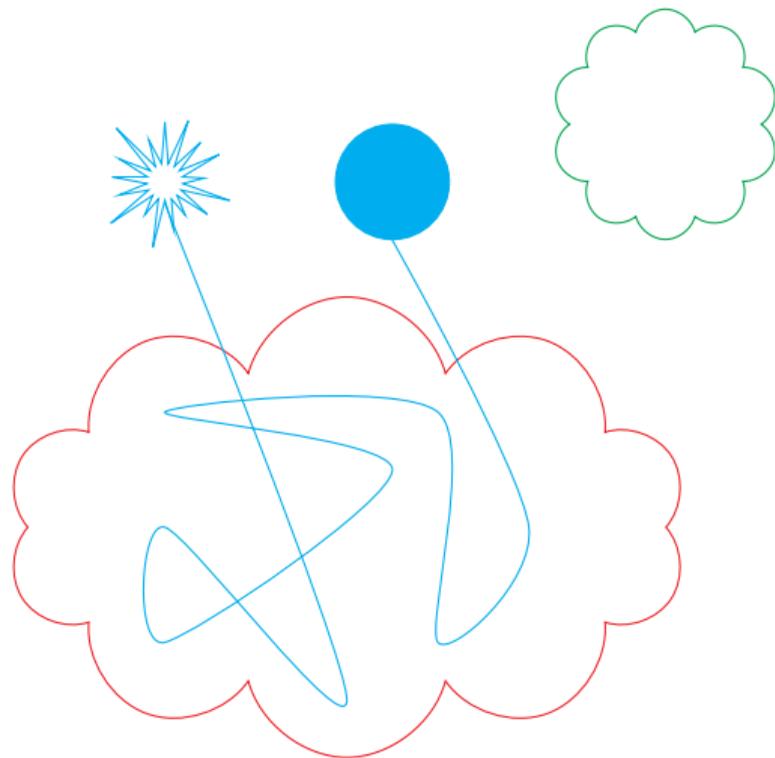
Intuition

- ▶ **Easy** part + **Hard** part.
- ▶ Conflict in hard part \implies More conflicts in hard part.
- ▶ But hard formulas are global.
- ▶ Eventually stabilize.
- ▶ Then chance to hit easy formula.



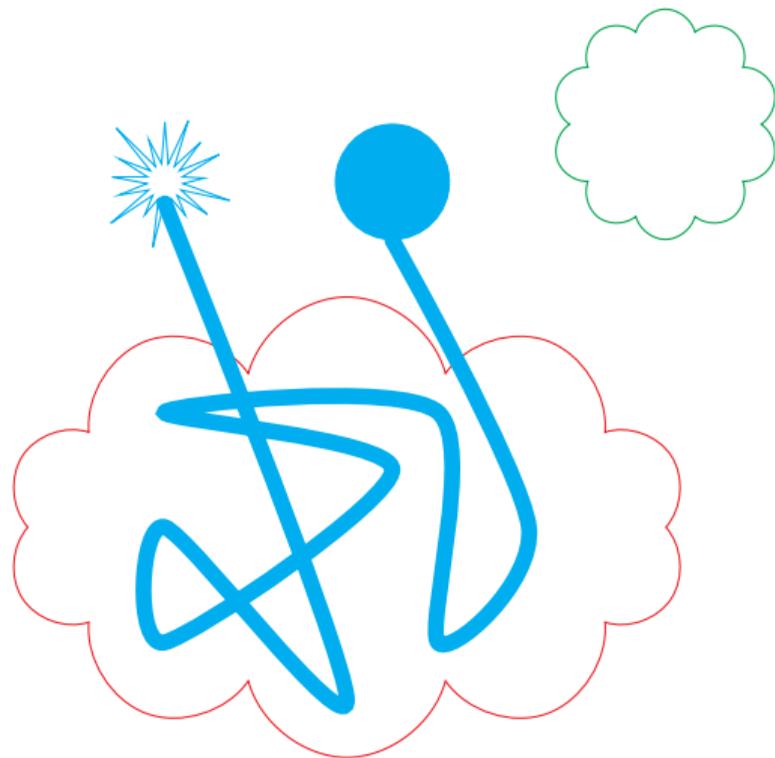
Intuition (II)

- ▶ **Pitfall gadget** produces a conflict involving all hard variables.



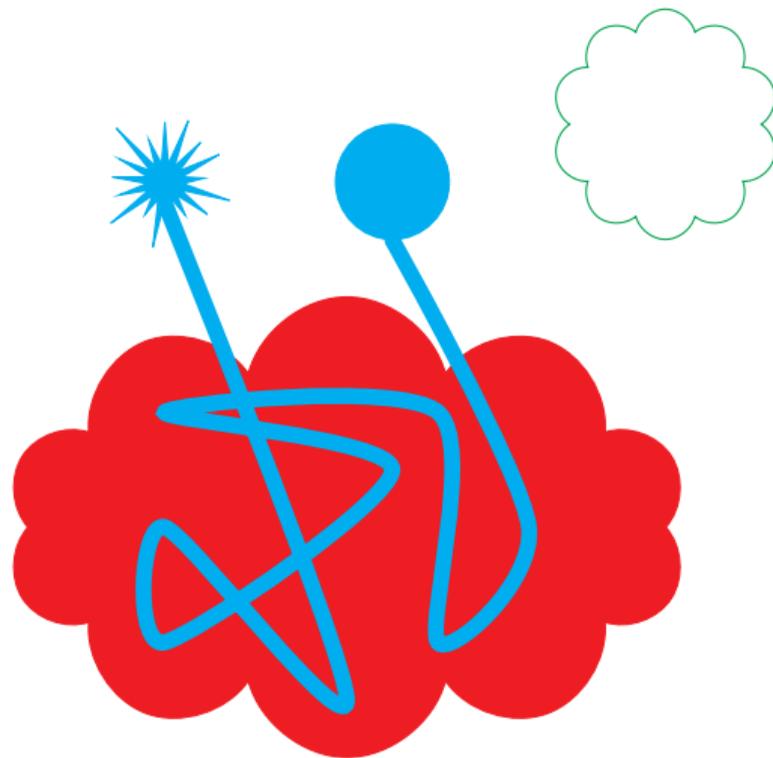
Intuition (II)

- ▶ **Pitfall gadget** produces a conflict involving all hard variables.



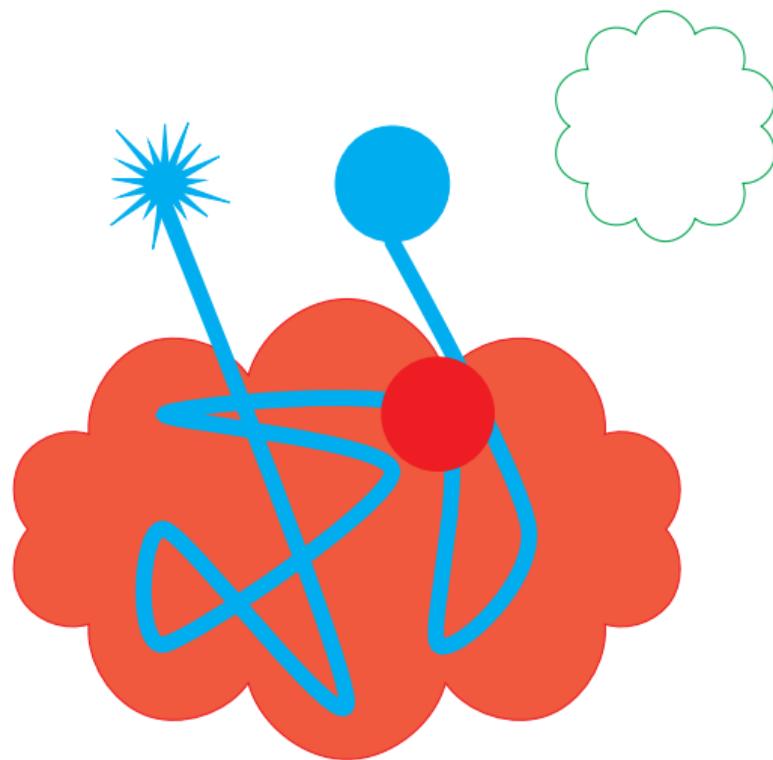
Intuition (II)

- ▶ **Pitfall gadget** produces a conflict involving all hard variables.



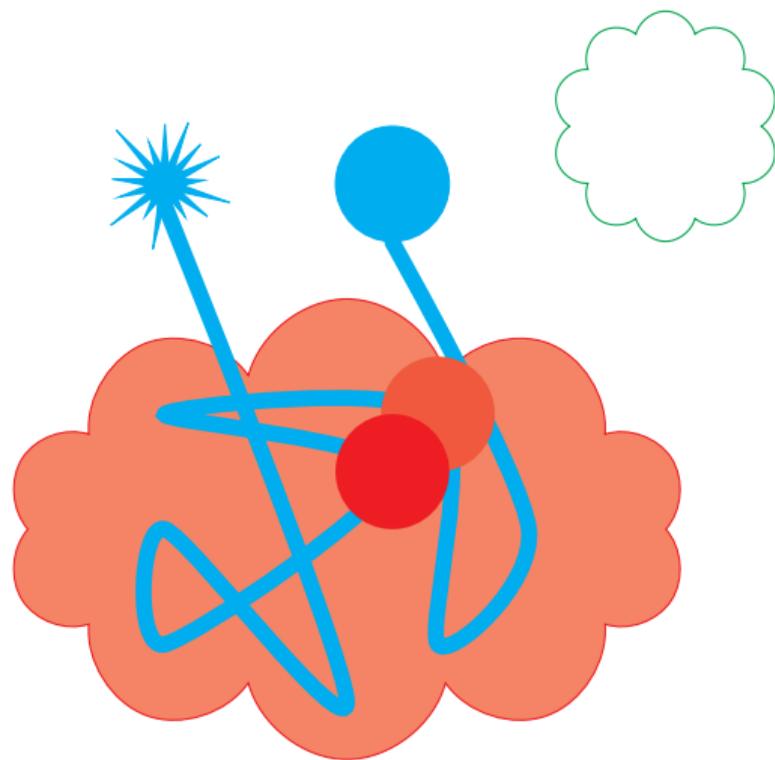
Intuition (II)

- ▶ **Pitfall gadget** produces a conflict involving all hard variables.
- ▶ Solver stuck with hard variables!



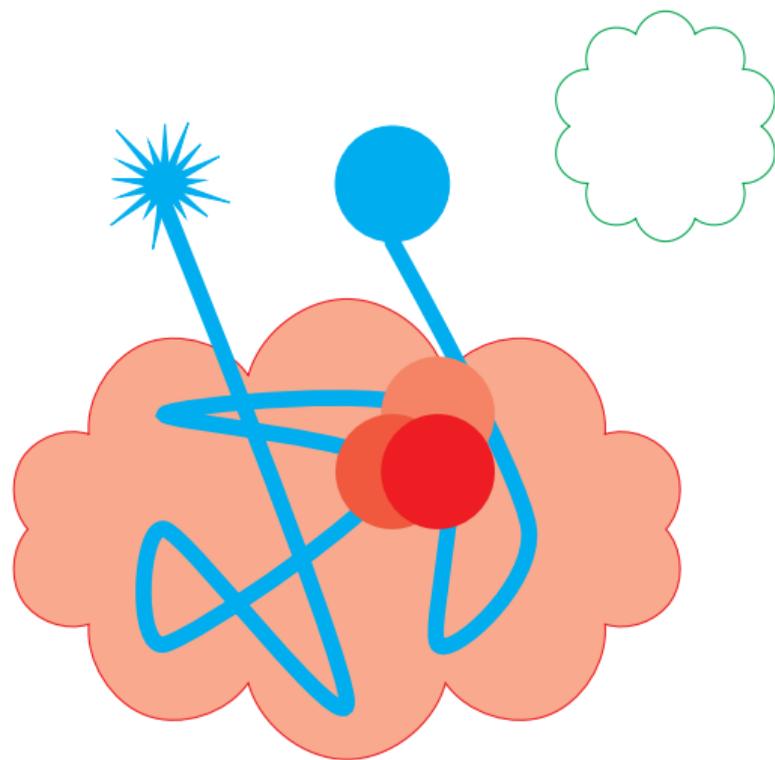
Intuition (II)

- ▶ **Pitfall gadget** produces a conflict involving all hard variables.
- ▶ Solver stuck with hard variables!



Intuition (II)

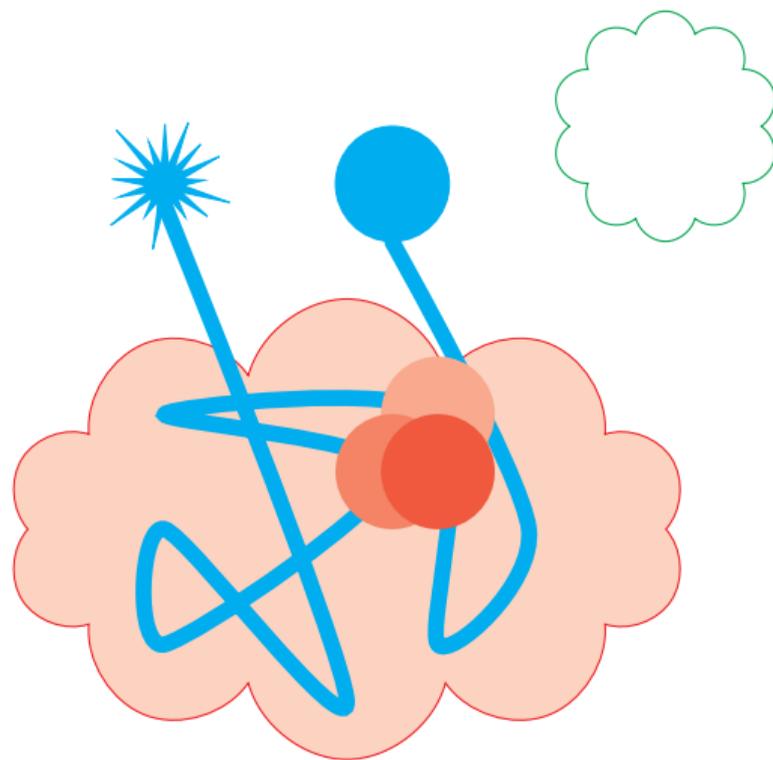
- ▶ **Pitfall gadget** produces a conflict involving all hard variables.
- ▶ Solver stuck with hard variables!



Intuition (II)

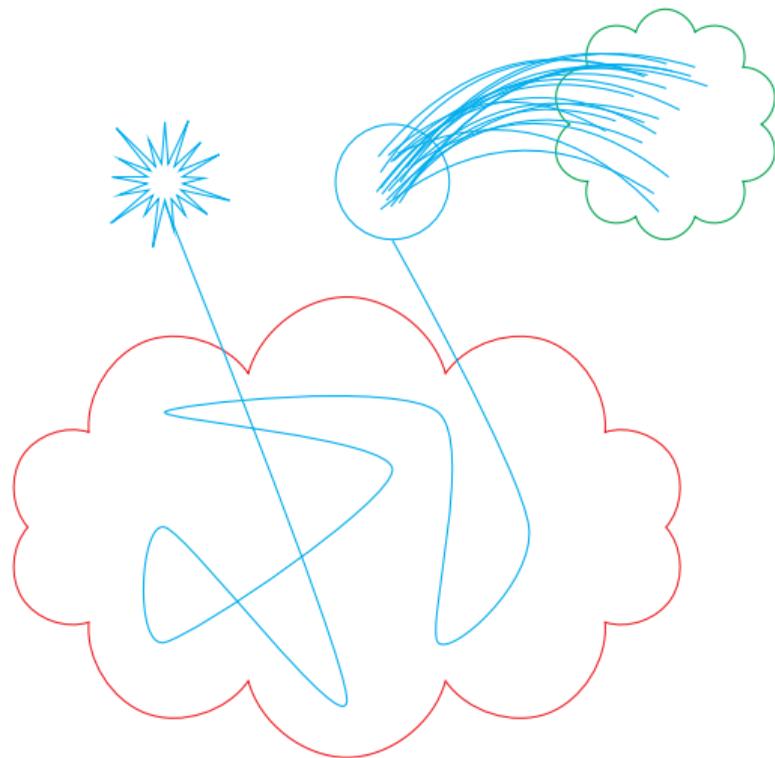
- ▶ **Pitfall gadget** produces a conflict involving all hard variables.
- ▶ Solver stuck with hard variables!

- ▶ But still $1/\text{poly}$ probability of solving easy part first.



Intuition (III)

- ▶ Make easy variables lead to pitfall gadget.



Experiments

Mean CPU time to solve (s)

Formula	CaDiCaL VMTF	Glucose VSIDS	MapleSAT CHB	MapleSAT LRB	Static
Hard(45)	3331	754	621	424	3600
Hard(50)	3600	3600	3600	3600	3600

Experiments

Mean CPU time to solve (s)

Formula	CaDiCaL VMTF	Glucose VSIDS	MapleSAT CHB	MapleSAT LRB	Static
Hard(45)	3331	754	621	424	3600
Pitfall(45)	1963	2273	607	2650	< 1
Hard(50)	3600	3600	3600	3600	3600
Pitfall(50)	3600	3600	3600	3600	< 1

Take Home

Result

- ▶ CDCL with VSIDS not equivalent to Resolution

Take Home

Result

- ▶ CDCL with VSIDS not equivalent to Resolution

Open Problems

- ▶ Proof robust wrt score precision?
- ▶ Simpler construction?
- ▶ Improve VSIDS?

Take Home

Result

- ▶ CDCL with VSIDS not equivalent to Resolution

Open Problems

- ▶ Proof robust wrt score precision?
- ▶ Simpler construction?
- ▶ Improve VSIDS?

Thanks!