

# 7. Logic and reasoning

# 7.1 What is logic?

- Study of criteria for sound reasoning:
  - through formal systems of inference
  - through arguments in natural language
- Interested in *validity*, not *truth*:
  - ✓ All lemons are blue
  - ✓ Mary is a lemon
  - ✓ Therefore, Mary is blue.
- Consists of
  - Alphabet – symbols from which formulas are built.
  - Syntax — defines the form of legal sentences.
  - *Semantics* — define the meaning of sentences with respect to some state of affairs.
  - *Proof theory* — contains set of rules for deducing the implications of set of sentences.

# 7.2 Logic in AI

- Predominant theory 1960 – 1980, recently losing to connectionism, GA and information theoretic machine learning.
- Knowledge representation formalism
- Means for reasoning about knowledge
- Computational tool (e.g., Prolog)
- Scope of logic is large:
  - Fallacies and paradoxes
  - Analysis of reasoning (e.g., probability and causality)

# 7.3 Logical alphabet

- Truth values: *true* and *false*
- Variables:  $x, y, w, \dots$
- Connectives:  $\wedge, \vee, \rightarrow, \leftrightarrow, \neg$
- Quantifiers:  $\forall, \exists$
- Parenthesis:  $(, )$
- Propositional symbols:  $P, Q, R, \dots$  (or  $p_1, p_2, \dots$ )
- Predicates:  $L(x, y, \dots)$
- Formulas (sentences):  $\varphi, \phi, \dots \in \Phi$

# 7.4 Logical inference

- Derivation of new sentences from old ones.
- A sentence *logically follows* from set of sentences, if it is true whenever the sentences are true.
- *Soundness*: false sentences cannot be proved.
- *Completeness*: any true derivation can be proved.
- *Monotonicity*: no valid proof in the system can be made invalid by adding new premises or assumptions.
- *Tautology* is a sentence that is always true, e.g., “it is raining or it is not raining.”
- Contradictory sentences are never true, e.g.,  $A \wedge \neg A$ .
- Sentence is satisfiable, if it is true under some interpretation, but not all, e.g.,  $A \vee B$ .

# 7.5 Propositional logic

- Propositions:  $P = \text{‘I like cheese.’}$
- Syntax



- *Well-formed formulas (wff)* are the following:  $P, Q, R, \dots, \text{true}, \text{false}, (P), P \wedge Q, P \vee Q, P \rightarrow Q, P \leftrightarrow Q, \neg P$

e.g.:

$P$	$Q$	$P \wedge Q$
$T$	$T$	$T$
$T$	$f$	$F$
$F$	$T$	$F$
$F$	$F$	$F$

$P$	$Q$	$P \rightarrow Q$
$T$	$T$	$T$
$T$	$F$	$F$
$F$	$T$	$T$
$F$	$F$	$T$

- Semantics

- Defined by truth tables.
- Independent of what propositions  $P$  and  $Q$  mean.

# 7.6 Inference rules

	Introduction	Elimination
$\wedge$	$\frac{P, Q}{P \wedge Q}$	$\frac{P \wedge Q}{P} \quad \frac{P \wedge Q}{Q}$
$\vee$	$\frac{P}{P \vee Q} \quad \frac{Q}{P \vee Q}$	$\frac{P \rightarrow R, Q \rightarrow R, P \vee Q}{R}$
$\rightarrow$	$\frac{\neg P}{P \rightarrow Q} \quad \frac{Q}{P \rightarrow Q}$	$\frac{P, P \rightarrow Q}{Q}$ <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 100px;">Modus ponens</div> $\frac{\neg \neg P}{P}$ <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 100px;">Reductio ad absurdum</div> $\frac{\neg A}{\dots} \perp$ $A$

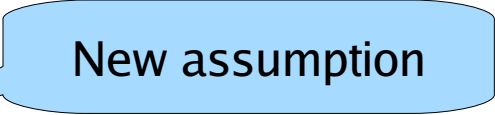
# 7.7 Example proof

Prove:  $(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow ((C \rightarrow D) \rightarrow (A \rightarrow D)))$

<u>A</u>	<u>A → B</u>	(assumptions)
<u>B</u>	<u>B → C</u>	(modus ponens)
<u>C</u>	<u>C → D</u>	(modus ponens)
<u>D</u>		(modus ponens)
<u>A → D</u>		(→ introduction)
<u>(C → D) → (A → D)</u>		(→ introduction)
<u>(B → C) → ((C → D) → (A → D))</u>		(→ introduction)
<u>(A → B) → ((B → C) → ((C → D) → (A → D)))</u>		(→ introduction)

# 7.8 Another example

Prove:  $(\neg A \rightarrow B) \rightarrow (\neg B \rightarrow A)$

$\frac{\neg A \quad \neg A \rightarrow B}{\neg A}$		(assumptions)
$\frac{B \quad \neg B}{\perp}$		(modus ponens)
$\frac{B \quad B \rightarrow \perp}{\perp}$	(rewriting $\neg B$ )	
$\perp$	(modus ponens)	
$A$	(reductio ad absurdum)	
$\frac{\neg B \rightarrow A}{\neg B \rightarrow A}$	( $\rightarrow$ introduction)	
$(\neg A \rightarrow B) \rightarrow (\neg B \rightarrow A)$	( $\rightarrow$ introduction)	

# 7.9 First-order predicate logic

- Predicates

- Express properties of and relationships between objects:  
 $L(me, cheese) = \text{“I like cheese”}$ .
- Take one or more arguments.

- Variables and constants

- $L(x, cheese) = \text{“x likes cheese.”}$

- *Universal and existential* quantifiers:

$$\neg(\forall x) (P(x) \rightarrow L(x, cheese)) \leftrightarrow (\exists x) (P(x) \wedge \neg L(x, cheese))$$

- Syntax

- $P(x_1, x_2, x_3, \dots, x_n)$  is wff.

Atomic formula

- If  $P$  is wff, and  $x$  is a variable, then  $(\exists x)P$  and  $(\forall x)P$  are wff (in addition to those in propositional logic).

# 7.10 Semantics of predicate logic

- Semantics is defined in terms of truth values of sentences relative to an *interpretation* (model).
- Interpretation is a specific choice of assignment of entities of domain  $D$  to a set of constants, variables and predicates of a FOPL expression.
  - For instance, in order to find the truth value of a sentence  $(\forall x)(P(x) \rightarrow Q(x))$ , we need to know both the meanings of  $P$  and  $Q$ , and which values  $x$  can take in domain  $D$ .
- Predicate calculus sentence is *satisfiable* iff there is an interpretation and variable assignment that satisfies it.
- Sentence is *valid* if it is true for all possible interpretations.

# 7.11 Modal logic

- Classical method of reasoning about knowledge in philosophy. Modal logic of knowledge created by Jaakko Hintikka 1962.
- Extends first-order logic with operators that constrain the extent to which a sentence is true, and indicate that a sentence is consistent with a reasoner's beliefs.
- Allows us reason about universes or scenarios that could logically happen = “all possible worlds”
  - Possibility, impossibility and necessity
  - Obligation and norms
  - Time
- For instance, the following sentences can be true in some possible world:
  - ✓ Trees are all blue.
  - ✓ Dogs can fly.
  - ✓ People have no legs.
- Law of the excluded middle  $p \vee \neg p$  cannot be true in any world.

# 7.12 Necessity vs. possibility

- Something is
  - *Possible*, if it might be true.
  - *Necessary*, if it possibly cannot be false.
  - Is something is necessarily true, it is true. If it is true, it is possible.
- Modal operators:
  - for *necessarily*
  - $\diamond$  for *possibly*
  - Can be defined with each other and negation:  
$$\diamond p \leftrightarrow \neg \neg p.$$
- If  $\varphi$  is wff, then are  $\Box \varphi$  and  $\diamond \varphi$ .

# 7.13 Temporal logic

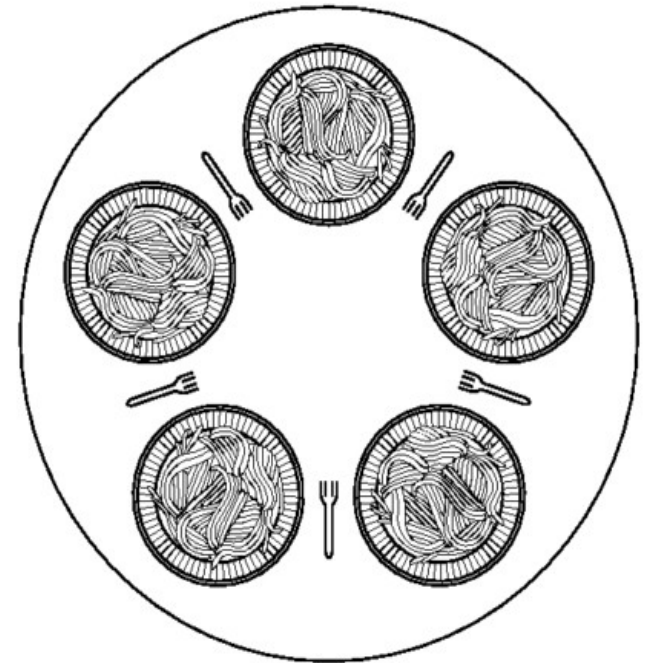
- Assume a concept of time that can be broken down into intervals.
- Modal operators:
  - $p = \text{“al ways”}$ :  $p$  will be true in all times in the future (from now on).
  - $\diamond p = \text{“ev entually”}$   $p$  will be true some times in the future.
  - $\circ p = \text{“at the nex t time”}$   $p$  will be true in the next time interval.
  - $p \mu q = \text{“until”}$ :  $p$  is true until  $q$  is true.

# 7.14 Use of temporal logic

- Specification and verification of software
- Conditions for a temporal logic system
  - *Safety conditions* define behaviors that should never occur.
  - *Liveness conditions* define the behaviors that the system should do.
  - *Fairness conditions* determine the system's behavior in non-deterministic situations.

# 7.15 Example: Dining philosophers

- Safety conditions:
  - $\neg(\text{eating}(i) \wedge \text{thinking}(i))$
  - $(\text{eating}(i) \vee \text{thinking}(i))$
  - $\neg(\text{eating}(i) \wedge \text{eating}(i+1))$
- Liveness conditions
  - $(\text{eating}(i) \rightarrow \diamond \text{thinking}(i))$
  - $(\text{thinking}(i) \rightarrow \diamond \text{eating}(i))$



# 7.16 Nonmonotonic logic

- Traditional logics are *monotonic*:
  - The truth of a proposition does not change if new information (axioms) is added to the system.
  - If sentence  $\varphi$  can be inferred from set of premises  $\Phi$ , it can be inferred also after a sentence  $\phi$  is added to  $\Phi$ , i.e., from any set that contains  $\Phi$  as its subset.
- Everyday thinking follows *defeasible inference*, i.e., the right to reverse or retract one's conclusions in light of new information.
- *Non-monotonic inference* = the set of conclusions warranted by knowledge does not increase with the knowledge (it may shrink).

# 7.17 Related issues

- *Closed-world assumption*; any fact not explicitly stated in the knowledge based is assumed false, i.e., all true facts need to be represented to the system.
- *Open-world assumption*; any fact not explicitly included in the data base is assumed true.
- *Ramification problem*
  - Has to do with additional consequences of an action that may not be immediately obvious.
  - Related to frame problem.

# 7.18 Nonmonotonic inference I

- Abduction

- Not logically sound
- $A = \text{“Fred is sick.”}$
- $B = \text{“Fred is not at work”}$

$B$	$A \rightarrow B$
$A$	

- Modal operator  $\mathcal{M}$

- Extend first-order logic with an operator that indicates if statement is consistent with our beliefs.
- Example:  $Exist(food) \wedge \mathcal{M}(Working(oven)) \rightarrow meal(Y)$
- Checking the consistency may be tricky.

# 7.19 Nonmonotonic inference II

- Default logic
  - Presumes certain facts by default, unless disapproved by some other facts.
  - Default rules of the form:

Prerequisite      Justification      Conclusion  
 $\phi(x) : \beta_1, \dots, \beta_m(x) \leftarrow \varphi(x)$

- Example:  $\text{Car}(x) \wedge \text{has\_driver}(x) \rightarrow \text{has\_driver}(x)$

- Truth maintenance systems

- Systems in which *belief revision* is important.
- Stores how each belief is derived, i.e., reasons or justifications for beliefs:

Belief  $Q$  has the reason  $(\{P, R\}, \{\neg S\})$

IN set: all true

OUT set: all false

# 7.20 Frame problem revisited

- Introduced by McCarthy and Hayes in 1969 in the context of the situation calculus:
  - Problem of expressing a dynamical domain in logic without explicitly specifying which conditions are not affected by an action.
- Later broader meaning in Philosophy: how to limit the beliefs that need to be updated in response to an action.
- After action is executed, the state of the world changes according to the action's effect.
- However, most actions do not change most of the world.

- Example:

$\neg \text{open}(t=0)$	(door is closed at time $t=0$ )
$\neg \text{on}(t=0)$	(lights are off at time $t=0$ )
$\text{true} \rightarrow \text{open}(t=1)$	(door is open at time $t=1$ )

- These formulae do not suffice to draw correct conclusions, i.e., specifying the changes caused by the action does not allow to conclude that all other conditions are not changed.

$\neg \text{Open}(0)$	$\text{Open}(1)$
$\neg \text{On}(0)$	$\neg \text{On}(1)$

$\neg \text{Open}(0)$	$\text{Open}(1)$
$\neg \text{On}(0)$	$\text{On}(1)$

- Non-effects of an action need to be specified for an artificial system by *frame axioms*.
- One such axiom needed for each action-condition pair.
- In the above example we need the axiom:  $\text{on}(0) \equiv \text{on}(1)$
- Frame problem: how to formalize a dynamical domain without explicitly specifying the frame actions.

# 7.21 Frame problem is

- Problem in mathematical logic.
- Representational problem, not inductive problem.
- Not about predicting changes, but how to deal with non-changes.
- Not problem of relevance.
- Not problem of qualification: impossibility of stating all preconditions of an action.

## 7.22 Initial solution

- *Circumscription* formalized by John McCarthy: things are as expected unless otherwise specified.
  - Those things that can be proved to be true, are assumed to be the only ones.
  - Related to closed world assumption.
- Circumscription failed; Yale shooting problem (Steve Hanks & Drew McDermott, 1987)
  - $\text{alive}(0)$
  - $\neg\text{loaded}(0)$
  - $\text{true} \rightarrow \text{loaded}(1)$
  - $\text{loaded}(2) \rightarrow \neg\text{alive}(3)$

## 7.23 Successor state axiom solution

- The value of a condition after an action is determined by the fact that the condition is true if and only if:
  - The action made the condition true.
  - The condition is true before the action, and the action does not make it false.
- Example:
  - $\neg\text{open}(t=0)$
  - $\neg\text{on}(t=0)$
  - $\text{open\_door}(t=0)$
  - $\forall t \text{ open}(t+1) \equiv \text{open\_door}(t) \vee (\text{open}(t) \wedge \neg\text{close\_door}(t))$

# 7.24 Fluent occlusion solution

- Not only the value of a condition is represented, but also if it can be affected by the last action ← *occlusion*
- Permission to change; the condition can change value only if the corresponding occlusion predicate is true at the next time point.

- Example:

$\neg \text{open}(t=0)$

$\neg \text{on}(t=0)$

$\text{true} \rightarrow \text{open}(t=1) \wedge \text{occlude\_open}(t=1)$

$\forall t (\neg \text{occlude\_open}(t)) \rightarrow (\text{open}(t-1) \equiv (\text{open}(t)))$

$\forall t (\neg \text{occlude\_on}(t)) \rightarrow (\text{on}(t-1) \equiv (\text{on}(t)))$

# 7.25 Predicate completion solution

- Additional predicates denote change, not permission to change.
- Predicate change only if the corresponding change predicate is true.
- Example:

$\neg \text{open}(t=0)$

$\neg \text{on}(t=0)$

$\neg \text{open}(t=0) \wedge \text{true} \rightarrow \text{change\_open}(t=0)$

$\forall t (\text{change\_open}(t)) \equiv (\text{open}(t) \neq (\text{open}(t+1)))$

$\forall t (\text{change\_on}(t)) \equiv (\text{on}(t) \neq (\text{on}(t+1)))$

# 7.26 What is reasoning?

- Process by which a conclusion is reached.
- Act of using reason to derive conclusion from certain premises using a given methodology:

– Deductive reasoning

- Does not increase knowledge
- Exact inference

$$\frac{P, P \rightarrow Q}{Q} \quad \frac{\forall x P(x)}{P(c)}$$

– Inductive reasoning

- Infer generalization from instances

$$\frac{P(c)}{\forall x P(x)}$$

– Abductive reasoning:

- Inference to the best explanation

$$\frac{Q(c), P(a) \rightarrow Q(a)}{P(c)}$$

– Reasoning by analogy

# 7.27 Reasoning in propositional logic

- Truth tables can be used to mechanically prove that all conclusions logically follow from premises → manual use of huge tables can be cumbersome.
- The goal is to automate reasoning.
  - Rules of deduction (introduction and elimination)
  - Refutation
    - Show that the negated conclusion is inconsistent with its premises.
    - Example: When trying to prove  $\{(P \vee Q), \neg P\} \Rightarrow Q$ , show the inconsistency by  $\{(P \vee Q) \wedge \neg P \wedge \neg Q\} \Rightarrow \perp$
  - Resolution:

$$\frac{P \vee Q \quad \neg Q}{P}$$

$$\frac{P \vee Q \quad \neg Q \vee R}{P \vee R}$$

Proposition symbols are called atoms.

- $P \vee Q$

- $\neg P$  ← Negative

- $P$  ← Positive

An atom, or its negation is called literal.

- Conjunctive normal form (CNF): conjunction of disjunctions of literals:  $(P \vee Q \vee R) \wedge (S \vee \neg P) \wedge (P \vee T)$ .

- CNF is of the form  $C_1 \wedge C_2 \wedge C_3 \wedge \dots \wedge C_n$

Clauses

- Disjunctive normal form (DNF): disjunction of conjunctions of literals:  $(P \wedge Q \wedge R) \vee (S \wedge \neg P) \vee (P \wedge T)$ .

- DNF is of the form  $C_1 \vee C_2 \vee C_3 \vee \dots \vee C_n$

# 7.28 Transformation of wff to CNF

1. Eliminate equivalence:  $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$

2. Eliminate implication:  $P \rightarrow Q \equiv \neg P \vee Q$

3. Move negation in (De Morgan's law):

- $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$

- $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$

4. Eliminate double negation:  $\neg\neg P \equiv P$

5. Distribute  $\wedge$  over  $\vee$ :  $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$

## 7.29 Resolution refutation

1. Each clause is represented as a set, and the whole expression as a set of sets:

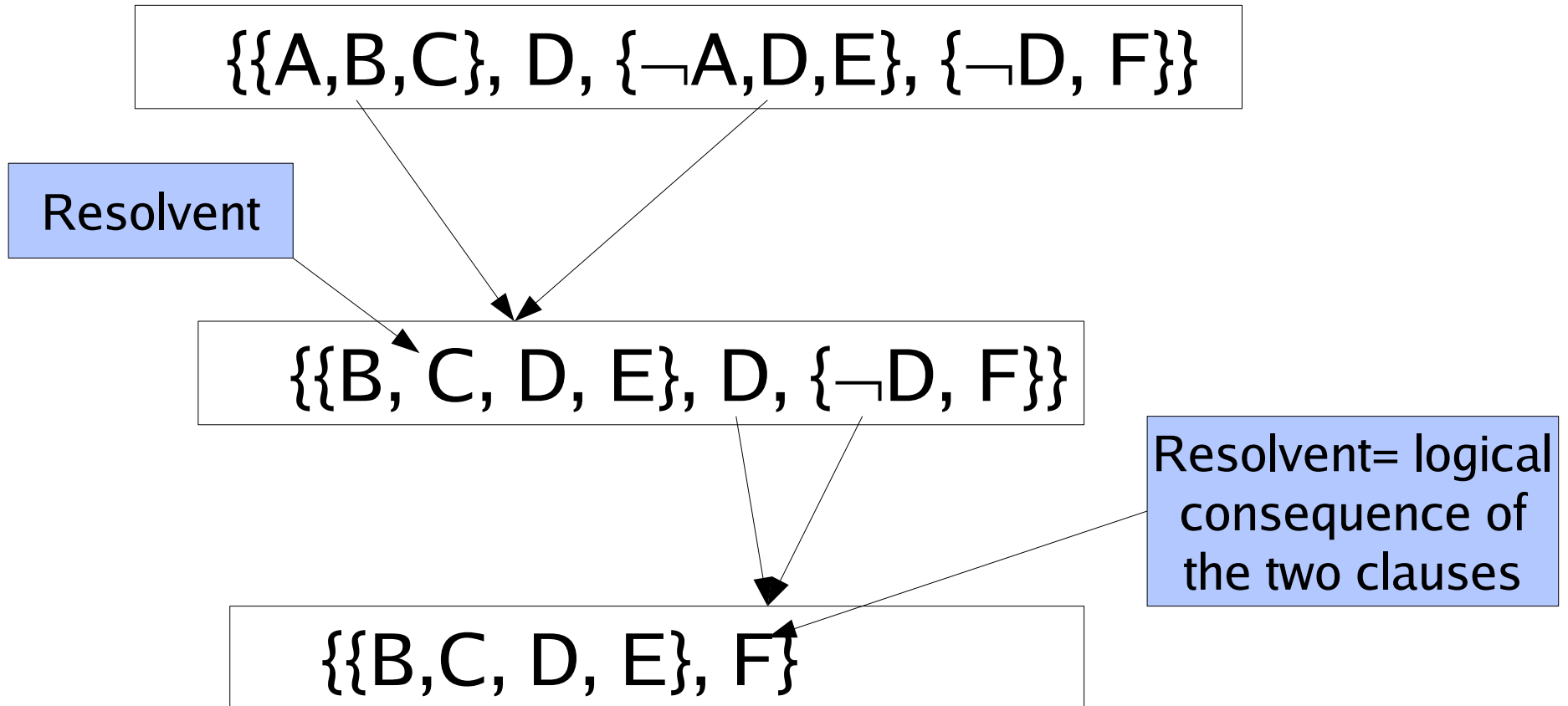
$$(P \vee Q \vee R) \wedge (S \vee \neg P) \wedge (P \vee T) = \{\{P, Q, R\}, \{S, \neg P\}, \{P, T\}\}$$

2. Resolve clauses containing complementary pairs of literals ( $L$  and  $\neg L$ ):

$$\text{resolve}(C_1, C_2) = C_1 \setminus \{L\} \cup C_2 \setminus \{\neg L\}$$

3. Continue until no clauses can be resolved or an empty clause is derived.

# 7.30 Example



# 7.31 Resolution in FOPL

- We can apply the same method as to propositional formulas.
- Here literals are predicates or their negations:

$$P(x_1, x_2, \dots, x_n), \neg P(x_1, x_2, \dots, x_n)$$

Positive

Negative

Clause containing  
a single literal=  
unit clause

- Clause is finite disjunction of literals:  $P(x_1, x_2)$
- CNF as defined before.
- First, we need to handle variables and quantifiers.

# 7.32 Normalization in FOPL

1. Eliminate equivalence and implication as before.

2. Move negation in :

- $\neg \exists x P(x) \equiv \forall x \neg P(x)$
- $\neg \forall x P(x) \equiv \exists x \neg P(x)$

3. Eliminate double negation.

4. Rename bound variables:

$\forall x P(x) \vee \exists x Q(x)$  becomes  $\forall x P(x) \vee \exists y Q(y)$ .

5. Move quantifiers to the front. For instance,

$$1. \forall x P(x) \vee R \equiv \forall x (P(x) \vee R)$$

$$2. \exists x P(x) \vee \exists x Q(x) \equiv \exists x (P(x) \vee Q(x))$$

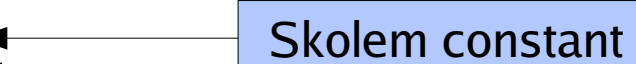
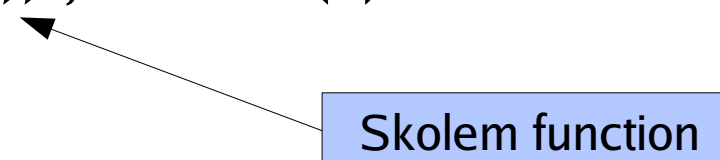
$$3. \exists x P(x) \wedge \forall y Q(y) \equiv \exists x \forall y (P(x) \wedge Q(y)) \text{ (altogether >10 rules)}$$

6. Eliminate existential quantifier by skolemization.

7. Drop universal quantifier, since after the previous step all variables are universally bound.

8. Distribute  $\wedge$  over  $\vee$  as before. Rename variables if required: note, that  $\forall x P(x) \vee \forall x Q(x)$  is not equivalent to  $\forall x (P(x) \vee Q(x))$ .

# 7.33 Skolemization

- Removes existential quantifiers by replacing existentially quantified variables with *skolem constants*:
  - For instance,  $\exists x P(x)$  becomes  $P(c)$ ; 
  - $c$  can be thought of as an instant that makes predicate true.
  - **Important!**  $c$  should not appear anywhere else in the sentence.
- Replaces existentially quantified variables with *skolem functions* when  $\exists$  follows a universal quantifier:
  - Replacing  $\forall x \exists y P(x,y)$  with  $\forall x P(x,c)$  would be wrong in most cases, e.g.,  $\forall x \exists y \text{mother}(y,x)$  (‘everybody has a mother’)
  - Correct skolemization:  $\forall x P(x,f(x))$ , where  $f(x)$  returns for instance, the mother of  $x$ . 

# 7.34 Example of normalization

- Convert  $\forall x (P(x) \rightarrow \exists y (Q(y) \wedge R(x,y)))$  to normal form:

$$\forall x (\neg P(x) \vee \exists y (Q(y) \wedge R(x,y))) \quad (\text{remove implication})$$

$$\forall x \exists y (\neg P(x) \vee (Q(y) \wedge R(x,y))) \quad (\text{quantifiers to front})$$

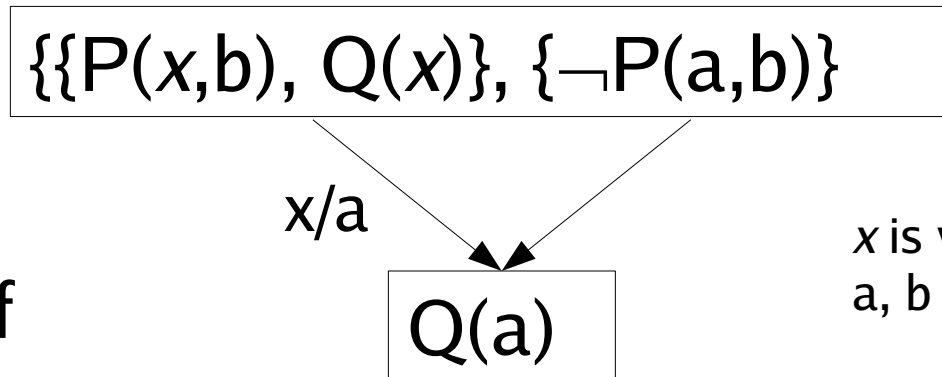
$$\forall x (\neg P(x) \vee (Q(f(x)) \wedge R(x,f(x)))) \quad (\text{remove } \exists)$$

$$(\neg P(x) \vee (Q(f(x)) \wedge R(x,f(x)))) \quad (\text{drop prefix})$$

$$(\neg P(x) \vee Q(f(x))) \wedge (\neg P(x) \vee R(x,f(x))) \quad (\text{distribute})$$

# 7.35 Unification

- Resolution is reasoning about individual objects or subsets of objects → need substitutions between matching literals. For instance, when resolving clauses:

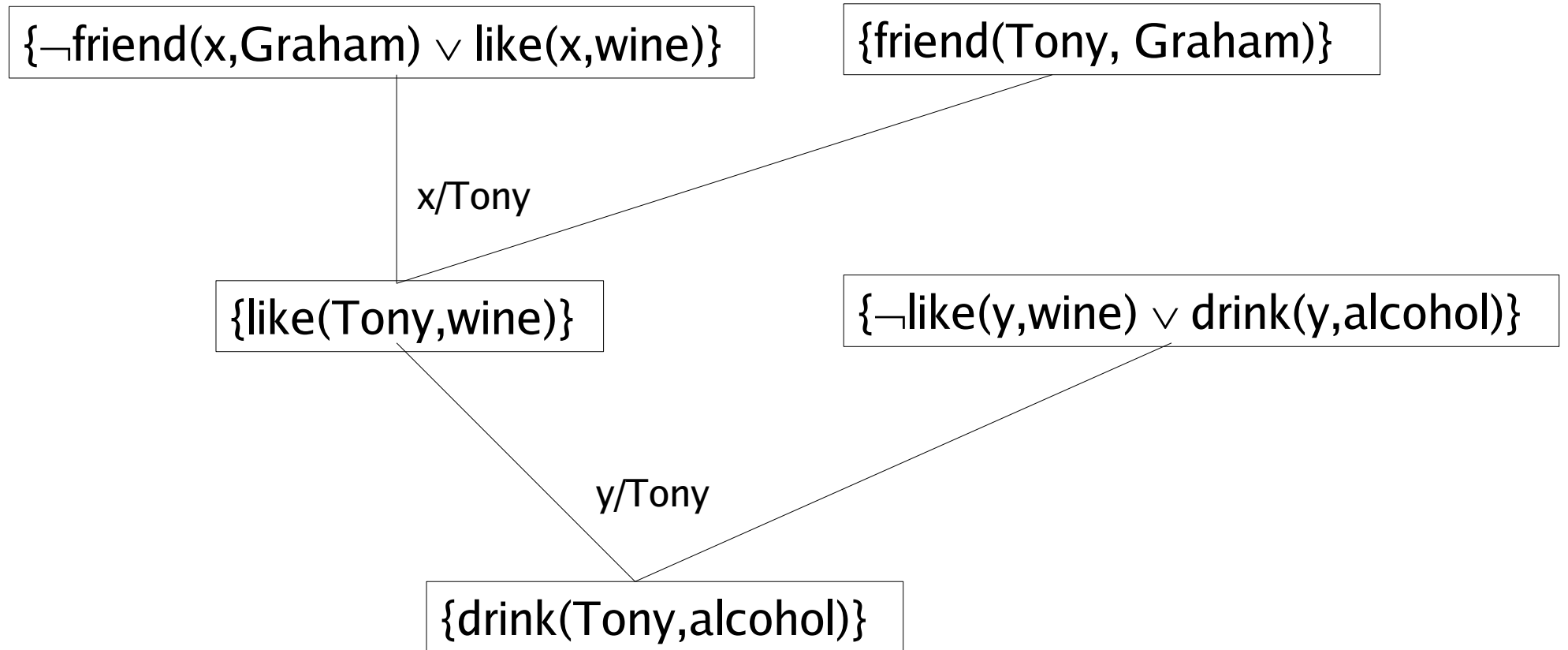


- Literals match if
  - Predicates have the same name and number of arguments.
  - Constants match to constants with the same name.
  - Constants match unbound variables.
  - Unbound variables match.

# 7.36 Example of resolution

- Premises & question:
  - All friends of Graham like wine.
  - Everyone who likes wine drinks alcohol.
  - Tony is a friend of Graham.
  - Does Tony like alcohol?
- Sentences is first-order predicate calculus:
  - $\forall x (\text{friend}(x, \text{Graham}) \rightarrow \text{like}(x, \text{wine}))$
  - $\forall y (\text{like}(x, \text{wine}) \rightarrow \text{drink}(y, \text{alcohol}))$
  - $\text{friend}(\text{Tony}, \text{Graham})$
- In normal form
  - $\neg \text{friend}(x, \text{Graham}) \vee \text{like}(x, \text{wine})$
  - $\neg \text{like}(y, \text{wine}) \vee \text{drink}(y, \text{alcohol})$
  - $\text{friend}(\text{Tony}, \text{Graham})$

# 7.37 Example (cont.)



# 7.38 Lottery paradox

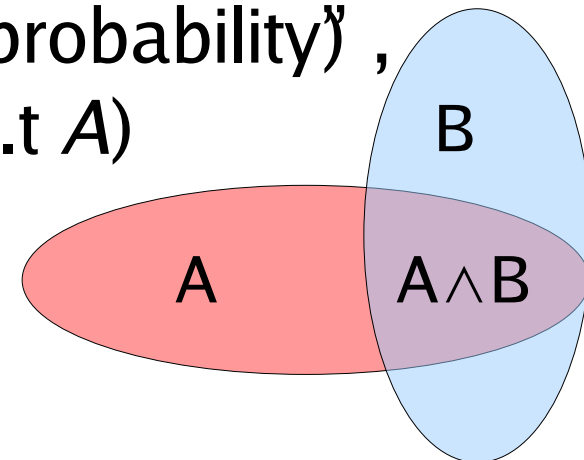
- You have a ticket  $t_k$ ,  $k \in \{1, \dots, 1,000,000\}$ , in a lottery with 1 million tickets.
- One and only one ticket wins.
- Probability that you win with  $t_k$  is  $p(t_k) = 10^{-6}$ .
- It seems reasonable to believe that  $t_k$  won't win, as well as  $t_1$  and  $t_2, \dots$ , and  $t_{1,000,000}$  won't win, i.e.,  $\neg \exists t_i (t_i \text{ will win})$ .
- But we know that  $\exists t_i (t_i \text{ will win})$ .
- How would you design an artificial reasoner that escapes the paradox?

# 7.39 Probabilistic reasoning

- Now we are talking about degrees of belief and probabilities instead of truths.
- We can still use (almost) the same vocabulary:

Do not count the same thing twice

- $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$  (“probability that either  $A$  or  $B$  (or both) is true”)
- $P(B|A) = P(A \wedge B)/P(A)$  (“conditional probability”, i.e., the relative volume of  $A \wedge B$ . (w.r.t  $A$ ))



# 7.40 Bayesian Probability

- Probability measures the degree of belief
  - $P(A)=0.7$  means “my belief in  $A$  is 70%”.
  - For instance,  $A$  could be “Boston is bigger than Helsinki” (by population count).
- Actually, all probabilities are conditional on some background knowledge  $K$ , (which often is not explicitly marked, i.e.,  $P(A)$  means actually  $P(A|K)$ ).
- So probability is an epistemic concept. It measures subjective knowledge.

# 7.41 Bayesian Modeling

- We are interested in some aspect  $\theta$  (theta) of the world, but we cannot observe it directly.
- We only have observations about the  $x$  that depends probabilistically on  $\theta$ .
- For example,  $\theta$  is the proportion of white balls in a bucket, and  $x$  is a color of a ball randomly drawn from the bucket.
  - If  $x$  is white, what should we think about  $\theta$ ?

# 7.42 Belief update

- If  $x$  is white, what should we think about  $\theta$ ?
- The answer (posterior) depends on two things:
  1. What we think about  $P(\theta)$  before (prior) we see the color of  $x$ :
    - i.e.,  $P(\theta=0.4)=0.3$  means that we were 30% sure that 40% of the balls in the bucket are white.
  2. For each possible value of  $\theta$ , how likely it is to see a white  $x$ :
    - $P(x=\text{white}|\theta=0.4)=0.4$  by definition.

# 7.43 Belief update

- Bayesian update rule says that after seeing a white ball  $x$ , our degree of belief about  $\theta$  being 0.4 should be:

$$P(\theta=0.4|x=\text{white})$$

$$= P(x=\text{white}|\theta=0.4) \times P(\theta=0.4) \times C,$$

- where  $C$  ensures that probabilities sum up to 1.
- This serves as a prior for the next ball.