

Backward chaining

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B

Backward chaining

KB \models Q

→ $\boxed{P} \Rightarrow Q$

$L \wedge M \Rightarrow \boxed{P}$

$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B

→ $C \wedge D \Rightarrow R$

$B \Rightarrow S$

→ $R \wedge S \Rightarrow T$

C

D

Backward chaining

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B

$$C \wedge D \Rightarrow R$$

$$B \Rightarrow S$$

$$R \wedge S \Rightarrow T$$

C

D

- ▶ Checking whether $KB \Rightarrow Q$ is valid.

Backward chaining

$$P \Rightarrow \underline{Q}$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

~~$$A \wedge B \Rightarrow L$$~~

A

~~B~~

$$C \wedge D \Rightarrow R$$

$$B \Rightarrow S$$

$$R \wedge S \Rightarrow T$$

C

D

→ A

→ B

X ⇒ B

- ▶ Checking whether $KB \Rightarrow Q$ is valid.

- ▶ What would happen if B was not known?

$$KB \models Q$$

$$KB : P \Rightarrow Q$$

Backward chaining

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B

$$C \wedge D \Rightarrow R$$

$$B \Rightarrow S$$

$$R \wedge S \Rightarrow T$$

C

D

- ▶ Checking whether $KB \Rightarrow Q$ is valid.
- ▶ What would happen if B was not known?
- ▶ Goal-driven inferencing

Effective Propositional Model Checking

Davis, Putnam, Logemann and Loveland (DPLL) Algorithm

Input : A sentence in CNF

Output : Is the sentence satisfiable?

Davis, Putnam, Logemann and Loveland (DPLL) Algorithm

Input : A sentence in CNF

Output : Is the sentence satisfiable?

- ▶ Early termination

Davis, Putnam, Logemann and Loveland (DPLL) Algorithm

Input : A sentence in CNF

Output : Is the sentence satisfiable?

- ▶ Early termination
- ▶ Pure symbol heuristic

Davis, Putnam, Logemann and Loveland (DPLL) Algorithm

Input : A sentence in CNF

Output : Is the sentence satisfiable?

- ▶ Early termination
- ▶ Pure symbol heuristic
- ▶ Unit clause heuristic

$$\underline{(a \vee b \vee c)}$$
$$a = F \quad b = F$$

DPLL Algorithm

... $\vee (a \vee b \vee c) \vee (\neg a \vee e \vee f)$

$b = f$
 $e = f$

function DPLL-SATISFIABLE?(s) **returns** *true* or *false*

inputs: s , a sentence in propositional logic

→ $clauses \leftarrow$ the set of clauses in the CNF representation of s

$symbols \leftarrow$ a list of the proposition symbols in s

return DPLL($clauses, symbols, \{\}$)

→ **function** DPLL($clauses, symbols, model$) **returns** *true* or *false*

if every clause in $clauses$ is *true* in $model$ **then return** *true*

if some clause in $clauses$ is *false* in $model$ **then return** *false*

→ $P, value \leftarrow$ FIND-PURE-SYMBOL($symbols, clauses, model$)

if P is *non-null* **then return** DPLL($clauses, symbols - P, model \cup \{P=value\}$) ✓

→ $P, value \leftarrow$ FIND-UNIT-CLAUSE($clauses, model$)

if P is *non-null* **then return** DPLL($clauses, symbols - P, model \cup \{P=value\}$)

→ $P \leftarrow$ FIRST($symbols$); $rest \leftarrow$ REST($symbols$)

return DPLL($clauses, rest, model \cup \{P=true\}$), **or**

DPLL($clauses, rest, model \cup \{P=false\}$)

DPLL Algorithm

Further enhancements:

DPLL Algorithm

Further enhancements:

► Component Analysis

$$\begin{array}{l} \cdot \underbrace{C_1 \wedge C_2 \wedge C_3 \wedge C_4}_{v_1 - v_5} \wedge \underbrace{C_6 \wedge C_7 \wedge C_8 \wedge C_9}_{v_6 - v_{10}} \\ \rightarrow \end{array}$$

$$2^{10} = 1024$$

$$2^5 = \underline{32 + 32}$$

DPLL Algorithm

Further enhancements:

- ▶ Component Analysis
- ▶ Variable and value ordering

DPLL Algorithm

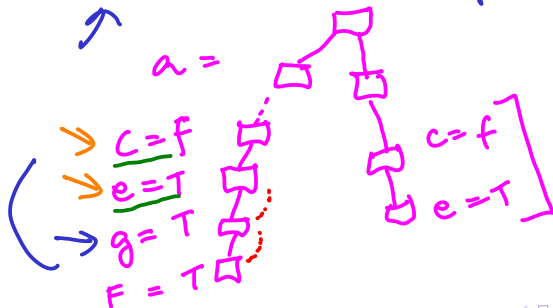
$$c=f \quad e=\top$$

Further enhancements:

- ▶ Component Analysis
- ▶ Variable and value ordering
- ▶ Intelligent Backtracking

$$\dots \wedge \underbrace{(c \vee \neg e \vee g)} \wedge \underbrace{(c \vee \neg e \vee \neg f)} \wedge \underbrace{(\neg g \vee f)} \wedge \dots$$

Annotations: Blue arrows point to the first and second clauses. Purple arrows point to the literals c , $\neg e$, and $\neg f$. An orange arrow points to the literal g in the third clause, and a blue arrow points to the literal f in the same clause.



DPLL Algorithm

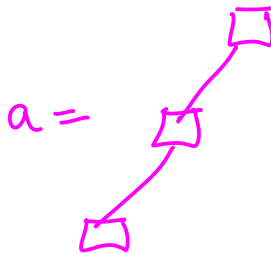
Further enhancements:

- ▶ Component Analysis
- ▶ Variable and value ordering
- ▶ Intelligent Backtracking
- ▶ Conflict clause learning

DPLL Algorithm

Further enhancements:

- ▶ Component Analysis
- ▶ Variable and value ordering
- ▶ Intelligent Backtracking
- ▶ Conflict clause learning
- ▶ Random restarts



DPLL Algorithm

Further enhancements:

- ▶ Component Analysis
- ▶ Variable and value ordering
- ▶ Intelligent Backtracking
- ▶ Conflict clause learning
- ▶ Random restarts
- ▶ Clever indexing

φ

$a = T$

$\boxed{a} \rightarrow$

$\neg a$