COURSE STRUCTURE

CONSTRAINT SOLVERS

- Propositional Logic, SAT solving, DPLL
- First-Order Logic, SMT
- First-Order Theories

DEDUCTIVE VERIFICATION

- Operational Semantics
- Strongest Post-condition, Weakest Pre-condition
- Hoare Logic

FULLY AUTOMATED VERIFICATION

- Abstract Interpretation/Static Analysis
- Model Checking, Predicate Abstraction, CEGAR
- Property-directed Reachability

PROPOSITIONAL LOGIC

MATHEMATICAL LOGIC

- Logic is the foundation of computation.
- We will use logic for multiple purposes:
  - Describing specifications
  - Describing program executions
  - Mathematical guarantees of logic will translate to guarantees of program correctness
  - Decision procedures for logic will be used for verification.

PROPOSITIONAL LOGIC

Is $(p \wedge (p \rightarrow q)) \rightarrow q$ valid?

Is $p \wedge \bot \rightarrow \neg q \vee \top$ satisfiable?

SYNTAX

Atom	Truth Values - $\perp$: False, $\top$: True Propositional Variables - p,q,r…
Logical Connectives	$\wedge$: and, $\vee$: or, $\neg$: not, $\rightarrow$: implies, $\leftrightarrow$: if and only if(iff)
Literal	Atom or its negation
Formula	A literal or the application of logical connectives to formulae

SEMANTICS

Interpretation I

I : Set of Propositional Variables → { ⊥ , ⊤ }

MODEL OF

Given an interpretation I and Formula F,

$I \models F$	F evaluates to ⊤ under I
$I \not\models F$	F evaluates to ⊥ under I

SEMANTICS: INDUCTIVE DEFINITION

Base Case:

$I \models \top$	
$I \not\models \bot$	
$I \models p$	iff I(p)=$\top$
$I \not\models p$	iff I(p)=$\bot$

Inductive Case:

$I \models \neg F$	iff $I \not\models F$
$I \models F_1 \wedge F_2$	iff $I \models F_1$ and $I \models F_2$
$I \models F_1 \vee F_2$	iff $I \models F_1$ or $I \models F_2$
$I \models F_1 \rightarrow F_2$	iff $I \not\models F_1$ or $I \models F_2$
$I \models F_1 \leftrightarrow F_2$	iff $I \models F_1$ and $I \models F_2$, or $I \not\models F_1$ and $I \not\models F_2$

EXAMPLE

$I = \{p : \top, q : \bot\}$

$F = p \wedge q \rightarrow p \vee \neg q$

Is $I \vDash F$?

1. $I \nvDash q$
2. $I \nvDash p \wedge q$
3. $I \vDash p \wedge q \rightarrow p \vee \neg q$

PRECEDENCE OF LOGICAL CONNECTIVES

- We assume the following precedence from highest to lowest:
  - $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
  - Example: $\neg p \wedge q \rightarrow p \vee q \wedge r$ is the same as $((\neg p) \wedge q) \rightarrow (p \vee (q \wedge r))$.
- We assume that all logical connectives associate to the right.
  - Example: $p \rightarrow q \rightarrow r$ is the same as $p \rightarrow (q \rightarrow r)$
- Parenthesis can be used to change precedence or associativity.

SATISFIABILITY AND VALIDITY

- A formula F is satisfiable iff there exists an interpretation I such that $I \vDash F$.
- A formula F is valid iff for all interpretations I, $I \vDash F$.
- A formula F is valid iff $\neg F$ is unsatisfiable.
  - A Decision Procedure for satisfiability is therefore also a decision procedure for validity. How?

QUESTIONS

- A formula can either be SAT, UNSAT or VALID.
  - Does Validity ⇒ Satisfiability?
  - Does Satisfiability ⇒ Validity?
- Can a decision procedure for Validity be used as a decision procedure for Satisfiability?
  - F is satisfiable iff ¬F is not valid.
- Homework: Are the following formulae are sat, unsat or valid?
  - $p \wedge q \rightarrow p \vee q$
  - $p \vee q \rightarrow \neg p \vee \neg q$
  - $(p \rightarrow q \rightarrow r) \wedge (p \wedge q \wedge \neg r)$

MORE TERMINOLOGY

- Formulae F_1 and F_2 are equivalent (denoted by $F_1 \Leftrightarrow F_2$) when the formula $F_1 \leftrightarrow F_2$ is valid.
  - Example: $p \rightarrow q \Leftrightarrow \neg p \vee q$
  - Another definition: F_1 and F_2 are equivalent if for all interpretations I, $I \vDash F_1$ if and only if $I \vDash F_2$.
- Formula F_1 implies F_2 (denoted by $F_1 \Rightarrow F_2$) when the formula $F_1 \rightarrow F_2$ is valid.
  - Example: $(p \rightarrow q) \wedge p \Rightarrow q$
- Formulae F_1 and F_2 are equisatisfiable when F_1 is satisfiable if and only if F_2 is satisfiable.
  - Example: $p \wedge (q \vee r)$ and $q \vee r$ are equisatisfiable

MORE EXAMPLES

- Which of the following are true?
  - $\neg(F_1 \wedge F_2) \Leftrightarrow \neg F_1 \vee \neg F_2$
  - $(F_1 \leftrightarrow F_2) \wedge (F_2 \leftrightarrow F_3) \Rightarrow (F_1 \leftrightarrow F_3)$
  - $p \Leftrightarrow p \wedge q$
  - p and q are equisatisfiable.
- What is the simplest example of two formulae which are not equisatisfiable?

DECISION PROCEDURES FOR SATISFIABILITY AND VALIDITY

- Two methods
  - Truth Tables: Search for satisfying interpretation
  - Semantic Argument: Rule-based deductive approach
- Modern SAT solvers use combination of both approaches

TRUTH TABLES - EXAMPLE

$$p \wedge q \to p \vee \neg q$$

p	q	$\neg q$	$p \wedge q$	$p \vee \neg q$	$p \wedge q \to p \vee \neg q$
0	0	1	0	1	1
0	1	0	0	0	1
1	0	1	0	1	1
1	1	0	1	1	1

TRUTH TABLES - EXAMPLE

$p \wedge q \rightarrow p \vee \neg q$ is valid

p	q	$\neg q$	$p \wedge q$	$p \vee \neg q$	$p \wedge q \rightarrow p \vee \neg q$
0	0	1	0	1	1
0	1	0	0	0	1
1	0	1	0	1	1
1	1	0	1	1	1

SEMANTIC ARGUMENT METHOD

- Deductive approach for showing validity based on proof rules
- Main Idea: Proof by Contradiction.
  - Assume that a falsifying interpretation exists.
  - Use proof rules to deduce more facts.
  - Find contradictory facts.

PROOF RULES (NEGATION)

$$\frac{I \models \neg F}{I \not\models F}$$

PROOF RULES (NEGATION)

PREMISE

$$\frac{I \models \neg F}{I \not\models F}$$

PROOF RULES (NEGATION)

PREMISE

$$\frac{I \models \neg F}{I \not\models F}$$

DEDUCTION

PROOF RULES (NEGATION)

$$\frac{I \models \neg F}{I \not\models F}$$

PREMISE

DEDUCTION

$$\frac{I \not\models \neg F}{I \models F}$$

PROOF RULES (CONJUNCTION)

$$\frac{I \models F \land G}{\begin{array}{l} I \models F \\ I \models G \end{array}}$$

PROOF RULES (CONJUNCTION)

$$\frac{I \models F \land G}{\begin{array}{l} I \models F \\ I \models G \end{array}}$$

$$\frac{I \not\models F \land G}{I \not\models F \quad | \quad I \not\models G}$$

PROOF RULES (CONJUNCTION)

$$\frac{I \models F \land G}{\begin{array}{l} I \models F \\ I \models G \end{array}}$$

$$\frac{I \not\models F \land G}{I \not\models F \quad | \quad I \not\models G}$$

BRANCHING:

Need to show a contradiction in every branch

PROOF RULES (DISJUNCTION)

$$\frac{I \models F \vee G}{I \models F \quad | \quad I \models G}$$

$$\frac{I \not\models F \vee G}{\begin{array}{l} I \not\models F \\ I \not\models G \end{array}}$$

PROOF RULES (IMPLICATION)

$$\frac{I \models F \to G}{I \not\models F \quad | \quad I \models G}$$

$$\begin{array}{l} I \not\models F \to G \\ \hline I \models F \\ I \not\models G \end{array}$$

PROOF RULES (IFF)

$$\frac{I \models F \leftrightarrow G}{I \models F \land G \quad | \quad I \not\models F \lor G}$$

$$\frac{I \not\models F \leftrightarrow G}{I \models F \land \neg G \quad | \quad I \models \neg F \land G}$$

PROOF RULES (CONTRADICTION)

$$\frac{\begin{array}{ccc} I & \models & F \\ I & \not\models & F \end{array}}{\begin{array}{ccc} I & \models & \bot \end{array}}$$

EXAMPLE

Prove that $p \wedge q \rightarrow p \vee \neg q$ is valid

EXAMPLE

Prove that $p \land q \rightarrow p \lor \neg q$ is valid

$$I \not\models p \land q \rightarrow p \lor \neg q$$

EXAMPLE

Prove that $p \land q \to p \lor \neg q$ is valid

$$\frac{I \not\models p \land q \to p \lor \neg q}{I \models p \land q \qquad I \not\models p \lor \neg q}$$

EXAMPLE

Prove that $p \wedge q \rightarrow p \vee \neg q$ is valid

$$\frac{I \not\models p \wedge q \rightarrow p \vee \neg q}{\dfrac{I \models p \wedge q}{\begin{array}{c} I \models p \\ I \models q \end{array}} \qquad \dfrac{I \not\models p \vee \neg q}{\begin{array}{c} I \not\models p \\ I \not\models \neg q \end{array}}}$$

EXAMPLE

Prove that $p \land q \rightarrow p \lor \neg q$ is valid

$$\cfrac{I \not\models p \land q \rightarrow p \lor \neg q}{\cfrac{I \models p \land q}{\begin{array}{c}\boxed{I \models p} \\ I \models q\end{array}} \quad \cfrac{I \not\models p \lor \neg q}{\begin{array}{c}\boxed{I \not\models p} \\ I \not\models \neg q\end{array}}}$$

CONTRADICTION

ANNOUNCEMENTS

- Lectures slides are available on the course moodle page.
    - Self-enroll is enabled on moodle.
- The text book (BM) is also uploaded on moodle.
    - Please try Exercises 1.1-1.5.

EXAMPLE WITH BRANCHING

Prove that $(p \to q) \land p \to q$ is valid

EXAMPLE WITH BRANCHING

Prove that $(p \rightarrow q) \wedge p \rightarrow q$ is valid

$$I \not\models (p \rightarrow q) \wedge p \rightarrow q$$

EXAMPLE WITH BRANCHING

Prove that $(p \rightarrow q) \land p \rightarrow q$ is valid

$$\dfrac{I \not\models (p \rightarrow q) \land p \rightarrow q}{I \models (p \rightarrow q \land p) \qquad I \not\models q}$$

EXAMPLE WITH BRANCHING

Prove that $(p \rightarrow q) \wedge p \rightarrow q$ is valid

$$\frac{I \nvDash (p \rightarrow q) \wedge p \rightarrow q}{\dfrac{I \vDash (p \rightarrow q \wedge p)}{I \vDash (p \rightarrow q) \quad I \vDash p} \quad I \nvDash q}$$

EXAMPLE WITH BRANCHING

Prove that $(p \to q) \land p \to q$ is valid

$$\dfrac{\dfrac{\dfrac{I \models (p \to q)}{I \not\models p \quad \mid \quad I \models q} \qquad I \models p}{I \models (p \to q \land p)} \qquad I \not\models q}{I \not\models (p \to q) \land p \to q}$$

EXAMPLE WITH BRANCHING

Prove that $(p \to q) \land p \to q$ is valid

$$\cfrac{I \nvDash (p \to q) \land p \to q}{\cfrac{I \vDash (p \to q \land p)}{\cfrac{I \vDash (p \to q)}{\boxed{I \nvDash p} \;\Big|\; I \vDash q} \quad \boxed{I \vDash p}} \quad I \nvDash q}$$

CONTRADICTION

EXAMPLE WITH BRANCHING

Prove that $(p \to q) \land p \to q$ is valid

$$I \not\models (p \to q) \land p \to q$$

$$I \models (p \to q \land p) \qquad \boxed{I \not\models q}$$

$$I \models (p \to q) \qquad I \models p$$

$$I \not\models p \;\Big|\; \boxed{I \models q}$$

CONTRADICTION

Each branch should lead to a contradiction

QUESTIONS

- Is the semantic argument method complete?
- Can we use the semantic argument method for satisfiability?
- What is the time complexity of the semantic argument method?

DECISION PROCEDURES FOR SAT

- We will go through the DPLL algorithm.
  - Davis-Putnam-Logemann-Loveland Algorithm
  - Combines truth table and deductive approaches
  - Requires formulae in Conjunctive Normal Form (CNF)
  - Forms the basis of modern SAT solvers

NORMAL FORMS

- A Normal Form of a formula F is another equivalent formula F′ which obeys some syntactic restrictions.
- Three important normal forms:
  - Negation Normal Form (NNF): Should use only $\neg$, $\wedge$, $\vee$ as the logical connectives, and $\neg$ should only be applied to literals
  - Disjunctive Normal Form (DNF): Should be a disjunction of conjunction of literals
  - Conjunctive Normal Form (CNF): Should be a conjunction of disjunction of literals

CONJUNCTIVE NORMAL FORM

- A conjunction of disjunction of literals

$$\bigwedge_i \bigvee_j \ell_{i,j} \quad \text{for literals } \ell_{i,j}$$

- Each inner disjunct is also called a clause
- Is every formula in CNF also in NNF?

CNF CONVERSION

- We can use distribution of $\vee$ over $\wedge$ to obtain formula in CNF
  - $F_1 \vee (F_2 \wedge F_3) \Leftrightarrow (F_1 \vee F_2) \wedge (F_1 \vee F_3)$
  - Causes exponential blowup.
- Tseitin's transformation algorithm can be used to obtain an equisatisfiable CNF formula linear in size
  - BM Chapter 1

TRUTH TABLE BASED METHOD

Decision Procedure for Satisfiability:
Returns true if F is SAT, false if F is UNSAT

```
SAT(F){
  if (F = ⊤) return true;
  if (F = ⊥) return false;
  Choose a variable p in F;
  return SAT(F[⊤/p]) ∨ SAT(F[⊥/p]);
}
```

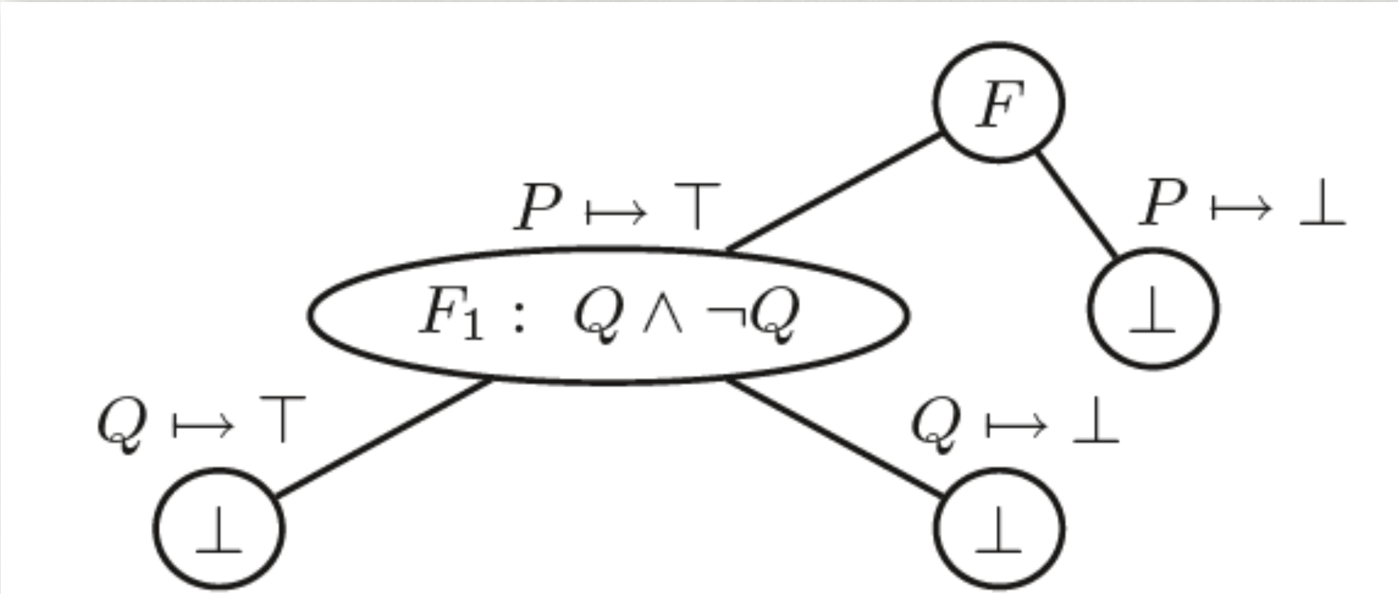
F[G/P] : G REPLACES EVERY OCCURRENCE OF P IN F, THEN SIMPLIFY

SIMPLIFICATION

- Following equivalences can be used to simplify:
  - $F \wedge \bot \Leftrightarrow \bot$
  - $F \wedge \top \Leftrightarrow F$
  - $F \vee \bot \Leftrightarrow F$
  - $F \vee \top \Leftrightarrow \top$
- Note that these equivalences would be applied syntactically.
  - That is, if the formula contains a $\top$ or $\bot$, it would be re-written according to the above equivalences.

EXAMPLE

- SAT$((P \to Q) \land P \land \neg Q)$
- $F = (\neg P \lor Q) \land P \land \neg Q$
- $F[\top / P] \triangleq (\bot \lor Q) \land \top \land \neg Q \equiv Q \land \neg Q$



SIMPLIFICATION MAY SAVE BRANCHING ON SOME OCCASIONS

DEDUCTION: CLAUSAL RESOLUTION

FOR CNF

$$\frac{I \vDash p \vee F \qquad I \vDash \neg p \vee G}{I \vDash F \vee G}$$

[CLAUSAL RESOLUTION]

- Given a CNF Formula $F = C_1, C_2, \ldots C_n$, if C' is a resolvent deduced from F, then $F' = C_1, C_2, \ldots, C_n, C'$ is equivalent to F.
- Example: $F = (\neg P \vee Q) \wedge P \wedge \neg Q$
    - Rewritten as $F = (\neg P \vee Q) \wedge (P \vee \bot) \wedge \neg Q$
    - Resolvent: $(Q \vee \bot) = Q$
    - $F' = (\neg P \vee Q) \wedge P \wedge \neg Q \wedge Q \rightarrow$ The next resolvent will be $\bot$.
- Idea: Repeatedly apply clausal resolution until no more new clauses can be deduced. If $\bot$ is never deduced, then the formula is satisfiable.

DEDUCTION: UNIT RESOLUTION

FOR CNF

$$\frac{I \models p \qquad I \models \neg p \vee F}{I \models F} \quad \text{[UNIT RESOLUTION]}$$

In Unit Resolution, the resolvent replaces the original clause

BOOLEAN CONSTRAINT PROPAGATION (BCP)

FOR CNF

$$I \models p \wedge (\neg p \vee q) \wedge (r \vee \neg q \vee s)$$

BOOLEAN CONSTRAINT PROPAGATION (BCP)

FOR CNF

$$\frac{I \vDash p \wedge (\neg p \vee q) \wedge (r \vee \neg q \vee s)}{I \vDash q \wedge (r \vee \neg q \vee s)} \quad \text{[UNIT RESOLUTION]}$$

BOOLEAN CONSTRAINT PROPAGATION (BCP)

FOR CNF

$$\frac{\dfrac{I \vDash p \wedge (\neg p \vee q) \wedge (r \vee \neg q \vee s)}{I \vDash q \wedge (r \vee \neg q \vee s)}}{I \vDash r \vee s}$$

[UNIT RESOLUTION]

BOOLEAN CONSTRAINT PROPAGATION (BCP)

FOR CNF

$$\begin{array}{c} I \models p \wedge (\neg p \vee q) \wedge (r \vee \neg q \vee s) \\ \hline I \models q \wedge (r \vee \neg q \vee s) \\ \hline I \models r \vee s \end{array}$$

[UNIT RESOLUTION]

FIND A SATISFYING INTERPRETATION

PURE LITERAL PROPAGATION (PLP)

FOR CNF

- If a variable appears only positively or negatively in a formula, then all clauses containing the variable can be removed.
  - p appears positively if every $p-$literal is just p
  - p appears negatively if every $p-$literal is $\neg p$
- Removing such clauses from F results in a equisatisfiable formula F'
  - Why?
  - Are F and F' equivalent?

DPLL

FOR CNF

Decision Procedure for Satisfiability of CNF Formula:
Returns true if F is SAT, false if F is UNSAT

```
SAT(F){

  F’ = PLP(F);

  F’’ = BCP(F’);

  if (F’’ = ⊤) return true;

  if (F’’ = ⊥) return false;

  Choose a variable p in F’’;

  return SAT(F’’[⊤/p]) ∨ SAT(F’’[⊥/p]);

}
```

EXAMPLE

$$F : (\neg p \vee q \vee r) \wedge (\neg q \vee r) \wedge (\neg q \vee \neg r) \wedge (p \vee \neg q \vee \neg r)$$

- SAT(F)
  - No PLP or BCP.
  - q ← CHOOSE.
  - F[True/q] = $r \wedge \neg r \wedge (p \vee \neg r)$
- SAT(F[True/q])
  - After PLP: $r \wedge \neg r$
  - After BCP: False
  - Return False and backtrack to previous call

```
SAT(F){
  F' = PLP(F);
  F'' = BCP(F');
  if (F'' = ⊤) return true;
  if (F'' = ⊥) return false;
  Choose a variable p in
F'';
  return SAT(F''[⊤/p]) ∨
  SAT(F''[⊥/p]);
}
```

EXAMPLE

$$F : (\neg p \vee q \vee r) \wedge (\neg q \vee r) \wedge (\neg q \vee \neg r) \wedge (p \vee \neg q \vee \neg r)$$

- SAT(F)
  - No PLP or BCP.
  - q ← CHOOSE.
  -

```
SAT(F){

  F’ = PLP(F);

  F’’ = BCP(F’);

  if (F’’ = ⊤) return true;

  if (F’’ = ⊥) return false;

  Choose a variable p in
F’’;

  return SAT(F’’[⊤/p]) ∨
  SAT(F’’[⊥/p]);
}
```

EXAMPLE

$$F : (\neg p \vee q \vee r) \wedge (\neg q \vee r) \wedge (\neg q \vee \neg r) \wedge (p \vee \neg q \vee \neg r)$$

- SAT(F)
  - No PLP or BCP.
  - q ← CHOOSE.
  - F[False/q] = $\neg p \vee r$

```
SAT(F){
  F' = PLP(F);
  F'' = BCP(F');
  if (F'' = ⊤) return true;
  if (F'' = ⊥) return false;
  Choose a variable p in F'';
  return SAT(F''[⊤/p]) ∨
  SAT(F''[⊥/p]);
}
```

EXAMPLE

$$F : (\neg p \vee q \vee r) \wedge (\neg q \vee r) \wedge (\neg q \vee \neg r) \wedge (p \vee \neg q \vee \neg r)$$

- SAT(F)
  - No PLP or BCP.
  - q ← CHOOSE.
  - F[False/q] = $\neg p \vee r$
- SAT(F[False/q])
  -

```
SAT(F){

  F’ = PLP(F);

  F’’ = BCP(F’);

  if (F’’ = ⊤) return true;

  if (F’’ = ⊥) return false;

   Choose a variable p in
F’’;

   return SAT(F’’[⊤/p]) ∨
  SAT(F’’[⊥/p]);
}
```

EXAMPLE

$$F : (\neg p \vee q \vee r) \wedge (\neg q \vee r) \wedge (\neg q \vee \neg r) \wedge (p \vee \neg q \vee \neg r)$$

- SAT(F)
  - No PLP or BCP.
  - q ← CHOOSE.
  - F[False/q] = $\neg p \vee r$
- SAT(F[False/q])
  - After PLP: True
  - Satisfiable!

```
SAT(F){
  F’ = PLP(F);
  F’’ = BCP(F’);
  if (F’’ = ⊤) return true;
  if (F’’ = ⊥) return false;
  Choose a variable p in
F’’;
  return SAT(F’’[⊤/p]) ∨
  SAT(F’’[⊥/p]);
}
```

DPLL IS JUST THE STARTING POINT!

- Modern SAT solvers use a variety of approaches to further improve performance
  - Non-chronological back tracking
  - Conflict-driven clause learning (CDCL)
  - Heuristics to CHOOSE appropriate variables and assignments
- Current SAT solvers can solve problems with millions of clauses in reasonable amount of time on average.

ENCODING PROBLEMS IN PL

- Even though PL is relatively straightforward, many problems in diverse areas can be encoded in PL.
  - Problems in graph theory and combinatorics, games such as Sudoku, problems in biotechnology and bioinformatics, etc.
  - There exists a reduction from every NP-Complete problem to SAT.
- As an example, let us try to encode the graph-colouring problem in PL.

GRAPH COLOURING IN PL

- In the graph colouring problem, the goal is to assign colours to vertices such that no two adjacent vertices have the same colour.
- Formally, consider graph $G = \langle V, E \rangle$
  - Vertices, $V = \{v_1, \ldots, v_n\}$
  - Edges, $E = \{e_1, \ldots, e_l\} \subseteq V \times V$
  - Colours, $C = \{c_1, \ldots, c_m\}$
- Assign each vertex $v \in V$ a color $\mathrm{color}(v) \in C$ such that
  - for edge $e = (v, w) \in E$, $\mathrm{color}(v) \neq \mathrm{color}(w)$.

GRAPH COLOURING IN PL

- We use binary variable p_v^c to denote that vertex v has been assigned color c.
- Properties that the colouring should satisfy:
  - Each vertex must be coloured from the set C.
  - Each vertex must be assigned at most one colour.
  - Two adjacent vertices must be assigned different colours.

GRAPH COLOURING IN PL

- Each vertex must be coloured from the set C.

$$(p_{v_1}^{c_1} \vee p_{v_1}^{c_2} \vee \ldots \vee p_{v_1}^{c_m}) \wedge \ldots \wedge (p_{v_n}^{c_1} \vee p_{v_n}^{c_2} \vee \ldots \vee p_{v_n}^{c_m})$$

- Each vertex must be assigned at most one colour.

$$\bigwedge_{i=1}^{n} \bigwedge_{1 \leq j < k \leq m} p_{v_i}^{c_j} \rightarrow \neg p_{v_i}^{c_k}$$

- Two adjacent vertices must be assigned different colours.

$$\bigwedge_{(v,v') \in E} \bigwedge_{k=1}^{m} \neg (p_{v}^{c_k} \wedge p_{v'}^{c_k})$$

GRAPH COLOURING IN PL

- An optimisation: We can omit the at-most one colour constraint.
  - This is because if there is a valid colouring which assigns more than one colour, then there is also a valid colouring assigning exactly one colour.
  - The original formula and the optimised formula are equisatisfiable.

FIRST-ORDER LOGIC

SYNTAX

Term	Constants - a,b,c… Variables - x,y,z…
Function	Arity *n*: Takes *n* terms as input, and forms a term
Predicate	Arity *n*: Takes *n* terms as input, and forms an atom

SYNTAX

Atom	Predicate: p,q,r…
Logical Connectives	$\wedge$: and, $\vee$: or, $\neg$: not, $\rightarrow$: implies, $\leftrightarrow$: if and only if(iff)
Quantifier	$\forall$: Universal $\exists$: Existential
Literal	Atom or its negation
Formula	A literal or the application of logical connectives and quantifiers to formulae

EXAMPLE

$$\forall x\,.\,((\exists y\,.\,p(f(x), y)) \rightarrow q(x))$$

EXAMPLE

$$\forall \boxed{x} \,.\, ((\exists \boxed{y} \,.\, p(f(\boxed{x}), \boxed{y})) \to q(\boxed{x}))$$

Variables

EXAMPLE

$$\forall x \,.\, ((\exists y \,.\, p(\boxed{f(x)}, y)) \rightarrow q(x))$$

Function

EXAMPLE

$$\forall x \,.\, ((\exists y \,.\, \boxed{p(f(x), y)}) \to \boxed{q(x)})$$

Predicate

EXAMPLE

$$\forall x\,.\,((\exists y\,.\,p(f(x), y)) \rightarrow q(x))$$

Quantifier

EXAMPLE

$$\boxed{\forall x} \, . \, ((\exists y \, . \, p(f(x), y)) \rightarrow q(x))$$

Scope of x

EXAMPLE

$$\forall x\,.\,((\boxed{\exists y}\,.\,\underbrace{p(f(x), y))}_{\text{Scope of y}} \to q(x))$$

EXAMPLE

$$\forall x \,.\, ((\boxed{\exists y} \,.\, \underbrace{p(f(x), y))}_{\text{Scope of y}} \to q(x))$$

An occurrence of a variable is bound if it is in the scope of some quantifier

EXAMPLE

$$\forall x \,.\, ((\boxed{\exists y} \,.\, \underbrace{p(f(x), y)}_{\text{Scope of y}}) \rightarrow q(x))$$

An occurrence of a variable is bound if it is in the scope of some quantifier

An occurrence of a variable is free if it is not in the scope of some quantifier

SEMANTICS - EXAMPLES

- All Humans are mortal.
  - Assume unary predicates *human* and *mortal.*

$$\forall x \,.\, human(x) \rightarrow mortal(x)$$

SEMANTICS - EXAMPLES

- In a class, there always exists someone such that if (s)he laughs, then everyone laughs.
    - Assume unary predicate *laughs*.

$$\exists x \,.\, (laughs(x) \rightarrow \forall y \,.\, laughs(y))$$

SEMANTICS - EXAMPLES

- Every dog has its day.
  - $\forall x . dog(x) \rightarrow \exists y . day(y) \wedge itsDay(x, y)$
- Some dogs have more days than others.
  - $\exists x, y . dog(x) \wedge dog(y) \wedge \#days(x) > \#days(y)$
- All cats have more days than dogs.
  - $\forall x, y . (dog(x) \wedge cat(y)) \rightarrow \#days(y) > \#days(x)$

INTERPRETATIONS

- An interpretation I is an assignment from variables (in general, terms) to values in a specified domain.
- Domain, D_I
  - A nonempty set of values or objects. Also called universe of discourse.
  - Numbers, humans, students, courses, animals,…
- Assignment, α_I
  - Maps constants and variables to elements of the domain D_I (i.e. values)
  - Maps functions and predicate symbols to functions and predicates (of the same arity) over D_I

INTERPRETATIONS - EXAMPLE 1

- Suppose $D_I = \{A, B\}$
- Constants a and b are mapped to following elements in D_I
  - $\alpha_I(a) = B \quad \alpha_I(b) = A$
- A binary function symbol f is mapped to the following actual function on D_I:
  - $\alpha_I(f) = \{(A, A) \to B,\ (A, B) \to B,\ (B, A) \to A,\ (B, B) \to B\}$
- A unary predicate symbol p is mapped to the following actual predicate on D_I
  - $\alpha_I(p) = \{A \to True,\ B \to False\}$

INTERPRETATIONS - EXAMPLE 2

- Consider the formula: $x + y > z \rightarrow y > z - x$
  - Here, $+, -$ are functions and $>$ is a predicate.
  - Equivalent to $p(f(x, y), z) \rightarrow p(y, g(z, x))$.
- A standard interpretation for this formula would be:
  - Domain: $\mathbb{Z}$
  - $+, -$ would be mapped to the standard integer addition and subtraction functions.
  - $>$ would be mapped to the standard greater-than relation over integers.
  - x, y, z could be mapped to $5,10,9$ resp.

SEMANTICS: INDUCTIVE DEFINITION

Base Case:

$I \models \top$	
$I \not\models \bot$	
$I \models p$	iff I[p]=⊤
$I \not\models p$	iff I[p]=⊥

Inductive Case:

$I \models \neg F$	iff $I \not\models F$
$I \models F_1 \wedge F_2$	iff $I \models F_1$ and $I \models F_2$
$I \models F_1 \vee F_2$	iff $I \models F_1$ or $I \models F_2$
$I \models F_1 \rightarrow F_2$	iff $I \not\models F_1$ or $I \models F_2$
$I \models F_1 \leftrightarrow F_2$	iff $I \models F_1$ and $I \models F_2$, or $I \not\models F_1$ and $I \not\models F_2$

SEMANTICS: INDUCTIVE DEFINITION

Base Case:

What does this mean?

$I \vDash \top$	
$I \nvDash \bot$	
$I \vDash p$	iff I[p]=$\top$
$I \nvDash p$	iff I[p]=$\bot$

Inductive Case:

$I \vDash \neg F$	iff $I \nvDash F$
$I \vDash F_1 \wedge F_2$	iff $I \vDash F_1$ and $I \vDash F_2$
$I \vDash F_1 \vee F_2$	iff $I \vDash F_1$ or $I \vDash F_2$
$I \vDash F_1 \rightarrow F_2$	iff $I \nvDash F_1$ or $I \vDash F_2$
$I \vDash F_1 \leftrightarrow F_2$	iff $I \vDash F_1$ and $I \vDash F_2$, or $I \nvDash F_1$ and $I \nvDash F_2$

SEMANTICS - CONTINUED...

$$I \models p(t_1, \ldots, t_n) \text{ iff } \alpha_I[p](\alpha_I[t_1], \ldots, \alpha_I[t_n]) = \top$$

$$\alpha_I[f(t_1, \ldots, t_n)] = \alpha_I[f](\alpha_I[t_1], \ldots, \alpha_I[t_n])$$

SEMANTICS - EXAMPLE

INTERPRETATION I

$D_I = \{A, B\}$

$\alpha_I(a) = B \quad \alpha_I(b) = A$

$\alpha_I(f) = \{(A, A) \to B,\ (A, B) \to B,\ (B, A) \to A,\ (B, B) \to B\}$

$\alpha_I(p) = \{A \to True,\ B \to False\}$

$$I \vDash p(b)$$

$$I \vDash p(f(a, b))$$

$$I \nvDash p(f(b, a))$$

SEMANTICS - QUANTIFIERS

- An $x-$variant of interpretation $I = (D_I, \alpha_I)$ is an interpretation $J = (D_J, \alpha_J)$ such that
  - $D_I = D_J$;
  - and $\alpha_I[y] = \alpha_J[y]$ for all constant, free variable, function, and predicate symbols y, except possibly x.
- An $x-$variant of I, where x is mapped to some $v \in D_I$ is denoted by $I[x \mapsto v]$.

$$I \vDash \forall x \,.\, F \text{ iff for all } v \in D_I, I[x \mapsto v] \vDash F$$

$$I \vDash \exists x \,.\, F \text{ iff there exists } v \in D_I, I[x \mapsto v] \vDash F$$

SEMANTICS - QUANTIFIERS - EXAMPLE

INTERPRETATION I

$D_I = \{A, B\}$

$\alpha_I(a) = B \quad \alpha_I(b) = A$

$\alpha_I(f) = \{(A, A) \rightarrow B,\ (A, B) \rightarrow B,\ (B, A) \rightarrow A,\ (B, B) \rightarrow B\}$

$\alpha_I(p) = \{A \rightarrow True,\ B \rightarrow False\}$

$$I \vDash \exists x\,.\,p(x)$$

$$I \vDash \forall x\,.\,\neg p(f(b, x))$$

SATISFIABILITY AND VALIDITY

- A FOL formula F is satisfiable if there exists an interpretation I such that $I \vDash F$.
  - If no such interpretation exists, then it is unsatisfiable
- A FOL formula F is valid if for all interpretations I, $I \vDash F$
  - Technically, only for interpretations which assign to all the constants, variables, predicates, functions used in F.
- F is valid iff $\neg F$ is unsatisfiable.

FREE VARIABLES

- Given a FOL formula F, a variable x is free in F if there is a use of x in F which is not bound to any quantifier.
  - $free(F)$ denotes all variables free in F.
- A FOL formula F is closed if it does not contain any free variables.
- Technically, satisfiability and validity are only applicable for closed FOL formulae.
- However, we can extend these concepts to formulae with free variables by following the below convention:
  - For satisfiability, all free variables are implicitly existentially quantified.
  - For validity, all free variables are implicitly universally quantified.

SATISFIABILITY AND VALIDITY

EXAMPLES

- Is the formula $\forall x \,.\, \exists y \,.\, p(x, y)$ satisfiable?
    - Yes. A satisfying interpretation:
      $I = (\{A\}, < p \mapsto \{(A, A) \mapsto \top \} >)$
- Is the formula $\forall x \,.\, \exists y \,.\, p(x, y)$ valid?
    - No. A falsifying interpretation:
      $I = (\{A\}, < p \mapsto \{(A, A) \mapsto \bot \} >)$
- Is the formula $(\forall x \,.\, p(x)) \to (\exists y \,.\, p(y))$ valid?
- Is the formula $\forall x \,.\, (p(x) \to (\exists y \,.\, p(y)))$ valid?
    - What about $\forall x \,.\, (p(x) \to (\forall y \,.\, p(y)))$?

DECISION PROCEDURE FOR VALIDITY

- Semantic Argument Method
  - Deductive Approach
  - Proof by Contradiction
  - Assume that a falsifying interpretation exists.
  - Use proof rules to deduce more facts.
  - The goal is to find contradictory facts in each branch (also called closing the branch).
- Proof rules for negation, conjunction, disjunction, implication, iff carry over from Propositional logic

PROOF RULES

UNIVERSAL QUANTIFICATION

$$\frac{I \models \forall x \,.\, F}{I[x \mapsto v] \models F} \quad \text{For any } v \in D_I$$

PROOF RULES

UNIVERSAL QUANTIFICATION

$$\frac{I \models \forall x \,.\, F}{I[x \mapsto v] \models F}$$ For any $v \in D_I$

$$\frac{I \not\models \forall x \,.\, F}{I[x \mapsto v] \not\models F}$$ For a fresh $v \in D_I$

PROOF RULES

EXISTENTIAL QUANTIFICATION

$$\frac{I \models \exists x \,.\, F}{I[x \mapsto v] \models F} \quad \text{For a fresh } v \in D_I$$

PROOF RULES

EXISTENTIAL QUANTIFICATION

$$\frac{I \models \exists x \,.\, F}{I[x \mapsto v] \models F} \quad \text{For a fresh } v \in D_I$$

$$\frac{I \not\models \exists x \,.\, F}{I[x \mapsto v] \not\models F} \quad \text{For any } v \in D_I$$

PROOF RULES

CONTRADICTION

$$\frac{\begin{array}{c} J \models p(s_1, \ldots, s_n) \quad K \not\models p(t_1, \ldots, t_n) \\ J = I[\ldots] \quad K = I[\ldots] \\ \alpha_J[s_i] = \alpha_K[t_i] \text{ for all } i = 1,\ldots,n \end{array}}{I \models \bot}$$

EXAMPLE

Prove that $(\forall x \,.\, p(x)) \rightarrow (\forall y \,.\, p(y))$ is valid

$$\frac{I \not\models (\forall x \,.\, p(x)) \rightarrow (\forall y \,.\, p(y))}{I \models (\forall x \,.\, p(x)) \qquad I \not\models (\forall y \,.\, p(y))}$$

EXAMPLE

Prove that $(\forall x \,.\, p(x)) \rightarrow (\forall y \,.\, p(y))$ is valid

$$\dfrac{I \not\models (\forall x \,.\, p(x)) \rightarrow (\forall y \,.\, p(y))}{I \models (\forall x \,.\, p(x)) \qquad \dfrac{I \not\models (\forall y \,.\, p(y))}{I[y \mapsto v] \not\models p(y)}\ \text{[for a fresh } v\text{]}}$$

EXAMPLE

Prove that $(\forall x \,.\, p(x)) \rightarrow (\forall y \,.\, p(y))$ is valid

$$\dfrac{I \not\models (\forall x \,.\, p(x)) \rightarrow (\forall y \,.\, p(y))}{\dfrac{I \models (\forall x \,.\, p(x))}{I[x \mapsto v] \models p(x)} \qquad \dfrac{I \not\models (\forall y \,.\, p(y))}{I[y \mapsto v] \not\models p(y)}}$$

[for a fresh v]

EXAMPLE

Prove that $(\forall x\,.\,p(x)) \to (\forall y\,.\,p(y))$ is valid

$$\dfrac{I \not\models (\forall x\,.\,p(x)) \to (\forall y\,.\,p(y))}{\dfrac{I \models (\forall x\,.\,p(x))}{\boxed{I[x \mapsto v] \models p(x)}} \qquad \dfrac{I \not\models (\forall y\,.\,p(y))}{\boxed{I[y \mapsto v] \not\models p(y)}}}$$

[for a fresh v]

CONTRADICTION

EXAMPLE - 2

Prove that $\exists x . F \rightarrow G \leftrightarrow (\forall x . F) \rightarrow (\exists x . G)$ is valid

EXAMPLE - 2

Prove that $\exists x \, . \, F \rightarrow G \leftrightarrow (\forall x \, . \, F) \rightarrow (\exists x \, . \, G)$ is valid

$$I \not\models \exists x \, . \, F \rightarrow G \leftrightarrow (\forall x \, . \, F) \rightarrow (\exists x \, . \, G)$$

EXAMPLE - 2

Prove that $\exists x . F \rightarrow G \leftrightarrow (\forall x . F) \rightarrow (\exists x . G)$ is valid

$$\dfrac{I \not\models \exists x . F \rightarrow G \leftrightarrow (\forall x . F) \rightarrow (\exists x . G)}{I \models \exists x . F \rightarrow G \quad I \not\models (\forall x . F) \rightarrow (\exists x . G) \;\Big|\; I \not\models \exists x . F \rightarrow G \quad I \models (\forall x . F) \rightarrow (\exists x . G)}$$

EXAMPLE - 2

Prove that $\exists x \,.\, F \to G \leftrightarrow (\forall x \,.\, F) \to (\exists x \,.\, G)$ is valid

$$\frac{I \not\models \exists x \,.\, F \to G \leftrightarrow (\forall x \,.\, F) \to (\exists x \,.\, G)}{I \models \exists x \,.\, F \to G \qquad\qquad I \not\models (\forall x \,.\, F) \to (\exists x \,.\, G)}$$

EXAMPLE - 2

Prove that $\exists x . F \to G \leftrightarrow (\forall x . F) \to (\exists x . G)$ is valid

$$\frac{I \not\models \exists x . F \to G \leftrightarrow (\forall x . F) \to (\exists x . G)}{\dfrac{I \models \exists x . F \to G}{I[x \mapsto v] \models F \to G} \qquad I \not\models (\forall x . F) \to (\exists x . G)}$$

EXAMPLE - 2

Prove that $\exists x\,.\,F \rightarrow G \leftrightarrow (\forall x\,.\,F) \rightarrow (\exists x\,.\,G)$ is valid

$$\dfrac{I \not\models \exists x\,.\,F \rightarrow G \leftrightarrow (\forall x\,.\,F) \rightarrow (\exists x\,.\,G)}{\dfrac{I \models \exists x\,.\,F \rightarrow G}{\dfrac{I[x \mapsto v] \models F \rightarrow G}{I[x \mapsto v] \not\models F \quad | \quad I[x \mapsto v] \models G}} \qquad I \not\models (\forall x\,.\,F) \rightarrow (\exists x\,.\,G) \quad |}$$

EXAMPLE - 2

Prove that $\exists x \,.\, F \rightarrow G \leftrightarrow (\forall x \,.\, F) \rightarrow (\exists x \,.\, G)$ is valid

$$I \not\models \exists x \,.\, F \rightarrow G \leftrightarrow (\forall x \,.\, F) \rightarrow (\exists x \,.\, G)$$

$$I \models \exists x \,.\, F \rightarrow G \qquad I \not\models (\forall x \,.\, F) \rightarrow (\exists x \,.\, G)$$

$$I[x \mapsto v] \models F \rightarrow G \qquad I \models (\forall x \,.\, F) \quad I \not\models (\exists x \,.\, G)$$

$$I[x \mapsto v] \not\models F \quad | \quad I[x \mapsto v] \models G$$

EXAMPLE - 2

Prove that $\exists x . F \rightarrow G \leftrightarrow (\forall x . F) \rightarrow (\exists x . G)$ is valid

$$I \not\models \exists x . F \rightarrow G \leftrightarrow (\forall x . F) \rightarrow (\exists x . G)$$

$$I \models \exists x . F \rightarrow G \qquad I \not\models (\forall x . F) \rightarrow (\exists x . G)$$

$$I[x \mapsto v] \models F \rightarrow G \qquad I \models (\forall x . F) \quad I \not\models (\exists x . G)$$

$$I[x \mapsto v] \not\models F \quad | \quad I[x \mapsto v] \models G \qquad I[x \mapsto v] \models F$$

EXAMPLE - 2

Prove that $\exists x . F \rightarrow G \leftrightarrow (\forall x . F) \rightarrow (\exists x . G)$ is valid

$$
\dfrac{I \not\models \exists x . F \rightarrow G \leftrightarrow (\forall x . F) \rightarrow (\exists x . G)}{\dfrac{\dfrac{I \models \exists x . F \rightarrow G}{\dfrac{I[x \mapsto v] \models F \rightarrow G}{\boxed{I[x \mapsto v] \not\models F} \quad \Big| \quad I[x \mapsto v] \models G}} \qquad \dfrac{I \not\models (\forall x . F) \rightarrow (\exists x . G)}{\dfrac{I \models (\forall x . F)}{\boxed{I[x \mapsto v] \models F}} \quad \dfrac{I \not\models (\exists x . G)}{}} \quad \Big|}{}}
$$

CONTRADICTION

EXAMPLE - 2

Prove that $\exists x . F \to G \leftrightarrow (\forall x . F) \to (\exists x . G)$ is valid

$$I \not\models \exists x . F \to G \leftrightarrow (\forall x . F) \to (\exists x . G)$$

$$I \models \exists x . F \to G \qquad I \not\models (\forall x . F) \to (\exists x . G)$$

$$I[x \mapsto v] \models F \to G \qquad I \models (\forall x . F) \quad I \not\models (\exists x . G)$$

$$I[x \mapsto v] \not\models F \quad | \quad I[x \mapsto v] \models G \qquad I[x \mapsto v] \models F \quad I[x \mapsto v] \not\models G$$

EXAMPLE - 2

Prove that $\exists x \,.\, F \to G \leftrightarrow (\forall x \,.\, F) \to (\exists x \,.\, G)$ is valid

$$I \not\models \exists x \,.\, F \to G \leftrightarrow (\forall x \,.\, F) \to (\exists x \,.\, G)$$

$$I \models \exists x \,.\, F \to G \qquad\qquad I \not\models (\forall x \,.\, F) \to (\exists x \,.\, G)$$

$$I[x \mapsto v] \models F \to G \qquad\qquad I \models (\forall x \,.\, F) \quad I \not\models (\exists x \,.\, G)$$

$$I[x \mapsto v] \not\models F \quad \Big| \quad \boxed{I[x \mapsto v] \models G} \qquad I[x \mapsto v] \models F \quad \boxed{I[x \mapsto v] \not\models G}$$

CONTRADICTION

Homework: Complete the proof in the other branch

MORE EXAMPLES

- Prove or disprove validity of following FOL formulae
  - $\forall x \,.\, F \rightarrow G \leftrightarrow (\exists x \,.\, F) \rightarrow (\forall x \,.\, G)$
  - $(\forall x \,.\, p(x)) \leftrightarrow \neg(\exists x \,.\, \neg p(x))$
  - $(\exists x \,.\, p(x)) \rightarrow (\forall y \,.\, p(y))$
  - $\exists x \,.\, (p(x) \rightarrow \forall y \,.\, p(y))$

DECIDABILITY OF VALIDITY OF FOL

- Church and Turing showed that it is undecidable to find whether a first-order formula is valid or not.
- But we have just seen the Semantic Argument-based decision procedure!
  - How to instantiate domain values in Proof rules for quantifiers?
  - What order should proof rules be applied in?
- The semantic argument-based method can be augmented to make the validity of FOL problem semi-decidable.
  - If the input formula is valid, then the method will halt and answer positive.
  - If the input formula is not valid, then the method may never halt.
  - More details in the BM book [Chapter 2, Section 2.7].

NORMAL FORMS OF FOL

- Negation Normal Form (NNF)
  - Should use only $\neg$, $\wedge$, $\vee$ as the logical connectives, and $\neg$ should only be applied to literals
  - $\neg(\forall x\,.\,F) \Leftrightarrow \exists x\,.\,\neg F$ and $\neg(\exists x\,.\,F) \Leftrightarrow \forall x\,.\,\neg F$

PRENEX NORMAL FORM

- A formula is in Prenex Normal Form (PNF) if all of its quantifiers appear at the beginning of the formula:
  - $Q_1 x_1 \ldots Q_n x_n \,.\, F[x_1, \ldots, x_n]$, where F is quantifier-free and may have $x_1, \ldots, x_n$ as free variables.
- How to convert an arbitrary formula F to PNF?
  1. First, convert F to NNF (call it F_1).
  2. If two quantified variables in F_1 have the same name, then rename them to fresh variables (obtaining the formula F_2).
  3. Remove all quantifiers in F_2 to obtain F_3.
  4. Add all the removed quantifiers at the beginning of F_3, ensuring that if Q_j was in the scope of Q_i in F_2, then Q_i occurs before Q_j

PRENEX NORMAL FORM

EXAMPLE

$$F : \ \forall x.\ \neg(\exists y.\ p(x,y) \ \wedge \ p(x,z)) \ \vee \ \exists y.\ p(x,y)$$

STEP-1

$$F_1 : \ \forall x.\ (\forall y.\ \neg p(x,y) \ \vee \ \neg p(x,z)) \ \vee \ \exists y.\ p(x,y)$$

STEP-2

$$F_2 : \ \forall x.\ (\forall y.\ \neg p(x,y) \ \vee \ \neg p(x,z)) \ \vee \ \exists w.\ p(x,w)$$

STEP-3

$$F_3 : \ \neg p(x,y) \ \vee \ \neg p(x,z) \ \vee \ p(x,w)$$

STEP-4

$$\forall x.\ \forall y.\ \exists w.\ \neg p(x,y) \ \vee \ \neg p(x,z) \ \vee \ p(x,w)$$

QUESTIONS

- Semantic Argument-based method for Validity of FOL Formula
  - Can it be for checking satisfiability of FOL Formula?
  - Yes, but we may not get a satisfying interpretation. Consider the formula $\forall x \,.\, \exists y \,.\, p(x, y)$.
- Why do we insist on fresh values in some proof rules?
  - Example: $\forall x \,.\, \forall y \,.\, \forall z \,.\, p(x, y) \vee \neg p(x, z)$.
- Prenex Normal Form
  - Is $\forall x \,.\, \exists y \,.\, p(x, y) \Leftrightarrow \exists y \,.\, \forall x \,.\, p(x, y)$?
  - What about $\forall x \,.\, \exists y \,.\, p(x) \wedge q(y)$ and $\exists y \,.\, \forall x \,.\, p(x) \wedge q(y)$?

HOMEWORK

- Please try Exercises 2.1-2.4 in the BM Book, Chapter 2.