Introduction to Logic
Relational Analysis

Michael Genesereth
Computer Science Department
Stanford University

Truth Assignment

A *truth assignment / interpretation* is an association between ground relational sentences and the truth values *true* and *false* or, equivalently 1 and 0.

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

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 $\varphi^i = \psi^i$

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.

Properties of Sentences

Valid
Contingent
Unsatisfiable

A sentence is *valid* if and only if *every* interpretation satisfies it.

A sentence is *contingent* if and only if *some* interpretation satisfies it and *some* interpretation falsifies it.

A sentence is *unsatisfiable* if and only if *no* interpretation satisfies it.

Properties of Sentences

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.

Relationships

A sentence ϕ is *logically equivalent* to a sentence ψ if and only if ϕ and ψ have the *same value* for every truth assignment.

A sentence ϕ is *consistent with* a sentence ψ if and only if there is a truth assignment that satisfies both ϕ and ψ.

A sentence ϕ *logically entails* a sentence ψ (written $\phi \models \psi$) if and only if every truth assignment that satisfies ϕ also satisfies ψ.

Sets of Sentences

A *set* of sentences Γ *logically entails* a set of sentences Δ (written as $\Gamma \vDash \Delta$) if and only if every truth assignment that satisfies *all* of the sentences in Γ satisfies *all* of the sentences in Δ. (Think of sets as conjunctions.)

Ditto for equivalence and consistency.

Metatheorems

Metatheorems

Propositional Metatheorems:

Monotonicity Theorem (*More premises mean more conclusions*.)

Ramification Theorem (*If many conclusions, then few conclusions*.)

Equivalence Theorem (φ *equivalent to* ψ *iff* ($\varphi \Leftrightarrow \psi$) *is valid*.)

Substitution Theorem (*If* ($\varphi \Leftrightarrow \psi$) *is valid*, $\chi_{\varphi \leftarrow \psi}$ *equivalent to* χ.)

Deduction Theorem ($\varphi \vDash \psi$ *iff* ($\varphi \Rightarrow \psi$) *is valid*.)

Unsatisfiability Theorem ($\Delta \vDash \varphi$ *iff* $\Delta \cup \{\neg\varphi\}$ is unsatisfiable.)

Consistency Theorem (φ is *consistent* ψ *iff* ($\varphi \wedge \psi$) is *satisfiable*.)

These theorems also hold in Relational Logic provided that all sentences are ***closed*** *(i.e. they have no* ***free*** *variables).*

Distributing Quantifiers over Quantifiers

Common Quantifier Reversal:

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

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

Distributing Existentials over Universals:

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

Distributing Universals over Existentials *not cool*:

No! No!! No!!! $\forall x.\exists y.q(x,y) \vDash \exists y.\forall x.q(x,y)$ *No! No!! No!!!*

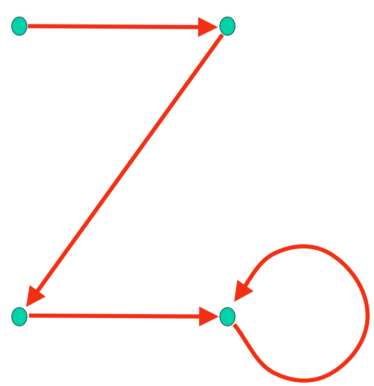
Distributing Quantifiers over Quantifiers

Distributing Existentials over Universals:

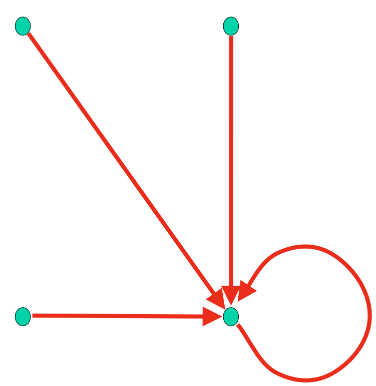
$$\exists y.\forall x.q(x,y) \models \forall x.\exists y.q(x,y)$$

Distributing Universals over Existentials *not cool*:

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



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



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

Distributing Quantifiers over Operators

Distributing Quantifiers over Negations:

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

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

Distributing Quantifiers over Conjunctions:

$$\forall x.(p(x) \wedge q(x)) \models \forall x.p(x) \wedge \forall x.q(x))$$

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

Distributing Quantifiers over Disjunctions:

$$\exists x.(p(x) \vee q(x)) \models \exists x.p(x) \vee \exists x.q(x))$$

No! No!! No!!! $\forall x.(p(x) \vee q(x)) \models \forall x.p(x) \vee \forall x.q(x))$ *No! No!! No!!!*

$$\forall x.(p(x) \vee q(b)) \models \forall x.p(x) \vee q(b)$$

Distributing Operators over Quantifiers

No! No!! No!!! $\forall x.(p(x) \vee q(x)) \vDash \forall x.p(x) \vee \forall x.q(x)$ *No! No!! No!!!*

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

In this case:

$\forall x.(p(x) \vee q(x))$ is *true* but

$\forall x.p(x)$ is *false* and $\forall x.q(x)$ is *false*

Distributing Quantifiers over Implications

Implication Distribution:

$$\forall y.(p(a) \Rightarrow q(y)) \vDash (p(a) \Rightarrow \forall y.q(y))$$

$$\forall x.(p(x) \Rightarrow q(b)) \vDash (\exists x.p(x) \Rightarrow q(b))$$

$$\forall x.\forall y.(p(x) \Rightarrow q(y)) \vDash (\exists x.p(x) \Rightarrow \forall y.q(y))$$

Distributing Quantifiers over Implications

Implication Distribution:

$$\forall y.(p(a) \Rightarrow q(y)) \models (p(a) \Rightarrow \forall y.q(y))$$

$$\forall x.(p(x) \Rightarrow q(b)) \models (\exists x.p(x) \Rightarrow q(b))$$

$$\forall x.\forall y.(p(x) \Rightarrow q(y)) \models (\exists x.p(x) \Rightarrow \forall y.q(y))$$

Derivation:

$$\begin{aligned} \forall x.(p(x) \Rightarrow q(b)) &\models \forall x.(\neg p(x) \vee q(b)) \\ &\models (\forall x.\neg p(x) \vee q(b)) \\ &\models (\neg \exists x.p(x) \vee q(b)) \\ &\models (\exists x.p(x) \Rightarrow q(b)) \end{aligned}$$

Common, very useful distribution.

Distributing Operators over Quantifiers

Distributing Negations over Quantifiers:

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

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

Distributing Conjunctions over Quantifiers:

$$\forall x.p(x) \wedge \forall x.q(x)) \vDash \forall x.(p(x) \wedge q(x))$$

No! No!! No!!! $\exists x.p(x) \wedge \exists x.q(x) \vDash \exists x.(p(x) \wedge q(x))$ *No! No!! No!!!*

$$\exists x.p(x) \wedge q(b) \vDash \exists x.(p(x) \wedge q(b))$$

Distributing Disjunctions over Quantifiers:

$$\exists x.p(x) \vee \exists x.q(x)) \vDash \exists x.(p(x) \vee q(x))$$

$$\forall x.p(x) \vee \forall x.q(x)) \vDash \forall x.(p(x) \vee q(x))$$

Distributing Operators over Quantifiers

$$\forall x.p(x) \wedge \forall x.q(x)) \models \forall x.(p(x) \wedge q(x))$$

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

No! No!! No!!! $\exists x.p(x) \wedge \exists x.q(x) \models \exists x.(p(x) \wedge q(x))$ *No! No!! No!!!*

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

Truth Table Method

Determining Logical Entailment

$$\{m \Rightarrow p \vee q, p \Rightarrow q\} \models m \Rightarrow q?$$

m	p	q	$m \Rightarrow p \vee q$	$p \Rightarrow q$	$m \Rightarrow q$
1	1	1	1	1	1
1	1	0	1	0	0
1	0	1	1	1	1
1	0	0	0	1	0
0	1	1	1	1	1
0	1	0	1	0	1
0	0	1	1	1	1
0	0	0	1	1	1

Determining Logical Entailment

Question:

$$\{p(a) \vee p(b), \forall x.(p(x) \Rightarrow q(x))\} \models \exists x.q(x)?$$

Determining Logical Entailment

Question:

$$\{p(a) \vee p(b), \forall x.(p(x) \Rightarrow q(x))\} \models \exists x.q(x)?$$

Object Constants: a, b
Unary Relation Constants: p, q
Herbrand Base: $\{p(a), p(b), q(a), q(b)\}$

Determining Logical Entailment

$$\{p(a) \vee p(b), \forall x.(p(x) \Rightarrow q(x))\} \models \exists x.q(x)?$$

$p(a)$	$p(b)$	$q(a)$	$q(b)$	$p(a) \vee p(b)$	$\forall x.(p(x) \Rightarrow q(x))$	$\exists x.q(x)$
1	1	1	1	1	1	1
1	1	1	0	1	0	1
1	1	0	1	1	0	1
1	1	0	0	1	0	0
1	0	1	1	1	1	1
1	0	1	0	1	1	1
1	0	0	1	1	0	1
1	0	0	0	1	0	0
0	1	1	1	1	1	1
0	1	1	0	1	0	1
0	1	0	1	1	1	1
0	1	0	0	1	0	0
0	0	1	1	0	1	1
0	0	1	0	0	1	1
0	0	0	1	0	1	1
0	0	0	0	0	1	0

Determining Logical Entailment

$$\{p(a) \vee p(b), \forall x.(p(x) \Rightarrow q(x))\} \models \exists x.q(x)?$$

$p(a)$	$p(b)$	$q(a)$	$q(b)$	$p(a) \vee p(b)$	$\forall x.(p(x) \Rightarrow q(x))$	$\exists x.q(x)$
1	1	1	1	1	1	1
1	1	1	0	1	0	1
1	1	0	1	1	0	1
1	1	0	0	1	0	0
1	0	1	1	1	1	1
1	0	1	0	1	1	1
1	0	0	1	1	0	1
1	0	0	0	1	0	0
0	1	1	1	1	1	1
0	1	1	0	1	0	1
0	1	0	1	1	1	1
0	1	0	0	1	0	0
0	0	1	1	0	1	1
0	0	1	0	0	1	1
0	0	0	1	0	1	1
0	0	0	0	0	1	0

Analysis

Object constants: n
Binary relation constants: k
Factoids in Herbrand Base: $k*n^2$
Interpretations: 2^{k*n^2}

Object constants: 4
Binary relation constants: 4
Factoids in Herbrand Base: 64
Interpretations: 2^{64} = 18,446,744,073,709,551,616

Methods

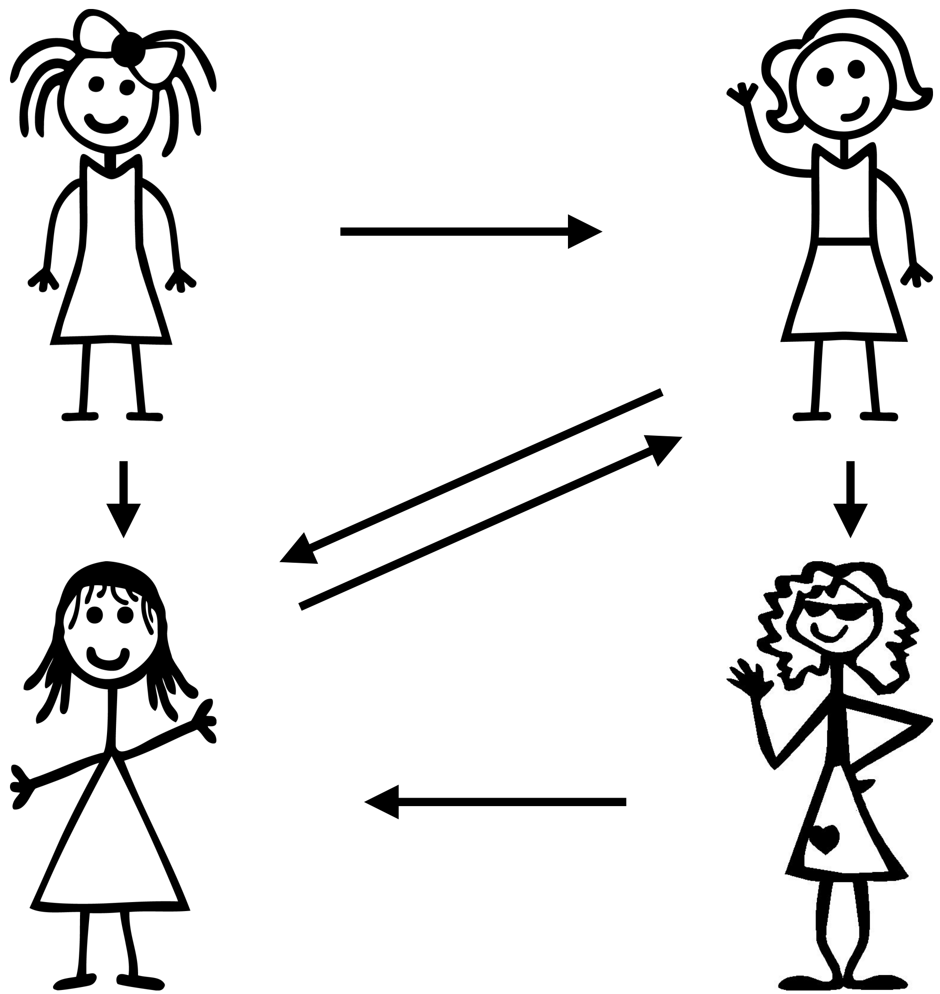
Truth Tables / Models
- Guaranteed
- Often Impractical

Proofs
- Guaranteed
- Usually smaller than models — *Next time*
- Sometimes not as intuitive as models

Constraint Satisfaction (model creation + proofs)
- Boolean Grids (aka Logic Grids)
- Non-Boolean Grids

Boolean Grids / Logic Grids

Friends



Logical Sentences

Dana likes Cody.
Abby does not like Dana.
Dana does not like Abby.
Abby likes everyone that Bess likes.
Bess likes Cody or Dana.
Abby and Dana both dislike Bess.
Cody likes everyone who likes her.
Nobody likes herself.

Logical Sentences

$likes(dana,cody)$

$\neg likes(abby,dana)$

$\neg likes(dana,abby)$

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

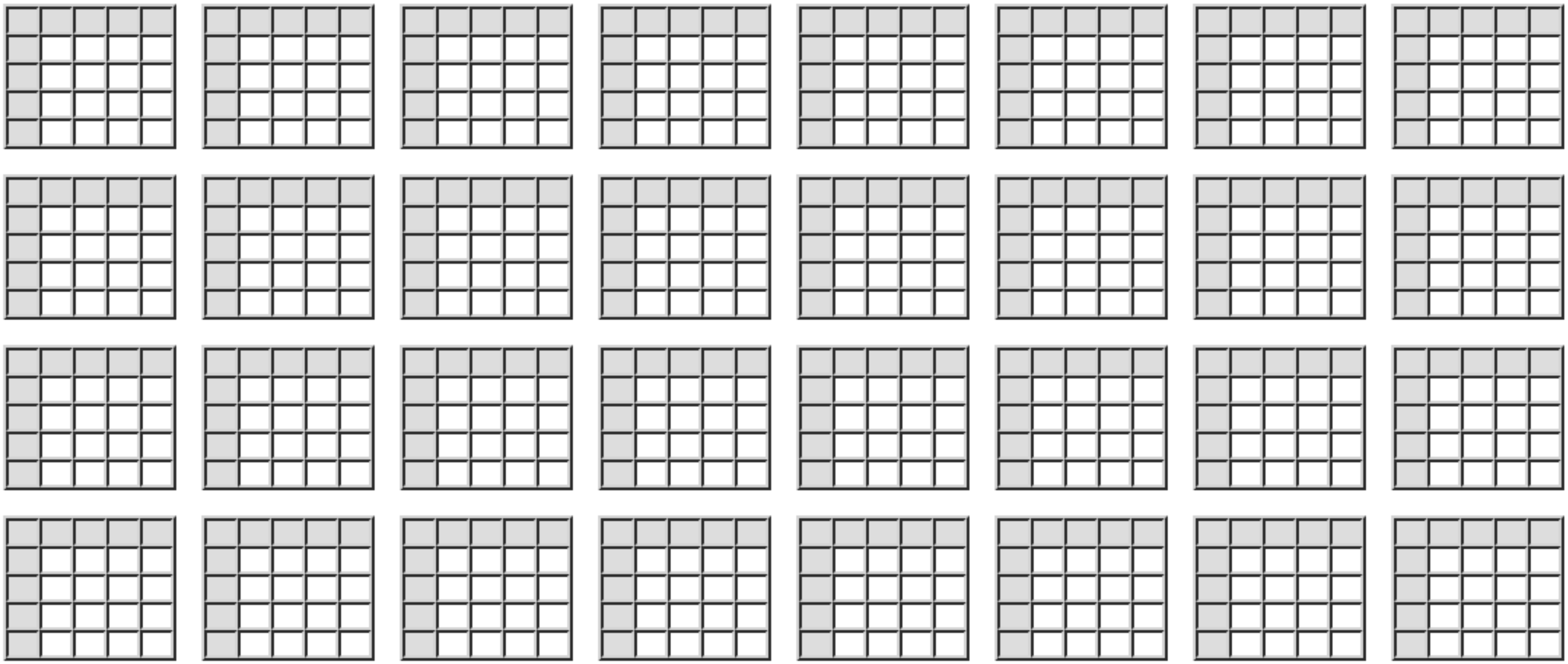
$likes(bess,cody) \vee likes(bess,dana)$

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

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

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

All Possible Truth Assignments



2^16 (65,536) truth assignments.

Only One Truth Assignment Works

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

Logic Grid

	Abby	Bess	Cody	Dana
Abby				
Bess				
Cody				
Dana				

Logic Grid

	Abby	Bess	Cody	Dana
Abby				0
Bess				
Cody				
Dana	0		1	

$likes(dana, cody)$

$\neg likes(abby, dana)$

$\neg likes(dana, abby)$

Logic Grid

	Abby	Bess	Cody	Dana
Abby				0
Bess				0
Cody				
Dana	0		1	

$likes(dana, cody)$

$\neg likes(abby, dana)$

$\neg likes(dana, abby)$

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

Logic Grid

	Abby	Bess	Cody	Dana
Abby			1	0
Bess			1	0
Cody				
Dana	0		1	

$likes(dana,cody)$

$\neg likes(abby,dana)$

$\neg likes(dana,abby)$

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

$likes(bess,cody) \vee likes(bess,dana)$

Logic Grid

	Abby	Bess	Cody	Dana
Abby		0	1	0
Bess			1	0
Cody				
Dana	0	0	1	

$likes(dana,cody)$

$\neg likes(abby,dana)$

$\neg likes(dana,abby)$

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

$likes(bess,cody) \vee likes(bess,dana)$

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

Logic Grid

	Abby	Bess	Cody	Dana
Abby		0	1	0
Bess			1	0
Cody	1	1		1
Dana	0	0	1	

$$likes(dana, cody)$$
$$\neg likes(abby, dana)$$
$$\neg likes(dana, abby)$$
$$\forall y.(likes(bess, y) \Rightarrow likes(abby, y))$$
$$likes(bess, cody) \vee likes(bess, dana)$$
$$\neg likes(abby, bess) \wedge \neg likes(dana, bess)$$
$$\forall x.(likes(x, cody) \Rightarrow likes(cody, x))$$

Logic Grid

	Abby	Bess	Cody	Dana
Abby	0	0	1	0
Bess		0	1	0
Cody	1	1	0	1
Dana	0	0	1	0

$likes(dana,cody)$

$\neg likes(abby,dana)$

$\neg likes(dana,abby)$

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

$likes(bess,cody) \vee likes(bess,dana)$

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

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

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

Logic Grid

	Abby	Bess	Cody	Dana
Abby	0	0	1	0
Bess	0	0	1	0
Cody	1	1	0	1
Dana	0	0	1	0

$$likes(dana,cody)$$

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

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

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

$$likes(bess,cody) \lor likes(bess,dana)$$

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

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

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

Course Website

http:/logica.stanford.edu

127.0.0.1

Logica

Tools for Thought

Clarke

Show Instructions

Universe:

```
abby bess cody dana
```

Logic Grid

	abby	**bess**	**cody**	**dana**
abby				
bess				
cody				
dana				

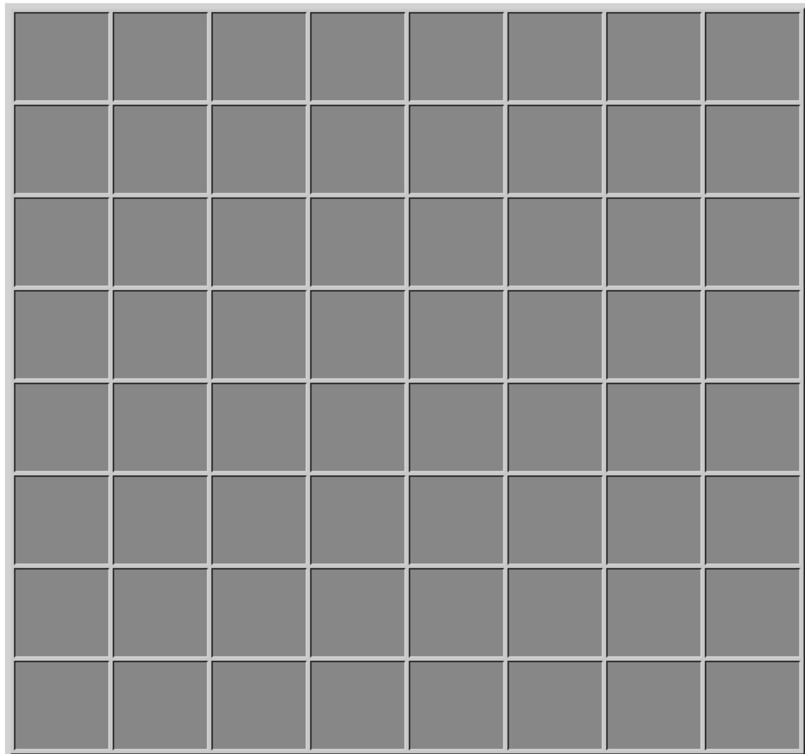
Course Website

http://intrologic.stanford.edu

127.0.0.1

Introduction to Logic

Mineplanner



```
~EY:mine(1,Y)
AX:(mine(X,1) => mine(1,X))
AY:(mine(8,Y) => mine(1,Y))
AY:(~mine(1,Y) & ~mine(8,Y) => ~mine(Y,8))
~EX:mine(X,X)
EY:mine(5,Y)
mine(5,3) | mine(5,4)
~mine(5,3)
```

Show Free Cells | Show Instructions | Reset

Non-Boolean Grids

Sudoku

	6		1		4		5	
		8	3		5	6		
2								1
8			4		7			6
		6				3		
7			9		1			4
5								2
		7	2		6	9		
	4		5		8		7	

Sukoshi

	4		1
2			
			3
		4	

Axiomatization

$cell(1,2,4)$
$cell(1,4,1)$
$cell(2,1,2)$
$cell(3,4,3)$
$cell(4,3,4)$

$same(1,1)$	$\neg same(2,1)$	$\neg same(3,1)$	$\neg same(4,1)$
$\neg same(1,2)$	$same(2,2)$	$\neg same(3,2)$	$\neg same(4,2)$
$\neg same(1,3)$	$\neg same(2,3)$	$same(3,3)$	$\neg same(4,3)$
$\neg same(1,4)$	$\neg same(2,4)$	$\neg same(3,4)$	$same(4,4)$

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

Analysis

Object constants: 4

Ternary relation constants: 1

Factoids in Herbrand Base: 64

Truth Assignments: 2^{64} = 18,446,744,073,709,551,616

Analysis

Object constants: 4

Ternary relation constants: 1

Factoids in Herbrand Base: 64

Truth Assignments: 2^{64} = 18,446,744,073,709,551,616

Non-Boolean Grids: 4^{16} = 4,294,967,296

Sukoshi

	4		1
2			
			3
		4	

Sukoshi

	4		1
2			4
			3
		4	

Sukoshi

	4		1
2			4
4			3
		4	

Sukoshi

	4		1
2			4
4			3
		4	2

Sukoshi

	4		1
2			4
4			3
1		4	2

Sukoshi

3	4		1
2			4
4			3
1		4	2

Sukoshi

3	4	2	1
2			4
4			3
1		4	2

Sukoshi

3	4	2	1
2			4
4			3
1	3	4	2

Sukoshi

3	4	2	1
2			4
4	2		3
1	3	4	2

Sukoshi

3	4	2	1
2			4
4	2	1	3
1	3	4	2

Sukoshi

3	4	2	1
2		3	4
4	2	1	3
1	3	4	2

Sukoshi

3	4	2	1
2	1	3	4
4	2	1	3
1	3	4	2

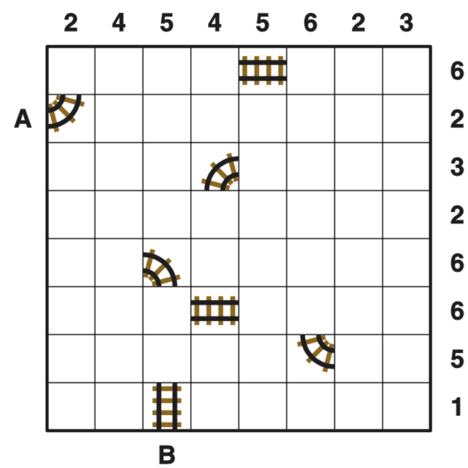
Course Website

http://intrologic.stanford.edu

Introduction to Logic

Train Puzzle

A train is departing from village A to village B but there are no rails. Can you find out where the rails must go? The numbers at the top and right border indicate how many rails must go in the corresponding row or column. There are only straight rails and curves. Draw the rails into the cells so that the train can go from village A to village B!



Show Answer Reset Answer

For more train puzzles, visit Puzzle Phil or check out The Times Train Puzzle books.

Zebra Puzzle

There is a row of five houses.
The Englishman lives in the red house.
The Spaniard owns the dog.
Coffee is drunk in the green house.
The Ukrainian drinks tea.
The green house is immediately to the right of the ivory house.
The Old Gold smoker owns snails.
Kools are smoked in the yellow house.
Milk is drunk in the middle house.
The Norwegian lives in the first house.
The man who smokes Chesterfields lives in the house next to the man with the fox.
Kools are smoked in the house next to the house where the horse is kept.
The Lucky Strike smoker drinks orange juice.
The Japanese smokes Parliaments.
The Norwegian lives next to the blue house.

Zebra Puzzle

There is a row of five houses.
The Englishman lives in the red house.
The Spaniard owns the dog.
Coffee is drunk in the green house.
The Ukrainian drinks tea.
The green house is immediately to the right of the ivory house.
The Old Gold smoker owns snails.
Kools are smoked in the yellow house.
Milk is drunk in the middle house.
The Norwegian lives in the first house.
The man who smokes Chesterfields lives in the house next to the man with the fox.
Kools are smoked in the house next to the house where the horse is kept.
The Lucky Strike smoker drinks orange juice.
The Japanese smokes Parliaments.
The Norwegian lives next to the blue house.

Who owns the Zebra?

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 RL in 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

Instantiate all quantified sentences.

(1) Leave all ground sentences as is.

(2) Replace every universally quantified sentence by all instances of its scope.

(3) Replace every existentially quantified sentence by a disjunction of instances of its scope.

Grounding

Object constants: a, b
Unary Relations constants: p, q

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

$p(a)$	$p(a)$
$\forall x.(p(x) \Rightarrow q(x))$	$p(a) \Rightarrow q(a)$ $p(b) \Rightarrow q(b)$
$\exists x.q(x)$	$q(a) \vee q(b)$

Renaming RL to PL

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

RL 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 and logical entailment for Relational Logic (RL) is also decidable.