Resolution: Motivation

- Steps in inferencing (e.g., forward-chaining)
  1. Define a set of inference rules
  2. Define a set of axioms
  3. Repeatedly choose one inference rule & one or more axioms (or premices) to derive new sentences until the conclusion sentence is formed

- Basic requirement:

  Rules + axioms should constitute a complete proof system

- Observation:

  Automated inferencing could be a lot more efficient & easy to implement if there was just a single inference rule in the proof system!

Resolution

- Resolution (Robinson, 1965):

  A form of inference that relies on a single rule to prove the truth or falsity of logic sentences

- Because of its simplicity, efficiency & completeness properties, resolution has dominated reasoning in AI

Key characteristics:

- Resolution produces proofs by refutation:

  “To prove a statement, assume that the negation of the statement is true & try to arrive at a contradiction”

- Simplicity achieved by forcing inference rule to operate on sentences that have a very special form called Clause Normal Form (CNF)

- Completeness achieved because every logic sentence can be converted to CNF

The Resolution Rule

Resolution relies on the following rule:

$$\frac{\neg\alpha \Rightarrow \beta,\ \beta \Rightarrow \gamma}{\neg\alpha \Rightarrow \gamma} \qquad \text{Resolution rule}$$

equivalently,

$$\frac{\alpha \vee \beta,\ \neg\beta \vee \gamma}{\alpha \vee \gamma} \qquad \text{Resolution rule}$$

Applying the resolution rule:

1. Find two sentences that contain the same literal, once in its positive form & once in its negative form:

CNF sentences

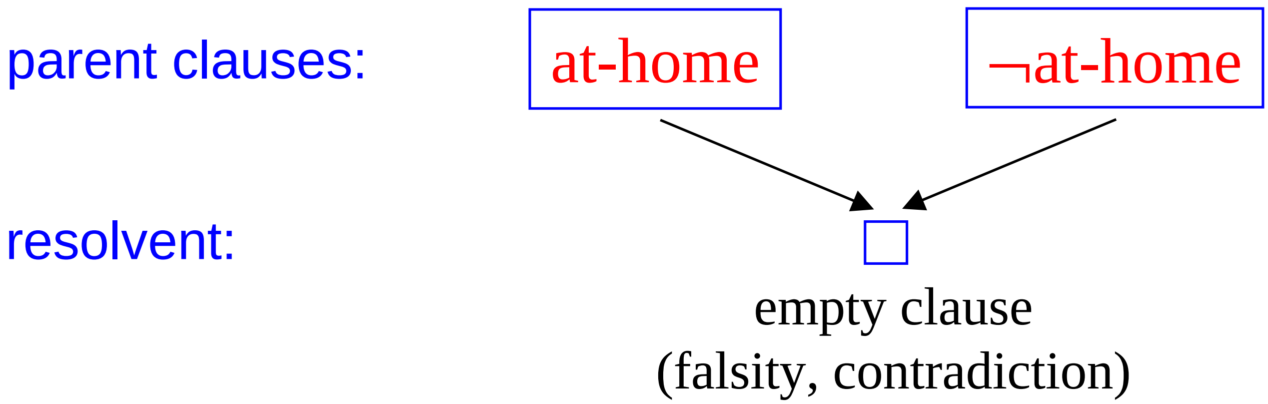
$\boxed{\text{summer} \vee \text{winter},\ \neg\text{winter} \vee \text{cold}}$

2. Use the resolution rule to eliminate the literal from both sentences

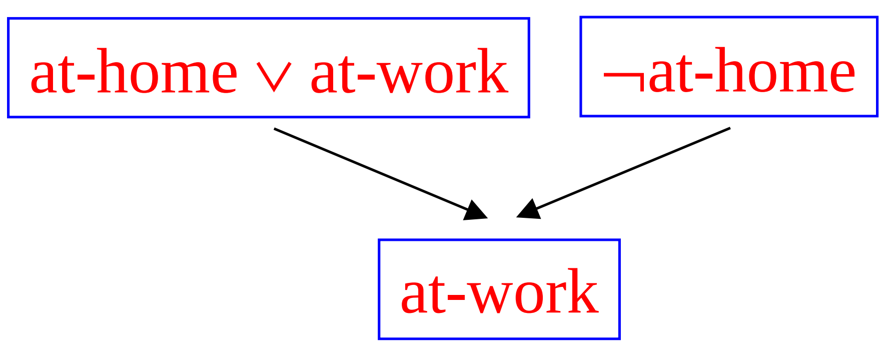
$\boxed{\text{summer} \vee \text{cold}}$

The Resolution Rule (cont.)

A resolution example:



Another example:



Observations:

- Resolution reduces the length of parent clauses by one literal
- Resolution applied after first converting all sentences to CNF form:
  - Disjunctions only
  - Negations of atoms only

Resolution in Propositional Logic

Basic steps for proving a proposition S:

1. Convert all propositions in premises to CNF

p		p
$(p \wedge q) \Rightarrow r$	$\neg(p \wedge q) \vee r$	$\neg p \vee \neg q \vee r$
$(s \vee t) \Rightarrow q$	$\neg(s \vee t) \vee q$	$\neg s \vee q$
		$\neg t \vee q$
t	t	t

CNF

2. Negate S & convert result to CNF

3. Add negated S to premises

4. Repeat until contradiction or no progress is made:
   a. Select 2 clauses (call them parent clauses)
   b. Resolve them together
   c. If resolvent is the empty clause, a contradiction has been found (i.e., S follows from the premises)
   d. If not, add resolvent to the premises

Resolution in Propositional Logic

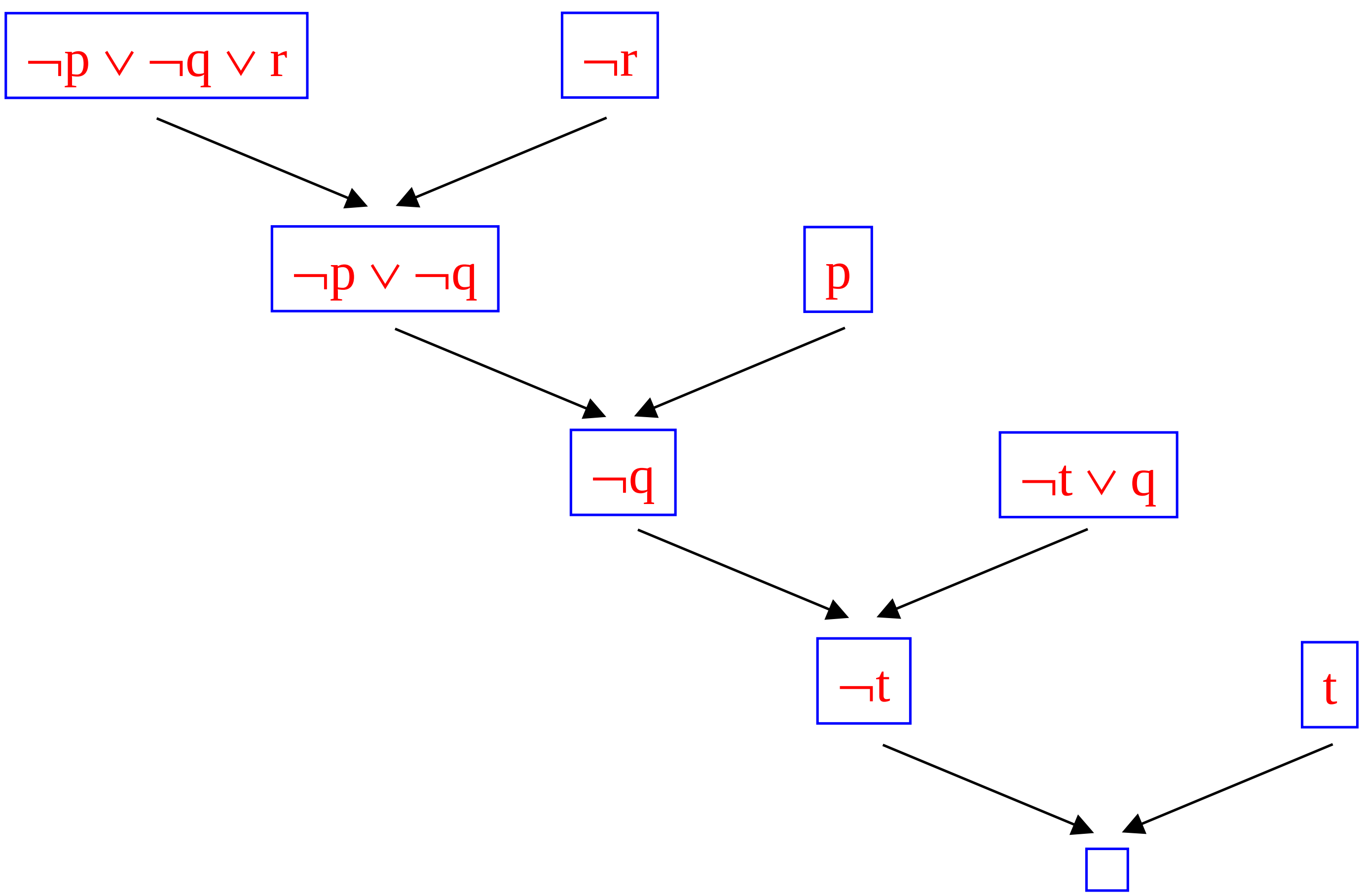
Premises:

p
$(p \wedge q) \Rightarrow r$
$(s \vee t) \Rightarrow q$
t

$\neg p \vee \neg q \vee r$
$\neg s \vee q$
$\neg t \vee q$
t

CNF

A resolution proof of r:



Resolution in First-Order Logic

In propopositional logic:

$\boxed{\text{at-home} \lor \text{at-work}}$ $\boxed{\neg\text{at-home}}$

$\downarrow$

$\boxed{\text{at-work}}$

In first-order logic:

$\boxed{\forall x.\text{at-home}(x) \lor \text{at-work}(x)}$ $\boxed{\neg\text{at-home}(y)}$

$\downarrow$

$\boxed{?}$

To generalize resolution proofs to FOL
we must account for

- Predicates
- Unbound variables
- Existential & universal quantifiers

Clause Form in First-Order Logic

- Disjunctions only
- Negations of atoms only

  $\neg P(A,B)$

- No quantifiers:
  - universal quantification implicit

    $\forall x.P(x) \quad \rightarrow \quad P(x)$

  - existential quantification replaced by Skolem constants/functions

    $\exists x.P(x) \quad \rightarrow \quad P(E)$

    $\forall y \exists x.P(x,y) \quad \rightarrow \quad P(E(y),y)$

Ordinary FOL		Clause Form
$P(A)$	none	$P(A)$
$\neg\neg Q(A,B)$	$\neg\neg$ elimination	$Q(A,B)$
$\neg(P(A) \wedge Q(B,C))$	deMorgan	$\neg P(A) \vee \neg Q(B,C)$
$\neg(P(A) \vee Q(B,C))$	deMorgan, $\wedge$ dropping	$\boxed{\neg P(A)}$, $\boxed{\neg Q(B,C)}$

2 unit clauses

Clause Form in First-Order Logic

Ordinary FOL		Clause Form
$P(A) \Rightarrow Q(B,C)$	$\xrightarrow{\Rightarrow \text{ elimination}}$	$\neg P(A) \vee Q(B,C)$
$\neg(P(A) \Rightarrow Q(B,C))$	$\xrightarrow[\text{deMorgan, } \wedge \text{ drop}]{\Rightarrow \text{ elimination}}$	$P(A), \neg Q(B,C)$
$P(A) \wedge (Q(B,C) \vee R(D))$	$\xrightarrow{\wedge \text{ drop}}$	$P(A), Q(B,C) \vee R(D)$
$P(A) \vee (Q(B,C) \wedge R(D))$	$\xrightarrow[\vee \text{ distribution}]{\wedge \text{ drop}}$	$P(A) \vee Q(B,C),$ $P(A) \vee R(D)$
$\forall x.P(x)$	$\xrightarrow{\forall \text{ drop}}$	$P(x)$
$\forall x.P(x) \Rightarrow Q(x,A)$	$\xrightarrow[\forall \text{ drop}]{\Rightarrow \text{ elimination}}$	$\neg P(x) \vee Q(x,A)$
$\exists x.P(x)$	$\xrightarrow{\text{skolemization}}$	$P(E)$, where E is a new constant
$P(A) \Rightarrow \exists x.Q(x)$	$\xrightarrow[\text{skolemization}]{\Rightarrow \text{ elimination}}$	$\neg P(A) \vee Q(F)$
$\neg\forall x.P(x)$ $\exists x.\neg P(x)$	$\xrightarrow[\text{skolemization}]{\text{deMorgan}}$	$\neg P(G)$

Clause Form in First-Order Logic

Ordinary FOL | **Clause Form**

$\neg\exists x.P(x)$

$\xrightarrow{\text{deMorgan}} \forall x.\neg P(x)$

$\xrightarrow[\forall\text{ drop}]{} \neg P(G)$

$\neg(\exists x.P(x) \wedge \forall x.Q(x))$

$\xrightarrow{\text{variable rename}} \neg(\exists x.P(x) \wedge \forall y.Q(y))$

$\xrightarrow{\text{deMorgan}} \neg\exists x.P(x) \vee \neg\forall y.Q(y)$

$\xrightarrow{\text{deMorgan}} \forall x.\neg P(x) \vee \exists y.\neg Q(y)$

$\xrightarrow[\text{skolemization}]{\forall\text{ drop}} \neg P(x) \vee \neg Q(H)$

$\forall x\exists y.P(x,y)$

$\xrightarrow{\text{fun. skolemization}} \forall x.P(x,K(x))$

$\xrightarrow{\forall\text{ drop}} P(x,K(x))$

$\forall x\forall y\exists z.P(x,y,z) \xrightarrow[\forall\text{ drop}]{\text{skolemization}} P(x,y,L(x,y))$

Clause Form in First-Order Logic

Ordinary FOL | Clause Form

$\forall x.P(x) \Rightarrow \exists y.Q(x,y) \xrightarrow[\text{skol., } \forall \text{ drop}]{\Rightarrow \text{ elimination}} \neg P(x) \lor Q(x,M(x))$

$(\forall x.P(x)) \Rightarrow \exists y.P(y)$

$\xrightarrow{\Rightarrow \text{ elimination}} (\neg\forall x.P(x)) \lor \exists y.P(y)$

$\xrightarrow{\text{deMorgan}} \exists x.\neg P(x) \lor \exists y.P(y)$

$\xrightarrow[\text{skolemization}]{} \neg P(N) \lor P(O)$

Conversion to Clause Form

Steps in general case:

1. Rename all variables so that all quantifiers bind distinct variables

2. $\Rightarrow$-elimination

3. deMorgan ($\neg\vee$, $\neg\wedge$, $\neg\forall$, $\neg\exists$)

4. Skolemization ($\exists$-elimination)

5. $\forall$-dropping

6. $\vee$-distribution

7. $\wedge$-dropping

Resolution in First-Order Logic

In propopositional logic:

at-home ∨ at-work   ¬at-home

at-work

In first-order logic:

To generalize resolution proofs to FOL
we must account for

- Predicates
- Unbound variables
- Existential & universal quantifiers

Idea: First convert sentences to clause form
Then unify variables

UNIFY

∀x.at-home(x) ∨ at-work(x)   ¬at-home(y)

?

Resolution Steps

Resolution steps for 2 clauses containing

P(arg.list1), ¬P(arg.list2)

1. Make the variables in the 2 clauses distinct

2. Find the "most general unifier" of arg.list1 & arg.list2:

   go through the lists "in parallel," making substitutions for variables only, so as to make the 2 lists the same

3. Make the substitutions corresponding to the m.g.u. throughout both clauses

4. The resolvent is the clause consisting of all the resulting literals except P & ¬P