Logical Inference 3 resolution

Chapter 9

Some material adopted from notes by Andreas Geyer-Schulz,, Chuck Dyer, and Mary Getoor

Resolution

- Resolution is a **sound** and **complete** inference procedure for unrestricted FOL
- Reminder: Resolution rule for propositional logic:
  - $P_1 \vee P_2 \vee \ldots \vee P_n$
  - $\neg P_1 \vee Q_2 \vee \ldots \vee Q_m$
  - Resolvent: $P_2 \vee \ldots \vee P_n \vee Q_2 \vee \ldots \vee Q_m$
- We'll need to extend this to handle quantifiers and variables

Two Common Normal Forms for a KB

Implicative normal form

- Set of sentences expressed as implications where left hand sides are conjunctions of 0 or more literals

P

Q

$P \wedge Q \Rightarrow R$

Conjunctive normal form

- Set of sentences expressed as disjunctions literals

P

Q

$\sim P \vee \sim Q \vee R$

- Recall: literal is an atomic expression or its negation
  e.g., loves(john, X), ~ hates(mary, john)
- Any KB of sentences can be expressed in either form

Resolution covers many cases

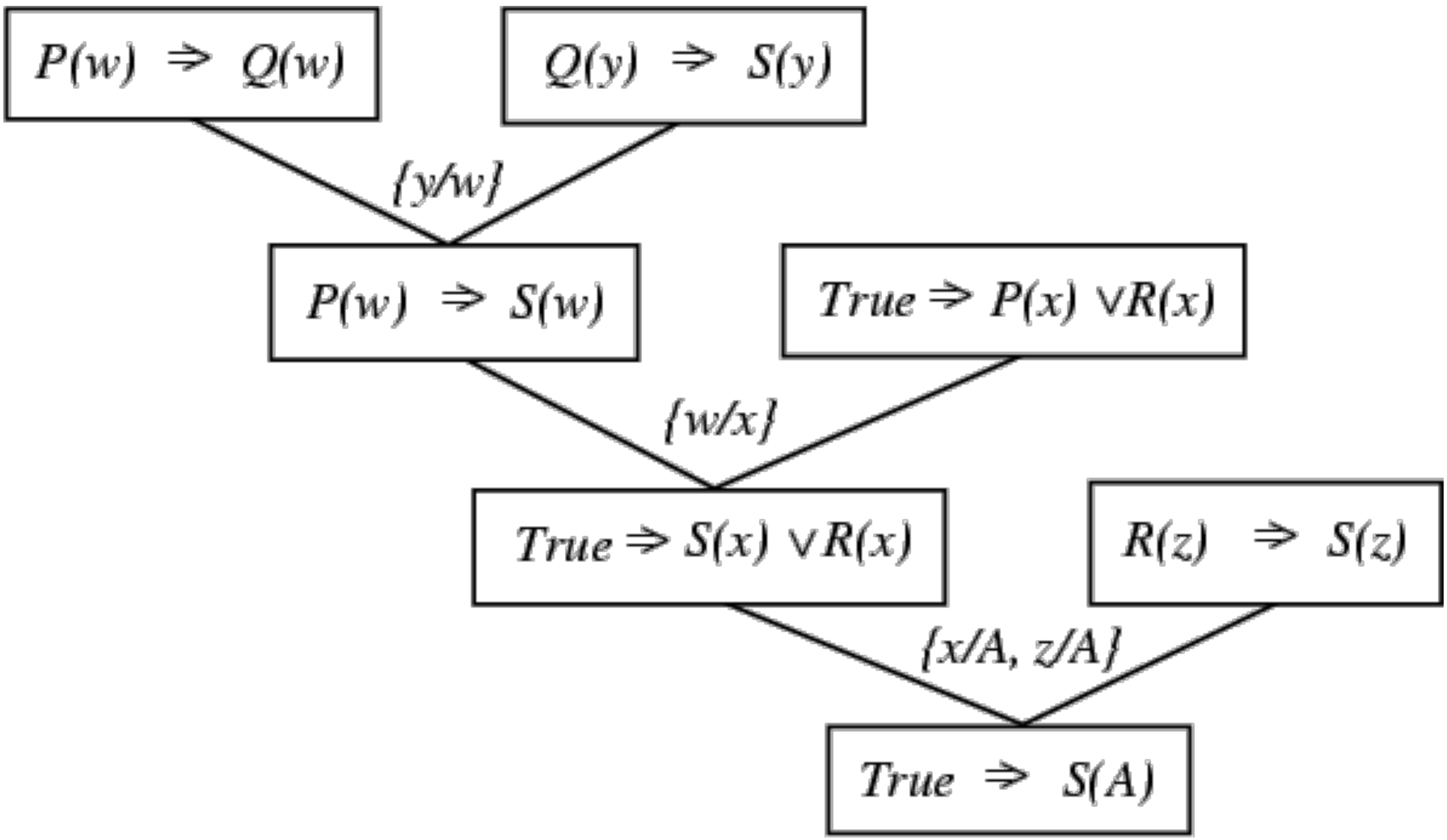
- Modes Ponens
  - from P and $P \rightarrow Q$ derive Q
  - from P and $\neg P \vee Q$ derive Q
- Chaining
  - from $P \rightarrow Q$ and $Q \rightarrow R$ derive $P \rightarrow R$
  - from $(\neg P \vee Q)$ and $(\neg Q \vee R)$ derive $\neg P \vee R$
- Contradiction detection
  - from P and $\neg P$ derive false
  - from P and $\neg P$ derive the empty clause (= false)

Resolution in first-order logic

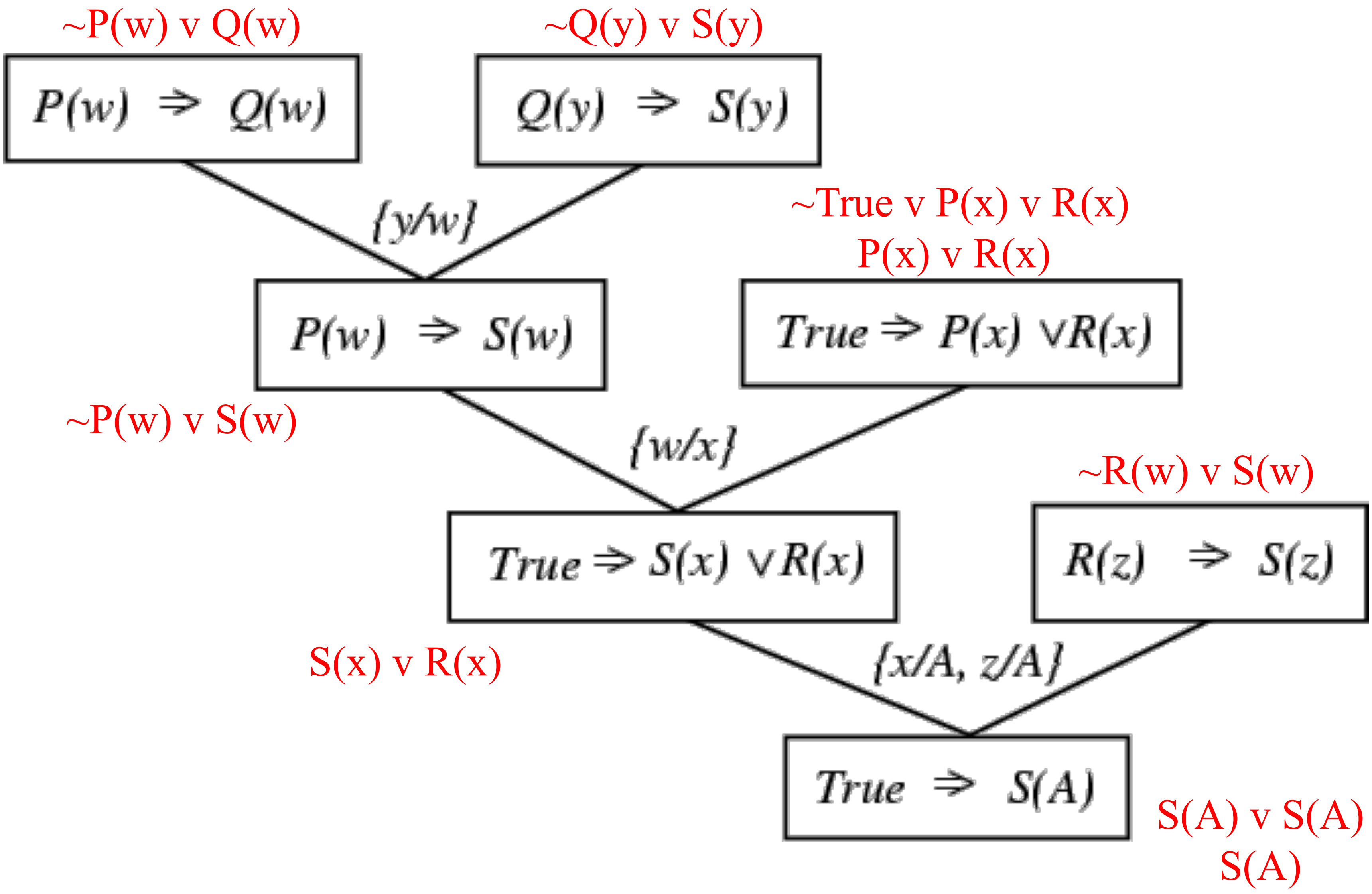
- Given sentences in *conjunctive normal form:*
  - $P_1 \vee \ldots \vee P_n$ and $Q_1 \vee \ldots \vee Q_m$
  - P_i and Q_i are literals, i.e., positive or negated predicate symbol with its terms
- if P_j and $\neg Q_k$ **unify** with substitution list θ, then derive the resolvent sentence:

  $\text{subst}(\theta, P_1 \vee \ldots \vee P_{j-1} \vee P_{j+1} \ldots P_n \vee Q_1 \vee \ldots Q_{k-1} \vee Q_{k+1} \vee \ldots \vee Q_m)$
- Example
  - from clause $\mathbf{P(x, f(a)) \vee P(x, f(y)) \vee Q(y)}$
  - and clause $\mathbf{\neg P(z, f(a)) \vee \neg Q(z)}$
  - derive resolvent $\mathbf{P(z, f(y)) \vee Q(y) \vee \neg Q(z)}$
  - Using $\boldsymbol{\theta = \{x/z\}}$

A resolution proof tree



A resolution proof tree



Resolution refutation (1)

- Given a consistent set of axioms KB and goal sentence Q, show that KB |= Q
- **Proof by contradiction:** Add ¬Q to KB and try to prove false, i.e.:

  (KB |- Q) ↔ (KB ∧ ¬Q |- False)

Resolution refutation (2)

- Resolution is **refutation complete**: can show sentence Q is entailed by KB, but can't always generate all consequences of a set of sentences
- Can't prove Q is **not entailed** by KB
- Resolution **won't always give an answer** since entailment is only semi-decidable
  - And you can't just run two proofs in parallel, one trying to prove Q and the other trying to prove ¬Q, since KB might not entail either one

Resolution example

- KB:
  - allergies(X) $\rightarrow$ sneeze(X)
  - cat(Y) $\wedge$ allergicToCats(X) $\rightarrow$ allergies(X)
  - cat(felix)
  - allergicToCats(mary)
- Goal:
  - sneeze(mary)

Refutation resolution proof tree

¬allergies(w) v sneeze(w)   ¬cat(y) v ¬allergicToCats(z) ∨ allergies(z)

w/z

¬cat(y) v sneeze(z) ∨ ¬allergicToCats(z)   cat(felix)

y/felix

sneeze(z) v ¬allergicToCats(z)   allergicToCats(mary)

z/mary

sneeze(mary)   ¬sneeze(mary)

false

negated query

Notation

old/new

Some tasks to be done

- Convert FOL sentences to conjunctive normal form (aka CNF, clause form): **normalization and skolemization**
- Unify two argument lists, i.e., how to find their most general unifier (**mgu**) q: **unification**
- Determine which two clauses in KB should be resolved next (among all resolvable pairs of clauses) : **resolution (search) strategy**