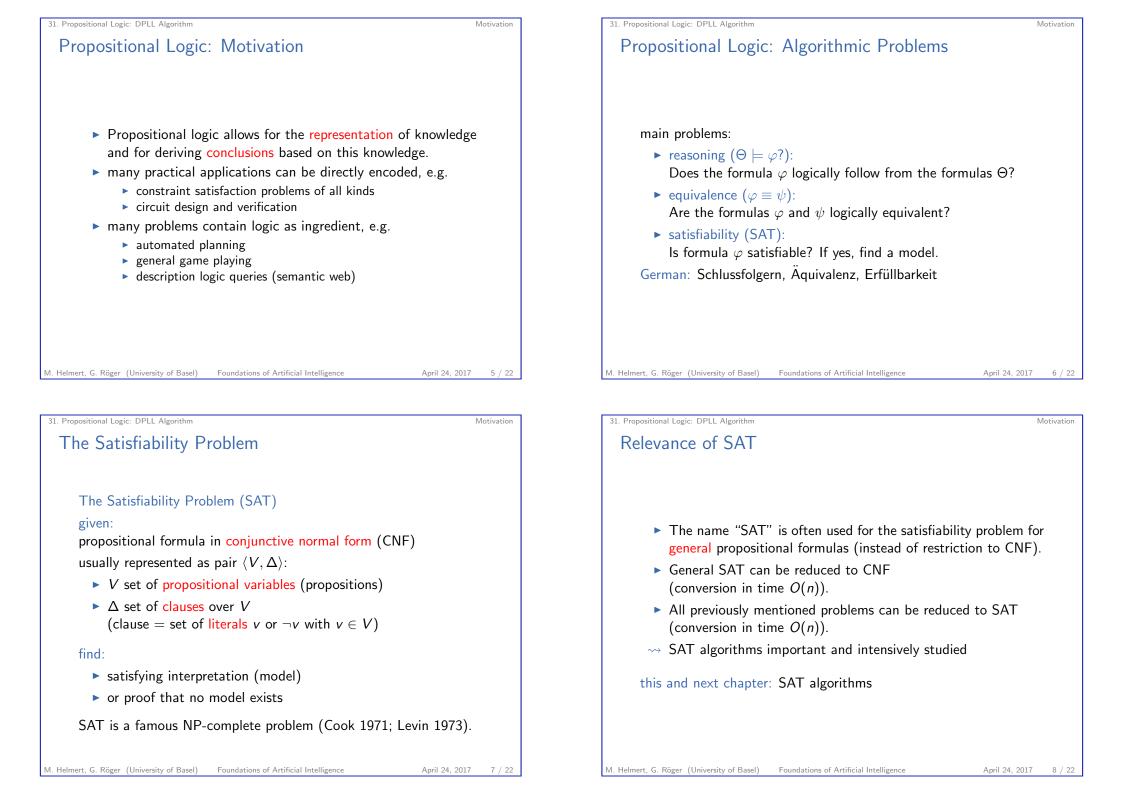


Foundations of Artificial Intelligence April 24, 2017 — 31. Propositional Logic: DPLL Algorithm		
31.1 Motivation		
31.2 Systematic Search: DPLL		
31.3 DPLL on Horn Formulas		
31.4 Summary		
M. Helmert, G. Röger (University of Basel) Foundations of Artificial Intelligence	April 24, 2017	2 / 22

# 31.1 Motivation

31. Propositional Logic: DPLL Algorithm

M. Helmert, G. Röger (University of Basel) Foundations of Artificial Intelligence



# 31.2 Systematic Search: DPLL

M. Helmert, G. Röger (University of Basel) Foundations of Artificial Intelligence

31. Propositional Logic: DPLL Algorithm

Systematic Search: DPLL

9 / 22

April 24, 2017

### 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
  - $\rightsquigarrow$  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

#### 31. Propositional Logic: DPLL Algorithm

## 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.

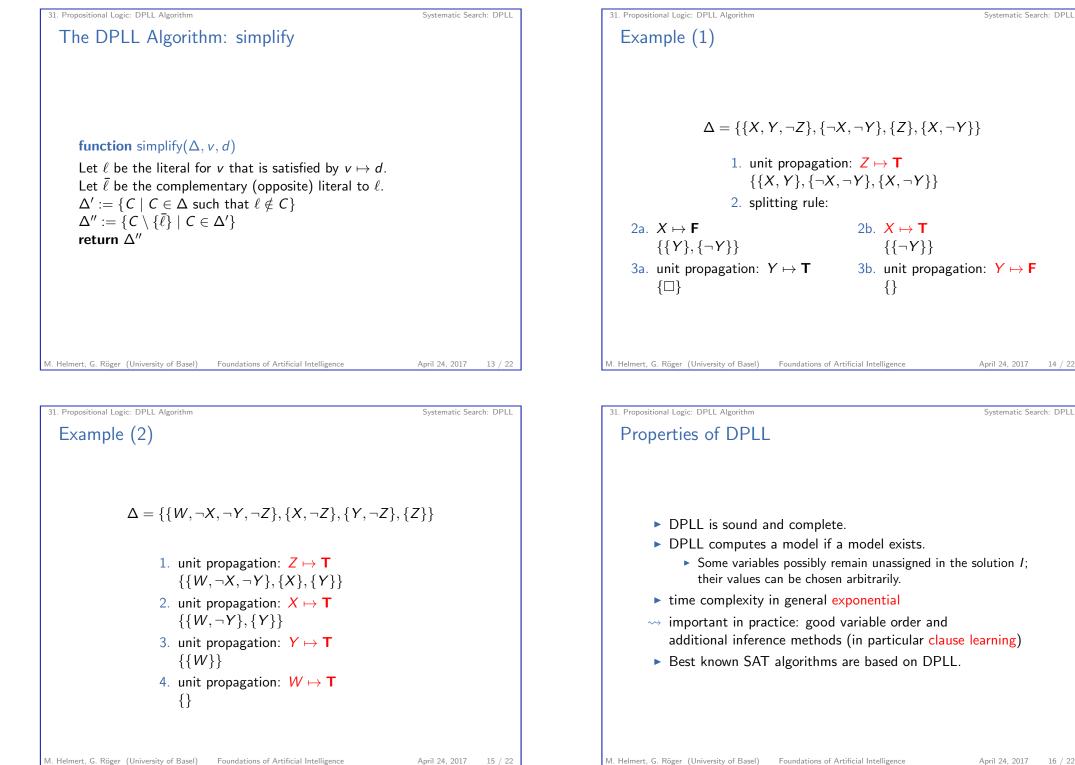
Due to this relationship, all ideas for CSPs are applicable to SAT:

- ► search
- ► inference
- variable and value orders

M. Helmert, G. Röger (University of Basel) Foundations of Artificial Intelligence

April 24, 2017 10 / 22

#### 31. Propositional Logic: DPLL Algorithm Systematic Search: DPLL The DPLL Algorithm: Pseudo-Code function DPLL( $\Delta$ , *I*): if $\Box \in \Delta$ : [empty clause exists $\rightsquigarrow$ unsatisfiable] return unsatisfiable else if $\Delta = \emptyset$ : [no clauses left $\rightsquigarrow$ 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 $l' \neq$ unsatisfiable return /' return unsatisfiable



April 24, 2017 16 / 22

14 / 22

April 24, 2017

DPLL on Horn Formulas

17 / 22

# 31.3 DPLL on Horn Formulas

M. Helmert, G. Röger (University of Basel) Foundations of Artificial Intelligence

31. Propositional Logic: DPLL Algorithm

### 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:

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

### 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 is allowed.)

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

German: Hornformel

- foundation of logic programming (e.g., PROLOG)
- hot research topic in program verification

M. Helmert, G. Röger (University of Basel) Foundations of Artificial Intelligence

April 24, 2017 18 / 22

# 31. Propositional Logic: DPLL Algorithm

DPLL on Horn Formulas

## DPLL on Horn Formulas (Continued)

### Proof (continued).

- 3. 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).
- 4. 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).
- 5. It follows that the search tree is of polynomial size, and hence the runtime is polynomial.

. . .

