Today

• Representation in Propositional Logic
• Semantics & Deduction
• Axioms and Facts
• Boolean Algebras
• Generate and Test
  – Checking Satisfiability (SAT) using DPLL
Representing Knowledge

• Propositional symbols represent facts under consideration:
  – there_is_rain, there_are_clouds, door1_open, robot_in_pos_56_210

• Not propositions:
  – is_there_rain?
  – location_of_robot
  – Dan_Roth
Truth Values

• Logical formulas: atoms, connectives, sentences
• Truth Assignments
• Evaluating the truth value of a formula
• Truth tables
Propositional Logic

- **Semantics:**
  - Truth assignments that satisfy KB/formula

| \( I_1 \) | -a | -b |
| \( I_2 \) | a  | -b |
| \( I_3 \) | -a | b  |
| \( I_4 \) | a  | b  |

\((a \land b) \lor (\neg a \land b)\)

**Interpretations:** \( I_1[a]=FALSE \quad I_1[b]=FALSE \)

assign truth values to propositional symbols
Propositional Logic

- **Semantics:**
  - Truth assignments that satisfy KB/formula

\[
\begin{array}{c|c|c}
\text{l}_1 & -a & b \\
\text{l}_2 & a & b \\
\hline
\text{M}_1 = \text{l}_3 & -a & b \\
\text{M}_2 = \text{l}_4 & a & b \\
\end{array}
\]

\[\models (a \land b) \lor (\neg a \land b)\]

**Models of f:** Interpretations that satisfy f
Propositional Logic

- Semantics:
  - Truth assignments that satisfy KB/formula

\[ \models (a \land b) \lor (\neg a \land b) \]

\[ M_1 = I_3 \]
\[ M_2 = I_4 \]

Models of \( f \): Interpretations that satisfy \( f \)
Propositional Logic

• Semantics:
  – Truth assignments that satisfy KB/formula

\[ M_1 \models (a \land b) \lor (\neg a \land b) \]

\[ M_2 = I_4 \]

Models of \( f \): Interpretations that satisfy \( f \)
Propositional Logic

• Semantics:
  – Truth assignments that satisfy KB/formula

Logical Entailment

\[ M_1 \models (a \land b) \lor (\neg a \land b) \]
Propositional Logic

• Semantics:
  – Truth assignments that satisfy KB/formula

Logical Entailment

\[ M_1 \models (a \land b) \lor (\neg a \land b) \]

\[ b \models (a \land b) \lor (\neg a \land b) \]
Propositional Logic

• Semantics:
  – Truth assignments that satisfy KB/formula

Logical Entailment

\[ M_1 \models (a \land b) \lor (\neg a \land b) \]

\[ b \models (a \land b) \lor (\neg a \land b) \]

\[ a \models (a \land b) \lor (\neg a \land b) \]
Propositional Logic

• Semantics:
  - Truth assignments that satisfy KB/formula

Logical Entailment

\[ M_1 \models (a \land b) \lor (\neg a \land b) \]
\[ b \models (a \land b) \lor (\neg a \land b) \]
\[ a \not\models (a \land b) \lor (\neg a \land b) \]
Propositional Logic

• Semantics:
  – Truth assignments that satisfy KB/formula

Logical
Entailment

\[ M_1 \models (a \land b) \lor (\neg a \land b) \]
\[ b \models (a \land b) \lor (\neg a \land b) \]
\[ a \not\models (a \land b) \lor (\neg a \land b) \]
\[ b \land a \models (a \land b) \lor (\neg a \land b) \]
Propositional Logic

• Semantics:
  – Truth assignments that satisfy KB/formula

Logical Entailment

\[ M_1 \models (a \land b) \lor (\neg a \land b) \]
\[ b \models (a \land b) \lor (\neg a \land b) \]
\[ a \not\models (a \land b) \lor (\neg a \land b) \]
\[ b \land a \models (a \land b) \lor (\neg a \land b) \]
\[ TRUE \models (a \land b) \lor (\neg a \land b) \]
Propositional Logic

• Semantics:
  – Truth assignments that satisfy KB/formula

Logical Entailment

\[ M_1 \models (a \land b) \lor (\neg a \land b) \]
\[ b \models (a \land b) \lor (\neg a \land b) \]
\[ a \not\models (a \land b) \lor (\neg a \land b) \]
\[ b \land a \models (a \land b) \lor (\neg a \land b) \]
\[ TRUE \not\models (a \land b) \lor (\neg a \land b) \]
Examples

• Double negation, double disjunction

• De-Morgan’s law

\[ \neg(p \land q) \iff (\neg p) \lor (\neg q) \]

• Distributivity of conjunction and disjunction
Propositional Logic

• Semantics:
  – Truth assignments that satisfy KB/formula

Logical Entailment

\[ b \models (a \land b) \lor (\neg a \land b) \]
Propositional Logic

• Semantics:
  – Truth assignments that satisfy KB/formula

Logical Entailment

\[ b \models (a \land b) \lor (\neg a \land b) \]

Deduction (inference)

\[ b \vdash (a \land b) \lor (\neg a \land b) \]
Notations

- Interpretations ~ Models
- Axioms – formulae that are “assumed”
- Signature – the symbols used by a KB
- Theory ~ KB (a set of axioms), or
- Theory ~ the complete set of sentences entailed by the axioms
- The value that symbol p takes in model M: $M[p]$
- Clauses: {lit1, lit2, lit3,...} or lit1 ∨ lit2 ∨ lit3...
Representing Knowledge

• Knowledge bases are sets of formulae
  – There_is_rain → there_are_clouds
  – Robot_in_pos_3_1 → ¬Position_3_1_empty
  – Has_drink → coffee ∨ tea
Knowledge Engineering

- Select a language: set of features
- Examine cases
- Decide on dependencies between features
- Write dependencies formally
- Test
Clausal Form

• Every formula can be reformulated into an equivalent CNF formula (conjunction of clauses).

• Examples (by Distributivity):

\[(a \land b) \lor (\neg a \land b)\]
Clausal Form

• Every formula can be reformulated into an equivalent CNF formula (conjunction of clauses).

• Examples (by Distributivity):

\[(a \land b) \lor (\neg a \land b) \equiv (a \lor \neg a)\]
Clausal Form

• Every formula can be reformulated into an equivalent CNF formula (conjunction of clauses).

• Examples (by Distributivity):

\[(a \land b) \lor (\neg a \land b) \equiv (a \lor \neg a) \land (a \lor b)\]
Clausal Form

• Every formula can be reformulated into an equivalent CNF formula (conjunction of clauses).

• Examples (by Distributivity):

\[(a \land b) \lor (\neg a \land b) \equiv (a \lor \neg a) \land (a \lor b) \land (b \lor \neg a)\]
Clausal Form

• Every formula can be reformulated into an equivalent CNF formula (conjunction of clauses).

• Examples (by Distributivity):

\[(a \land b) \lor (\neg a \land b) \equiv (a \lor \neg a) \land (a \lor b) \land (b \lor \neg a) \land (b \lor b)\]
Clausal Form

• Every formula can be reformulated into an equivalent CNF formula (conjunction of clauses).

• Examples:

  Face_door_t1 ^ turn_cl_90_t1 -> ~face_door_t2

  ~Face_door_t1 v ~turn_cl_90_t1 v ~face_door_t2
Clausal Form

• Every formula can be reformulated into an equivalent CNF formula (conjunction of clauses).

• Examples:

\[ \text{face\_door\_t1} \land \text{move\_fwd\_t1} \rightarrow \text{at\_corridor\_t2} \land \neg \text{face\_door\_t2} \equiv \]

\[ \neg \text{face\_door\_t1} \lor \neg \text{move\_fwd\_t1} \lor \neg \text{at\_corridor\_t2} \]

\[ \neg \text{face\_door\_t1} \lor \neg \text{move\_fwd\_t1} \lor \neg \text{face\_door\_t2} \]

\[ \neg \text{face\_door\_t1} \lor \neg \text{move\_fwd\_t1} \lor \neg \text{face\_door\_t2} \]
Application: Hardware Verification

\[ x_1 \rightarrow \text{AND} \rightarrow f_1 \rightarrow \text{not} \rightarrow f_3 \rightarrow \text{AND} \rightarrow f_5 \]

\[ x_2 \rightarrow \text{not} \rightarrow f_2 \rightarrow \text{OR} \rightarrow f_4 \]

\[ x_3 \]
Application: Hardware Verification

Question: Can we set this boolean circuit to TRUE?
Application: Hardware Verification

Question: Can we set this boolean circuit to TRUE?

\[ f_5(x_1, x_2, x_3) = \text{a function of the input signal} \]
Application: Hardware Verification

\[ f_5(x_1, x_2, x_3) = f_3 \land f_4 = \neg f_1 \land (f_2 \lor x_3) = \neg(x_1 \land x_2) \land (\neg x_2 \lor x_3) \]

\[ M[x_1] = \text{FALSE} \]
\[ M[x_2] = \text{FALSE} \]
\[ M[x_3] = \text{FALSE} \]

Question: Can we set this boolean circuit to TRUE?

\[ \text{SAT}(f_5) ? \]
Hardware Verification

• Questions in logical circuit verification
  – Equivalence of circuits
  – Arrival of the circuit to a state (required a temporal model, which gets propositionalized)
  – Achieving an output from the circuit
SATisfiability of Logical Formulas

• Given KB in CNF
  – If we have a truth table of KB, then we can check that KB satisfiable by looking at it.
SATisfiability of Logical Formulas

• Given KB in CNF
  – If we have a truth table of KB, then we can check that KB satisfiable by looking at it.

• Problem: n propositional symbols $\rightarrow 2^n$ rows in truth table
  – Checking truth table with KB takes time $O(|KB|)$
  – Generating table is expensive: $O(2^n |KB|)$ time
SATisfiability of Logical Formulas

• Given KB in CNF
  – If we have a truth table of KB, then we can check that KB satisfiable by looking at it.

• Problem: n propositional symbols $\rightarrow 2^n$ rows in truth table
  – Checking truth table with KB takes time $O(|KB|)$
  – Generating table is expensive: $O(2^n |KB|)$ time

• Observation: SAT requires us to look only for one model
DPLL (1960) Procedure for CNF

1. If KB is empty: “True”
2. If KB has empty clause: “False”
3. If KB has a unit clause (a literal)
   – propagate literal to simplify clauses
4. else: Choose a variable heuristically
   – Set var True, Recurse
   – Set var False, Recurse
   – Return value of recursion

Simplify if possible.
Search for satisfying assignment.
Just a recursive search.
Propagating a Truth-Value

• KB in CNF, and we observe \( a = \text{TRUE} \)
• Then, removing clauses with \( a \) positive from KB gives an equivalent theory.
• Example:

\[
\text{KB} \\
\vdash a \lor b \lor c \\
\vdash \neg a \lor b \lor d
\]
Propagating a Truth-Value

• KB in CNF, and we observe $a=\text{TRUE}$
• Then, removing clauses with $a$ positive from KB gives an equivalent theory.
• Example:

$$KB$$

$$a \lor b \lor c$$

$$\neg a \lor b \lor d$$

Observe

$$a$$
Propagating a Truth-Value

• KB in CNF, and we observe a=TRUE
• Then, removing clauses with a positive from KB gives an equivalent theory.
• Example:

\[
\begin{align*}
\text{KB} & \quad \left\{ \begin{array}{c}
a \lor b \lor c \\
\neg a \lor b \lor d \\
a
\end{array} \right\} \\
\text{Observe} & \quad \left\{ \begin{array}{c}
a \lor b \lor c \\
\neg a \lor b \lor d \\
a
\end{array} \right\}
\end{align*}
\]
Propagating a Truth-Value

- KB in CNF, and we observe \( a = \text{TRUE} \)
- Then, \textbf{removing clauses with} \( a \) \textbf{positive} \textbf{from} KB gives an equivalent theory.
- Example:

\[
\text{KB} = \{ a \lor b \lor c, \neg a \lor b \lor d \} \\
\text{Observe} = \{ a \}
\]
Propagating a Truth-Value

• KB in CNF, and we observe $a=\text{TRUE}$
• Then, removing negative instances of $a$ from KB gives an equivalent theory.
• Example:

\[
\begin{align*}
\text{KB:} & \quad \{a \lor b \lor c\} \\
\text{Observe:} & \quad \{\neg a \lor b \lor d\} \\
& \quad a
\end{align*}
\]
Propagating a Truth-Value

- KB in CNF, and we observe $a=\text{TRUE}$
- Then, removing negative instances of $a$ from KB gives an equivalent theory.
- Example:
DPLL Example
DPLL Example

• $p \text{ XOR } q = (p \lor q) \land (-p \lor -q)$
DPLL Example

• \( p \text{ XOR } q = (p \lor q) \land (-p \lor -q) \)
• \( p \text{ IFF } q = (p \text{ XOR } -q) \)
DPLL Example

• \( p \text{ XOR } q = (p \lor q) \land (\neg p \lor \neg q) \)
• \( p \text{ IFF } q = (p \text{ XOR } \neg q) \)

Not SAT
DPLL Example

• $p \text{ XOR } q = (p \lor q) \land (-p \lor -q)$
• $p \text{ IFF } q = (p \text{ XOR } \neg q)$

• $(p \text{ XOR } q) \land (p \text{ XOR } \neg q)$

Not SAT
DPLL Example

• $p \text{ XOR } q = (p \lor q) \land (\neg p \lor \neg q)$
• $p \text{ IFF } q = (p \text{ XOR } \neg q)$

• $(p \text{ XOR } q) \land (p \text{ XOR } \neg q)$
• $\cdots((p_1 \text{ XOR } p_2) \text{ XOR } p_3) \text{ XOR } p_4\cdots) \land$
DPLL Example

• \( p \ XOR \ q = (p \lor q) \land (-p \lor -q) \)
• \( p \ IFF \ q = (p \ XOR \ -q) \)

\[(p \ XOR \ q) \land (p \ XOR \ -q)\]
\[\ldots(((p1 \ XOR \ p2) \ XOR \ p3) \ XOR \ p4\ldots) \land \ldots(((p1 \ XOR \ p2) \ XOR \ p3) \ XOR \ -p4\ldots)\]
DPLL Example

- $p \text{ XOR } q = (p \lor q) \land (-p \lor -q)$
- $p \text{ IFF } q = (p \text{ XOR } -q)$
- $(p \text{ XOR } q) \land (p \text{ XOR } -q)$
- $…(((p_1 \text{ XOR } p_2) \text{ XOR } p_3) \text{ XOR } p_4…) \land \ldots (((p_1 \text{ XOR } p_2) \text{ XOR } p_3) \text{ XOR } -p_4…) \ldots$

DPLL takes $O(2^n)$ time, sometimes
SAT Solving Topics

- Order of selection of variables
- Stochastic local search
- Binary Decision Diagrams
- Strategies other than unit
- 2-SAT is solvable in linear time
- Smart backtracking
- Clauses/Vars in Random SAT
- SAT via probabilistic models