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

$KB \models Q$

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B

$C \wedge D \Rightarrow R$

$B \Rightarrow S$

$R \wedge S \Rightarrow T$

C

D

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

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

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

A

~~B~~

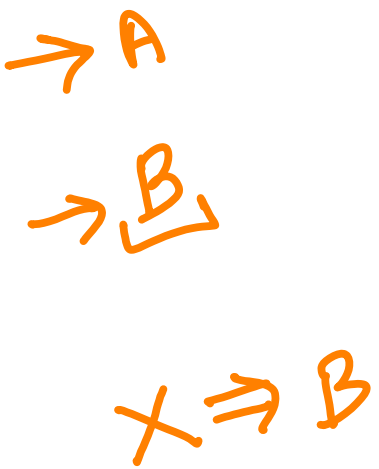
$C \wedge D \Rightarrow R$

$B \Rightarrow S$

$R \wedge S \Rightarrow T$

C

D



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

$KB \models Q$

$KB : P \Rightarrow Q$

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?

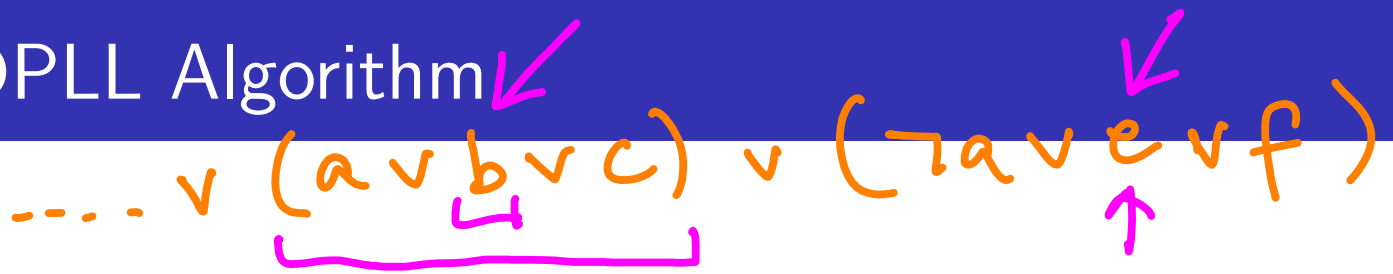
- Early termination
- Pure symbol heuristic

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

DPLL Algorithm



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

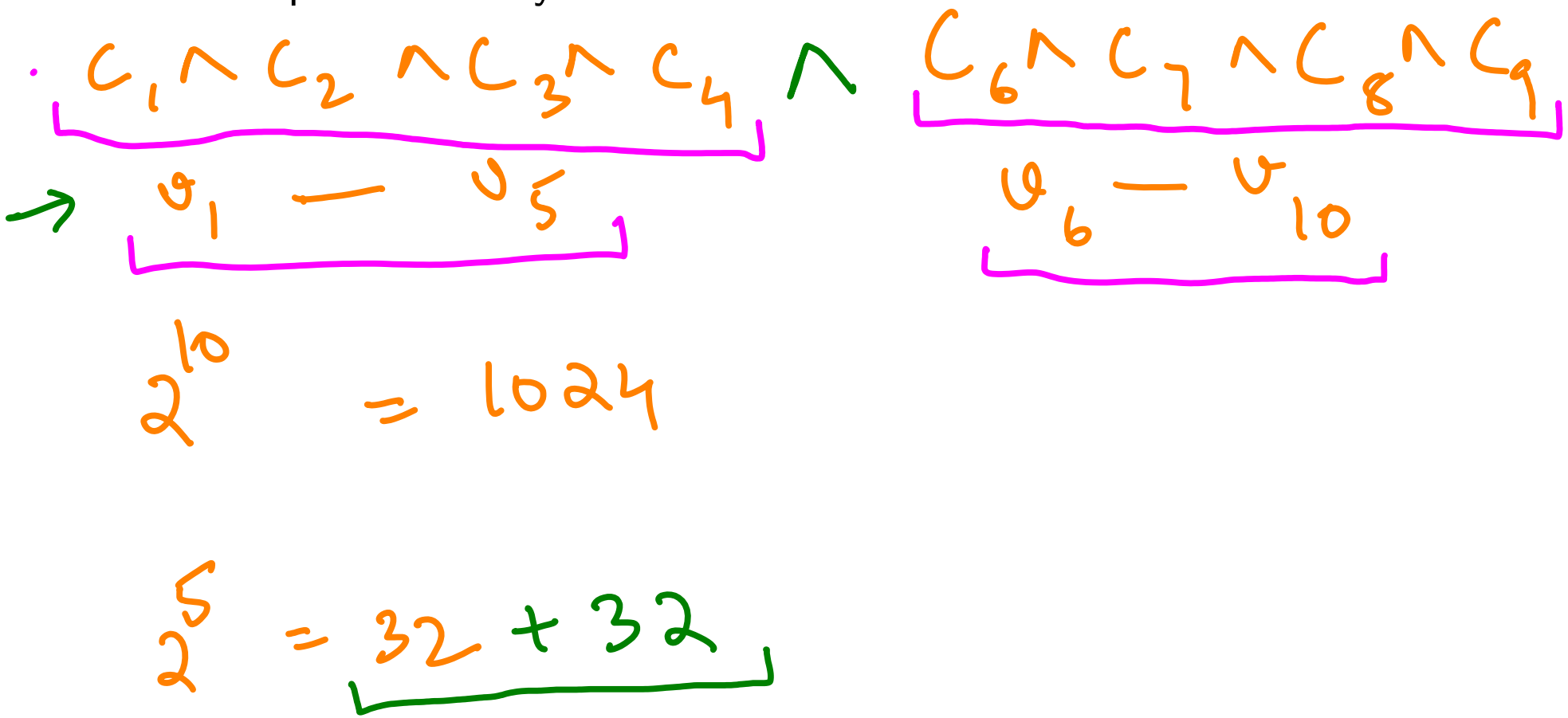
DPLL Algorithm

Further enhancements:

DPLL Algorithm

Further enhancements:

- Component Analysis



DPLL Algorithm

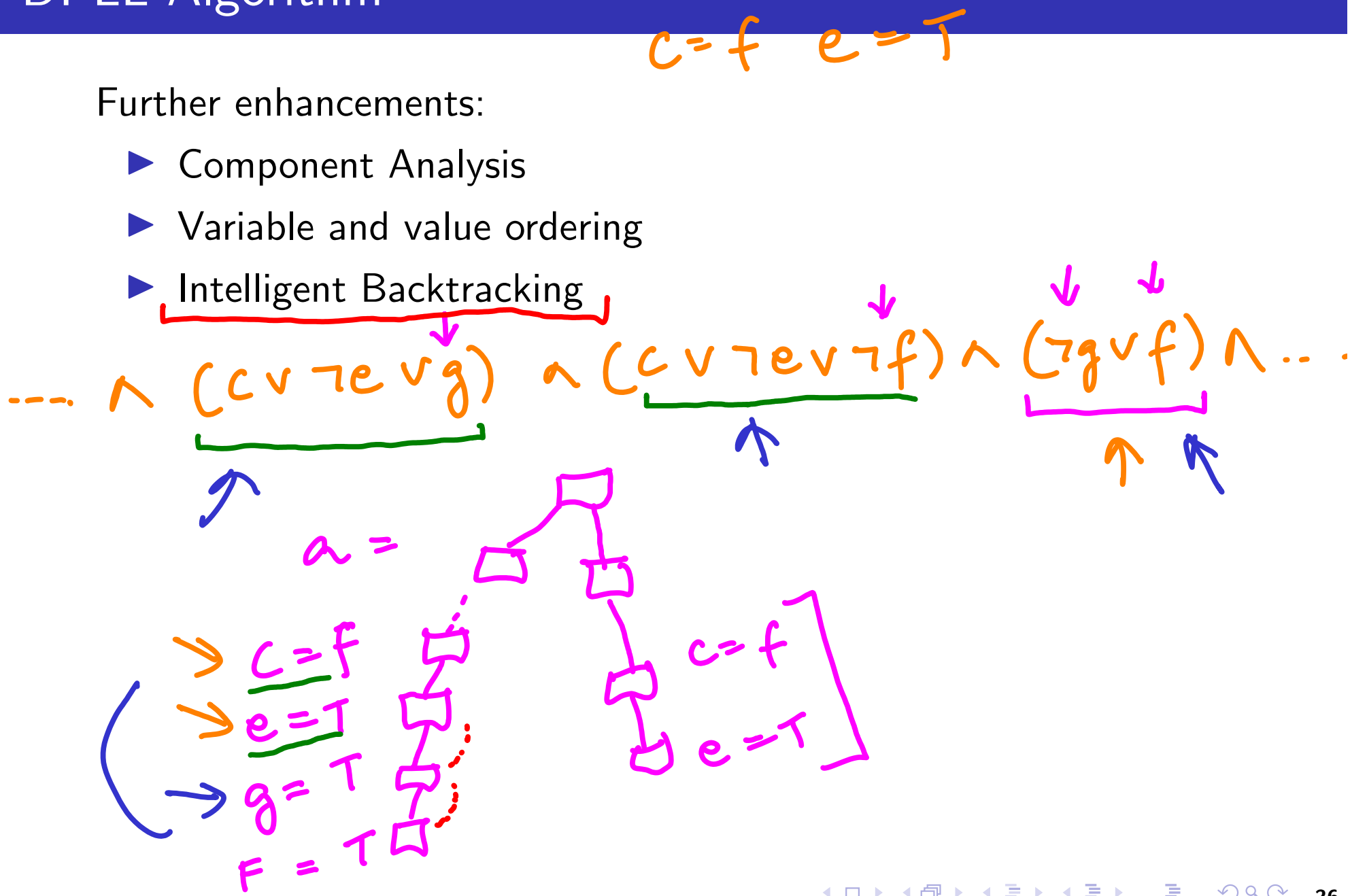
Further enhancements:

- Component Analysis
- Variable and value ordering

DPLL Algorithm

Further enhancements:

- Component Analysis
- Variable and value ordering
- Intelligent Backtracking



DPLL Algorithm

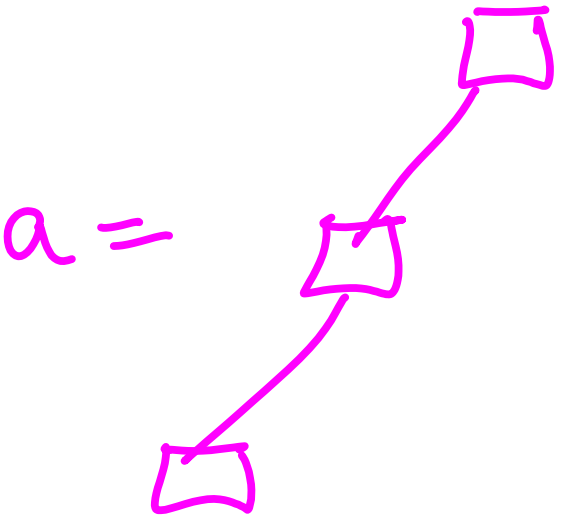
Further enhancements:

- 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