# Foundations of Artificial Intelligence 31. Propositional Logic: DPLL Algorithm

#### Thomas Keller and Florian Pommerening

University of Basel

April 26, 2023

DPLL on Horn Formulas

Summary 00

#### Propositional Logic: Overview

#### Chapter overview: propositional logic

- 29. Basics
- 30. Reasoning and Resolution
- 31. DPLL Algorithm
- 32. Local Search and Outlook

Systematic Search: DPL

DPLL on Horn Formulas

Summary 00

# Motivation

#### Propositional Logic: Motivation

- Propositional logic allows for the representation of knowledge and for deriving conclusions based on this knowledge.
- many practical applications can be directly encoded, e.g.
  - constraint satisfaction problems of all kinds
  - circuit design and verification
- many problems contain logic as ingredient, e.g.
  - automated planning
  - general game playing
  - description logic queries (semantic web)

Summary 00

#### Propositional Logic: Algorithmic Problems

main problems:

- reasoning ( $\Theta \models \varphi$ ?): Does the formula  $\varphi$  logically follow from the formulas  $\Theta$ ?
- equivalence ( $\varphi \equiv \psi$ ):

Are the formulas  $\varphi$  and  $\psi$  logically equivalent?

• satisfiability (SAT):

Is formula  $\varphi$  satisfiable? If yes, find a model.

Summary 00

# The Satisfiability Problem

The Satisfiability Problem (SAT)

```
given:
```

propositional formula in conjunctive normal form (CNF)

usually represented as pair  $\langle V, \Delta \rangle$ :

- V set of propositional variables (propositions)
- ∆ set of clauses over V
   (clause = set of literals v or ¬v with v ∈ V)

find:

- satisfying interpretation (model)
- or proof that no model exists

SAT is a famous NP-complete problem (Cook 1971; Levin 1973).

## Relevance of SAT

- The name "SAT" is often used for the satisfiability problem for general propositional formulas (instead of restriction to CNF).
- General SAT can be reduced to CNF (conversion in time O(n)).
- All previously mentioned problems can be reduced to SAT (conversion in time O(n)).
- $\rightsquigarrow$  SAT algorithms important and intensively studied

this and next chapter: SAT algorithms

Systematic Search: DPLL

DPLL on Horn Formulas

Summary 00

# Systematic Search: DPLL

## SAT vs. CSP

SAT can be considered as constraint satisfaction problem:

- CSP variables = propositions
- domains =  $\{\mathbf{F}, \mathbf{T}\}$
- constraints = clauses

However, we often have constraints that affect > 2 variables.

Due to this relationship, all ideas for CSPs are applicable to SAT:

- search
- inference
- variable and value orders

# The DPLL Algorithm

The DPLL algorithm (Davis/Putnam/Logemann/Loveland) corresponds to backtracking with inference for CSPs.

- recursive call DPLL(Δ, I)
   for clause set Δ and partial interpretation I
- result is consistent extension of *I*; unsatisfiable if no such extension exists
- first call  $DPLL(\Delta, \emptyset)$

inference in DPLL:

- simplify: after assigning value d to variable v, simplify all clauses that contain v
   → forward checking (for constraints of potentially higher arity)
- unit propagation: variables that occur in clauses without other variables (unit clauses) are assigned immediately
   minimum remaining values variable order

Summary 00

# The DPLL Algorithm: Pseudo-Code

#### function DPLL( $\Delta$ , *I*): if $\Box \in \Delta$ : [empty clause exists ↔ unsatisfiable] return unsatisfiable else if $\Delta = \emptyset$ : [no clauses left $\rightarrow$ interpretation I satisfies formula] return / else if there exists a unit clause $\{v\}$ or $\{\neg v\}$ in $\Delta$ : [unit propagation] Let v be such a variable, d the truth value that satisfies the clause. $\Delta' := \operatorname{simplify}(\Delta, v, d)$ return DPLL( $\Delta', I \cup \{v \mapsto d\}$ ) else: splitting rule Select some variable v which occurs in $\Delta$ . for each $d \in \{F, T\}$ in some order: $\Delta' := \operatorname{simplify}(\Delta, v, d)$ $I' := \mathsf{DPLL}(\Delta', I \cup \{v \mapsto d\})$ if $I' \neq$ unsatisfiable return /' return unsatisfiable

Systematic Search: DPLL

DPLL on Horn Formulas

Summary

## The DPLL Algorithm: simplify

#### **function** simplify( $\Delta$ , v, d)

Let  $\ell$  be the literal for v that is satisfied by  $v \mapsto d$ .  $\Delta' := \{C \mid C \in \Delta \text{ such that } \ell \notin C\}$   $\Delta'' := \{C \setminus \{\overline{\ell}\} \mid C \in \Delta'\}$ return  $\Delta''$ 

- Remove clauses containing ℓ

   → clause is satisfied by v → d
- Remove ℓ from remaining clauses
   → clause has to be satisfied with other variable

Systematic Search: DPLL

DPLL on Horn Formulas

Summary 00

# Example (1)

## $\Delta = \{\{X, Y, \neg Z\}, \{\neg X, \neg Y\}, \{Z\}, \{X, \neg Y\}\}$

Systematic Search: DPLL

DPLL on Horn Formulas

Summary 00

# Example (1)

$$\Delta = \{\{X, Y, \neg Z\}, \{\neg X, \neg Y\}, \{Z\}, \{X, \neg Y\}\}$$

• unit propagation:  $Z \mapsto \mathbf{T}$ 

Systematic Search: DPLL

DPLL on Horn Formulas

Summary 00

# Example (1)

$$\Delta = \{\{X, Y, \neg Z\}, \{\neg X, \neg Y\}, \{Z\}, \{X, \neg Y\}\}$$

• unit propagation:  $Z \mapsto \mathbf{T}$  $\{\{X, Y\}, \{\neg X, \neg Y\}, \{X, \neg Y\}\}$ 

# Example (1)

$$\Delta = \{\{X, Y, \neg Z\}, \{\neg X, \neg Y\}, \{Z\}, \{X, \neg Y\}\}$$

- unit propagation:  $Z \mapsto \mathbf{T}$  $\{\{X, Y\}, \{\neg X, \neg Y\}, \{X, \neg Y\}\}$
- splitting rule:

Systematic Search: DPLL

DPLL on Horn Formulas

Summary 00

# Example (1)

$$\Delta = \{\{X, Y, \neg Z\}, \{\neg X, \neg Y\}, \{Z\}, \{X, \neg Y\}\}$$

- unit propagation:  $Z \mapsto \mathbf{T}$  $\{\{X, Y\}, \{\neg X, \neg Y\}, \{X, \neg Y\}\}$
- splitting rule:

2a.  $X \mapsto \mathbf{F}$  $\{\{Y\}, \{\neg Y\}\}$ 

Systematic Search: DPLL

DPLL on Horn Formulas

Summary 00

# Example (1)

$$\Delta = \{\{X, Y, \neg Z\}, \{\neg X, \neg Y\}, \{Z\}, \{X, \neg Y\}\}$$

- unit propagation:  $Z \mapsto \mathbf{T}$  $\{\{X, Y\}, \{\neg X, \neg Y\}, \{X, \neg Y\}\}$
- Splitting rule:

2a.  $X \mapsto \mathbf{F}$   $\{\{Y\}, \{\neg Y\}\}\$ 3a. unit propagation:  $Y \mapsto \mathbf{T}$  $\{\Box\}$ 

Systematic Search: DPLL

DPLL on Horn Formulas

Summary 00

# Example (1)

 $\{\Box\}$ 

$$\Delta = \{\{X, Y, \neg Z\}, \{\neg X, \neg Y\}, \{Z\}, \{X, \neg Y\}\}$$

- unit propagation:  $Z \mapsto \mathbf{T}$  $\{\{X, Y\}, \{\neg X, \neg Y\}, \{X, \neg Y\}\}$
- Splitting rule:
- 2a.  $X \mapsto \mathbf{F}$ 2b.  $X \mapsto \mathbf{T}$  $\{\{Y\}, \{\neg Y\}\}$  $\{\{\neg Y\}\}$ 3a. unit propagation:  $Y \mapsto \mathbf{T}$

Systematic Search: DPLL

DPLL on Horn Formulas

Summary 00

# Example (1)

$$\Delta = \{\{X, Y, \neg Z\}, \{\neg X, \neg Y\}, \{Z\}, \{X, \neg Y\}\}$$

- unit propagation:  $Z \mapsto \mathbf{T}$  $\{\{X, Y\}, \{\neg X, \neg Y\}, \{X, \neg Y\}\}$
- Splitting rule:
- 2a.  $X \mapsto \mathbf{F}$ <br/> $\{\{Y\}, \{\neg Y\}\}$ 2b.  $X \mapsto \mathbf{T}$ <br/> $\{\{\neg Y\}\}$ 3a. unit propagation:  $Y \mapsto \mathbf{T}$ <br/> $\{\Box\}$ 3b. unit propagation:  $Y \mapsto \mathbf{F}$ <br/> $\{\}$

Systematic Search: DPLL

DPLL on Horn Formulas

Summary 00

# Example (1)

$$\Delta = \{\{X, Y, \neg Z\}, \{\neg X, \neg Y\}, \{Z\}, \{X, \neg Y\}\}$$

- unit propagation:  $Z \mapsto T$ {{X, Y}, { $\neg X, \neg Y$ }, { $X, \neg Y$ }}
- Splitting rule:
- 2a.  $X \mapsto \mathbf{F}$ <br/> $\{\{Y\}, \{\neg Y\}\}$ 2b.  $X \mapsto \mathbf{T}$ <br/> $\{\{\neg Y\}\}$ 3a. unit propagation:  $Y \mapsto \mathbf{T}$ <br/> $\{\Box\}$ 3b. unit propagation:  $Y \mapsto \mathbf{F}$ <br/> $\{\}$

Systematic Search: DPLL

DPLL on Horn Formulas

Summary 00

# Example (2)

Systematic Search: DPLL

DPLL on Horn Formulas

Summary 00

# Example (2)

#### $\Delta = \{\{W, \neg X, \neg Y, \neg Z\}, \{X, \neg Z\}, \{Y, \neg Z\}, \{Z\}\}$

• unit propagation:  $Z \mapsto \mathbf{T}$ 

Systematic Search: DPLL

DPLL on Horn Formulas

Summary 00

## Example (2)

#### $\Delta = \{\{W, \neg X, \neg Y, \neg Z\}, \{X, \neg Z\}, \{Y, \neg Z\}, \{Z\}\}$

• unit propagation:  $Z \mapsto \mathbf{T}$ {{ $W, \neg X, \neg Y$ }, {X}, {Y}}

Systematic Search: DPLL

DPLL on Horn Formulas

Summary 00

# Example (2)

- unit propagation:  $Z \mapsto \mathbf{T}$ {{ $W, \neg X, \neg Y$ }, {X}, {Y}}
- unit propagation:  $X \mapsto \mathbf{T}$  $\{\{W, \neg Y\}, \{Y\}\}$

# Example (2)

- unit propagation:  $Z \mapsto \mathbf{T}$  $\{\{W, \neg X, \neg Y\}, \{X\}, \{Y\}\}$
- unit propagation:  $X \mapsto \mathbf{T}$  $\{\{W, \neg Y\}, \{Y\}\}$
- unit propagation:  $Y \mapsto \mathbf{T}$ { $\{W\}$ }

# Example (2)

- unit propagation:  $Z \mapsto \mathbf{T}$  $\{\{W, \neg X, \neg Y\}, \{X\}, \{Y\}\}$
- unit propagation:  $X \mapsto \mathbf{T}$  $\{\{W, \neg Y\}, \{Y\}\}$
- unit propagation:  $Y \mapsto \mathbf{T}$ {{W}}
- unit propagation:  $W \mapsto \mathbf{T}$ {}

# Example (2)

- unit propagation:  $Z \mapsto T$ { $\{W, \neg X, \neg Y\}, \{X\}, \{Y\}\}$
- unit propagation:  $X \mapsto T$  $\{\{W, \neg Y\}, \{Y\}\}$
- unit propagation:  $Y \mapsto T$ {{W}}
- unit propagation:  $W \mapsto T$ {}

## Properties of DPLL

- DPLL is sound and complete.
- DPLL computes a model if a model exists.
  - Some variables possibly remain unassigned in the solution *I*; their values can be chosen arbitrarily.
- time complexity in general exponential
- important in practice: good variable order and additional inference methods (in particular clause learning)
  - Best known SAT algorithms are based on DPLL.

Systematic Search: DPL

DPLL on Horn Formulas

Summary 00

# **DPLL** on Horn Formulas

DPLL on Horn Formulas

Summary 00

#### Horn Formulas

important special case: Horn formulas

#### Definition (Horn formula)

A Horn clause is a clause with at most one positive literal, i.e., of the form

$$\neg x_1 \lor \cdots \lor \neg x_n \lor y \text{ or } \neg x_1 \lor \cdots \lor \neg x_n$$

```
(n = 0 \text{ is allowed.})
```

A Horn formula is a propositional formula in conjunctive normal form that only consists of Horn clauses.

- foundation of logic programming (e.g., PROLOG)
- critical in many kinds of practical reasoning problems

DPLL on Horn Formulas

Summary

## DPLL on Horn Formulas

#### Proposition (DPLL on Horn formulas)

If the input formula  $\varphi$  is a Horn formula, then the time complexity of DPLL is polynomial in the length of  $\varphi$ .

#### Proof.

#### properties:

- If  $\Delta$  is a Horn formula, then so is simplify $(\Delta, v, d)$ . (Why?)
  - → all formulas encountered during DPLL search are Horn formulas if input is Horn formula
- Every Horn formula without empty or unit clauses is satisfiable:
  - all such clauses consist of at least two literals
  - Horn property: at least one of them is negative
  - assigning F to all variables satisfies formula

DPLL on Horn Formulas

Summary 00

## DPLL on Horn Formulas (Continued)

#### Proof (continued).

- From 2. we can conclude:
  - if splitting rule applied, then current formula satisfiable, and
  - if a wrong decision is taken, then this will be recognized without applying further splitting rules (i.e., only by applying unit propagation and by deriving the empty clause).
- Hence the generated search tree for *n* variables can only contain at most *n* nodes where the splitting rule is applied (i.e., where the tree branches).
- It follows that the search tree is of polynomial size, and hence the runtime is polynomial.

Systematic Search: DPL

DPLL on Horn Formulas

Summary •0

# Summary



- satisfiability basic problem in propositional logic to which other problems can be reduced
- here: satisfiability for CNF formulas
- Davis-Putnam-Logemann-Loveland procedure (DPLL): systematic backtracking search with unit propagation as inference method
- DPLL successful in practice, in particular when combined with other ideas such as clause learning
- polynomial on Horn formulas
  - (= at most one positive literal per clause)