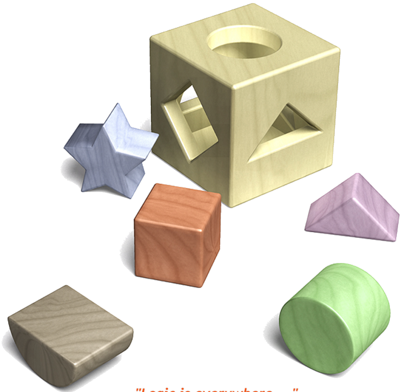
SAT Solving – Algorithms

Steffen Hölldobler and Norbert Manthey
International Center for Computational Logic
Technische Universität Dresden
Germany

- **DPLL**
- **CDCL**
- **A solving abstraction**



"Logic is everywhere ..."

Warm Up

- **Used programming languages**

Warm Up

- Used programming languages
- Size of implemented projects

Warm Up

- **Used programming languages**
- **Size of implemented projects**
- **Parallel computing (multi-core, GPGPU, cluster)**

Warm Up

- Used programming languages
- Size of implemented projects
- Parallel computing (multi-core, GPGPU, cluster)
- Interest in computer architecture

Revision

- **Used Data Types**
- **Semantics**

Formulas and Interpretations

- Let F be a formulas an I be an interpretation
- I can
  - **satisfy** F, if $F|_I \equiv \top$
  - **falsify** F, if $F|_I \equiv \bot$

Formulas and Interpretations

- Let F be a formulas an I be an interpretation
- I can
  - satisfy F, if $F|_I \equiv \top$
  - falsify F, if $F|_I \equiv \bot$
- A formula can be
  - unsatisfiable, $F \equiv \bot$
  - satisfiable
  - tautologic, $F \equiv \top$

Formulas and Interpretations

- Let F be a formulas an I be an interpretation
- I can
  - satisfy F, if $F|_I \equiv \top$
  - falsify F, if $F|_I \equiv \bot$
- A formula can be
  - unsatisfiable, $F \equiv \bot$
  - satisfiable
  - tautologic, $F \equiv \top$
- Property: $F \equiv \top$, then $\neg F \equiv \bot$.

Clauses and Conjunctive Normal Forms

- **Definition**
  - A clause is a generalized disjunction $[L_1, \ldots, L_n]$, $n \geq 0$, where every L_i, $1 \leq i \leq n$, is a literal
  - A clause is a unit clause if it contains precisely one literal
  - A clause is a binary clause if it contains precisely two literals

Clauses and Conjunctive Normal Forms

- **Definition**
  - A **clause** is a generalized disjunction $[L_1, \ldots, L_n]$, $n \geq 0$, where every L_i, $1 \leq i \leq n$, is a literal
  - A clause is a **unit clause** if it contains precisely one literal
  - A clause is a **binary clause** if it contains precisely two literals
- **Definition**
  - A formula is in **conjunctive normal form** (**clause form**, **CNF**) **iff** it is of the form $\langle C_1, \ldots, C_m \rangle$, $m \geq 0$, and every C_j, $1 \leq j \leq m$, is a clause
- Implementation and **working assumptions**
  - A clause is an array of literals
    - Maintained to be a **set** of literals (no duplicates)
    - Clauses are **no tautologies** (excluded during parsing)
  - A formula is an array of (pointers/references to) clauses
    - Maintained to be a **multi set**

Propositional Resolution

- **Remind: clauses are considered to be sets**
- **Definition** Let C_1 be a clause containing L and C_2 be a clause containing $\overline{L}$;
  The (propositional) resolvent of C_1 and C_2 with respect to L is the clause

$$(C_1 \setminus \{L\}) \cup (C_2 \setminus \{\overline{L}\})$$

  C is said to be a resolvent of C_1 and C_2 iff
  there exists a literal L such that C is the resolvent of C_1 and C_2 wrt L

- **Examples when resolving on a**
- $(a \vee \neg a) \otimes (\neg a \vee a) = (a \vee \neg a)$
- $(a \vee \neg b) \otimes (\neg a \vee b) = (b \vee \neg b)$
- $(a \vee b) \otimes (\neg a \vee b) = (b)$

Propositional Resolution

- Remind: clauses are considered to be sets
- Definition Let C_1 be a clause containing L and C_2 be a clause containing $\overline{L}$; The (propositional) resolvent of C_1 and C_2 with respect to L is the clause

$$(C_1 \setminus \{L\}) \cup (C_2 \setminus \{\overline{L}\})$$

  C is said to be a resolvent of C_1 and C_2 iff there exists a literal L such that C is the resolvent of C_1 and C_2 wrt L

- Examples when resolving on a
- $(a \vee \neg a) \otimes (\neg a \vee a) = (a \vee \neg a)$
- $(a \vee \neg b) \otimes (\neg a \vee b) = (b \vee \neg b)$
- $(a \vee b) \otimes (\neg a \vee b) = (b)$
- Resolvents can subsume antecedents

Propositional Resolution

- Remind: clauses are considered to be sets
- Definition Let C_1 be a clause containing L and C_2 be a clause containing $\overline{L}$; The (propositional) resolvent of C_1 and C_2 with respect to L is the clause

$$(C_1 \setminus \{L\}) \cup (C_2 \setminus \{\overline{L}\})$$

  C is said to be a resolvent of C_1 and C_2 iff there exists a literal L such that C is the resolvent of C_1 and C_2 wrt L

- Examples when resolving on a
- $(a \vee \neg a) \otimes (\neg a \vee a) = (a \vee \neg a)$
- $(a \vee \neg b) \otimes (\neg a \vee b) = (b \vee \neg b)$
- $(a \vee b) \otimes (\neg a \vee b) = (b)$

- Resolvents can subsume antecedents
- Usually, resolvents have more literals than antecedents

SAT Solving

SAT Solving - Example

- **Given: Conjunction of clauses**
- **Task: Find satisfying interpretation for variables if possible!**

$$F = (a \lor c) \land (\bar{b} \lor \bar{e} \lor \bar{f}) \land (\bar{a} \lor \bar{d} \lor f) \land (\bar{a} \lor \bar{b} \lor \bar{d} \lor e) \land (\bar{a} \lor b)$$

- **How to find a solution?**
- **Some questions:**

SAT Solving - Example

- **Given: Conjunction of clauses**
- **Task: Find satisfying interpretation for variables if possible!**

$$F = (a \lor c) \land (\bar{b} \lor \bar{e} \lor \bar{f}) \land (\bar{a} \lor \bar{d} \lor f) \land (\bar{a} \lor \bar{b} \lor \bar{d} \lor e) \land (\bar{a} \lor b)$$

- **How to find a solution?**
- **Some questions:**

1. **How many combinations (solution candidates) exist for 6 Boolean variables?**

SAT Solving - Example

- **Given: Conjunction of clauses**
- **Task: Find satisfying interpretation for variables if possible!**

$$F = (a \vee c) \wedge (\bar{b} \vee \bar{e} \vee \bar{f}) \wedge (\bar{a} \vee \bar{d} \vee f) \wedge (\bar{a} \vee \bar{b} \vee \bar{d} \vee e) \wedge (\bar{a} \vee b)$$

- **How to find a solution?**
- **Some questions:**

1. **How many combinations (solution candidates) exist for 6 Boolean variables?**
2. **How many percent of the candidates are cut by a unit clause?**

SAT Solving - Example

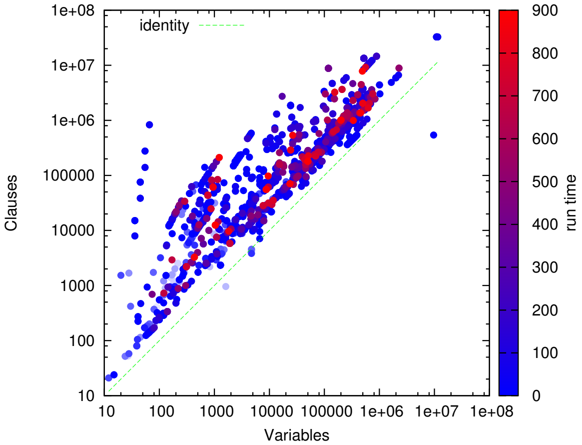
- **Given: Conjunction of clauses**
- **Task: Find satisfying interpretation for variables if possible!**

$$F = (a \lor c) \land (\bar{b} \lor \bar{e} \lor \bar{f}) \land (\bar{a} \lor \bar{d} \lor f) \land (\bar{a} \lor \bar{b} \lor \bar{d} \lor e) \land (\bar{a} \lor b)$$

- **How to find a solution?**
- **Some questions:**

1. **How many combinations (solution candidates) exist for 6 Boolean variables?**
2. **How many percent of the candidates are cut by a unit clause?**
3. **How many percent of the candidates are cut by a binary, ternary, ... clause?**

Power of Modern SAT Solvers



- **RISS 4.27, SAT Competition 2014, application track**
- **Formulas with several million clauses and variables can be solved**

SAT Solving – With Search

- **Assume a literal**
- **Propagate immediate consequences**
- **If a conflict, backtrack**

SAT Solving – With Search

- **Assume a literal**
- **Propagate immediate consequences**
- **If a conflict, backtrack**
- **Known as DPLL (Davis Putnam Logemann Loveland)**

SAT Solving – With Search

- **Assume a literal**
- **Propagate immediate consequences**
- **If a conflict, backtrack**
- **Known as DPLL (Davis Putnam Logemann Loveland)**
- **What are immediate consequences?**

SAT Solving – With Search

- Assume a literal
- Propagate immediate consequences
- If a conflict, backtrack
- Known as DPLL (Davis Putnam Logemann Loveland)

- What are immediate consequences?

$$F = (a \lor c) \land (\bar{b} \lor \bar{e} \lor \bar{f}) \land (\bar{a} \lor \bar{d} \lor f) \land (\bar{a} \lor \bar{b} \lor \bar{d} \lor e) \land (\bar{a} \lor b)$$

- Assume $\bar{b} = \top$, then we have $J = (\bar{b})$
- Are there variables with a forced assignment?

SAT Solving – With Search

- Assume a literal
- Propagate immediate consequences
- If a conflict, backtrack
- Known as DPLL (Davis Putnam Logemann Loveland)
- What are immediate consequences?

$$F|_{\bar{b}} = (a \vee c) \wedge \qquad (\bar{a} \vee \bar{d} \vee f) \wedge \qquad \wedge (\bar{a} \qquad)$$

- Assume $\bar{b} = \top$, then we have $J = (\bar{b})$
- Are there variables with a forced assignment?

SAT Solving – With Search

- Assume a literal
- Propagate immediate consequences
- If a conflict, backtrack
- Known as DPLL (Davis Putnam Logemann Loveland)

- What are immediate consequences?

$$F|_{\bar{b}} = (a \lor c) \land \qquad (\bar{a} \lor \bar{d} \lor f) \land \qquad \land (\bar{a} \qquad)$$

- Assume $\bar{b} = \top$, then we have $J = (\bar{b})$
- Are there variables with a forced assignment?
  - $\bar{a}$

SAT Solving – With Search

- Assume a literal
- Propagate immediate consequences
- If a conflict, backtrack
- Known as DPLL (Davis Putnam Logemann Loveland)
- What are immediate consequences?

$$F|_{\bar{b}\bar{a}} = (a \lor c) \land \qquad (\bar{a} \lor \bar{d} \lor f) \land \qquad \land (\bar{a} \qquad)$$

- Assume $\bar{b} = \top$, then we have $J = (\bar{b})$
- Are there variables with a forced assignment?
  - $\bar{a}$

SAT Solving – With Search

- Assume a literal
- Propagate immediate consequences
- If a conflict, backtrack
- Known as DPLL (Davis Putnam Logemann Loveland)
- What are immediate consequences?

  $F|_{\bar{b}\bar{a}} = (\quad c) \wedge$

- Assume $\bar{b} = \top$, then we have $J = (\bar{b})$
- Are there variables with a forced assignment?
  - $\bar{a}$ and $\bar{c}$

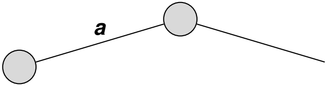
Davis Putnam Logemann Loveland (DPLL) in a Nutshell

$$= (\overline{a} \lor b) \land (\overline{b} \lor \overline{d} \lor e) \land (c \lor d) \land (\overline{a} \lor \overline{b} \lor \overline{e} \lor f) \land (\overline{a} \lor \overline{e} \lor \overline{f}) \land (d \lor \overline{f}) \land (\overline{c} \lor e \lor f)$$

Variable	a	b	c	d	e	f
Reason	-	-	-	-	-	-

Davis Putnam Logemann Loveland (DPLL) in a Nutshell

$$= (\overline{a} \vee b) \wedge (\overline{b} \vee \overline{d} \vee e) \wedge (c \vee d) \wedge (\overline{a} \vee \overline{b} \vee \overline{e} \vee f) \wedge (\overline{a} \vee \overline{e} \vee \overline{f}) \wedge (d \vee \overline{f}) \wedge (\overline{c} \vee e \vee f)$$

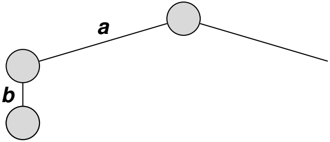


Variable	a	b	c	d	e	f
Reason	-	-	-	-	-	-

- **add a search decision**

Davis Putnam Logemann Loveland (DPLL) in a Nutshell

$$= (\overline{a} \vee b) \wedge (\overline{b} \vee \overline{d} \vee e) \wedge (c \vee d) \wedge (\overline{a} \vee \overline{b} \vee \overline{e} \vee f) \wedge (\overline{a} \vee \overline{e} \vee \overline{f}) \wedge (d \vee \overline{f}) \wedge (\overline{c} \vee e \vee f)$$

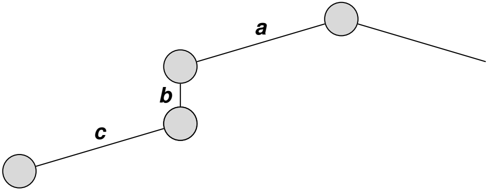


Variable	a	b	c	d	e	f
Reason	-	C_1	-	-	-	-

- **propagate consequences**

Davis Putnam Logemann Loveland (DPLL) in a Nutshell

$$= (\overline{a} \lor b) \land (\overline{b} \lor \overline{d} \lor e) \land (c \lor d) \land (\overline{a} \lor \overline{b} \lor \overline{e} \lor f) \land (\overline{a} \lor \overline{e} \lor \overline{f}) \land (d \lor \overline{f}) \land (\overline{c} \lor e \lor f)$$

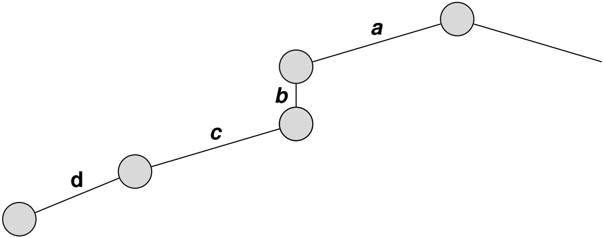


Variable	a	b	c	d	e	f
Reason	-	C_1	-	-	-	-

- **add a search decision**

Davis Putnam Logemann Loveland (DPLL) in a Nutshell

$= (\overline{a} \vee b) \wedge (\overline{b} \vee \overline{d} \vee e) \wedge (c \vee d) \wedge (\overline{a} \vee \overline{b} \vee \overline{e} \vee f) \wedge (\overline{a} \vee \overline{e} \vee \overline{f}) \wedge (d \vee \overline{f}) \wedge (\overline{c} \vee e \vee f)$

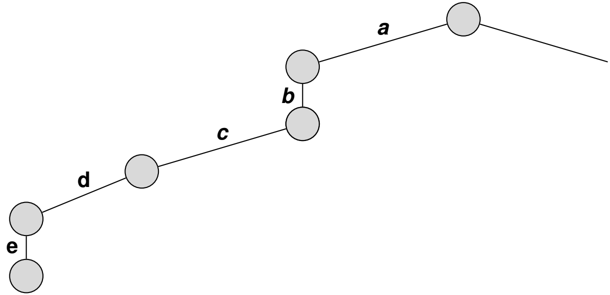


Variable	a	b	c	d	e	f
Reason	-	C_1	-	-	-	-

- add a search decision

Davis Putnam Logemann Loveland (DPLL) in a Nutshell

$$= (\overline{a} \lor b) \land (\overline{b} \lor \overline{d} \lor e) \land (c \lor d) \land (\overline{a} \lor \overline{b} \lor \overline{e} \lor f) \land (\overline{a} \lor \overline{e} \lor \overline{f}) \land (d \lor \overline{f}) \land (\overline{c} \lor e \lor f)$$

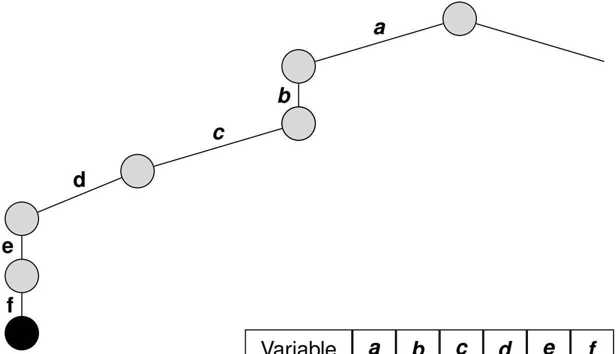


Variable	a	b	c	d	e	f
Reason	-	C_1	-	-	C_2	-

- **propagate consequences**

Davis Putnam Logemann Loveland (DPLL) in a Nutshell

$$= (\bar{a} \lor b) \land (\bar{b} \lor \bar{d} \lor e) \land (c \lor d) \land (\bar{a} \lor \bar{b} \lor \bar{e} \lor f) \land (\bar{a} \lor \bar{e} \lor \bar{f}) \land (d \lor \bar{f}) \land (\bar{c} \lor e \lor f)$$



found conflict

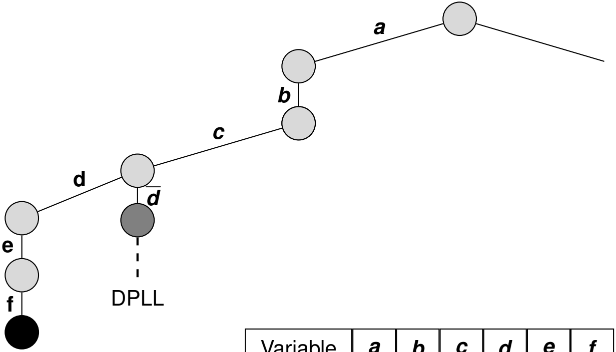
$C_5 = (\bar{a} \lor \bar{e} \lor \bar{f})$

Variable	a	b	c	d	e	f
Reason	-	C_1	-	-	C_2	C_4

- **propagate consequences**

Davis Putnam Logemann Loveland (DPLL) in a Nutshell

$$= (\bar{a} \vee b) \wedge (\bar{b} \vee \bar{d} \vee e) \wedge (c \vee d) \wedge (\bar{a} \vee \bar{b} \vee \bar{e} \vee f) \wedge (\bar{a} \vee \bar{e} \vee \bar{f}) \wedge (d \vee \bar{f}) \wedge (\bar{c} \vee e \vee f)$$



found conflict

$C_5 = (\bar{a} \vee \bar{e} \vee \bar{f})$

Variable	a	b	c	d	e	f
Reason	-	C_1	-	-	C_2	C_4

- **backtrack from conflict and proceed with search**

DPLL pseudo code

- **An iterative solving algorithm**

IDPLL (CNF formula $\boldsymbol{F}$)

Input: A formula $\boldsymbol{F}$ in CNF
Output: The solution **SAT** or **UNSAT** of this formula

$J := ()$	// start with empty interpretation
while true	// until we find a solution
if $F\|_J = \emptyset$ **then return SAT**	// satisfiability rule
if $[] \in F\|_J$ **then**	// there was a conflict
if $J = J'\dot{x}J''$ and $\nexists \dot{y} \in J''$ **then**	// backtrack and undo most recent decision
$J := J'\overline{x}$	// add the complement
continue	
else return UNSAT	// unsatisfiability rule
if $(x) \in F\|_J$ **then**	// unit rule
$J := Jx$	// extend the interpretation
continue	
if $x \in$ **lits**$(F\|_J)$ and $\overline{x} \notin$ **lits**$(F\|_J)$ **then**	// pure literal rule
$J := Jx$	
continue	
$J := J\dot{x}$ for some $x \in$ **lits**$(F\|_J)$	// decide rule

Conclusions of the DPLL Algorithm

- **Chronological backtracking**
- **Heavily depends on the order of the decision variables**

- **How to perform unit propagation? How to find a unit in the formula efficiently?**

Unit Propagation

- **How to perform unit propagation?**
- **How to find a unit in the formula efficiently?**

- **Assumption: we use the presented pseudo code as algorithm.**

Conflict Driven Clause Learning (CDCL) in a Nutshell

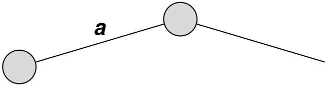
Conflict Driven Clause Learning (CDCL) in a Nutshell

$$F = (\overline{a} \vee b) \wedge (\overline{b} \vee \overline{d} \vee e) \wedge (c \vee d) \wedge (\overline{a} \vee \overline{b} \vee \overline{e} \vee f) \wedge (\overline{a} \vee \overline{e} \vee \overline{f}) \wedge (d \vee \overline{f}) \wedge (\overline{c} \vee e \vee f)$$

Variable	a	b	c	d	e	f
Reason	-	-	-	-	-	-

Conflict Driven Clause Learning (CDCL) in a Nutshell

$$F = (\overline{a} \vee b) \wedge (\overline{b} \vee \overline{d} \vee e) \wedge (c \vee d) \wedge (\overline{a} \vee \overline{b} \vee \overline{e} \vee f) \wedge (\overline{a} \vee \overline{e} \vee \overline{f}) \wedge (d \vee \overline{f}) \wedge (\overline{c} \vee e \vee f)$$

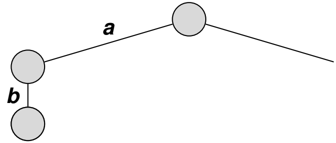


Variable	a	b	c	d	e	f
Reason	-	-	-	-	-	-

- **add a search decision**

Conflict Driven Clause Learning (CDCL) in a Nutshell

$$F = (\overline{a} \lor b) \land (\overline{b} \lor \overline{d} \lor e) \land (c \lor d) \land (\overline{a} \lor \overline{b} \lor \overline{e} \lor f) \land (\overline{a} \lor \overline{e} \lor \overline{f}) \land (d \lor \overline{f}) \land (\overline{c} \lor e \lor f)$$

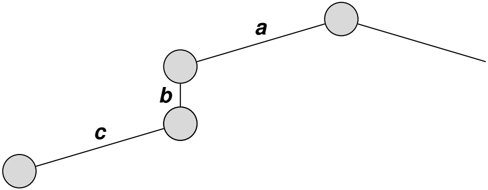


Variable	a	b	c	d	e	f
Reason	-	C_1	-	-	-	-

- **propagate consequences**

Conflict Driven Clause Learning (CDCL) in a Nutshell

$$F = (\overline{a} \vee b) \wedge (\overline{b} \vee \overline{d} \vee e) \wedge (c \vee d) \wedge (\overline{a} \vee \overline{b} \vee \overline{e} \vee f) \wedge (\overline{a} \vee \overline{e} \vee \overline{f}) \wedge (d \vee \overline{f}) \wedge (\overline{c} \vee e \vee f)$$

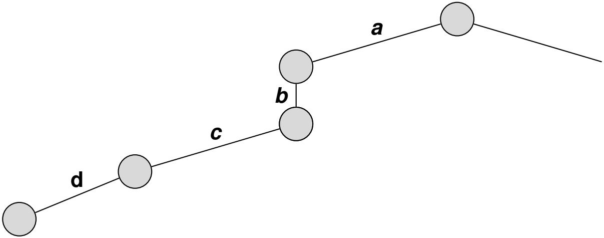


Variable	*a*	*b*	*c*	*d*	*e*	*f*
Reason	-	C_1	-	-	-	-

- **add a search decision**

Conflict Driven Clause Learning (CDCL) in a Nutshell

$$F = (\overline{a} \lor b) \land (\overline{b} \lor \overline{d} \lor e) \land (c \lor d) \land (\overline{a} \lor \overline{b} \lor \overline{e} \lor f) \land (\overline{a} \lor \overline{e} \lor \overline{f}) \land (d \lor \overline{f}) \land (\overline{c} \lor e \lor f)$$

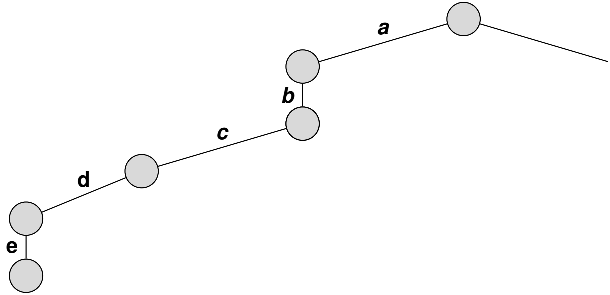


Variable	a	b	c	d	e	f
Reason	-	C_1	-	-	-	-

- **add a search decision**

Conflict Driven Clause Learning (CDCL) in a Nutshell

$$F = (\overline{a} \vee b) \wedge (\overline{b} \vee \overline{d} \vee e) \wedge (c \vee d) \wedge (\overline{a} \vee \overline{b} \vee \overline{e} \vee f) \wedge (\overline{a} \vee \overline{e} \vee \overline{f}) \wedge (d \vee \overline{f}) \wedge (\overline{c} \vee e \vee f)$$

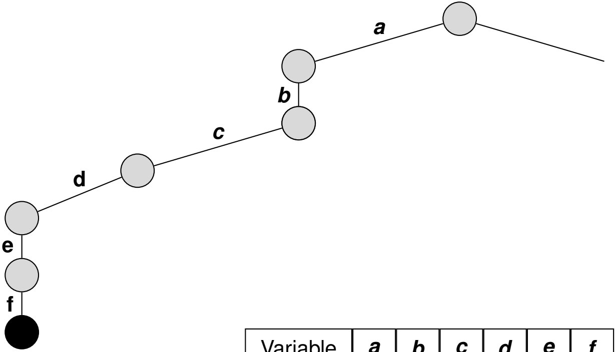


Variable	a	b	c	d	e	f
Reason	-	C_1	-	-	C_2	-

- **propagate consequences**

Conflict Driven Clause Learning (CDCL) in a Nutshell

$$F = (\overline{a} \vee b) \wedge (\overline{b} \vee \overline{d} \vee e) \wedge (c \vee d) \wedge (\overline{a} \vee \overline{b} \vee \overline{e} \vee f) \wedge (\overline{a} \vee \overline{e} \vee \overline{f}) \wedge (d \vee \overline{f}) \wedge (\overline{c} \vee e \vee f)$$



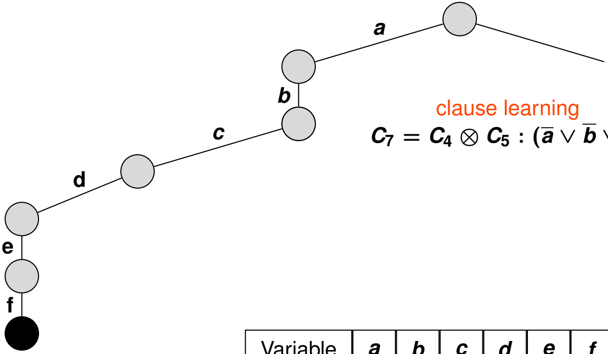
found conflict
$C_5 = (\overline{a} \vee \overline{e} \vee \overline{f})$

Variable	a	b	c	d	e	f
Reason	-	C_1	-	-	C_2	C_4

- **propagate consequences**

Conflict Driven Clause Learning (CDCL) in a Nutshell

$$F = (\overline{a} \vee b) \wedge (\overline{b} \vee \overline{d} \vee e) \wedge (c \vee d) \wedge (\overline{a} \vee \overline{b} \vee \overline{e} \vee f) \wedge (\overline{a} \vee \overline{e} \vee \overline{f}) \wedge (d \vee \overline{f}) \wedge (\overline{c} \vee e \vee f)$$



clause learning

$C_7 = C_4 \otimes C_5 : (\overline{a} \vee \overline{b} \vee \overline{e})$

found conflict

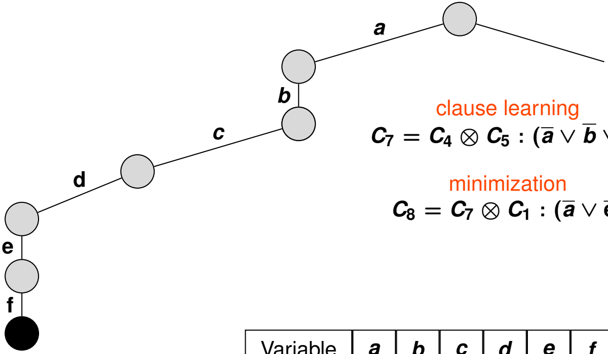
$C_5 = (\overline{a} \vee \overline{e} \vee \overline{f})$

Variable	a	b	c	d	e	f
Reason	-	C_1	-	-	C_2	C_4

- **create and add the learned clause to the formula**

Conflict Driven Clause Learning (CDCL) in a Nutshell

$$F = (\overline{a} \vee b) \wedge (\overline{b} \vee \overline{d} \vee e) \wedge (c \vee d) \wedge (\overline{a} \vee \overline{b} \vee \overline{e} \vee f) \wedge (\overline{a} \vee \overline{e} \vee \overline{f}) \wedge (d \vee \overline{f}) \wedge (\overline{c} \vee e \vee f)$$



clause learning

$C_7 = C_4 \otimes C_5 : (\overline{a} \vee \overline{b} \vee \overline{e})$

minimization

$C_8 = C_7 \otimes C_1 : (\overline{a} \vee \overline{e})$

found conflict

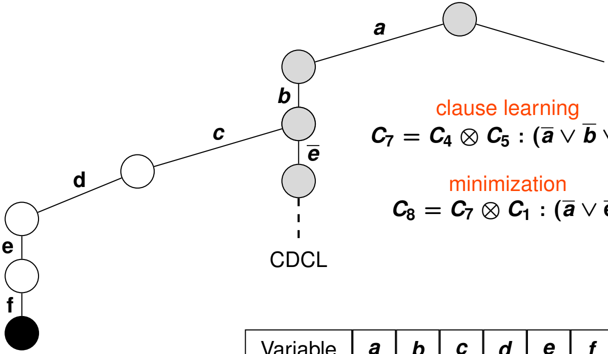
$C_5 = (\overline{a} \vee \overline{e} \vee \overline{f})$

Variable	a	b	c	d	e	f
Reason	-	C_1	-	-	C_2	C_4

- **create and add the learned clause to the formula**

Conflict Driven Clause Learning (CDCL) in a Nutshell

$$F = (\overline{a} \vee b) \wedge (\overline{b} \vee \overline{d} \vee e) \wedge (c \vee d) \wedge (\overline{a} \vee \overline{b} \vee \overline{e} \vee f) \wedge (\overline{a} \vee \overline{e} \vee \overline{f}) \wedge (d \vee \overline{f}) \wedge (\overline{c} \vee e \vee f)$$



clause learning

$$C_7 = C_4 \otimes C_5 : (\overline{a} \vee \overline{b} \vee \overline{e})$$

minimization

$$C_8 = C_7 \otimes C_1 : (\overline{a} \vee \overline{e})$$

found conflict

$$C_5 = (\overline{a} \vee \overline{e} \vee \overline{f})$$

Variable	a	b	c	d	e	f
Reason	-	C_1	-	-	C_8	-

- **backtrack, add C_8, and proceed with unit propagation**

The CDCL Algorithm

CDCL (CNF formula $\boldsymbol{F}$)

Input: A formula $\boldsymbol{F}$ in CNF
Output: The solution **SAT** or **UNSAT** of this formula

$\boldsymbol{J} := ()$ // start with empty interpretation
while true
  while $(\boldsymbol{x}) \in \boldsymbol{F}|_{\boldsymbol{J}}$ **do** // unit rule
    $\boldsymbol{J} := \boldsymbol{Jx}$
  if $[] \in \boldsymbol{F}|_{\boldsymbol{J}}$ **then** // conflict
    if $\exists \dot{\boldsymbol{y}} \in \boldsymbol{J}$, such that $\boldsymbol{J} = \boldsymbol{J}'\boldsymbol{J}''\dot{\boldsymbol{y}}\boldsymbol{J}'''$ **then**
      $\boldsymbol{F} := \boldsymbol{F} \cup \boldsymbol{C}$ with $\boldsymbol{F} \models \boldsymbol{C}$ and $\boldsymbol{C} \notin \boldsymbol{F}$ // learning
      $\boldsymbol{J} := \boldsymbol{J}'$ // backjumping
    else return UNSAT // unsatisfiability rule
  else // no empty clause in $\boldsymbol{F}|_{\boldsymbol{J}}$
    if atoms($\boldsymbol{J}$**)** $\supseteq$ **atoms(**$\boldsymbol{F}$**) then return SAT** // satisfiability rule
    else $\boldsymbol{J} := \boldsymbol{J}\dot{\boldsymbol{z}}$ with **atoms(**$\boldsymbol{z}$**)** $\subseteq$ **atoms(**$\boldsymbol{F}$**)** // decision rule

Conclusions of the CDCL Algorithm

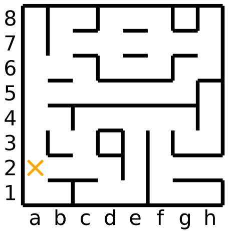
- Heavily depends on the order of the decision variables
- Non-Chronological backtracking, backjumping
- Learning of new clauses
- No mentioned here: restarts, clause removal

- Question: can the CDCL algorithm simulate the DPLL algorithm?

An Intuitive Abstraction of SAT Solver Techniques

Finding an Exit in a Maze

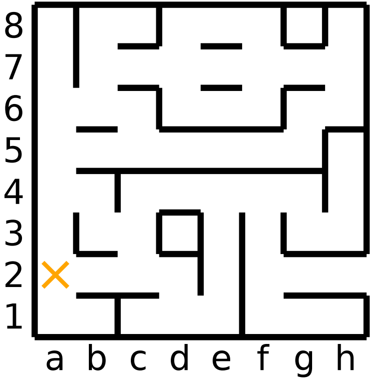
- **Some rules**
  - **starting point is located in the left column**
  - **exit is on the right side (if there exists one)**
  - **search decisions can be done only when moving right**
  - **when moving left, use backtracking**



- **Property: satisfiable formulas correspond to mazes that can be solved by searching right only**

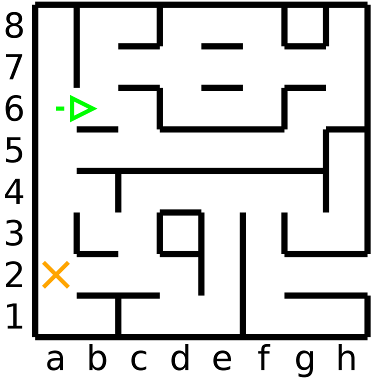
The DPLL Algorithm

- **Heuristics:**
  - **pick the highest possible column**
  - **then, pick the lowest column**



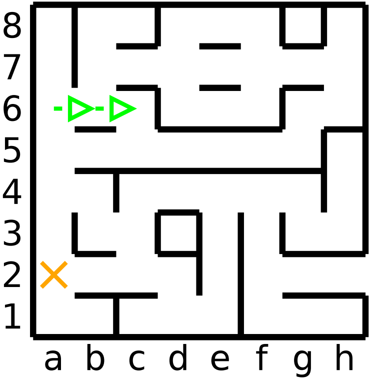
The DPLL Algorithm

- **Heuristics:**
  - **pick the highest possible column**
  - **then, pick the lowest column**



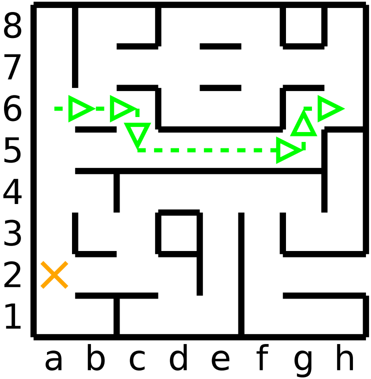
The DPLL Algorithm

- **Heuristics:**
  - **pick the highest possible column**
  - **then, pick the lowest column**



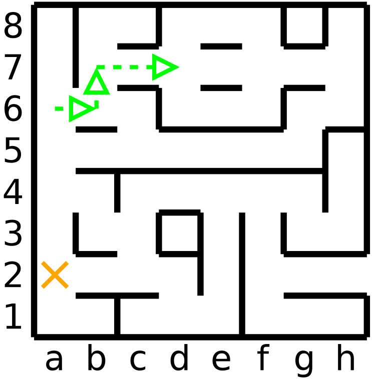
The DPLL Algorithm

- **No decision, hence propagate**



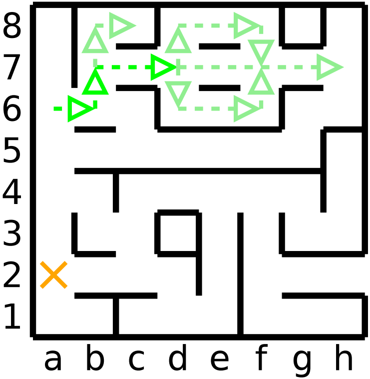
The DPLL Algorithm

- No exit (conflict), hence backtrack



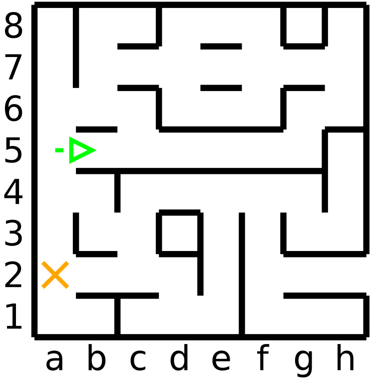
The DPLL Algorithm

- No exit in the upper search space, hence backtrack



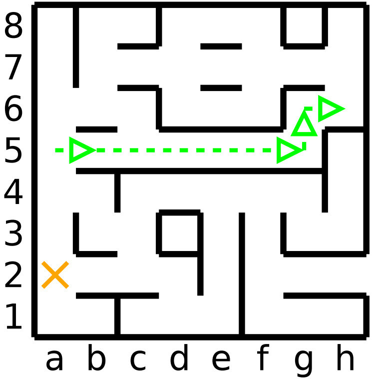
The DPLL Algorithm

- Do next search decision



The DPLL Algorithm

- Enter the same search space as before

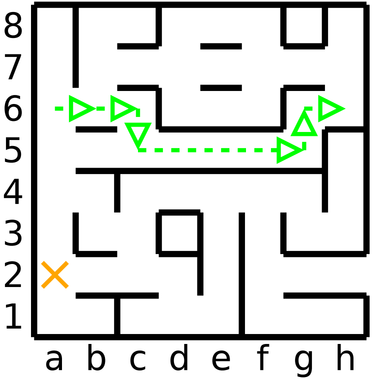


The CDCL Algorithm (conflict driven clause learning)

- **Choose,**
- **Propagate,**
- **and Backtrack after Conflict**
- **. . . enters the same search space again and again.**
- **Let's go a few steps back . . .**

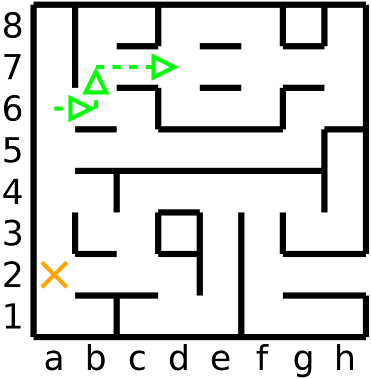
The CDCL Algorithm (conflict driven clause learning)

- No decision, hence propagate



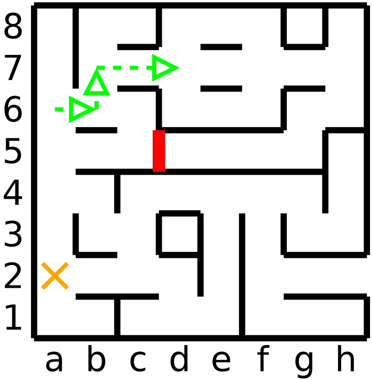
The CDCL Algorithm (conflict driven clause learning)

- No exit (conflict), hence backtrack



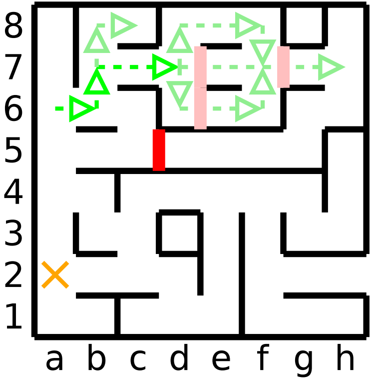
The CDCL Algorithm (conflict driven clause learning)

- . . . and learn a clause



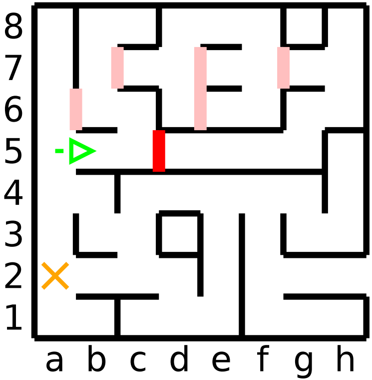
The CDCL Algorithm (conflict driven clause learning)

- No exit in upper search space, backtrack and learn



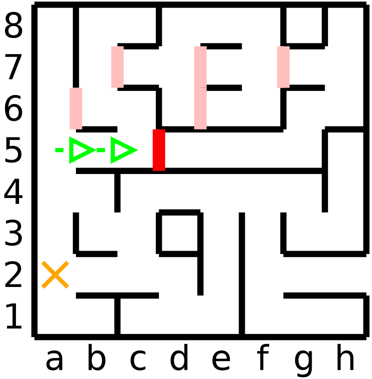
The CDCL Algorithm (conflict driven clause learning)

- Do next search decision

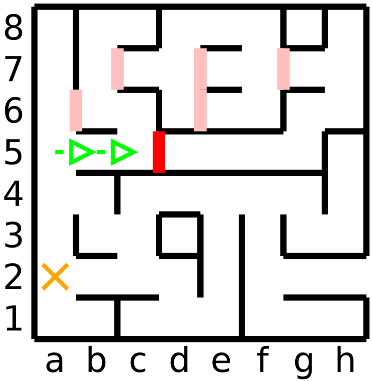


The CDCL Algorithm (conflict driven clause learning)

- **Does not** enter the same search space as before

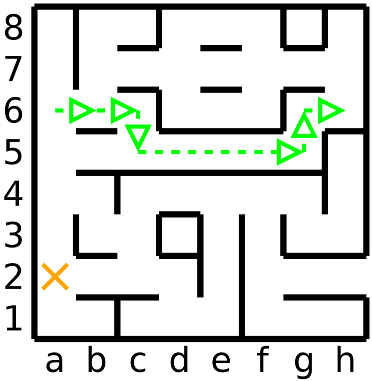


Motivating Clause Removal



How to perform Unit Propagation

- When is a unit clause in the reduct?
- How to find a unit clause in the reduct, especially in the CDCL algorithm?



Coming Next

- **Simplification**
- **Parallel Search**