# Cube and Conquer

## Guiding CDCL SAT Solvers by Lookaheads

Marijn Heule[1,2], Oliver Kullmann[3], Siert Wieringa[4] and Armin Biere[1]

[1] Johannes Kepler University, Linz, Austria

[2] Delft University of Technology, Delft, The Netherlands

[3] Swansea University, United Kingdom

[4] Aalto University, Helsinki, Finland

December 6, 2011 @ Haifa Verification Conference 2011

# Motivation

Context:

- Huge performance boost of CDCL solvers in the last decade
- CDCL solvers have become a crucial tool, e.g. in Formal Verification

Challenges:

- CDCL is not strong on small hard combinatorial problems
- CDCL is hard to parallelize effectively

CDCL: Conflict-Driven Clause Learning

# Satisfiability problem

Satisfiability (SAT) problem:

- Given a formula in Conjunctive Normal Form, is there a truth assignment to the Boolean variables satisfying all clauses?
- clause: $(l_a \vee l_b \vee l_c)$
- cube: $(l_d \wedge l_e \wedge l_f)$

Major SAT solver architectures:

- Conflict-Driven Clause Learning
- Lookahead

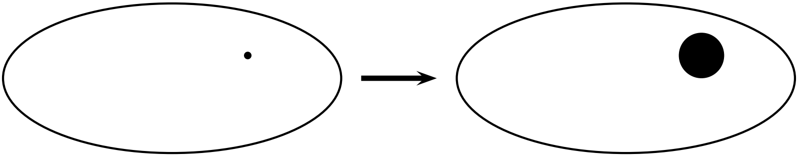
# Conflict-Driven Clause Learning solvers

Highlights:

- goal: find small effective conflict clauses
- decisions: assign variables that occur in recent conflicts
- strength: powerful on "easy" problems

Ideal CDCL situation:

- hit a conflict that can be generalized / analyzed to a small clause



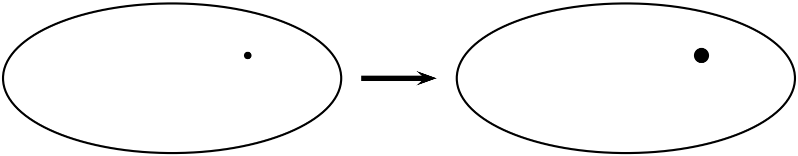
# Conflict-Driven Clause Learning solvers

Highlights:

- goal: find small effective conflict clauses
- decisions: assign variables that occur in recent conflicts
- strength: powerful on "easy" problems

General CDCL situation:

- hit a conflict that can be generalized / analyzed to a large clause



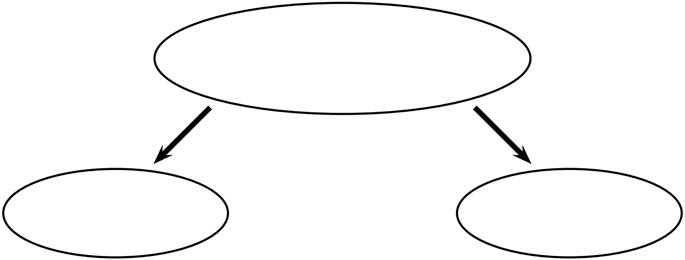
# Lookahead solvers

Highlights:

- goal: construct a small binary search tree
- decisions: assign variables that cause a large reduction
- strength: powerful on small hard problems

Ideal lookahead situation:

- split the search space into two equally large but smaller parts



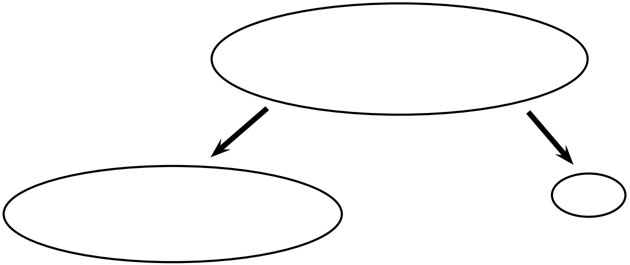
# Lookahead solvers

Highlights:

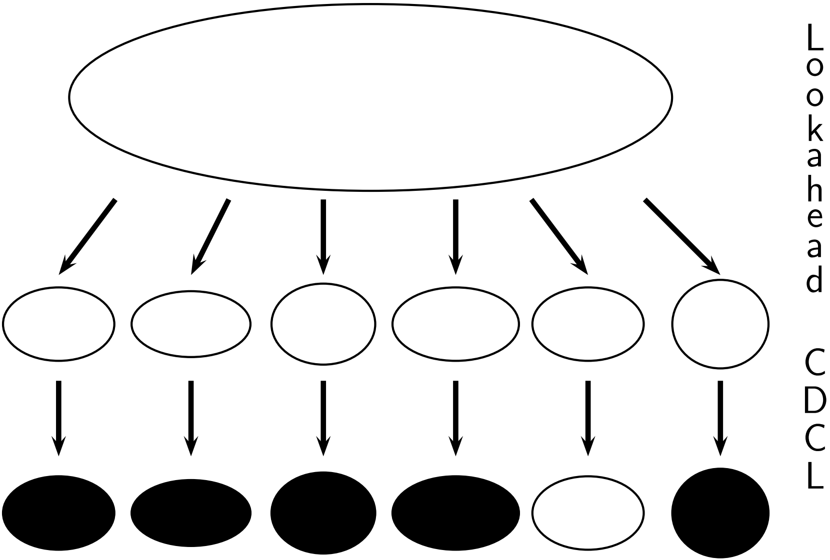
- goal: construct a small binary search tree
- decisions: assign variables that cause a large reduction
- strength: powerful on small hard problems

General lookahead situation:

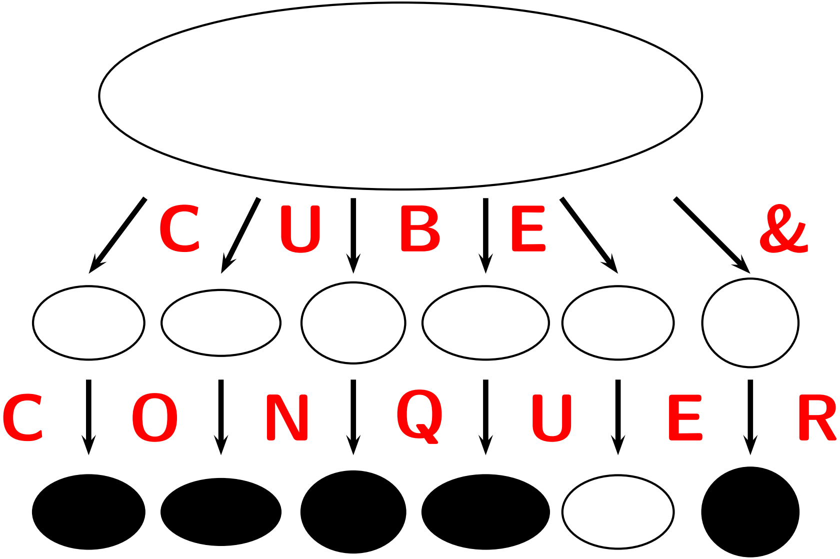
- the search space is split into a large and a small part



# Best of both worlds: Combining Lookahead and CDCL



# Best of both worlds: Cube and Conquer



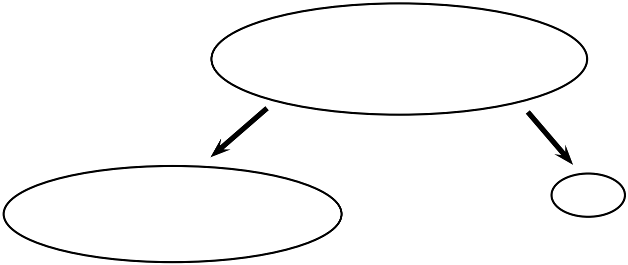
# Cube: key observation / contribution

Split until the problem becomes easy

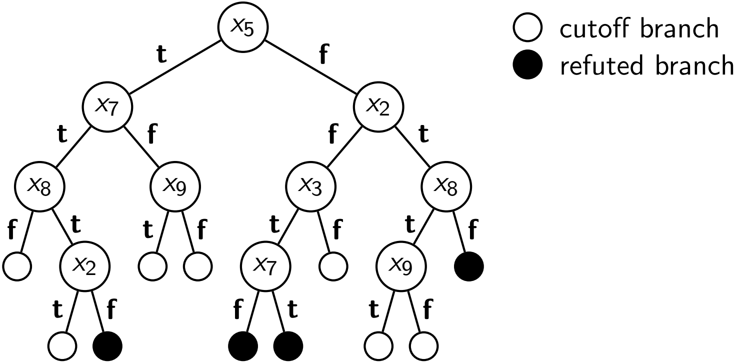
- do not have a fixed cut off depth
- determine hardness by number of assigned variables
- create many thousands or even millions of cubes

General lookahead situation:

- the search space is split into a large and a small part



# Cube: example




$F_1 := F \wedge (x_5 \wedge x_7 \wedge \neg x_8)$

$F_2 := F \wedge (x_5 \wedge x_7 \wedge x_8 \wedge x_2)$

$F_3 := F \wedge (x_5 \wedge \neg x_7 \wedge x_9)$

$F_4 := F \wedge (x_5 \wedge \neg x_7 \wedge \neg x_9)$

$F_5 := F \wedge (\neg x_5 \wedge \neg x_2 \wedge \neg x_3)$

$F_6 := F \wedge (\neg x_5 \wedge x_2 \wedge x_8 \wedge x_9)$

$F_7 := F \wedge (\neg x_5 \wedge x_2 \wedge x_8 \wedge \neg x_9)$

# Cube: pseudo-code (1)

*Cube* (CNF $F$, DNF $\mathcal{A}$, CNF $\mathcal{C}$, dec. lits. $\varphi_{\text{dec}}$, imp. lits. $\varphi_{\text{imp}}$)

$\theta := 1.05 \cdot \theta$

$\langle F, \varphi_{\text{imp}} \rangle :=$ *lookahead_simplify_and_learn* ($F$, $\varphi_{\text{dec}}$, $\varphi_{\text{imp}}$)

**if** $\varphi_{\text{dec}} \cup \varphi_{\text{imp}}$ falsify a clause in $F$ **or** $|\varphi_{\text{dec}}| > 20$ **then** $\theta := 0.7 \cdot \theta$

**if** $\varphi_{\text{dec}} \cup \varphi_{\text{imp}}$ falsify a clause in $F$ **then return** $\langle \mathcal{A}, \mathcal{C} \cup \{\neg \varphi_{\text{dec}}\} \rangle$

**if** cutoff heuristic is triggered **then return** $\langle \mathcal{A} \cup \{\varphi_{\text{dec}}\}, \mathcal{C} \rangle$

$l_{\text{dec}} :=$ *lookahead_decide* ($F$, $\varphi_{\text{dec}}$, $\varphi_{\text{imp}}$)

$\langle \mathcal{A}, \mathcal{C} \rangle :=$ *Cube* $(F, \mathcal{A}, \mathcal{C}, \varphi_{\text{dec}} \cup \{l_{\text{dec}}\}, \varphi_{\text{imp}})$

**return** *Cube* $(F, \mathcal{A}, \mathcal{C}, \varphi_{\text{dec}} \cup \{\neg l_{\text{dec}}\}, \varphi_{\text{imp}})$

# Cube: pseudo-code (2)

*Cube* (CNF $F$, DNF $\mathcal{A}$, CNF $\mathcal{C}$, dec. lits. $\varphi_{\rm dec}$, imp. lits. $\varphi_{\rm imp}$)

$\theta := 1.05 \cdot \theta$

$\langle F, \varphi_{\rm imp} \rangle :=$ *lookahead_simplify_and_learn* ($F$, $\varphi_{\rm dec}$, $\varphi_{\rm imp}$)

**if** $\varphi_{\rm dec} \cup \varphi_{\rm imp}$ falsify a clause in $F$ **or** $|\varphi_{\rm dec}| > 20$ **then** $\theta := 0.7 \cdot \theta$

**if** $\varphi_{\rm dec} \cup \varphi_{\rm imp}$ falsify a clause in $F$ **then return** $\langle \mathcal{A}, \mathcal{C} \cup \{\neg\varphi_{\rm dec}\} \rangle$

**if** $|\varphi_{\rm dec}| \cdot |\varphi_{\rm dec} \cup \varphi_{\rm imp}| > \theta \cdot |\text{vars}(F)|$ **then return** $\langle \mathcal{A} \cup \{\varphi_{\rm dec}\}, \mathcal{C} \rangle$

$l_{\rm dec} :=$ *lookahead_decide* ($F$, $\varphi_{\rm dec}$, $\varphi_{\rm imp}$)

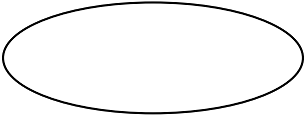
$\langle \mathcal{A}, \mathcal{C} \rangle :=$ *Cube* $(F, \mathcal{A}, \mathcal{C}, \varphi_{\rm dec} \cup \{l_{\rm dec}\}, \varphi_{\rm imp})$

**return** *Cube* $(F, \mathcal{A}, \mathcal{C}, \varphi_{\rm dec} \cup \{\neg l_{\rm dec}\}, \varphi_{\rm imp})$

# Conquer: describing cubes

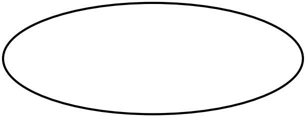
How much information to send to the CDCL solver?

- Only the decisions



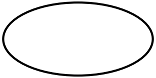
$\wedge \ \varphi_{\text{dec}}$

- The full assignment (including failed literals)



$\wedge \ \varphi_{\text{dec}} \ \wedge \ \varphi_{\text{imp}}$

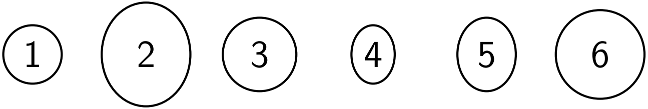
- The simplified formula (including local learnt clauses)



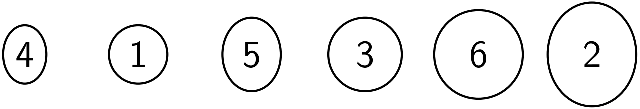
# Conquer: ordering cubes

What is the optimal order to solve the cubes?

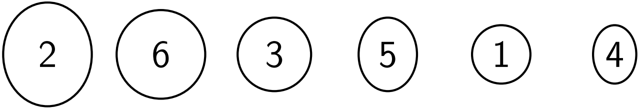
- Depth-first search (in lookahead order)




- Solves cubes with increasing (approximated) search space




- Solves cubes with decreasing (approximated) search space



# Conquer: pseudo-code

```
Conquer (CDCL solver S, CNF formula F, DNF of assumptions A)
  S.Load (F)
  while A is not empty do
    get a cube c from A and remove c from A
    if S.SolveWithAssumptions (c) = satisfiable then
      return satisfiable
    S.AnalyzeFinal ()
    S.ResetClauseDeletionPolicy ()
  return unsatisfiable
```

# Conquer: parallel solving

Strategies to solve cubes in parallel:

1. cores solve different cubes in parallel
2. cores solve the same cube in parallel
3. start with (1) till no new cubes are available, continue with (2)

What to share between cores?

- nothing, so hardly communication required (only ask / receive cubes)
- sharing the *AnalyzeFinal* clauses (maybe only to master)
- sharing the short conflict clauses, units (maybe also binaries)

# Results: two experiments

## $1^{st}$ experiment: single core on Van der Waerden numbers

- hard combinatorial problem in Ramsey Theory
- comparison with the best solver for each instance
- cube solver: `OKsolver`
- conquer solver: `minisat`

## $2^{nd}$ experiment: multi core on challenging applications

- unsolved application instances from the SAT09 benchmarks
- comparison with the best parallel solvers
- cube solver: `march`
- conquer solver: `lingeling`

# Results: palindromic Van der Waerden numbers

- $k_1$ : arithmetic progression of first set;
- $k_2$ : arithmetic progression of second set;
- $n$ : number of elements to partition;
- best solver : time of fastest sequential solver;
- $D$ : cut off depth.

| $k_1$ | $k_2$ | $n$ | #cls | ? | best solver | $D$ | #cubes | C&C |
|---|---|---|---|---|---|---|---|---|
| 3 | 25 | 586 | 45779 | S | $\sim$ 13 days | 45 | 9120 | 6.5 hours |
| 3 | 25 | 607 | 49427 | U | $\sim$ 13 days | 45 | 13462 | 2 days |
| 4 | 12 | 387 | 15544 | S | $>$ 14 days | 30 | 132131 | 2 days |
| 4 | 12 | 394 | 15889 | U | $>$ 14 days | 34 | 147237 | 8 hours |
| 5 | 8 | 312 | 9121 | S | 3.5 days | 20 | 2248 | 5 hours |
| 5 | 8 | 313 | 9973 | U | 53 days | 20 | 87667 | 40 hours |

# Results: parallel SAT solving

Portfolio solvers:

- run multiple versions of the same solver (different seeds)
- share short conflict clauses such as units
- solver `pLingeling` (`pLing`), on a 12-core machine

Grid based SAT solving approach:

- run solvers with different cubes on a grid
- grid constraints: limited communication, possible delay and timeout
- solver `PartitionTree` (`PTree`) on a grid, up to 60 jobs in parallel

# Results: hard application benchmarks

| Benchmark | ? | #cubes | I total | II total | II 12-core | pLing 12-core | PTree 60-core |
|---|---|---|---|---|---|---|---|
| 9dlx_vliw_at_b_iq8 | U | 121 | 150 | — | — | **3256** | — |
| 9dlx_vliw_at_b_iq9 | U | 100 | 179 | — | — | **5164** | — |
| AProVE07-25 | U | 84247 | 89 | 100340 | **8690** | — | 9967 |
| dated-5-19-u | U | 57716 | 418 | 3214 | **1451** | 4465 | 2522 |
| eq.atree.braun.12 | U | 86541 | 85 | 3261 | **273** | — | 4691 |
| eq.atree.braun.13 | U | 81313 | 77 | 18165 | **1517** | — | 9972 |
| gss-24-s100 | S | 18237 | 48 | 4975 | **415** | 2930 | 3492 |
| gss-26-s100 | S | 19455 | 57 | 37259 | **3108** | 18173 | 10347 |
| gus-md5-14 | U | 60102 | 961 | — | — | — | **13890** |
| ndhf_xits_09_UNS | U | 37358 | 82 | 71096 | 12041 | — | **9583** |
| rbcl_xits_09_UNK | U | 54669 | 132 | 94911 | 11542 | — | **9819** |
| rpoc_xits_09_UNS | U | 30681 | 114 | 48028 | **8366** | — | 8635 |
| sortnet-8-ipc5-h19 | S | 724 | 153 | 48668 | 4067 | **2700** | 4304 |
| total-10-17-u | U | 9192 | 288 | 5638 | 4517 | **3672** | 4447 |
| total-5-15-u | U | 14914 | 215 | — | — | — | **18670** |

# Results: hard application benchmarks

| Benchmark | ? | #cubes | I total | II total | II 12-core | pLing 12-core | PTree 60-core |
|---|---|---|---|---|---|---|---|
| 9dlx_vliw_at_b_iq8 | U | 121 | 150 | — | — | **3256** | — |
| 9dlx_vliw_at_b_iq9 | U | 100 | 179 | — | — | **5164** | — |
| AProVE07-25 | U | 84247 | 89 | 100340 | **8690** | — | 9967 |
| dated-5-19-u | U | 57716 | 418 | 3214 | **1451** | 4465 | 2522 |
| eq.atree.braun.12 | U | 86541 | 85 | 3261 | **273** | — | 4691 |
| eq.atree.braun.13 | U | 81313 | 77 | 18165 | **1517** | — | 9972 |
| gss-24-s100 | S | 18237 | 48 | 4975 | **415** | 2930 | 3492 |
| gss-26-s100 | S | 19455 | 57 | 37259 | **3108** | 18173 | 10347 |
| gus-md5-14 | U | 60102 | 961 | — | — | — | **13890** |
| ndhf_xits_09_UNS | U | 37358 | 82 | 71096 | 12041 | — | **9583** |
| rbcl_xits_09_UNK | U | 54669 | 132 | 94911 | 11542 | — | **9819** |
| rpoc_xits_09_UNS | U | 30681 | 114 | 48028 | **8366** | — | 8635 |
| sortnet-8-ipc5-h19 | S | 724 | 153 | 48668 | 4067 | **2700** | 4304 |
| total-10-17-u | U | 9192 | 288 | 5638 | 4517 | **3672** | 4447 |
| total-5-15-u | U | 14914 | 215 | — | — | — | **18670** |

# Results: hard application benchmarks

| Benchmark | ? | #cubes | I total | II total | II 12-core | pLing 12-core | PTree 60-core |
|---|---|---|---|---|---|---|---|
| 9dlx_vliw_at_b_iq8 | U | 121 | 150 | — | — | **3256** | — |
| 9dlx_vliw_at_b_iq9 | U | 100 | 179 | — | — | **5164** | — |
| AProVE07-25 | U | 84247 | 89 | 100340 | **8690** | — | 9967 |
| dated-5-19-u | U | 57716 | 418 | 3214 | **1451** | 4465 | 2522 |
| eq.atree.braun.12 | U | 86541 | 85 | 3261 | **273** | — | 4691 |
| eq.atree.braun.13 | U | 81313 | 77 | 18165 | **1517** | — | 9972 |
| gss-24-s100 | S | 18237 | 48 | 4975 | **415** | 2930 | 3492 |
| gss-26-s100 | S | 19455 | 57 | 37259 | **3108** | 18173 | 10347 |
| gus-md5-14 | U | 60102 | 961 | — | — | — | **13890** |
| ndhf_xits_09_UNS | U | 37358 | 82 | 71096 | 12041 | — | **9583** |
| rbcl_xits_09_UNK | U | 54669 | 132 | 94911 | 11542 | — | **9819** |
| rpoc_xits_09_UNS | U | 30681 | 114 | 48028 | **8366** | — | 8635 |
| sortnet-8-ipc5-h19 | S | 724 | 153 | 48668 | 4067 | **2700** | 4304 |
| total-10-17-u | U | 9192 | 288 | 5638 | 4517 | **3672** | 4447 |
| total-5-15-u | U | 14914 | 215 | — | — | — | **18670** |

# Conclusions

Cube and Conquer:

- effective combining lookahead and CDCL
- many thousands or even million of cubes
- natural to parallelize

Future work, online scheduling:

- adjust heuristics based on *AnalyzeFinal*
- communication between solvers
- all-in CDCL

# Cube and Conquer

## Guiding CDCL SAT Solvers by Lookaheads

Marijn Heule[1,2], Oliver Kullmann[3], Siert Wieringa[4] and Armin Biere[1]

[1]Johannes Kepler University, Linz, Austria

[2]Delft University of Technology, Delft, The Netherlands

[3]Swansea University, United Kingdom

[4]Aalto University, Helsinki, Finland

December 6, 2011 @ Haifa Verification Conference 2011