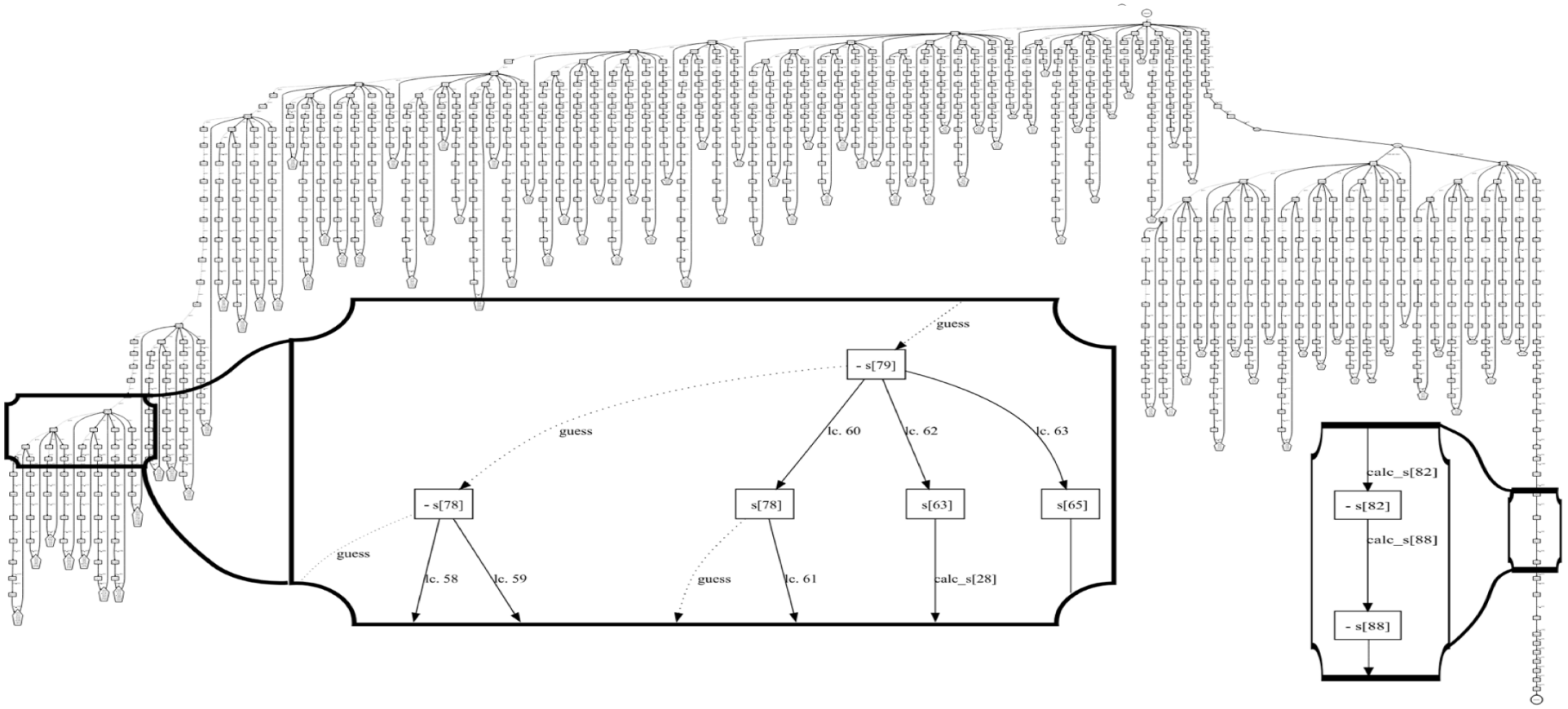
CDCL: Conflict Driven Clause Learning

1. UnitPropagation(m, F): applies unit propagation and extends m.

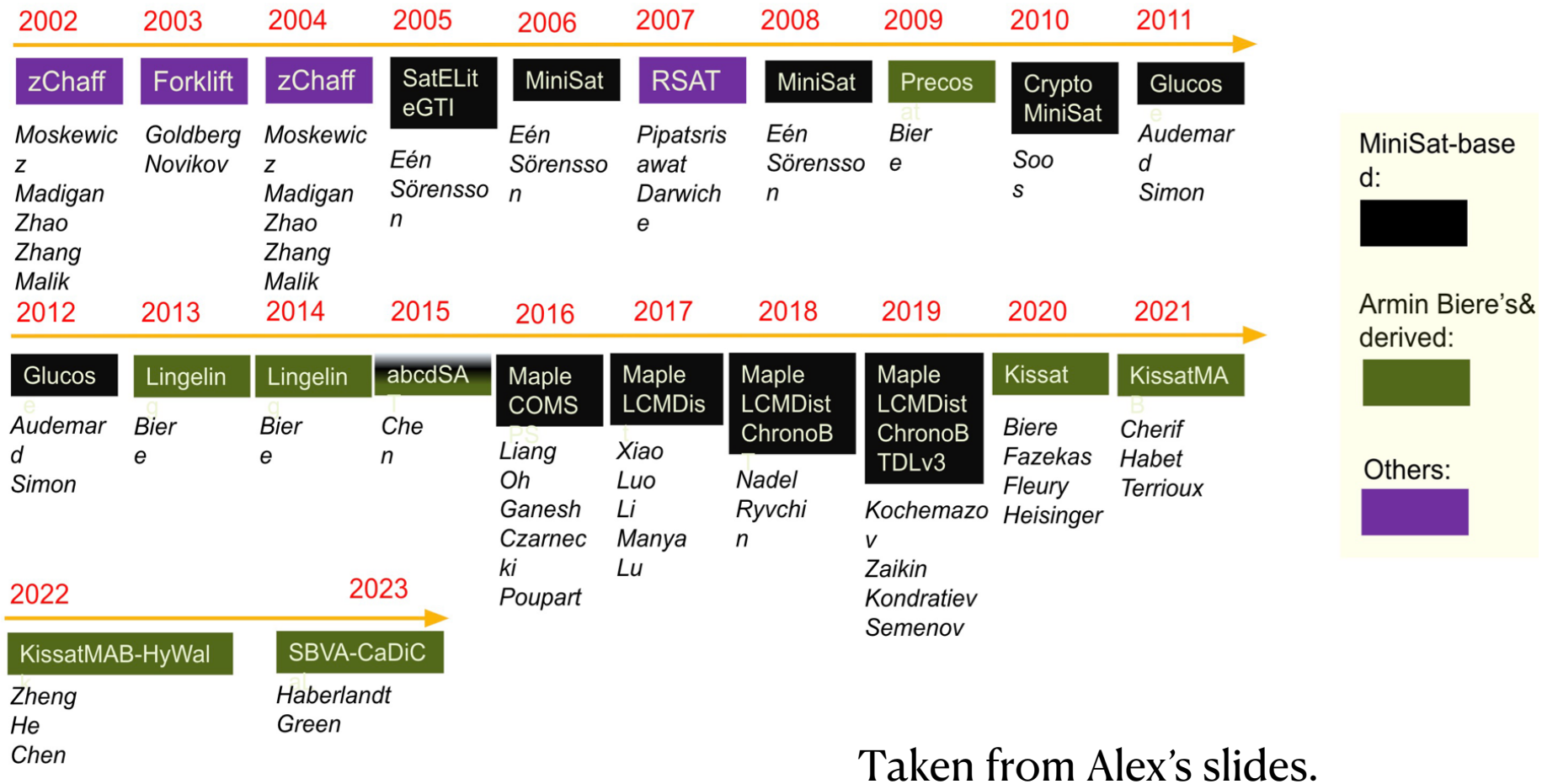
2. Decide(m, F): choose an unassigned variable in m and assign it a Boolean value.

3. AnalyzeConflict(m, F): returns a conflict clause learned using implication graph, and a decision level upto which the solver needs to backtrack.



Taken from Mate Soos 's slides.

SAT Competition & Race Winners (CNF & Appl. & Seq. & Non-incr. & All-inst.)



Taken from Alex's slides.

CDCL: Conflict Driven Clause Learning

1. UnitPropagation(m, F): applies unit propagation and extends m.

2. Decide(m, F): choose an unassigned variable in m and assign it a Boolean value.

Heuristics: which variables to pick, what value to assign?

3. ClauseLearning(m, F): returns a conflict clause learned using implication graph, and a decision level upto which the solver needs to backtrack.

Heuristics: how to learn a small conflict clause and unto which level to backtrack?

Heuristics: how to learn a small conflict clause and upto which level to backtrack?

AnalyzeConflict(m,F): some choices of clauses are found to be better than others.

Notations:

UIP (Unique Implication Point)

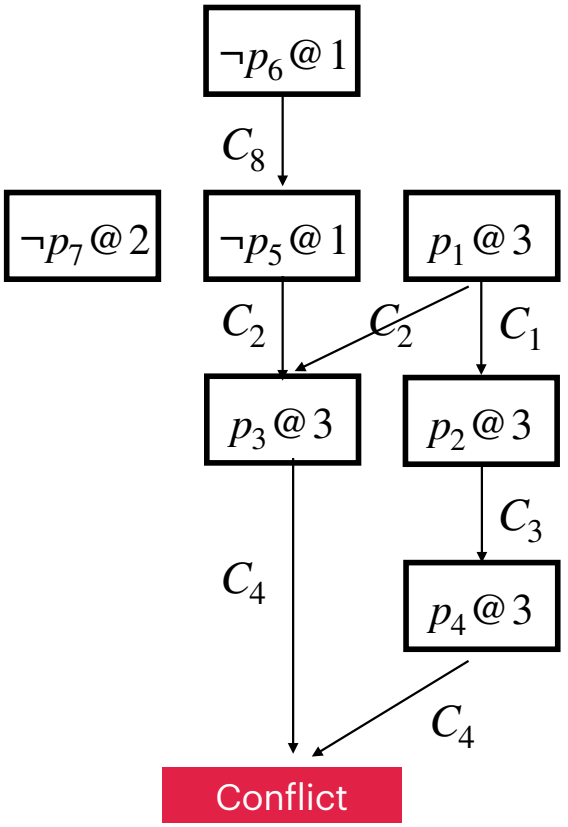
In an implication graph, node "$l@d$" is a UIP at decision level d if "$l@d$" occurs in each path from d^{th} decision literals to the conflict.

UIP points: In an implication graph, node *"$l@d$"* is a UIP at decision level d if *"$l@d$"* occurs in each path from d^{th} decision literals to the conflict.

UIP @ level 1:

UIP @ level 2:

UIP @ level 3:



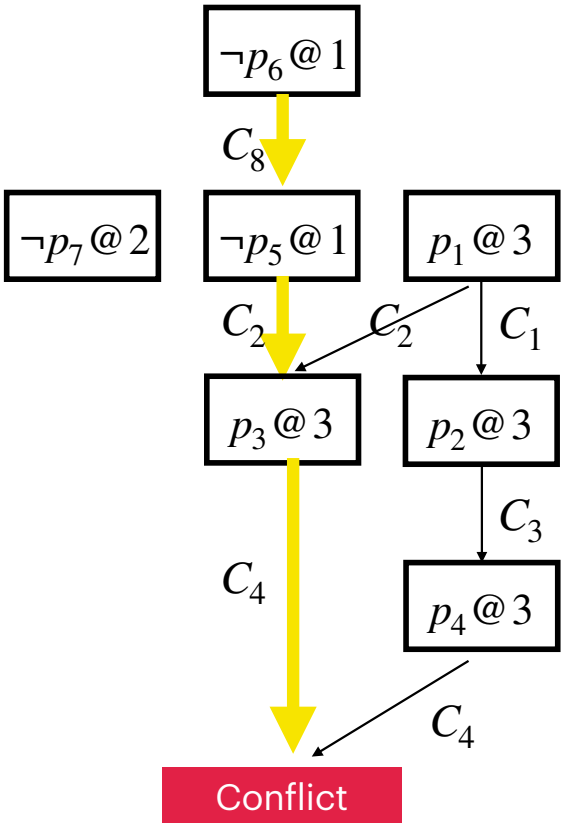
Implication Graph.

UIP points: In an implication graph, node *"$l@d$"* is a UIP at decision level d if *"$l@d$"* occurs in each path from d^{th} decision literals to the conflict.

UIP @ level 1: $\neg p_6@1, \neg p_5@1$

UIP @ level 2:

UIP @ level 3:



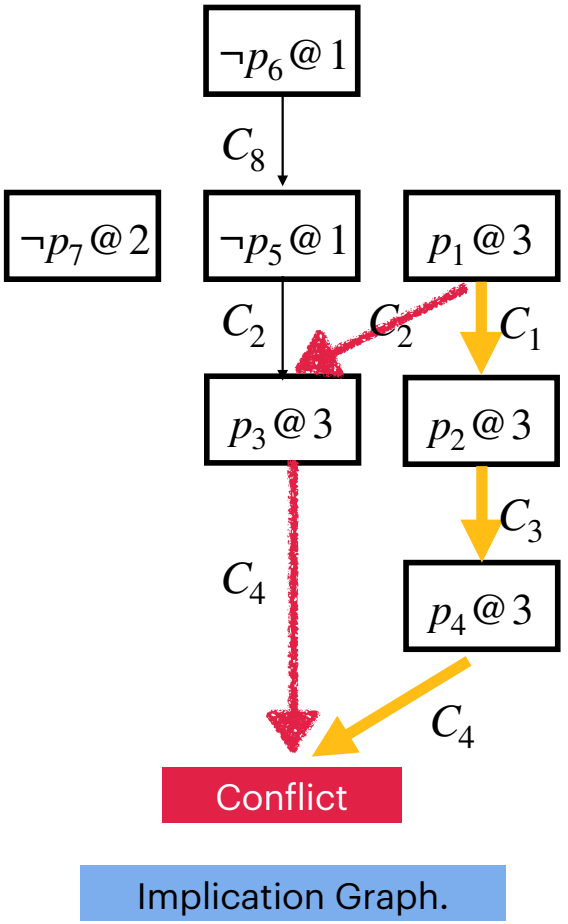
Implication Graph.

UIP points: In an implication graph, node *"$l@d$"* is a UIP at decision level d if *"$l@d$"* occurs in each path from d^{th} decision literals to the conflict.

UIP @ level 1: $\neg p_6@1, \neg p_5@1$

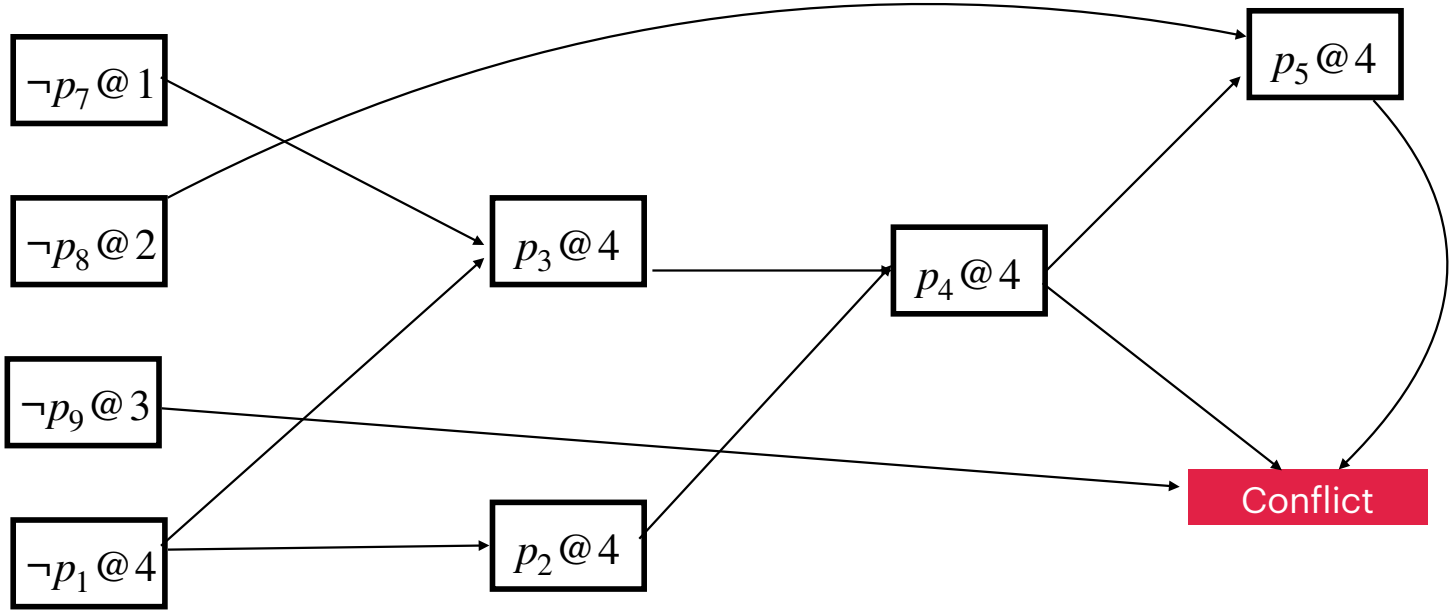
UIP @ level 2:

UIP @ level 3: $p_1@3$



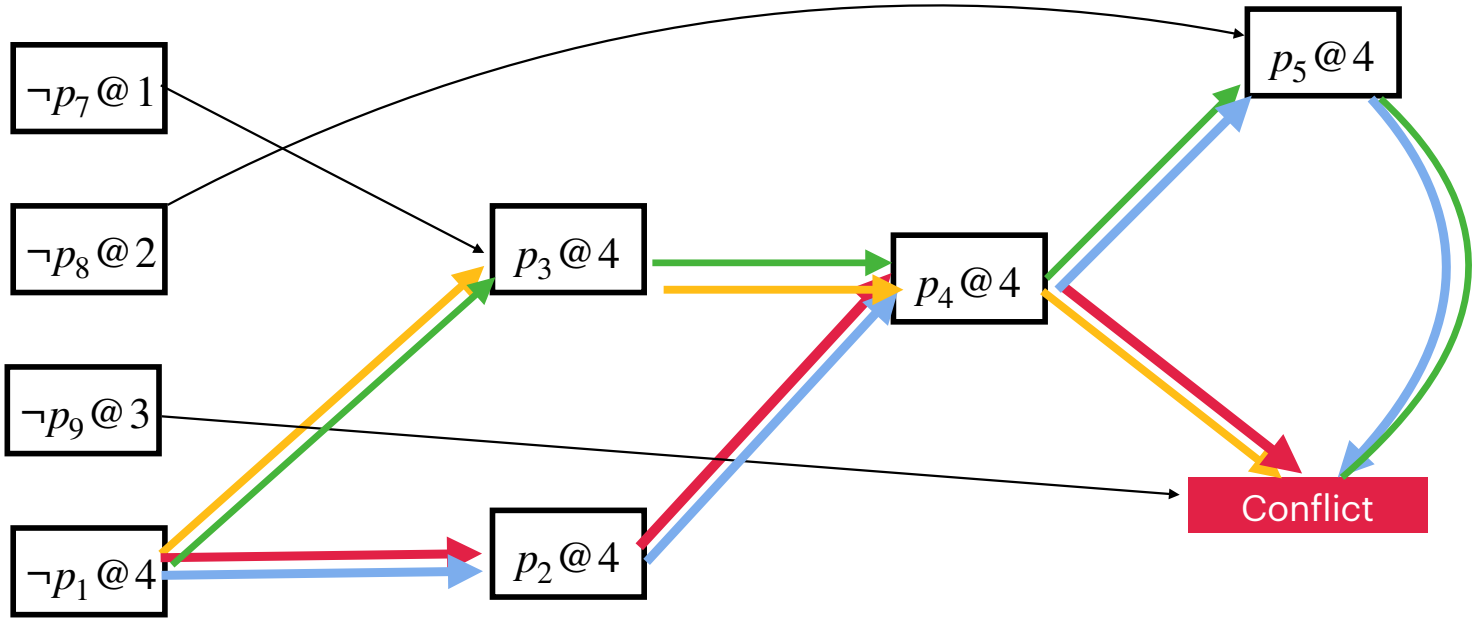
Implication Graph.

UIP points: In an implication graph, node *"$l@d$"* is a UIP at decision level d if *"$l@d$"* occurs in each path from d^{th} decision literals to the conflict.



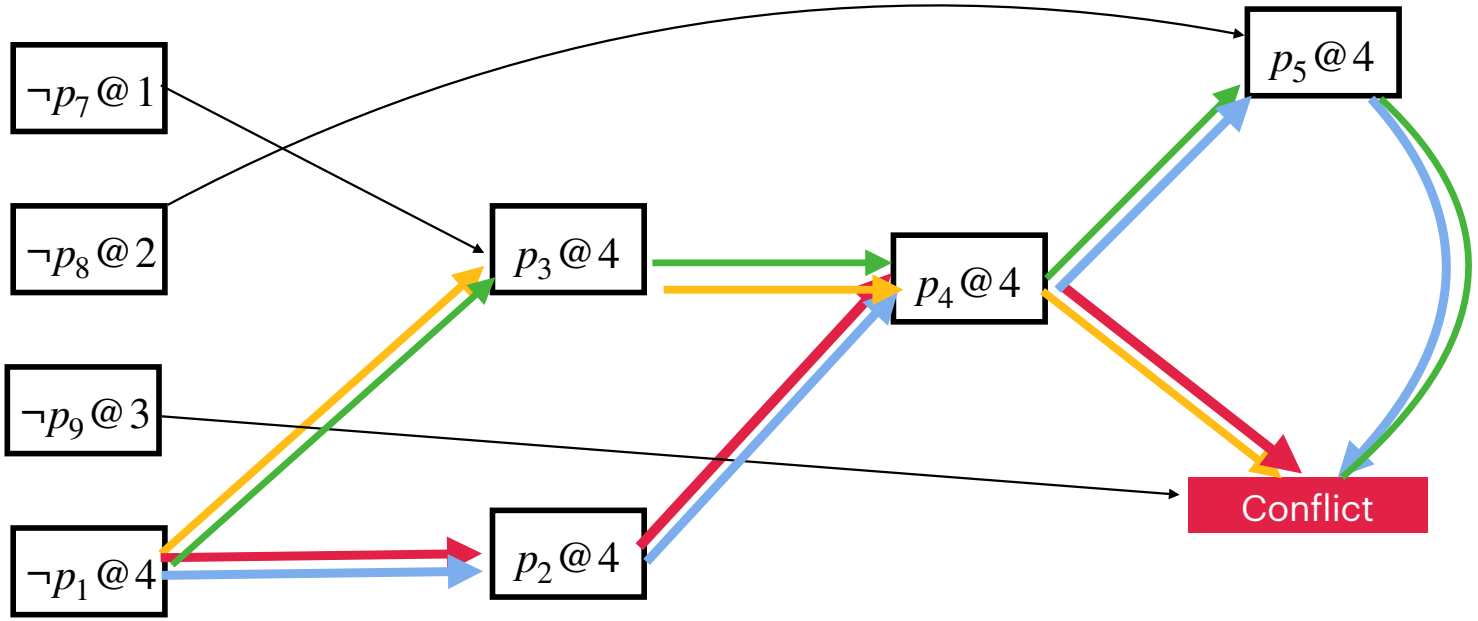
UIP @4 = ???

UIP points: In an implication graph, node *"$l@d$"* is a UIP at decision level d if *"$l@d$"* occurs in each path from d^{th} decision literals to the conflict.



UIP @4 = ???

UIP points: In an implication graph, node *"$l@d$"* is a UIP at decision level d if *"$l@d$"* occurs in each path from d^{th} decision literals to the conflict.



UIP $@4 = \neg p_1@4, p_4@4$

First UIP Point: $p_4@4$

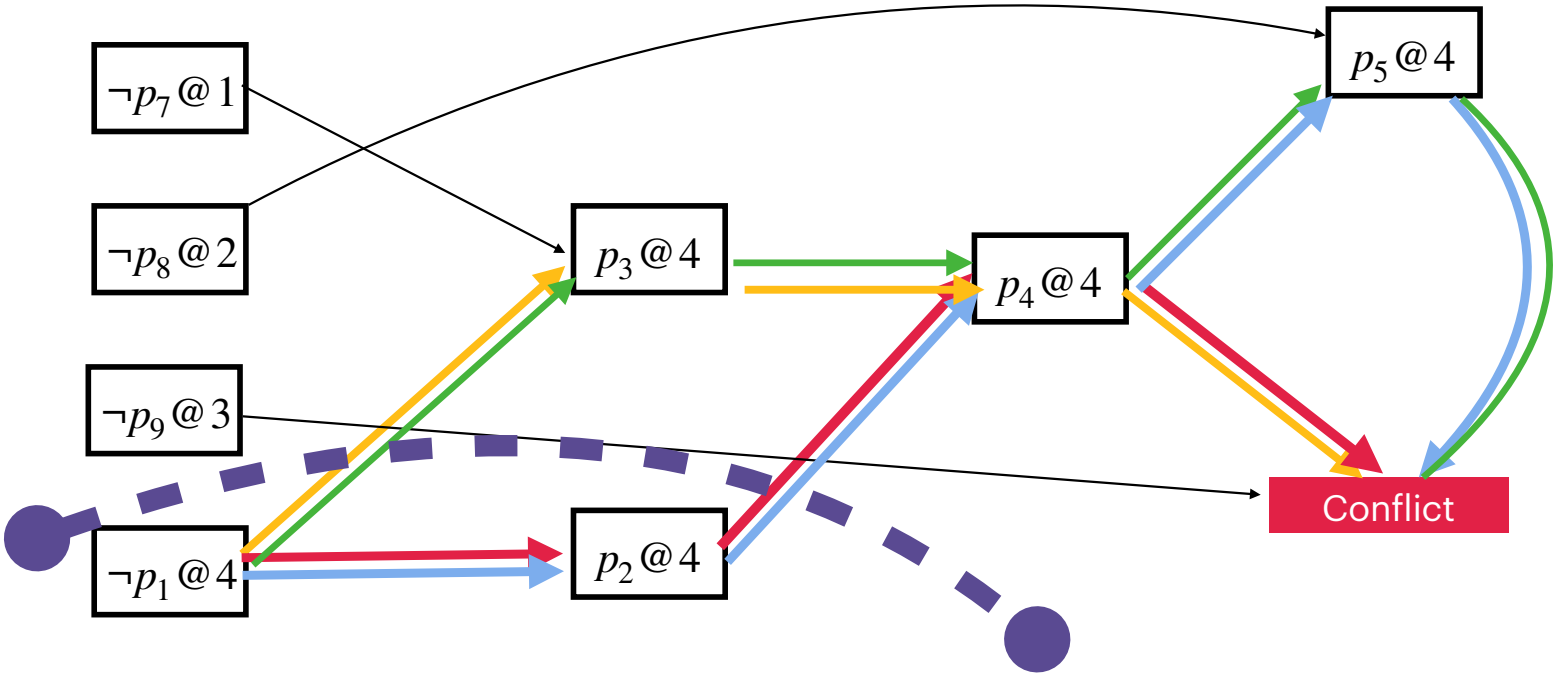
Last UIP Point: $\neg p_1@4$

UIP cuts to analyze conflicts:

If l is UIP, then corresponding UIP cut is (A,B) of the implication graph.
Where,

B contains all the successors of l from which there is a path to conflict.
A contains the rest.

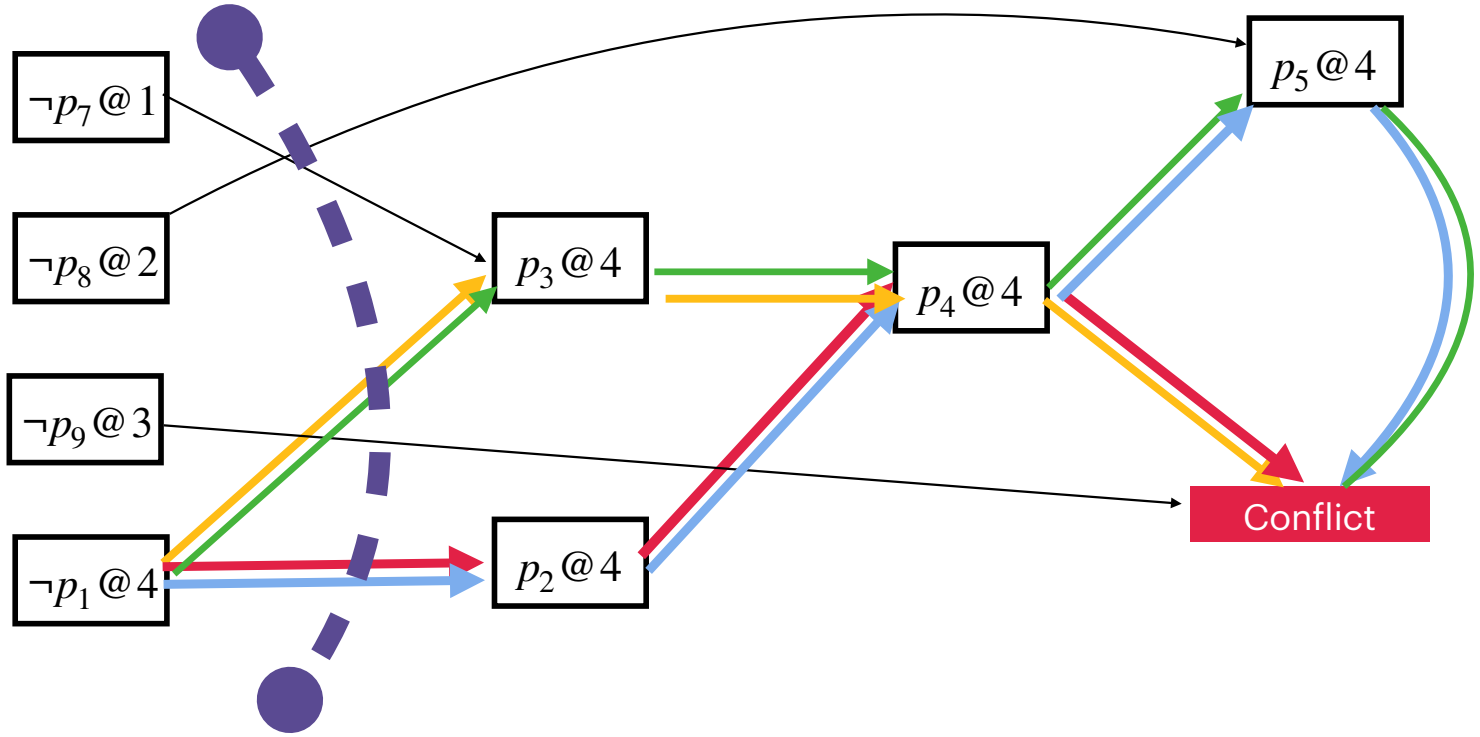
UIP cuts to analyze conflicts: If l is UIP, then corresponding UIP cut is (A,B) of the implication graph, where B contains all the successors of l from which there is a path to conflict, and A contains the rest.



UIP $@4 = \neg p_1 @ 4, p_4 @ 4$

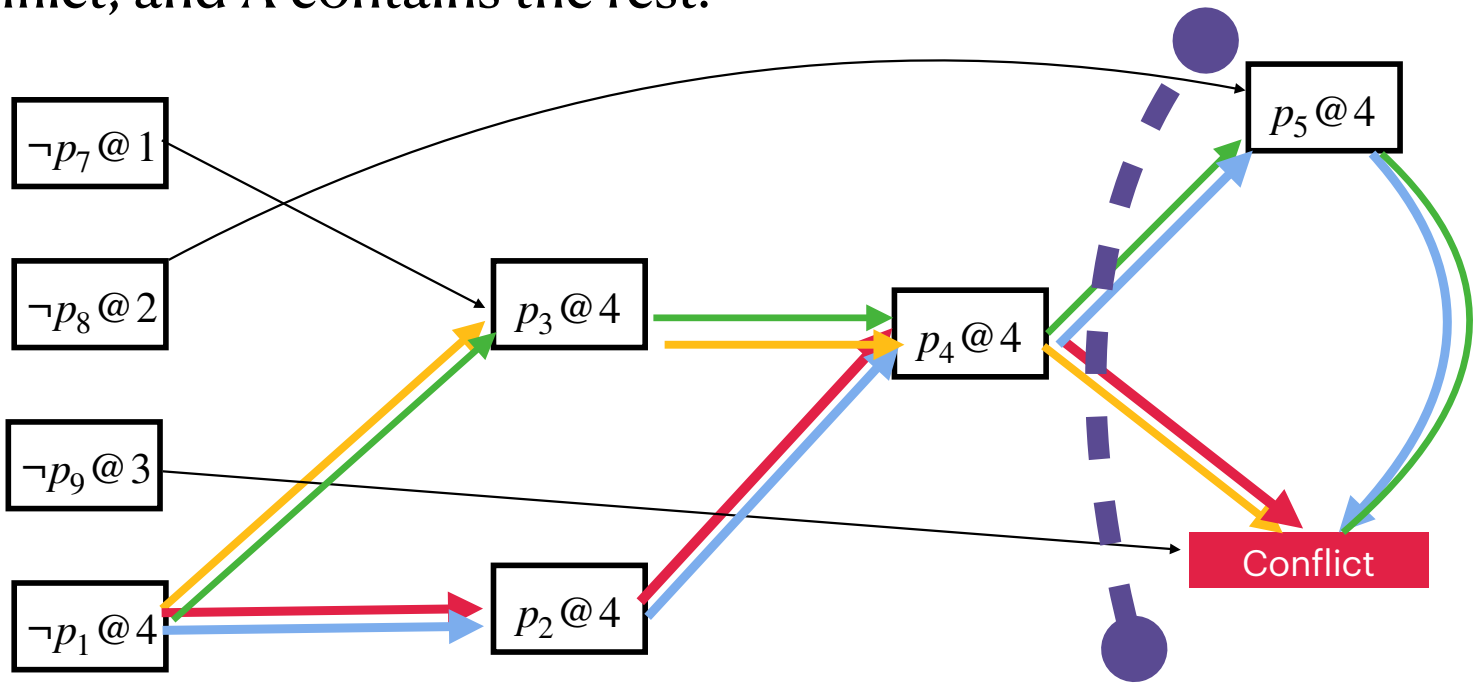
Is it a UIP cut?

UIP cuts to analyze conflicts: If l is UIP, then corresponding UIP cut is (A,B) of the implication graph, where B contains all the successors of l from which there is a path to conflict, and A contains the rest.



UIP $@4 = \neg p_1 @ 4, p_4 @ 4$ Is it a UIP cut? Yes, with respect to $\neg p_1 @ 4$

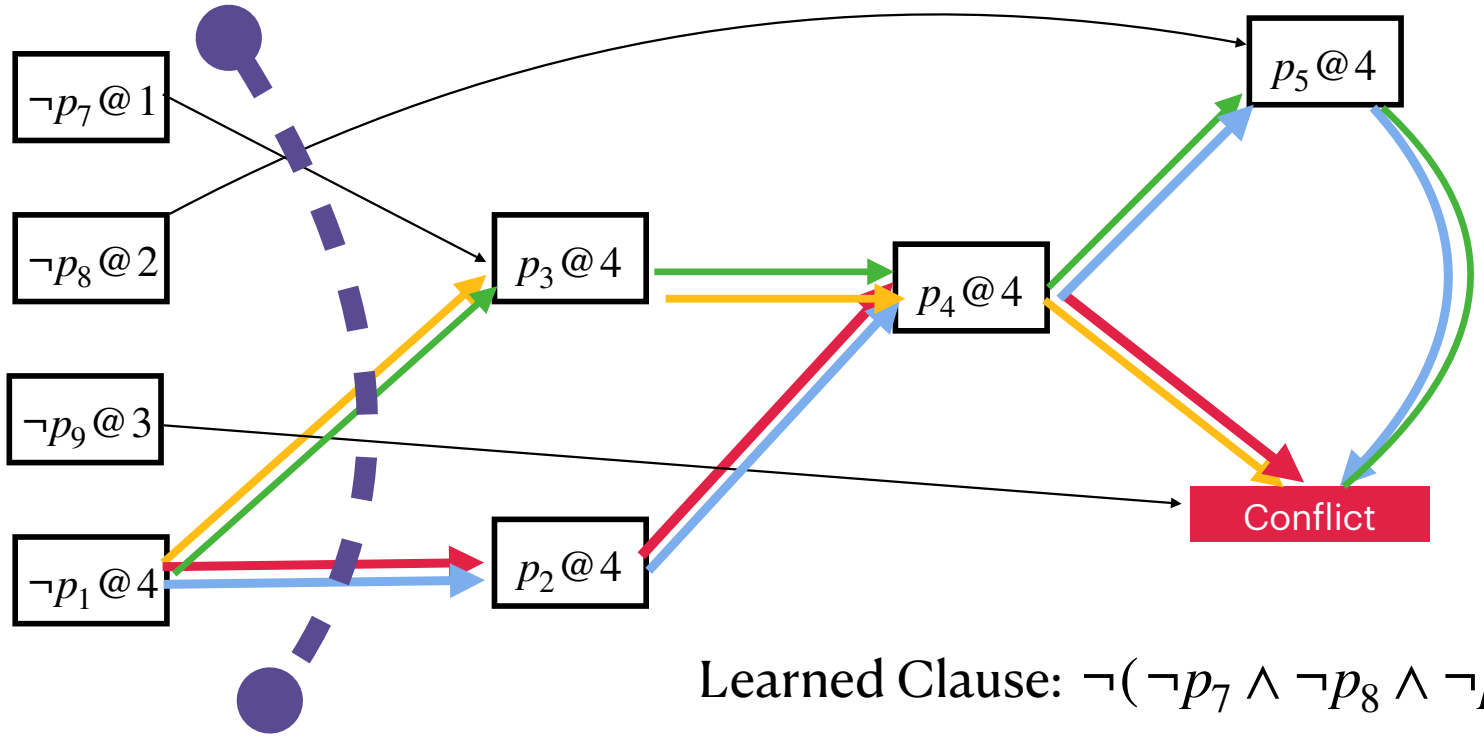
UIP cuts to analyze conflicts: If l is UIP, then corresponding UIP cut is (A,B) of the implication graph, where B contains all the successors of l from which there is a path to conflict, and A contains the rest.



UIP $@4 = \neg p_1 @ 4, p_4 @ 4$ Is it a UIP cut? Yes, with respect to $p_4 @ 4$

Learned Conflict Clause from UIP cut

The literals on the A side of the cut, which have an edge directed from A to B, form a clause. These literals are then negated and combined into a disjunction.

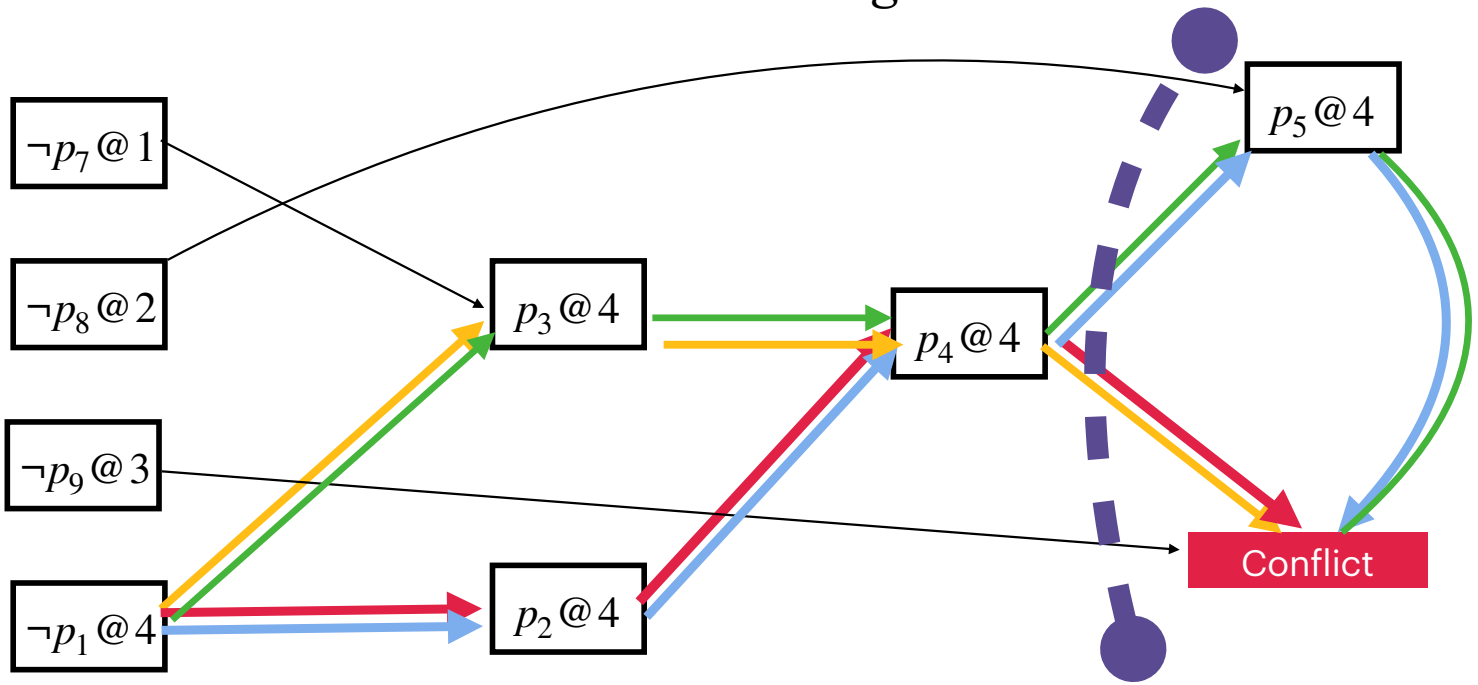


Learned Clause: $\neg(\neg p_7 \wedge \neg p_8 \wedge \neg p_9 \wedge \neg p_1)$

UIP $@4 = \neg p_1 @ 4, p_4 @ 4$

Learned Conflict Clause from UIP cut

The literals on the A side of the cut, which have an edge directed from A to B, form a clause. These literals are then negated and combined into a disjunction.



UIP $@4 = \neg p_1 @ 4, p_4 @ 4$ Learned Clause: $\neg(\neg p_8 \wedge p_4 \wedge \neg p_9)$

Heuristics: which variables to pick, what value to assign?

Variable ordering, Decision heuristics, Branching heuristics.

- # of variables occurrence in remaining unsatisfied clauses (different variants were studied in 90s).
- Dynamic heuristics:
    - Focus on variables which were useful recently in deriving learned clauses.
    - Can be interpreted as reinforcement learning.
    - VSIDS: Variable State Independent Decaying Sum.
- Look-ahead
    - Spent more time in selecting good variables.

VSIDS: Variable State Independent Decaying Sum

- Each literal l has a counter $S(l)$, initialized to zero.
- For every new clause $C = \{l_1, l_2, \ldots, l_n\}$, $S(l_i)$ is incremented if $l_i \in C$.
- The unassigned variable and polarity with highest counter is chosen.
- Ties are broken randomly.
- Periodically (once in 256 conflict), all counters are halved.

VSIDS: Variable State Independent Decaying Sum

Literals	Score
a	4
$\neg a$	5
b	3
$\neg b$	3
c	2
$\neg c$	3
d	2
$\neg d$	4
e	2
$\neg e$	6

.....

Initial value occurrences of “a” in formula F

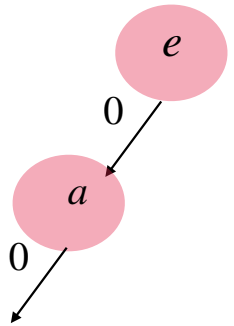
Count literal appearances in formula F

VSIDS: Variable State Independent Decaying Sum

Literals	Score
a	4
$\neg a$	5
b	3
$\neg b$	3
c	2
$\neg c$	3
d	2
$\neg d$	4
e	2
$\neg e$	6

.....

Initial value occurrences of “a” in formula F



.....

Conflict

Learned clause ($\neg h \vee a \vee c \vee \neg b \vee k$)

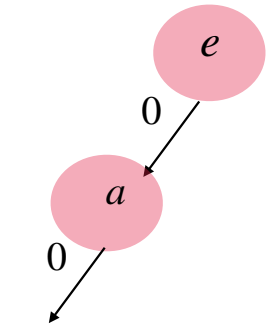
Count literal appearances in formula F

VSIDS: Variable State Independent Decaying Sum

Literals	Score
a	4 +1
$\neg a$	5
b	3+1
$\neg b$	3
c	2+1
$\neg c$	3
d	2
$\neg d$	4
e	2
$\neg e$	6

.....

Initial value occurrences of “a” in formula F



.....

Conflict

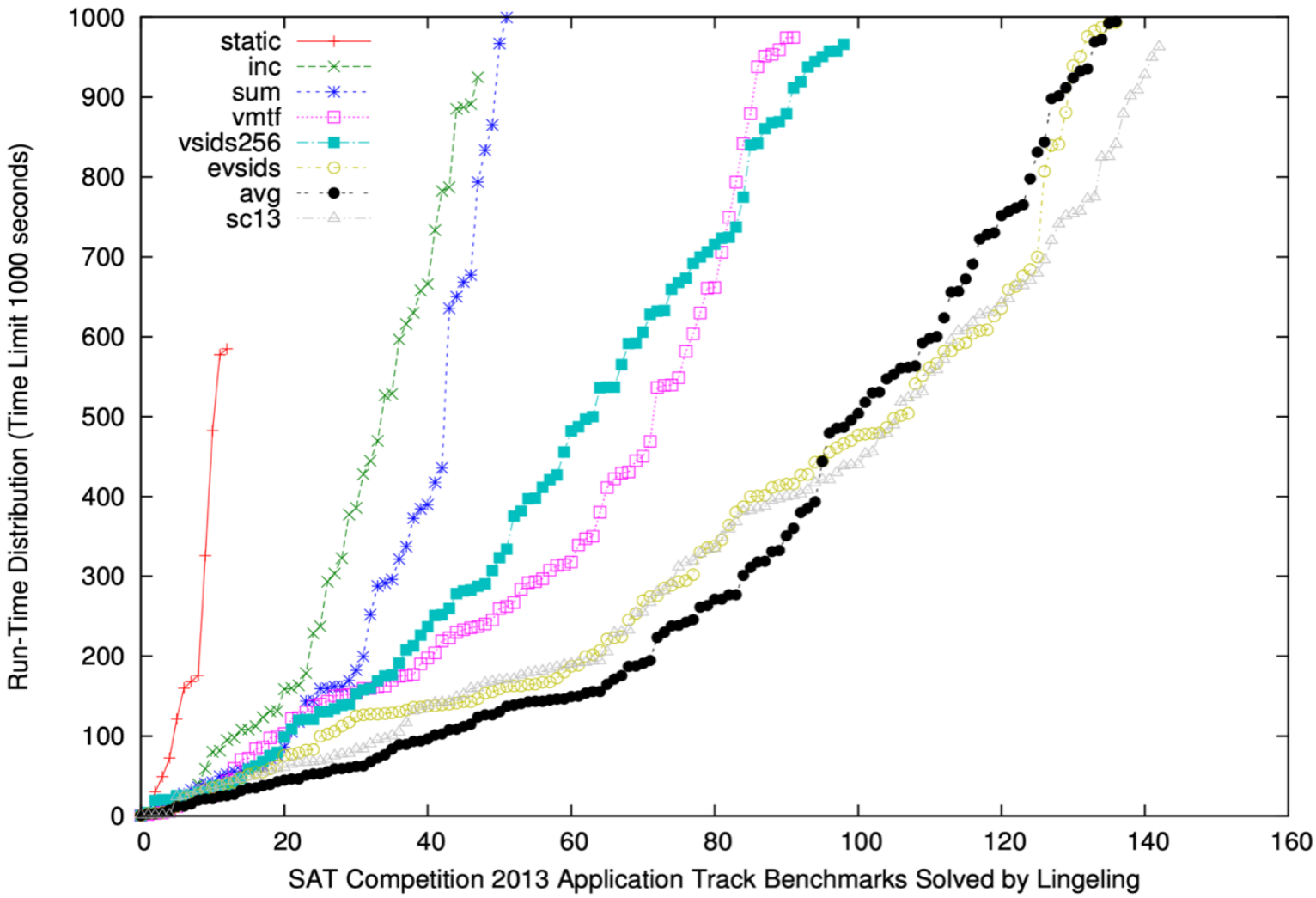
Learned clause $(\neg h \vee a \vee c \vee \neg b \vee k)$

Count literal appearances in formula F

VSIDS: Variable State Independent Decaying Sum

Why it was a breakthrough?

- Pre-chaff static heuristics — go over all clauses that are not satisfied and compute some function $f(a)$ for each literal “a”.
- VSLDS
  - Extremely low overhead.
  - Dynamic & local (conflict driven).
  - Focuses the search to learn from the local context.


static
inc
sum
vmtf
vsids256
evsids
avg
sc13
Run-Time Distribution (Time Limit 1000 seconds)
SAT Competition 2013 Application Track Benchmarks Solved by Lingeling
0
100
200
300
400
500
600
700
800
900
1000
0
20
40
60
80
100
120
140
160

Course Webpage

Thanks!