Relational Logic

Propositional Logic

Premises:

If Jack knows Jill, then Jill knows Jack.

Jack knows Jill.

Conclusion:

Is it the case that Jill knows Jack?

Problem

Premises:

If one person knows another, then the second person knows the first.

Jack knows Jill.

Conclusion:

Is it the case that Jill knows Jack?

How do we represent the first premise in a way that allows us to derive the desired conclusion?

Relational Logic

New Linguistic Features:

- Variables
- Quantifiers

Sample Sentence:

$$\forall x.\forall y.(knows(x,y) \Rightarrow knows(y,\mathrm{x}))$$

Programme

Syntax

Semantics

Examples, Examples, Examples

Properties of Sentences

Logical Entailment

Decidability

Syntax

Components of Language

Words

$$a, b, g, p$$

Terms

$$g(a,a)$$

Sentences

$$\forall x.(p(x) \Rightarrow p(x,g(x,x)))$$

Words

Words are strings of letters, digits, and occurrences of the underscore character.

Variables begin with characters from the end of the alphabet (from u through z).

$$u, v, w, x, y, z$$

Constants begin with digits or letters from the beginning of the alphabet (from a through t).

$$a, b, c, \mathit{123}, \mathit{comp225}, \mathit{barack\_obama}$$

Constants

Object constants represent objects.

joe, *stanford*, *usa*, 2345

Relation constants represent relations.

knows, *loves*

Arity

The *arity* of a relation constant is the number of arguments it takes.

Unary relation constant - 1 argument

Binary relation constant - 2 arguments

Ternary relation constant - 3 arguments

n-ary relation constant - *n* arguments

Signatures

A *signature* consist of a set of object constants and a set of relation constants together with a specification of arity for the relation constants.

Object Constants: a, b

Unary Relation Constant: p
Binary Relation Constant: q

Terms

A *term* is either a variable or an object constant,.

Terms represent objects.

Terms are analogous to noun phrases in natural language.

Sentences

Three types of sentences in Relational Logic:

Relational sentences - analogous to the simple sentences in natural language

Logical sentences - analogous to the logical sentences in natural language

Quantified sentences - sentences that express the significance of variables

Relational Sentences

A *relational sentence* is an expression formed from an n-ary relation constant and n terms enclosed in parentheses and separated by commas.

$$q(a,y)$$

Relational sentences are *not* terms and *cannot* be nested in relational sentences.

No! $q(a,q(a,y))$ No!

Logical Sentences

Logical sentences in Herbrand Logic are analogous to those in Propositional Logic.

$$(\neg q(a,b))$$
$$(p(a) \wedge p(b))$$
$$(p(a) \vee p(\mathrm{b}))$$
$$(q(x,y) \Rightarrow q(y,x))$$
$$(q(x,y) \Leftrightarrow q(y,x))$$

Quantified Sentences

Universal sentences assert facts about all objects.

$$(\forall x.(p(x) \Rightarrow q(x,x)))$$

Existential sentence assert the existence of objects with given properties.

$$(\exists x.(p(x) \wedge q(x,x)))$$

Quantified sentences can be nested within other sentences.

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

Parentheses

Parentheses can be removed when precedence allows us to reconstruct sentences correctly.

Precedence relations same as in Propositional Logic with quantifiers being of *higher* precedence than logical operators.

$$\forall x.p(x) \Rightarrow q(x,x) \rightarrow (\forall x.p(x)) \Rightarrow q(x,x)$$
$$\exists x.p(x) \wedge q(x,x) \rightarrow (\exists x.p(x)) \wedge q(x,x)$$

Ground and Non-Ground Expressions

An expression is *ground* if and only if it contains no variables.

Ground sentence:

$$p(a)$$

Non-Ground Sentence:

$$\forall x.p(x)$$

Bound and Free Variables

An occurrence of a variable is **bound** if and only if it lies in the scope of a quantifier of that variable. Otherwise, it is **free**.

$$\exists y.q(x,y)$$

In this example, x is free and y is bound.

Open and Closed Sentences

A sentence is **open** if and only if it has free variables. Otherwise, it is **closed**.

Open sentence:

$$\exists y.q(x,y)$$

Closed Sentence:

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

Semantics

Herbrand Base

The *Herbrand base* for a Relational language is the set of all ground relational sentences that can be formed from the vocabulary of the language.

Example

Object Constants: a, b
Unary Relation Constant: p
Binary Relation Constant: q

Herbrand Base:

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

Truth Assignment

A *truth assignment* is an association between ground atomic sentences and the truth values *true* or *false*. As with Propositional Logic, we use 1 as a synonym for *true* and 0 as a synonym for *false*.

$$p(a)^i = 1$$
$$p(b)^i = 0$$
$$q(a,a)^i = 1$$
$$q(a,b)^i = 0$$
$$q(b,a)^i = 1$$
$$q(b,b)^i = 0$$

Sentential Truth Assignment

A *sentential truth assignment* is an association between arbitrary sentences in a Herbrand language and the truth values 1 and 0.

Truth Assignment	Sentential Truth Assignment
$p(a)^i = 1$	$(p(a) \vee p(b))^i = 1$
$p(b)^i = 0$	$(p(a) \wedge \neg p(b))^i = 1$

Each base truth assignment leads to a particular sentential truth assignment based on the type of sentence.

Logical Sentences

$(\neg\varphi)^i = 1$ if and only if $\varphi^i = 0$

$(\varphi \wedge \psi)^i = 1$ if and only if $\varphi^i = 1$ and $\psi^i = 1$

$(\varphi \vee \psi)^i = 1$ if and only if $\varphi^i = 1$ or $\psi^i = 1$

$(\varphi \Rightarrow \psi)^i = 1$ if and only if $\varphi^i = 0$ or $\psi^i = 1$

$(\varphi \Leftrightarrow \psi)^i = 1$ if and only if $\varphi^i = \psi^i$

Instances

An *instance* of an expression is an expression in which all free variables have been consistently replaced by ground terms.

Consistent replacement here means that, if one occurrence of a variable is replaced by a ground term, then all occurrences of that variable are replaced by the same ground term.

Quantified Sentences

A *universally quantified sentence* is true for a truth assignment if and only if every instance of the scope of the quantified sentence is true for that assignment.

An *existentially quantified sentence* is true for a truth assignment if and only if some instance of the scope of the quantified sentence is true for that assignment.

Example

Truth Assignment:

$$p(a)^i = 1 \qquad q(a,a)^i = 1$$
$$p(b)^i = 0 \qquad q(a,b)^i = 0$$
$$q(b,a)^i = 1$$
$$q(b,b)^i = 0$$

Sentence:

$$\forall x.(p(x) \Rightarrow q(x,x))$$

Instances:

$$p(a) \Rightarrow q(a,a)$$
$$p(b) \Rightarrow q(b,b)$$

Example

Truth Assignment:

$$p(a)^i = 1 \qquad q(a,a)^i = 1$$
$$p(b)^i = 0 \qquad q(a,b)^i = 0$$
$$q(b,a)^i = 1$$
$$q(b,b)^i = 0$$

Sentence:

$$\forall x.(p(x) \Rightarrow q(x,x))$$

Instances:

$$p(a) \Rightarrow q(a,a) \checkmark$$
$$p(b) \Rightarrow q(b,b)$$

Example

Truth Assignment:

$$p(a)^i = 1 \qquad q(a,a)^i = 1$$
$$p(b)^i = 0 \qquad q(a,b)^i = 0$$
$$q(b,a)^i = 1$$
$$q(b,b)^i = 0$$

Sentence:

$$\forall x.(p(x) \Rightarrow q(x,x))$$

Instances:

$$p(a) \Rightarrow q(a,a) \ \surd$$
$$p(b) \Rightarrow q(b,b) \ \surd$$

Example

Truth Assignment:

$$p(a)^i = 1 \qquad q(a,a)^i = 1$$
$$p(b)^i = 0 \qquad q(a,b)^i = 0$$
$$q(b,a)^i = 1$$
$$q(b,b)^i = 0$$

Sentence:

$$\forall x.(p(x) \Rightarrow q(x,x)) \; \surd$$

Instances:

$$p(a) \Rightarrow q(a,a) \; \surd$$
$$p(b) \Rightarrow q(b,b) \; \surd$$

Example

Truth Assignment:

$$p(a)^i = 1 \qquad q(a,a)^i = 1$$
$$p(b)^i = 0 \qquad q(a,b)^i = 0$$
$$q(b,a)^i = 1$$
$$q(b,b)^i = 0$$

Sentence:

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

Instances:

$$\exists y.q(a,y) \qquad \exists y.q(b,y)$$
$$q(a,a) \qquad q(b,a)$$
$$q(a,b) \qquad q(b,b)$$

Open Sentences

A truth assignment satisfies *a sentence with free variables* if and only if it satisfies every instance of that sentence. (In other words, we can think of all free variables as being universally quantified.)

$$(\exists y.q(x,y))^i = (\forall x.\exists y.q(x,y))^i$$

A truth assignment satisfies *a set of sentences* if and only if it satisfies every sentence in the set.

Example - Sorority World

Sorority World

	Abby	Bess	Cody	Dana
Abby			✓	
Bess			✓	
Cody	✓	✓		✓
Dana			✓	

Signature

Object Constants: *abby*, *bess*, *cody*, *dana*

Binary Relation Constant: *likes*

Herbrand base has 16 ground relational sentences.

Data

$\neg likes(abby,abby)$
$\neg likes(abby,bess$
$likes(abby,cody)$
$\neg likes(abby,dana)$

$likes(cody,abby)$
$likes(cody,bess)$
$\neg likes(cody,cody)$
$likes(cody,dana)$

$\neg likes(bess,abby)$
$\neg likes(bess,bess)$
$likes(bess,cody)$
$\neg likes(bess,dana)$

$\neg likes(dana,abby)$
$\neg likes(dana,bess)$
$likes(dana,cody)$
$\neg likes(dana,dana)$

Sentences

Abby likes everyone Bess likes.
If Bess likes a girl, then Abby also likes her.

$$\forall y.(likes(bess,y) \Rightarrow likes(abby,y))$$

Cody likes everyone who likes her.
If some girl likes Cody, then Cody likes that girl.

$$\forall x.(likes(x,cody) \Rightarrow likes(cody,x))$$

Sentences

Cody likes somebody who likes her.
There is someone who likes cody and is liked by Cody.

$$\exists y.(likes(cody,y) \wedge likes(y,cody))$$

Nobody likes herself.
It is not the case that someone likes herself.

$$\neg \exists x.likes(x,x)$$

Sentences

Everybody likes somebody.

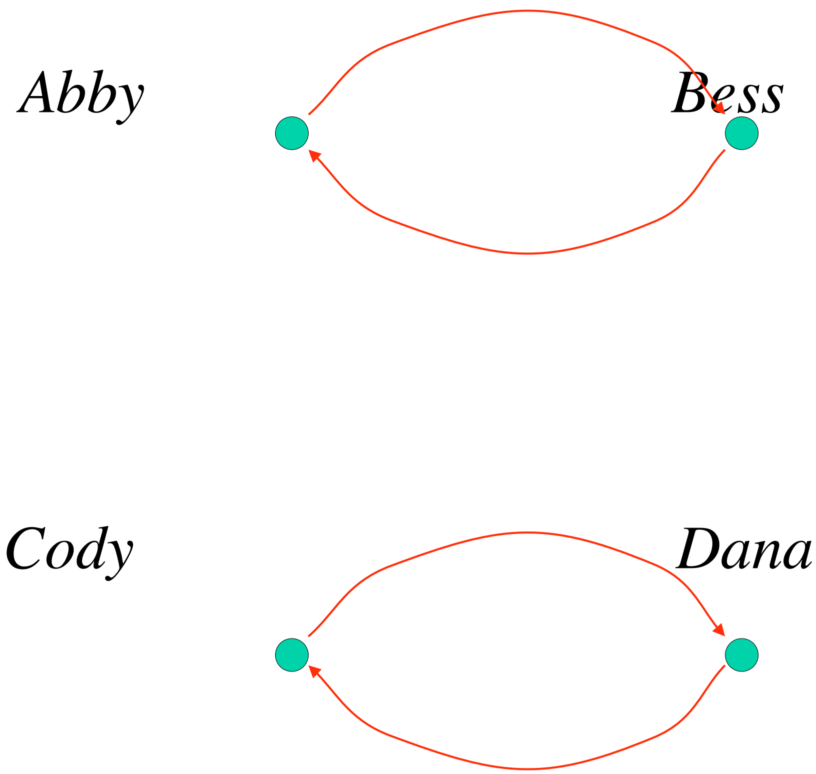
$$\forall x.\exists y.likes(x,y)$$

There is somebody whom everybody likes.

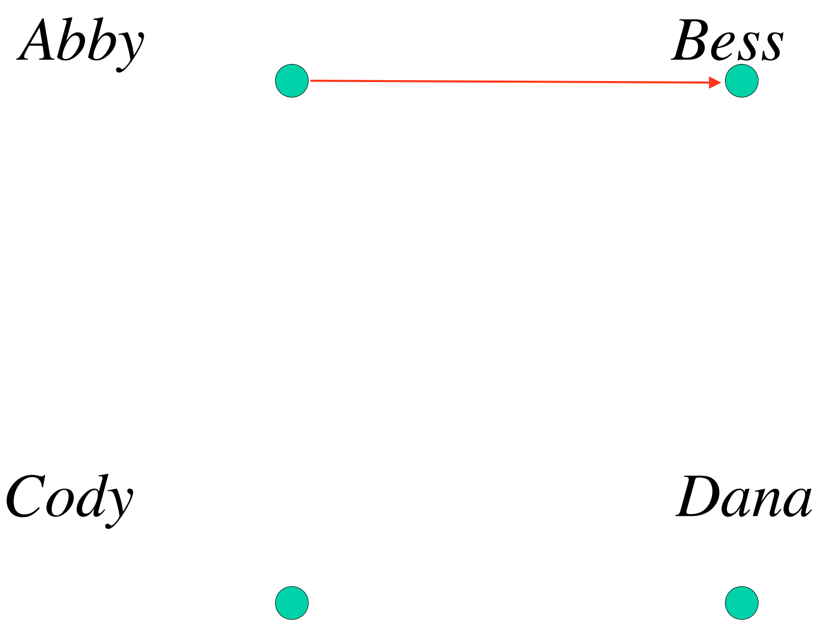
$$\exists y.\forall x.likes(x,y)$$

Example

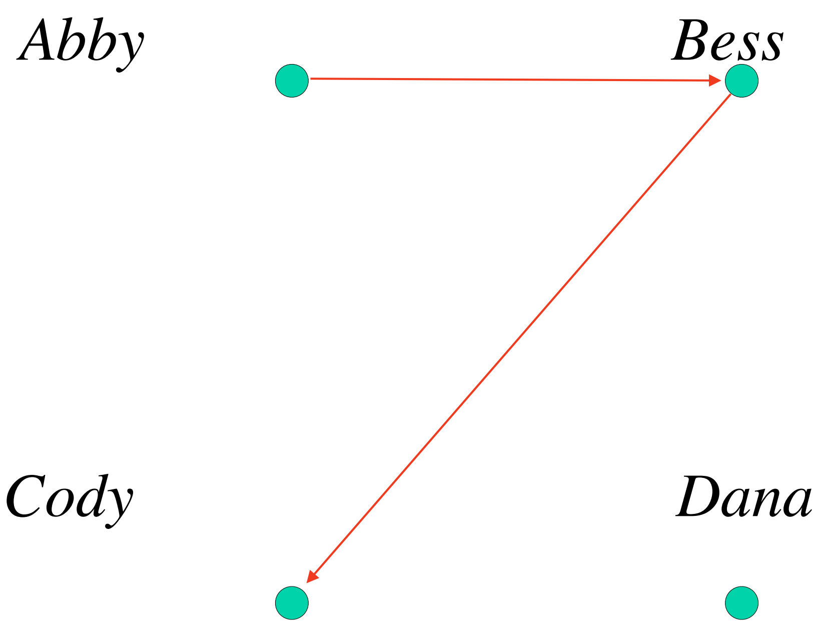
Everybody Likes Somebody



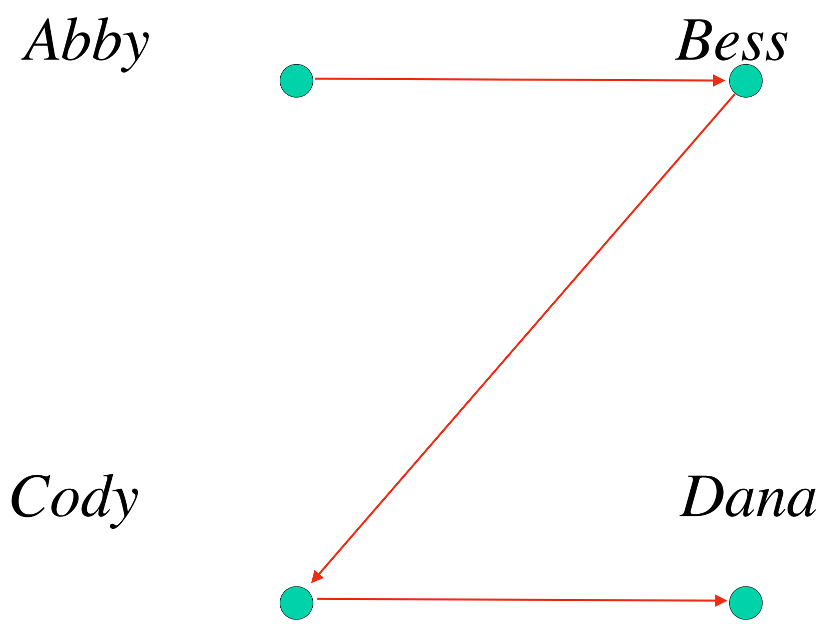
Everybody Likes Somebody



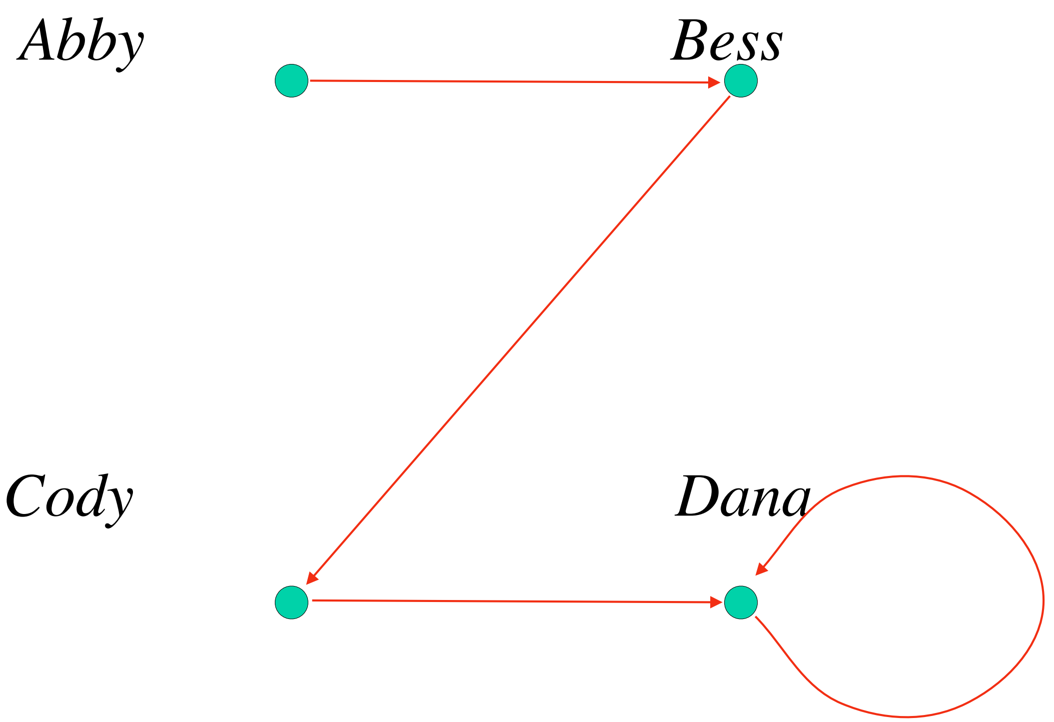
Everybody Likes Somebody



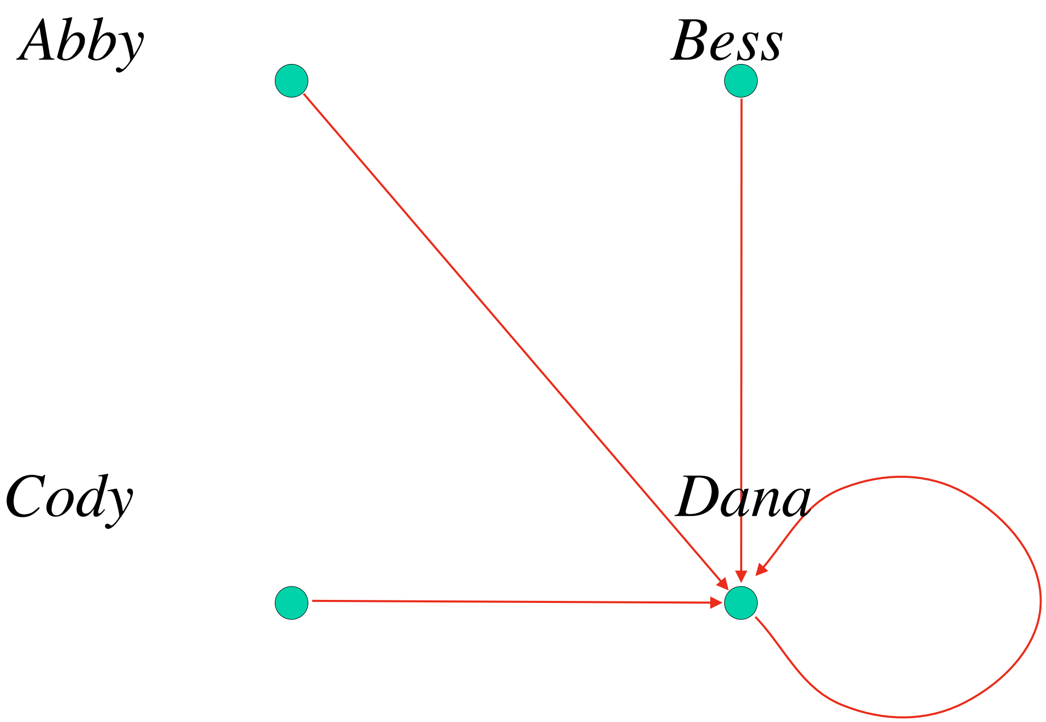
Everybody Likes Somebody



Everybody Likes Somebody

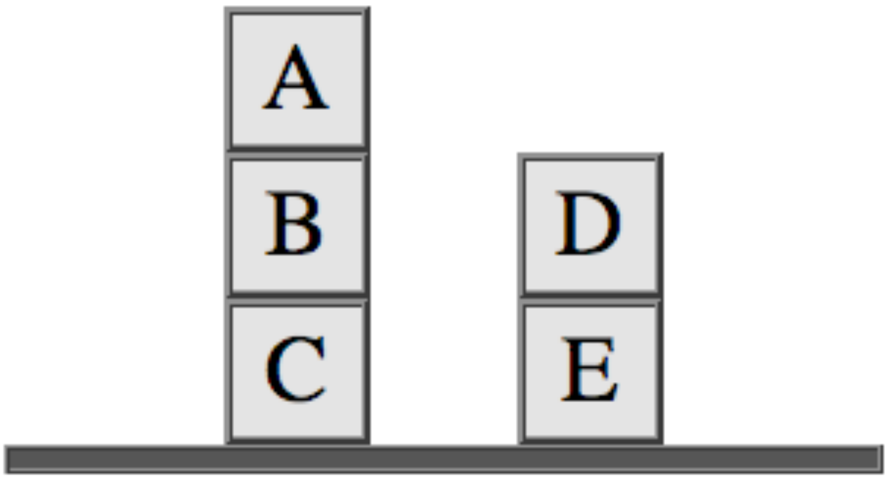


There is Somebody Whom Everyone Likes



Example - Blocks World

Blocks World



Signature

Object Constants: *a*, *b*, *c*, *d*, *e*

Unary Relation Constants:
clear - blocks with no blocks on top.
table - blocks on the table.

Binary Relation Constants:
on - pairs of blocks in which first is on the second.
above - pairs in which first block is above the second.

Ternary Relation Constant:
stack - triples of blocks arranged in a stack.

Data

$$\neg on(a,a) \quad \neg on(b,a) \quad \neg on(c,a)$$
$$on(a,b) \quad \neg on(b,b) \quad \neg on(c,b)$$
$$\neg on(a,c) \quad on(b,c) \quad \neg on(c,c)$$
$$\neg on(a,d) \quad \neg on(b,d) \quad \neg on(c,d)$$
$$\neg on(a,e) \quad \neg on(b,e) \quad \neg on(c,e)$$

$$\neg on(d,a) \quad \neg on(e,a)$$
$$\neg on(d,b) \quad \neg on(e,b)$$
$$\neg on(d,c) \quad \neg on(e,c)$$
$$\neg on(d,d) \quad \neg on(e,d)$$
$$on(d,e) \quad \neg on(e,e)$$

Definitions

Definition of *clear*:

$$\forall y.(clear(y) \Leftrightarrow \neg \exists x.on(x,y))$$

Definition of *table*:

$$\forall x.(table(x) \Leftrightarrow \neg \exists y.on(x,y))$$

Definitions

Definition of *stack*:

$$\forall x.\forall y.\forall z.(stack(x,y,z) \Leftrightarrow on(x,y) \wedge on(y,z))$$

Definition of *above*:

$$\forall x.\forall z.(above(x,z) \Leftrightarrow on(x,z) \vee \exists y.(on(x,y) \wedge above(y,z)))$$

$$\forall x.\neg above(x,x)$$

Example - Modular Arithmetic

Modular Arithmetic

In Modular Arithmetic of modulus 4 there are just 4 numbers (0,1, 2, 3).

0+0=0	1+0=1	2+0=2	3+0=3
0+1=1	1+1=2	2+1=3	3+1=0
0+2=2	1+2=3	2+2=0	3+2=1
0+3=3	1+3=0	2+3=1	3+3=2

Signature

Object Constants: 0, 1, 2, 3

Binary Relation Constants:

same - the first and second arguments are identical

next - the second argument is number after the first

Ternary Relation Constant:

plus - the third argument is the sum of the first two

plus(1,2,3)

Same

Ground Relational Data:

$same(0,0)$	$\neg same(1,0)$	$\neg same(2,0)$	$\neg same(3,0)$
$\neg same(0,1)$	$same(1,1)$	$\neg same(2,1)$	$\neg same(3,1)$
$\neg same(0,2)$	$\neg same(1,2)$	$same(2,2)$	$\neg same(3,2)$
$\neg same(0,3)$	$\neg same(1,3)$	$\neg same(2,3)$	$same(3,3)$

Next

Ground Relational Data:

$\neg next(0,0)$	$\neg next(1,0)$	$\neg next(2,0)$	$next(3,0)$
$next(0,1)$	$\neg next(1,1)$	$\neg next(2,1)$	$\neg next(3,1)$
$\neg next(0,2)$	$next(1,2)$	$\neg next(2,2)$	$\neg next(3,2)$
$\neg next(0,3)$	$\neg next(1,3)$	$next(2,3)$	$\neg next(3,3)$

Next

Ground Relational Data:

$$\begin{array}{c} next(0,1) \\ next(1,2) \\ next(2,3) \\ next(3,0) \end{array}$$

Deal with negative literals by saying that all other cases are false. How do we do this?

Functionality Axiom

For every x, there is just one y such that $next(x,y)$:

$$\forall x.\forall y.\forall z.(next(x,y) \wedge next(x,z) \Rightarrow same(y,z))$$

Logically equivalent formulation:

$$\forall x.\forall y.\forall z.(next(x,y) \wedge \neg same(y,z) \Rightarrow \neg next(x,z))$$

Addition

Ground Relational Data:

$plus(0,0,0)$	$plus(1,0,1)$	$plus(2,0,2)$	$plus(3,0,3)$
$plus(0,1,1)$	$plus(1,1,2)$	$plus(2,1,3)$	$plus(3,1,0)$
$plus(0,2,2)$	$plus(1,2,3)$	$plus(2,2,0)$	$plus(3,2,1)$
$plus(0,3,3)$	$plus(1,3,0)$	$plus(2,3,1)$	$plus(3,3,2)$

Functionality Axiom:

$$\forall x.\forall y.\ \forall z.\forall w.(plus(x,y,z) \wedge plus(x,y,w) \Rightarrow same(z,w))$$

Alternative Definition of Addition

Identity:

$$\forall y.plus(0,y,y)$$

Successor:

$$\forall x.\forall y.\forall z.(plus(x,y,z) \wedge next(x,x2) \wedge next(z,z2) \\ \Rightarrow plus(x2,y,z2))$$

Functionality:

$$\forall x.\forall y.\forall z.\forall w.(plus(x,y,z) \wedge plus(x,y,w) \Rightarrow same(z,w))$$

Properties of Sentences

Properties of Sentences

Valid	A sentence is *valid* if and only if *every* interpretation satisfies it.
Contingent	A sentence is *contingent* if and only if *some* interpretation satisfies it and *some* interpretation falsifies it.
Unsatisfiable	A sentence is *unsatisfiable* if and only if *no* interpretation satisfies it.

Properties of Sentences

Valid

Contingent

Unsatisfiable

A sentences is *satisfiable* if and only if it is either valid or contingent.

A sentences is *falsifiable* if and only if it is contingent or unsatisfiable.

Logical Validities

Law of the Excluded Middle:

$$p(a) \vee \neg p(a)$$

Double Negation:

$$p(a) \Leftrightarrow \neg\neg p(a)$$

deMorgan's Laws:

$$\neg(p(a) \wedge q(a,b)) \Leftrightarrow (\neg p(a) \vee \neg q(a,b))$$

$$\neg(p(a) \vee q(a,b)) \Leftrightarrow (\neg p(a) \wedge \neg q(a,b))$$

Quantificational Validities

Common Quantifier Reversal:

$$\forall x.\forall y.q(x,y) \Leftrightarrow \forall y.\forall x.q(x,y)$$

$$\exists x.\exists y.q(x,y) \Leftrightarrow \exists y.\exists x.q(x,y)$$

Existential Distribution:

$$\exists y.\forall x.q(x,y) \Rightarrow \forall x.\exists y.q(x,y)$$

Negation Distribution:

$$\neg\forall x.p(x) \Leftrightarrow \exists x.\neg p(x)$$

$$\neg\exists x.p(x) \Leftrightarrow \forall x.\neg p(x)$$

Logical Entailment

Logical Entailment

A set of premises Δ *logically entails* a conclusion φ (written as $\Delta \models \varphi$) if and only if every interpretation that satisfies the premises also satisfies the conclusion.

$$\{p(a)\} \models (p(a) \vee p(b))$$

$$\{p(a)\} \not\models (p(a) \wedge p(b))$$

$$\{p(a), p(b)\} \models (p(a) \wedge p(b))$$

Examples with Quantification

$$\exists y.\forall x.q(x,y) \models \forall x.\exists y.q(x,y)$$

$$\forall x.\forall y.q(x,y) \models \forall y.\forall x.q(x,y)$$

$$\forall x.\forall y.q(x,y) \models \forall x.\forall y.q(y,x)$$

Example with Free Variables

$$q(x,y) \models q(y,x)$$

$$\forall x.\forall y.q(x,y) \models \forall y.\forall x.q(y,x)$$

Relational Logic and Propositional Logic

Mapping

There is a simple procedure for mapping RL sentences to equivalent PL sentences.

(1) Convert to Prenex form.

(2) Compute the grounding.

(3) Rewrite from FHL to PL.

Prenex Form

A sentence is in *prenex form* if and only if (1) it is closed and (2) all of the quantifiers are outside of all logical operators.

Sentence in Prenex Form:

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

Sentences *not* in Prenex Form:

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

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

Conversion to Prenex Form

Rename duplicate variables.

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

Distribute logical operators over quantifiers.

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

Quantify any free variables.

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

Grounding

Let $\Delta_0 = \Delta$ and $\Gamma_0 = \{\}$. On each step, process φ in Δ until Δ is empty.

If φ is ground, we remove from Δ_i and add to Γ_i

$$\Delta_{i+1} = \Delta_i - \{\varphi\}$$
$$\Gamma_{i+1} = \Gamma_i \cup \{\varphi\}$$

If $\forall\upsilon.\varphi[\upsilon]$, replace on Δ_i with all instances

$$\Delta_{i+1} = \Delta_i - \{\forall\upsilon.\varphi[\upsilon]\} \cup \{\varphi[\tau_i] \mid \tau_i \text{ a constant}\}$$
$$\Gamma_{i+1} = \Gamma_i$$

If $\exists\upsilon.\varphi[\upsilon]$, replace on Δ_i with disjunction of instances

$$\Delta_{i+1} = \Delta_i - \{\exists\upsilon.\varphi[\upsilon]\} \cup \{\varphi[\tau_1] \vee \ldots \vee \varphi[\tau_n]\}$$
$$\Gamma_{i+1} = \Gamma_i$$

Example 1

$\Delta_0=\{p(a), \forall x.(p(x) \Rightarrow q(x))\ , \exists x.q(x)\}$
$\Gamma_0=\{\}$

$\Delta_1=\{\forall x.p(x) \Rightarrow q(x))\ , \exists x.q(x)\}$
$\Gamma_1=\{p(a)\}$

$\Delta_2=\{p(a) \Rightarrow q(a), p(b) \Rightarrow q(b), \exists x.q(x)\}$
$\Gamma_2=\{p(a)\}$

$\Delta_3=\{p(b) \Rightarrow q(b), \exists x.q(x)\}$
$\Gamma_3=\{p(a), p(a) \Rightarrow q(a)\}$

$\Delta_4=\{\exists x.q(x)\}$
$\Gamma_4=\{p(a), p(a) \Rightarrow q(a), p(b) \Rightarrow q(b)\}$

Example 2

Δ_4={$\exists x . q(x)$}

Γ_4={$p(a), p(a) \Rightarrow q(a), p(b) \Rightarrow q(b)$}

Δ_5={$q(a) \vee q(b)$}

Γ_5={$p(a), p(a) \Rightarrow q(a), p(b) \Rightarrow q(b)$}

Δ_6={}

Γ_6={$p(a), p(a) \Rightarrow q(a), p(b) \Rightarrow q(b), q(a) \vee q(b)$}

Renaming RL to PL

Select a proposition for each ground relational sentence and rewrite the grounding from FHL to PL.

FHL Grounding:

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

Corresponding PL:

$$p(a) \Leftrightarrow pa \qquad q(a) \Leftrightarrow qa$$
$$p(b) \Leftrightarrow pb \qquad q(b) \Leftrightarrow qb$$

Corresponding PL:

$$\{pa, pa \Rightarrow qa, pb \Rightarrow qb, qa \vee qb\}$$

Decidability

Unsatisfiability and logical entailment for Propositional Logic (PL) is decidable.

Given our mapping, we also know that unsatisfiability logical entailment for Relational Logic (RL) is also decidable.

Compactness

A logic is *compact* if and only if every unsatisfiable set of sentences (including infinite sets) has a finite subset that is unsatisfiable.

Propositional Logic is compact.

Given our mapping, we know that RL must also be compact.