Relational Logic: Logical Entailment with Resolution

Source: Computational Logic Lecture Notes
Stanford University

Informatics Engineering Study Program
School of Electrical Engineering and Informatics

Institute of Technology Bandung

Relational Logic: Logical Entailment with Resolution

Source: Computational Logic Lecture Notes
Stanford University

IF2121 Computational Logic 2023/2024

Informatics Engineering Study Program
School of Electrical Engineering and Informatics ITB

Agenda

- Relational Clausal Form: INSEADO
- Substitution
- Pattern Matching
- Unification
- Resolution Principle
- Resolution Theorem Proving

Resolution Principle

- The *Resolution Principle is a rule of inference.*
- Using the Resolution Principle alone (without axiom schemata or other rules of inference), it is possible to build a theorem prover that is sound and complete for all of Relational Logic.
- The search space using the Resolution Principle is much smaller than with standard axiom schemata.

Agenda

- Relational Clausal Form
- Substitution
- Pattern Matching
- Unification
- Resolution Principle
- Resolution Theorem Proving

Resolution: Clausal Form

- Relational resolution works only on expressions in *clausal form*.
- Fortunately, it is possible to convert any set of Relational Logic sentences into an equivalent set of sentences in clausal form.

Clausal Form

- A *literal is either an atomic sentence or a negation of an atomic* sentence.
- A *clausal sentence is either a literal or a disjunction of literals*.
- A *clause is a set of literals*.

$$\{p(a)\}$$

$$\{\neg p(a)\}$$

$$\{p(a), q(b)\}$$

- The empty clause {} is unsatisfiable.

INSEADO

- Implication Out

$$\phi 1 \Rightarrow \phi 2 \rightarrow \neg \phi 1 \vee \phi 2$$

$$\phi 1 \Leftarrow \phi 2 \rightarrow \phi 1 \vee \neg \phi 2$$

$$\phi 1 \Leftrightarrow \phi 2 \rightarrow (\neg \phi 1 \vee \phi 2) \wedge (\phi 1 \vee \neg \phi 2)$$

- Negation In

$$\neg \neg \phi \rightarrow \phi$$

$$\neg(\phi 1 \wedge \phi 2) \rightarrow \neg \phi 1 \vee \neg \phi 2$$

$$\neg(\phi 1 \vee \phi 2) \rightarrow \neg \phi 1 \wedge \neg \phi 2$$

$$\neg \forall v.\phi \rightarrow \exists v.\neg \phi$$

$$\neg \exists v.\phi \rightarrow \forall v.\neg \phi$$

INSEADO (2)

- Standardize variables

  $\forall x.p(x) \vee \forall x.q(x) \rightarrow \forall x.p(x) \vee \forall y.q(y)$

- Existentials Out

  $\exists x.p(x) \rightarrow p(a)$

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

INSEADO (3)

- Alls Out

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

- Distribution

$$\phi 1 \vee (\phi 2 \wedge \phi 3) \rightarrow (\phi 1 \vee \phi 2) \wedge (\phi 1 \vee \phi 3)$$

$$(\phi 1 \wedge \phi 2) \vee \phi 3 \rightarrow (\phi 1 \vee \phi 3) \wedge (\phi 2 \vee \phi 3)$$

- Operators Out

$$\phi 1 \wedge \ldots \wedge \phi n \rightarrow \phi 1$$

$$\ldots$$

$$\phi n$$

$$\phi 1 \vee \ldots \vee \phi n \rightarrow \{\phi 1, \ldots, \phi n\}$$

Example

	$\exists y.(g(y) \wedge \forall z.(r(z) \Rightarrow f(y,z)))$
I	$\exists y.(g(y) \wedge \forall z.(\neg r(z) \vee f(y, z)))$
N	$\exists y.(g(y) \wedge \forall z.(\neg r(z) \vee f(y, z)))$
S	$\exists y.(g(y) \wedge \forall z.(\neg r(z) \vee f(y, z)))$
E	$g(greg) \wedge \forall z.(\neg r(z) \vee f(greg, z)))$
A	$g(greg) \wedge (\neg r(z) \vee f(greg, z)))$
D	$g(greg) \wedge (\neg r(z) \vee f(greg, z)))$
O	$\{g(greg)\}$
	$\{\neg r(z), f(greg,z)\}$

Another Example

$\neg\exists y.(g(y) \wedge \forall z.(r(z) \Rightarrow f(y,z)))$	
I	$\neg\exists y.(g(y) \wedge \forall z.(\neg r(z) \vee f(y,z)))$
N	$\forall y.(\neg(g(y) \wedge \forall z.(\neg r(z) \vee f(y,z)))$
	$\forall y.(\neg g(y) \vee \neg\forall z.(\neg r(z) \vee f(y,z)))$
	$\forall y.(\neg g(y) \vee \exists z.\neg(\neg r(z) \vee f(y,z)))$
	$\forall y.(\neg g(y) \vee \exists z.(\neg\neg r(z) \wedge \neg f(y,z)))$
	$\forall y.(\neg g(y) \vee \exists z.(r(z) \wedge \neg f(y,z)))$
S	$\forall y.(\neg g(y) \vee \exists z.(r(z) \wedge \neg f(y,z)))$
E	$\forall y.(\neg g(y) \vee (r(k(y)) \wedge \neg f(y,k(y))))$
A	$\neg g(y) \vee (r(k(y)) \wedge \neg f(y,k(y)))$
D	$(\neg g(y) \vee r(k(y))) \wedge (\neg g(y) \vee \neg f(y,k(y)))$
O	$\{\neg g(y) \vee r(k(y))\}$
	$\{\neg g(y) \vee \neg f(y,k(y))\}$

Exercise 1

Ubah ke bentuk klausa

$\exists w.\forall x.(\exists y.(\neg p(x, y) \wedge r(y)) \leftarrow \exists z.(q(w, z)))$

Solution Exercise

	$\exists w.\forall x.(\exists y.(\neg p(x, y) \wedge r(y)) \leftarrow \exists z.(q(w, z)))$
I	$\exists w.\forall x.\ (\exists z.(q(w, z)) \rightarrow \exists y.(\neg p(x, y) \wedge r(y)))$
	$\exists w.\forall x.\ (\neg\exists z.(q(w, z)) \vee \exists y.(\neg p(x, y) \wedge r(y)))$
N	$\exists w.\forall x.\ (\forall z.\neg q(w, z) \vee \exists y.(\neg p(x, y) \wedge r(y)))$
S	$\exists w.\forall x.\ (\forall z.\neg q(w, z) \vee \exists y.(\neg p(x, y) \wedge r(y)))$
E	$\forall x.\ (\forall z.\neg q(a, z) \vee \exists y.(\neg p(x, y) \wedge r(y)))$
	$\forall x.\ (\forall z.\neg q(a, z) \vee (\neg p(x, f(x)) \wedge r(f(x))))$
A	$\neg q(a, z) \vee (\neg p(x, f(x)) \wedge r(f(x)))$
D	$(\neg q(a, z) \vee \neg p(x, f(x))) \wedge (\neg q(a, z) \vee r(f(x)))$
O	$\{\neg q(a, z), \neg p(x, f(x))\}$
	$\{\neg q(a, z), r(f(x))\}$

Exercise 2

Dari daftar kalimat berikut ini, tentukan variabel yang bebas (*free*) dan variabel yang terikat (*bound*), dan *quantifier* yang mana yang mengikat variabel tersebut. Catatan: jika diperlukan, diperbolehkan mengganti nama variabel untuk menghindari kebingungan

a. $\forall x.\ \forall z.\ (p(x,y) \vee q(y,z) \vee \exists x.\ (s(z,x)))$
b. $\forall z.\ (p(x,y) \vee q(y,z) \vee \exists x.\ (s(x,z)))$
c. $\forall x.\ \forall z.\ (p(x,y) \vee q(y,z)) \vee \exists x.\ (s(z,x))$

Solution Exercise 2

a. $\forall \underline{w}.\ \forall \underline{\underline{z}}.\ (p(\underline{w},y) \vee q(y,\underline{\underline{z}}) \vee \exists x.\ (s(\underline{\underline{z}}, x)))$

b. $\forall \underline{z}.\ (p(x,y) \vee q(y,\underline{z}) \vee \exists \underline{\underline{w}}.\ (s(\underline{\underline{w}},\underline{z})))$

c. $\forall \underline{x}.\ \forall \underline{\underline{w}}.\ (p(\underline{x},y) \vee q(y, \underline{\underline{w}})) \vee \exists x.\ (s(z, x))$

Exercise 3

Ubahlah ketiga kalimat relational logic berikut ke dalam bentuk klausa.

a. $\exists x. \forall y.(p(x,y) \leftrightarrow q(x,y))$

b. $\forall x.(\exists y.p(x,y) \vee \exists z.q(x,z))$

c. $\neg(\exists x.\exists y.(p(x,y) \wedge q(x,y)))$

Solution Exercise 3

a.

I $\exists x. \forall y.((\neg p(x,y) \vee q(x,y)) \wedge (p(x,y) \vee \neg q(x,y)))$

N $\exists x. \forall y.((\neg p(x,y) \vee q(x,y)) \wedge (p(x,y) \vee \neg q(x,y)))$

S $\exists x. \forall y.((\neg p(x,y) \vee q(x,y)) \wedge (p(x,y) \vee \neg q(x,y)))$

E $\forall y.((\neg p(a,y) \vee q(a,y)) \wedge (p(a,y) \vee \neg q(a,y)))$

A $(\neg p(a,y) \vee q(a,y)) \wedge (p(a,y) \vee \neg q(a,y))$

D $(\neg p(a,y) \vee q(a,y)) \wedge (p(a,y) \vee \neg q(a,y))$

O $\{\neg p(a,y), q(a,y)\}$

$\{ p(a,y), \neg q(a,y)\}$

Solution Exercise 3

b.

I $\forall x.(\exists y.p(x,y) \vee \exists z.q(x,z))$

N $\forall x.(\exists y.p(x,y) \vee \exists z.q(x,z))$

S $\forall x.(\exists y.p(x,y) \vee \exists z.q(x,z))$

E $\forall x.(p(x,f(x)) \vee q(x,h(x)))$

A $(p(x,f(x)) \vee q(x,h(x)))$

D $(p(x,f(x)) \vee q(x,h(x)))$

O $\{p(x,f(x)),q(x,h(x))\}$

Solution Exercise 3

c.

I $\neg(\exists x.\exists y.(p(x,y) \wedge q(x,y)))$

N $\forall x. \forall y.(\neg p(x,y) \vee \neg q(x,y))$

S $\forall x. \forall y.(\neg p(x,y) \vee \neg q(x,y))$

E $\forall x. \forall y.(\neg p(x,y) \vee \neg q(x,y))$

A $(\neg p(x,y) \vee \neg q(x,y))$

D $(\neg p(x,y) \vee \neg q(x,y))$

O $\{ \neg p(x,y), \neg q(x,y)\}$

Substitution

- A *substitution is a finite set of pairs of variables and* terms. The variables together constitute the *domain of the* substitution, and the terms are called *replacements*.

$$\{x \leftarrow a, y \leftarrow f(b), z \leftarrow v\}$$

- A substitution is *pure if and only if all replacements are* free of the variables in the domain of the substitution.
- Otherwise, the substitution is *impure*.

$$\{x \leftarrow a, y \leftarrow f(b), z \leftarrow x\}$$

Application

- The result of applying a substitution σ to an expression ϕ is the expression $\phi\sigma$ obtained from the original expression by replacing every occurrence of every variable in the substitution by the term with which it is associated.
  - $q(x, y)\{x\leftarrow a, y\leftarrow f(b), z\leftarrow v\} = q(a, f(b))$
  - $q(x, x)\{x\leftarrow a, y\leftarrow f(b), z\leftarrow v\} = q(a, a)$
  - $q(x,w)\{x\leftarrow a, y\leftarrow f(b), z\leftarrow v\} = q(a,w)$
  - $q(z,v)\{x\leftarrow a, y\leftarrow f(b), z\leftarrow v\} = q(v,v)$

Idempotence

- For pure substitution, application is idempotent.

$q(x, x, y, w, z)\{x \leftarrow a, y \leftarrow f(b), z \leftarrow v\} = q(a, a, f(b), w, v)$

$q(a, a, f(b), w, v)\{x \leftarrow a, y \leftarrow f(b), z \leftarrow v\} = q(a, a, f(b), w, v)$

- Not so for impure substitutions.

$q(x, x, y, w, z)\{x \leftarrow a, y \leftarrow f(b), z \leftarrow x\} = q(a, a, f(b), w, x)$

$q(a, a, f(b), w, x)\{x \leftarrow a, y \leftarrow f(b), z \leftarrow x\} = q(a, a, f(b), w, a)$

Composition of Substitutions

- The *composition of substitution σ and τ is the substitution* (written στ) obtained by

(1) applying τ to the replacements in σ

(2) adjoining to σ the pairs from τ with different variables

(3) deleting any assignments of variable to itself .

$\{x \leftarrow a, y \leftarrow f(u), z \leftarrow v\}\{u \leftarrow d, v \leftarrow e, z \leftarrow g\}$

$=\{x \leftarrow a, y \leftarrow f(d), z \leftarrow e\}\{u \leftarrow d, v \leftarrow e, z \leftarrow g\}$

$=\{x \leftarrow a, y \leftarrow f(d), z \leftarrow e, u \leftarrow d, v \leftarrow e\}$

Purity

- The composition of impure substitutions may be pure.

$\{x \leftarrow a, y \leftarrow f(x), z \leftarrow c\}\{x \leftarrow b, z \leftarrow g(x)\} = \{x \leftarrow a, y \leftarrow f(b), z \leftarrow c\}$

- The composition of pure substitutions may be impure.

$\{x \leftarrow a\}\{y \leftarrow f(x)\} = \{x \leftarrow a, y \leftarrow f(x)\}$

Composability

- A substitution σ and a substitution τ are composable if and only if the variables in the domain of σ do not appear among the replacements of τ.

$\{x \leftarrow a, y \leftarrow b, z \leftarrow v\}\{x \leftarrow u, v \leftarrow b\}$

- Otherwise, they are *noncomposable*.

$\{x \leftarrow a, y \leftarrow b, z \leftarrow v\}\{u \leftarrow x, v \leftarrow b\}$

- Theorem: The composition of composable pure substitutions must be pure.

Pattern Matching

- A substitution σ is a *matcher for a* pattern ϕ and an expression ψ if and only if $\phi\sigma=\psi$.
- An expression ψ *matches a pattern* ϕ *if and only if there is a* matcher for ϕ and ψ.
- Example:

p(a,b) matches p(x,y)

$p(x,y)\{x \leftarrow a, y \leftarrow b\} = p(a,b)$

Unification

- A substitution σ is a *unifier for an expression ϕ and an* expression ψ if and only if $\phi\sigma=\psi\sigma$.

$p(x,y)\{x\leftarrow a,y\leftarrow b,v\leftarrow b\}=p(a,b)$

$p(a,v)\{x\leftarrow a,y\leftarrow b,v\leftarrow b\}=p(a,b)$

- If two expressions have a unifier, they are said to be *unifiable*. Otherwise, they are *nonunifiable*.

$p(a,b)$

$p(b,a)$

Non-Uniqueness of Unification

- Unifier 1:

$$p(x,y)\{x\leftarrow a,y\leftarrow b,v\leftarrow b\}=p(a,b)$$

$$p(a,v)\{x\leftarrow a,y\leftarrow b,v\leftarrow b\}=p(a,b)$$

- Unifier 2:

$$p(x,y)\{x\leftarrow a,y\leftarrow f(w),v\leftarrow f(w)\}=p(a,f(w))$$

$$p(a,v)\{x\leftarrow a,y\leftarrow f(w),v\leftarrow f(w)\}=p(a,f(w))$$

- Unifier 3:

$$p(x,y)\{x\leftarrow a,y\leftarrow v\}=p(a,v)$$

$$p(a,v)\{x\leftarrow a,y\leftarrow v\}=p(a,v)$$

Generality of Unifiers

- A unifier σ is *as general as or more general than a unifier* τ if and only if there exists a substitution γ such that $\tau=\sigma\gamma$.

  $\{x \leftarrow a, y \leftarrow v\}\{v \leftarrow f(w)\}=\{x \leftarrow a, y \leftarrow f(w), v \leftarrow f(w)\}$

Most General Unifier (MGU)

- A substitution σ is a *most general unifier of two expressions if* and only if it is as general as or more general than any other unifier.
- Theorem: If two expressions are unifiable, then they have a most general unifier that is unique up to variable permutation.

$$p(x,y)\{x\leftarrow a, y\leftarrow v\}=p(a,v)$$

$$p(a,v)\{x\leftarrow a, y\leftarrow v\}=p(a,v)$$

$$p(x,y)\{x\leftarrow a, v\leftarrow y\}=p(a,y)$$

$$p(a,v)\{x\leftarrow a, v\leftarrow y\}=p(a,y)$$

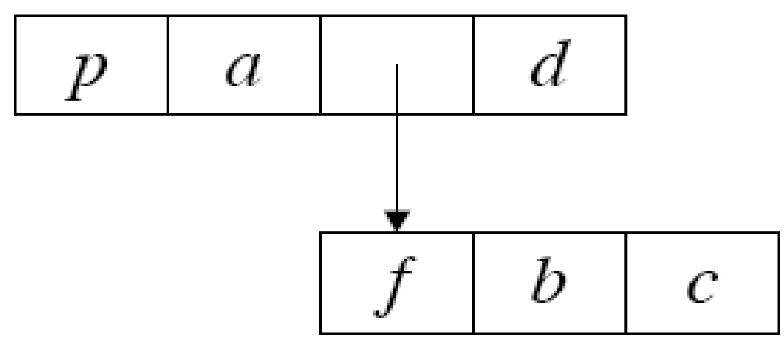
Expression Structure

Each expression is treated as a sequence of its immediate subexpressions.

Linear Version:

$$p(a, f(b, c), d)$$

Structured Version:



Procedure for computing MGU

- Start: two expressions and a substitution

- Steps:

1. If two expressions being compared are identical, then nothing more needs to be done.
2. If two expressions are not identical and both expressions are constants, then we fail, since there is no way to make them look alike.
3. If one of the expressions is a variable, we first check whether the variable has a binding in the current substitution. If so, we try to unify the binding with the second expression. If there is no binding, we check whether the second expression contains the variable. If the variable occurs within the expression, we fail; otherwise, we set the substitution to the composition of the old substitution and a new substitution in which we bind the variable to the second expression.
4. The only remaining possibility is that the two expressions are both sequences. In this case, we simply iterate across the expressions, comparing as described above.

Example

Find MGU of p(x,b) and p(a,y)

Compare: $p(x,b)$, $p(a,y)$, {}

  Compare: p, p, {}

  Result: {}

  Compare: x, a, {}

  Result: $\{x \leftarrow a\}$

  Compare: y, b, $\{x \leftarrow a\}$

  Result: $\{x \leftarrow a, y \leftarrow b\}$

Result: $\{x \leftarrow a, y \leftarrow b\}$

Another example

- Find MGU of p(x,x) and p(a,y)

Compare: $p(x,x)$, $p(a,y)$, {}

    Compare: p, p, {}

    Result: {}

    Compare: x, a, {}

    Result: $\{x \leftarrow a\}$

    Compare: x, y, $\{x \leftarrow a\}$

        Compare: a, y, $\{x \leftarrow a\}$

        Result: $\{x \leftarrow a, y \leftarrow a\}$

    Result: $\{x \leftarrow a, y \leftarrow a\}$

Result: $\{x \leftarrow a, y \leftarrow a\}$

Another Example

- Find MGU of p(x,x) and p(a,b)

Compare: $p(x,x)$, $p(a,$b$)$, {}

Compare: p, p, {}

Result: {}

Compare: x, a, {}

Result: $\{x \leftarrow a\}$

Compare: x, b, $\{x \leftarrow a\}$

Compare: a, b, $\{x \leftarrow a\}$ → If two expressions are not identical and both expressions are constants, then we fail

Result: Fail

Result: Fail

Result: Fail

Exercise

Tentukan MGU untuk no 1 – 3 jika unifiable

1. p(x,b,f(y)) dan p(a,y,f(z))
2. p(f(g(a)), x, f(h(z, z)), h(y, g(w))) dan p(y, g(z), f(v), h(f(w), x))
3. p(t(x, y), r(z, z)) dan p(t(t(w, z), v), w)

Solution Exercise

Tentukan MGU untuk no 1 - 3

1. p(x,b,f(y)) dan p(a,y,f(z))

unifiable dengan MGU: **{x←a, y←b, z←b}**

2. p(f(g(a)), x, f(h(z, z)), h(y, g(w))) dan p(y, g(z), f(v), h(f(w), x))

unifiable dengan MGU: **{v←h(g(a),g(a)), w←g(a), x←g(g(a)), y←f(g(a)), z←g(a) }**

3. p(t(x, y), r(z, z)) dan p(t(t(w, z), v), w)

unifiable dengan MGU: **{w←r(z,z), y←v, x←t(r(z,z),z)}**

Proof Method: Resolution

Previously:

- Relational Clausal Form (INSEADO)
- Substitution
- Pattern Matching
- Unification

Now:

- Resolution Principle
- Resolution Theorem Proving

Resolution: Clausal Form

- Relational resolution works only on expressions in *clausal form*.
- Fortunately, it is possible to convert any set of Relational Logic sentences into an equivalent set of sentences in clausal form → INSEADO
  - Implication Out
  - Negation In
  - Standardize variables
  - Existentials Out
  - Alls Out
  - Distribution
  - Operators Out

Propositional Resolution

$$\frac{\{\phi 1, \ldots, \phi, \ldots, \phi m\} \quad \{\psi 1, \ldots, \neg\phi, \ldots, \psi n\}}{\{\phi 1, \ldots, \phi m, \psi 1, \ldots, \psi n\}}$$

Relational Resolution Principle I

$\{\phi 1,\ldots,\ \phi,\ldots,\ \phi m\}$

$\{\psi 1,\ldots,\ \neg\psi,\ldots,\ \psi n\}$

$\{\phi 1,\ldots,\ \phi m, \psi 1,\ldots,\ \psi n\}\sigma$

Where $\sigma = mgu(\phi,\psi\ )$

Example:

$\{\ p(a, y), r(y)\}$

$\{\neg p(x,b)\}$

$\{r(y)\}\{x \leftarrow a, y \leftarrow b\}$

$\{r(b)\}$

Problem

$\{\ p(a, x), r(x)\}$

$\{\neg p(x,b)\}$

Failure

Solution: Relational Resolution Principle II

Relational Resolution Principle II

$\{\phi 1, \ldots, \phi, \ldots, \phi m\}$

$\{\psi 1, \ldots, \neg\psi, \ldots, \psi n\}$

$\{\phi 1\tau, \ldots, \phi m\tau, \psi 1, \ldots, \psi n\}\sigma$

Where $\sigma = mgu(\phi\tau, \psi)$

where τ is a variable renaming on ϕ

- Example:

$\{\, p(a, x), r(x)\}$

$\{\neg p(x,b)\}$

Failure

$\{\, p(a, y), r(y)\}$

$\{\neg p(x,b)\}$

$\{r(y)\}\{x \leftarrow a, y \leftarrow b\}$

$\{r(b)\}$

Provability

- A resolution derivation of a clause ϕ from a set Δ of clauses is a sequence of clauses terminating in ϕ in which each item is:

  (1) a member of Δ or

  (2) the result of applying the resolution principle to early items in sequence.
- A sentence ϕ is *provable from a set of sentences Δ* by resolution if and only if there is a derivation of the empty clause from the clausal form of $\Delta \cup \{\neg\phi\}$.
- A resolution proof is a derivation of the empty clause from the clausal form of the premises and the negation of the desired conclusion.

Example

- Everybody loves somebody. Everybody loves a lover. Show that everybody loves everybody.

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

  $\forall u. \forall v. \forall w. (loves(v,w) \Rightarrow loves(u, v))$

  $\neg \forall x. \forall y. loves(x, y)$

- *Clausal Form (INSEADO):*

  $\{loves(x, f(x))\}$

  $\{\neg loves(v,w), loves(u,v)\}$

  $\{\neg loves(jack, jill)\}$

Example (con't)

1. $\{loves(x, f(x))\}$	*Premise*
2. $\{\neg loves(v,w), loves(u,v)\}$	*Premise*
3. $\{\neg loves(jack, jill)\}$	*Negated Goal*
4. $\{loves(u, x)\}$	1,2 MGU: $\{v \leftarrow x, w \leftarrow f(x)\}$
5. $\{\}$	4,3 MGU: $\{u \leftarrow jack, x \leftarrow jill\}$

Relational Resolution Principle III

- Problem:

$\{p(x), p(y)\}$

$\{\neg p(u), \neg p(v)\}$

$\{p(y), \neg p(v)\}$

$\{p(x), \neg p(v)\}$

$\{p(y), \neg p(u)\}$

$\{p(x), \neg p(u)\}$

Factor

- If a subset of the literals in a clause Φ has a most general unifier γ, then the clause Φ' obtained by applying γ to Φ is called a *factor of Φ*.
- Clause

  $\{p(x), p(f(y)), r(x,y)\}$
- Factors

  $\{p(f(y)), r(f(y),y)\}$

  $\{p(x), p(f(y)), r(x,y)\}$

Relational Resolution Principle III (final)

Φ

Ψ

$((\Phi' - \{\phi\})\tau \cup (\Psi' - \{\neg\psi\}))\sigma$

Where $\phi \in \Phi'$, a factor of Φ

where $\neg\psi \in \Psi'$, a factor of Ψ

Where $\sigma = mgu(\phi\tau,\psi)$

where τ is a variable renaming on ϕ

Example

$\{p(x), p(y)\}$

$\{\neg p(u), \neg p(v)\}$

$\{p(y), \neg p(v)\}$

$\{p(x), \neg p(v)\}$

$\{p(y), \neg p(u)\}$

$\{p(x), \neg p(u)\}$

$\{p(x)\}$

$\{\neg p(u)\}$

$\{\}$

Review

- Unification → MGU
- Relational Resolution Proof:
  - Transform to Clausal Form → INSEADO
  - Apply Relational Resolution Princples to derive an empty clause

    → I, II, III

Exercise 1

- We know that horses are faster than dogs and that there is a greyhound that is faster than every rabbit. We know that Harry is a horse and that Ralph is a rabbit. Derive the fact that Harry is faster than Ralph.
- $\forall x,y.(horse(x) \wedge dog(y) \rightarrow faster(x,y))$
- $\exists x.(greyhound(x) \wedge \forall y.(rabbit(y) \rightarrow faster(x,y)))$
- horse(Harry)
- rabbit(Ralph)
- $\forall x.(greyhound(x) \rightarrow dog(x))$
- $\forall x,y,z.(faster(x,y) \wedge faster(y,z) \rightarrow faster(x,z))$
- Goal: faster(Harry,Ralph)

Exercise 2

- All hungry animals are caterpillars. All caterpillars have 42 legs. Edward is a hungry animal. Therefore, Edward has 42 legs.
- Relation constant: hungry(x), caterpillar(x), 42legs(x)

Exercise 3

1. **Art is the father of Jon, that Bob is the father of Kim, and that fathers are parents.**
   - Prove that Art is a parent of Jon.
   - Relation constants: father(x,y), parent(x,y)

Review

- **Logical Entailment:**

  To determine whether a set Δ of sentences logically entails a closed sentence φ, rewrite $\Delta \cup \{\varphi \rightarrow goal\}$ in clausal form and try to derive goal.

- **Answer Extraction:**

  To get values for free variables $v_1, \ldots, v_n$ in φ for which Δ logically entails φ, rewrite $\Delta \cup \{\varphi \rightarrow goal(v_1, \ldots, v_n)\}$ in clausal form and try to derive $goal(v_1, \ldots, v_n)$

- **Intuition**

  The sentence (q(z) → goal(z)) says that, whenever, z satisfies q, it satisfies the "goal"

Example

- Given (p(x) → q(x)) and p(a) and p(b), find a term T such that q(T) is true

1.	$\{\neg p(x), q(x)\}$	$p(x) \Rightarrow q(x)$
2.	$\{p(a)\}$	$p(a)$
3.	$\{p(b)\}$	$p(b)$
4.	$\{\neg q(z), goal(z)\}$	$q(z) \Rightarrow goal(z)$
5.	$\{\neg p(z), goal(z)\}$	1,4
6.	$\{goal(a)\}$	2,5
7.	$\{goal(b)\}$	3,5

Example

- Given (p(x) → q(x)) and (p(a) V p(b)), find a term τ such that q(τ) is true

1.	$\{\neg p(x), q(x)\}$	$p(x) \Rightarrow q(x)$
2.	$\{p(a), p(b)\}$	$p(a) \vee p(b)$
3.	$\{\neg q(z), goal(z)\}$	$q(z) \Rightarrow goal(z)$
4.	$\{\neg p(z), goal(z)\}$	1,3
5.	$\{p(b), goal(a)\}$	2,4
6.	$\{goal(a), goal(b)\}$	4,5

Example Kinship

- Art is the parent of Bob and Bud.
- Bob is the parent of Cal and Coe.
- A grandparent is a parent of a parent.

- p(art,bob)
- p(art,bud)
- p(bob,cal)
- p(bob,coe)
- $p(x, y) \land p(y,z) \rightarrow g(x,z)$

Example Kinship

- Is Art the Grandparent of Coe? (logical entailment)

1.	$\{p(art,bob)\}$	$p(art,bob)$
2.	$\{p(art,bud)\}$	$p(art,bud)$
3.	$\{p(bob,cal)\}$	$p(bob,cal)$
4.	$\{p(bob,coe)\}$	$p(bob,coe)$
5.	$\{\neg p(x,y), \neg p(y,z), g(x,z)\}$	$p(x,y) \wedge p(y,z) \Rightarrow g(x,z)$
6.	$\{\neg g(art,coe), goal\}$	$g(art,coe) \Rightarrow goal$
7.	$\{\neg p(art,y), \neg p(y,coe), goal\}$	5,6
8.	$\{\neg p(bob,coe), goal\}$	1,7
9.	$\{goal\}$	4,8

Example Kinship

- Who is the Grandparent of Coe? (Answer extraction)

1.	$\{p(art,bob)\}$	$p(art,bob)$
2.	$\{p(art,bud)\}$	$p(art,bud)$
3.	$\{p(bob,cal)\}$	$p(bob,cal)$
4.	$\{p(bob,coe)\}$	$p(bob,coe)$
5.	$\{\neg p(x,y), \neg p(y,z), g(x,z)\}$	$p(x,y) \wedge p(y,z) \Rightarrow g(x,z)$
6.	$\{\neg g(x,coe), goal(x)\}$	$g(x,coe) \Rightarrow goal(x)$
7.	$\{\neg p(x,y), \neg p(y,coe), goal(x)\}$	5,6
8.	$\{\neg p(bob,coe), goal(art)\}$	1,7
9.	$\{goal(art)\}$	4,8

Example Kinship

Who Are the Grandchildren of Art? (Answer Extraction)

1.	$\{p(art,bob)\}$	$p(art,bob)$
2.	$\{p(art,bud)\}$	$p(art,bud)$
3.	$\{p(bob,cal)\}$	$p(bob,cal)$
4.	$\{p(bob,coe)\}$	$p(bob,coe)$
5.	$\{\neg p(x,y), \neg p(y,z), g(x,z)\}$	$p(x,y) \wedge p(y,z) \Rightarrow g(x,z)$
6.	$\{\neg g(art,z), goal(z)\}$	$g(art,z) \Rightarrow goal(z)$
7.	$\{\neg p(art,y), \neg p(y,z), goal(z)\}$	5,6
8.	$\{\neg p(bob,z), goal(z)\}$	1,7
9.	$\{\neg p(bud,z), goal(z)\}$	2,7
10.	$\{goal(cal)\}$	3,8
11.	$\{goal(coe)\}$	4,8

Example Kinship

People and their Grandchildren?

1.	$\{p(art,bob)\}$	$p(art,bob)$
2.	$\{p(art,bud)\}$	$p(art,bud)$
3.	$\{p(bob,cal)\}$	$p(bob,cal)$
4.	$\{p(bob,coe)\}$	$p(bob,coe)$
5.	$\{\neg p(x,y), \neg p(y,z), g(x,z)\}$	$p(x,y) \wedge p(y,z) \Rightarrow g(x,z)$
6.	$\{\neg g(x,z), goal(x,z)\}$	$g(x,z) \Rightarrow goal(x,z)$
7.	$\{\neg p(x,y), \neg p(y,z), goal(x,z)\}$	5,6
8.	$\{\neg p(bob,z), goal(art,z)\}$	1,7
9.	$\{\neg p(bud,z), goal(art,z)\}$	2,7
10.	$\{goal(art,cal)\}$	3,8
11.	$\{goal(art,coe)\}$	4,8

THANK YOU