Introduction to Logic
Relational Logic

Michael Genesereth
Computer Science Department
Stanford University

Propositional Logic

Premises:

If Abby likes Bess, then Bess likes Abby.

Abby likes Bess.

Conclusion:

Bess likes Abby.

Symmetry of Affection

Propositional Logic:

If Abby likes Bess, then Bess likes Abby.
If Abby likes Cody, then Cody likes Abby.
If Abby likes Dana, then Dana likes Abby.
...
If Bess likes Abby, then Abby likes Bess.
If Cody likes Abby, then Abby likes Cody.
If Dana likes Abby, then Abby likes Dana.

Relational Logic:

If X likes Y, then Y likes X.

Relational Logic

Natural Language Sentence:

If X likes Y, then Y likes X.

For every X and for every Y, if X likes Y, then Y likes X.

New Linguistic Features:

Variables

Quantifiers

Relational Logic Sentence:

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

Syntax

Components of Language

Words

$$a \quad b \quad c \quad p \quad q \quad r \quad x \quad y \quad z$$

Sentences

$$\forall x.(p(x,a) \land q(x,b) \Rightarrow r(a,b))$$

Words

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

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

$$a, b, c, 123, cs157, barack\_obama$$

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

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

Note that, in the online tools, we use lower case and capital letters to distinguish constants and variables.

Constants

Object constants represent objects.

joe *stanford* *canada* 2345

Relation constants represent properties or relationships.

isaperson *isacountry* *knows* *likes* *between*

Arity

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

Unary relation constant - 1 argument
e.g. *isaperson*, *isacountry*

Binary relation constant - 2 arguments
e.g. *knows*, *likes*

Ternary relation constant - 3 arguments
e.g. *between*

n-ary relation constant - *n* arguments

Vocabularies

A *vocabulary* / *signature* consists of a finite, non-empty set of object constants and a finite, non-empty 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 (1) a variable or (2) an object constant.

Terms represent objects.

Terms are analogous to pronouns and nouns in English.

Sentences

Three types of sentences in Relational Logic:

Relational sentences - analogous to the proposition constants in Propositional Logic

Logical sentences - analogous to logical sentences in Propositional Logic

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!

Relational sentences are also called *atoms* or *atomic sentences*.

Logical Sentences

Logical sentences in Relational Logic are analogous to those in Propositional Logic (except with relational sentences in place of propositional constants)

$$(\neg q(a,b))$$

$$(p(a) \wedge p(b))$$

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

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

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

Quantified Sentences

Universal sentences assert facts about all objects.

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

Existential sentences assert the existence of objects with given properties.

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

The sentence contained *within* a quantified sentence is called the *scope* of that sentence.

Nesting

Quantified sentences can be nested within other sentences.

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

The sentence contained *inside* a quantified sentence is called the *scope* of that sentence.

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 Sentences:

$$q(a,x)$$

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

Bound and Free Variables

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

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

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

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

Exercise

Object Constants: $jim, molly$
Unary Relation Constant: $person$
Binary Relation Constant: $parent$

$parent(jim, molly)$
$parent(molly, molly)$
$\neg person(jim)$
$person(jim, molly)$
$parent(molly, z)$
$\exists x.parent(molly, x)$
$\forall y.parent(molly, jim)$
$\exists z.(z(jim, molly) \vee z(molly, jim))$

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:

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

Questions:

How large is the Herbrand base for a vocabulary with n object constants and 2 unary relation constants?

How large is the Herbrand base for a vocabulary with n object constants and 1 binary relation constant?

Truth Assignment

A *truth assignment / interpretation* 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*.

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

How many truth assignments are there for a language with n object constants and 1 binary relation constant?

Sentential Truth Assignment

A *sentential truth assignment* is an association between arbitrary sentences in a Relational 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 truth assignment gives rise to a unique 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.

Example:

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

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

Example:

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

$p(a) \Rightarrow \exists y.q(a,y)$
$p(b) \Rightarrow \exists y.q(b,y)$

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

$p(a)^i = 1$

$p(b)^i = 0$

$q(a,a)^i = 1$

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

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

$q(b,b)^i = 0$

Sentence:

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

Instances:

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

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

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:

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

Example

Truth Assignment:

$p(a)^i = 1$

$p(b)^i = 0$

$q(a,a)^i = 1$

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

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

$q(b,b)^i = 0$

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:

$p(a)^i = 1$

$p(b)^i = 0$

$q(a,a)^i = 1$

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

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

$q(b,b)^i = 0$

Sentence:

$\forall x.\exists y.q(x,y)$ √

Instances:

$\exists y.q(a,y)$ √

$q(a,a)$ √

$q(a,b)$ x

$\exists y.q(b,y)$ √

$q(b,a)$ √

$q(b,b)$ x

Open Sentences

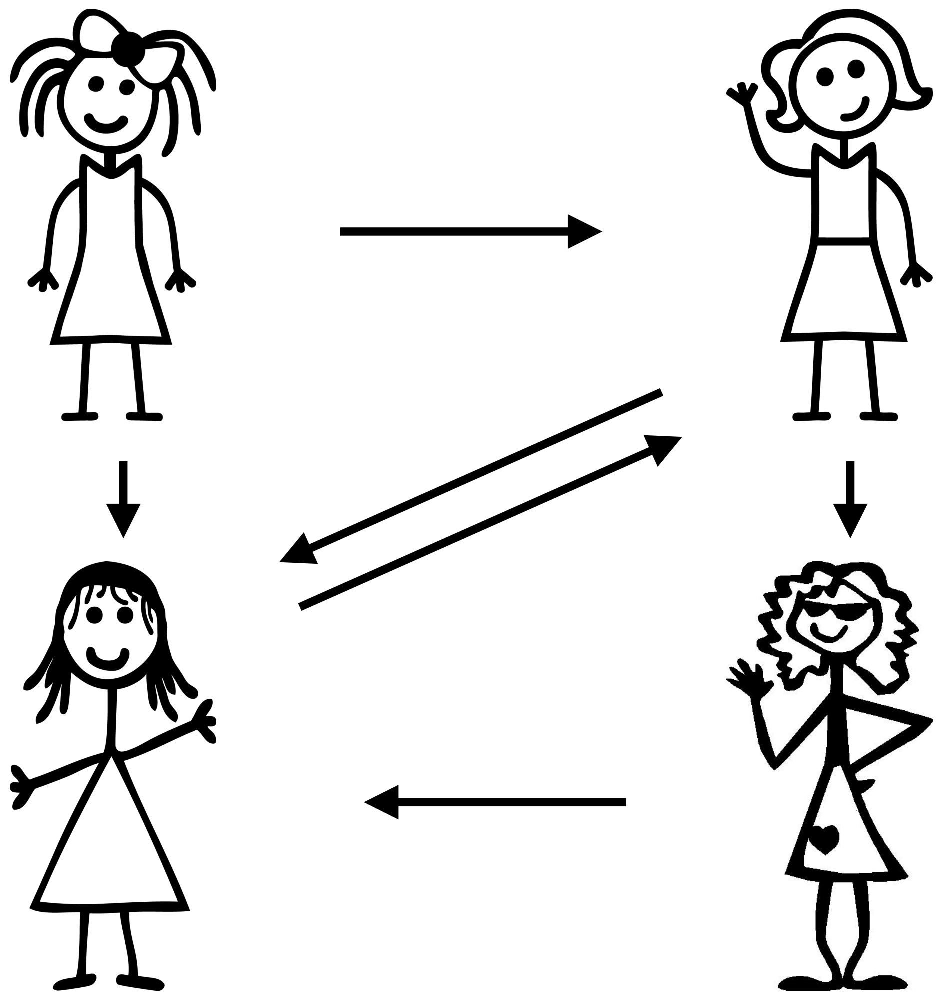
A truth assignment satisfies *a sentence with free variables* if and only if it satisfies every instance of that sentence. (In other words, we can think of all free variables as being universally quantified.)

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

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

Example - Friends

Friends



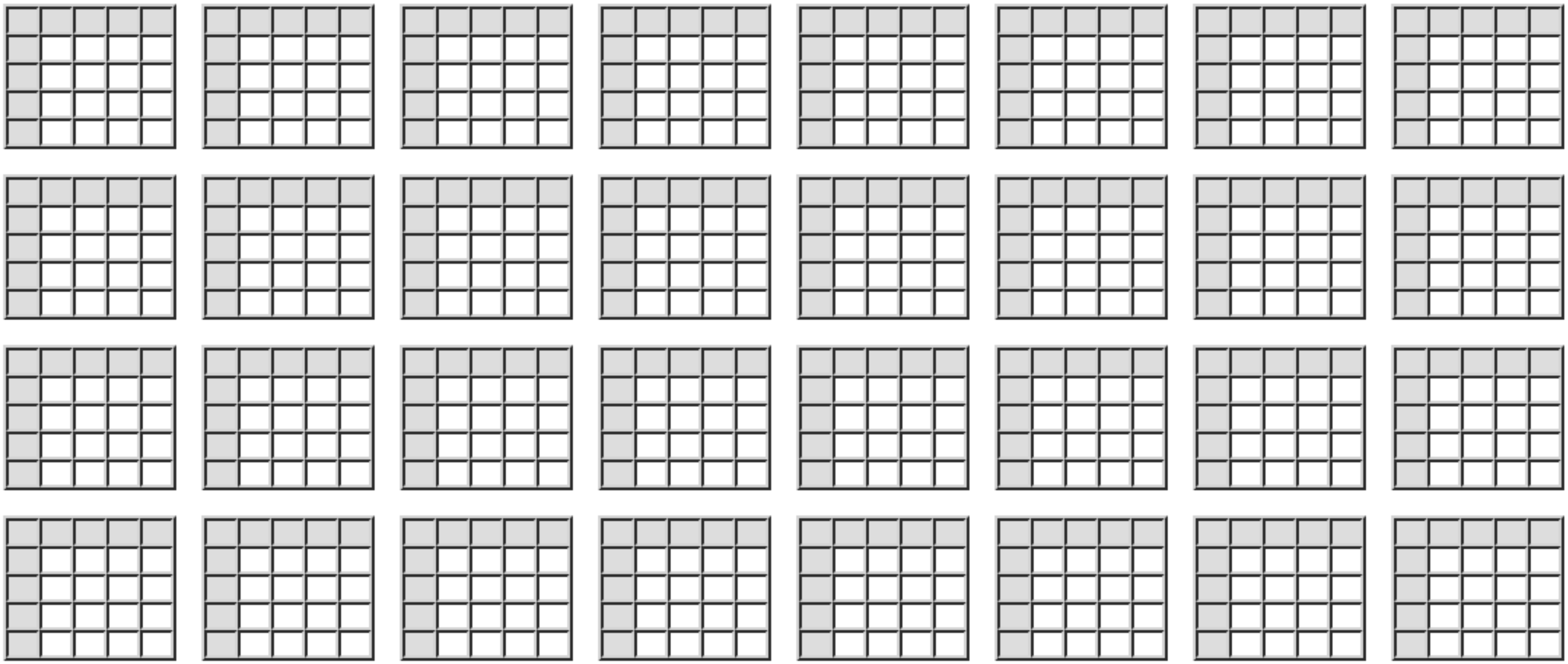
One Possible State

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

Another Possible State

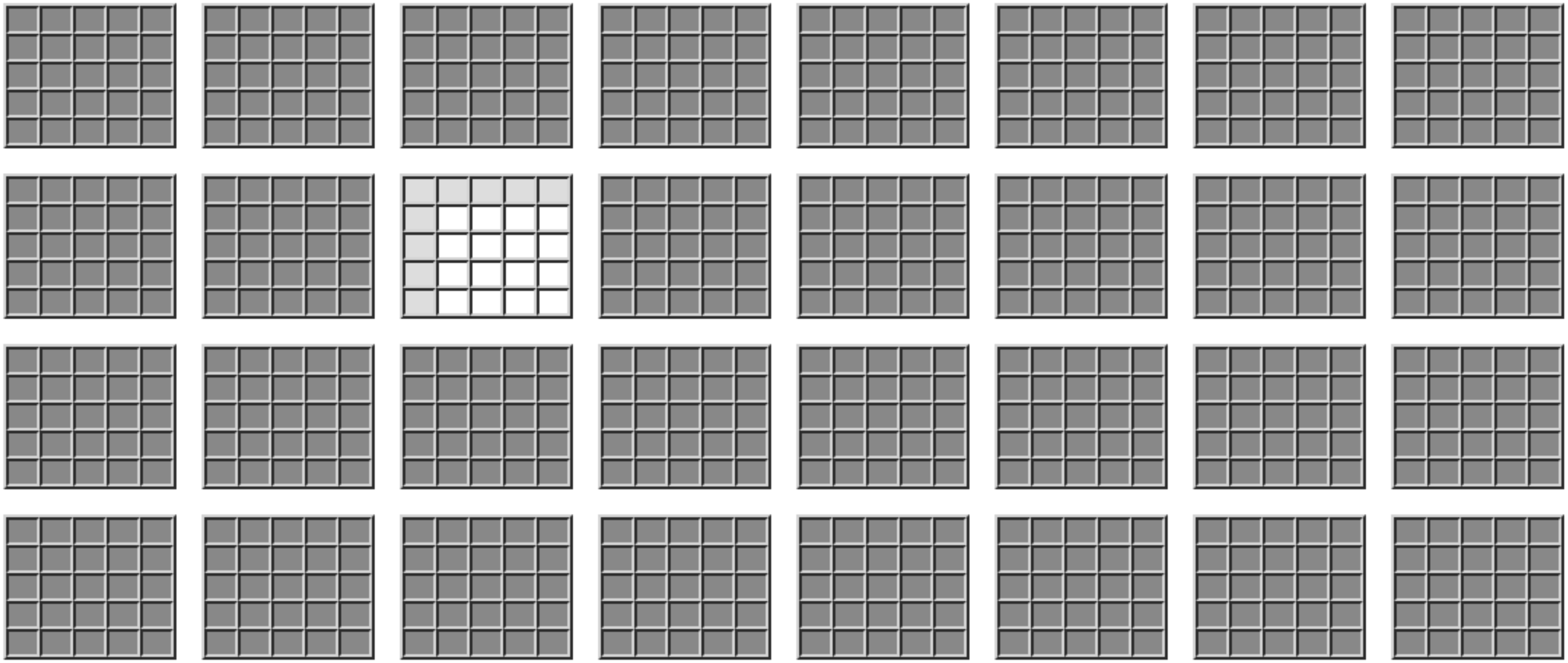
	Abby	Bess	Cody	Dana
Abby	✓		✓	
Bess		✓		✓
Cody	✓		✓	
Dana		✓		✓

Possible States



2^16 (65,536) possible worlds.

Actual State



Signature

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

Binary Relation Constant: *likes*

Herbrand base has 16 ground relational sentences.

Herbrand Base

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

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

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

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

State of Friends World

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

Ground Data

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

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

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

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

Sentences

Abby likes everyone Bess likes.

Sentences

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

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

Sentences

Abby likes everyone Bess likes.
If Bess likes someone, then Abby also likes her.

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

Cody likes everyone who likes her.

Sentences

Abby likes everyone Bess likes.
If Bess likes someone, then Abby also likes her.

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

Cody likes everyone who likes her.
If a person likes Cody, then Cody likes that person.

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

Sentences

Cody likes somebody who likes her.

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

Wrong!

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

Suppose no one likes Cody. All of these sentences are true!

Sentences

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

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

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.

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 there is someone who likes herself.

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

Sentences

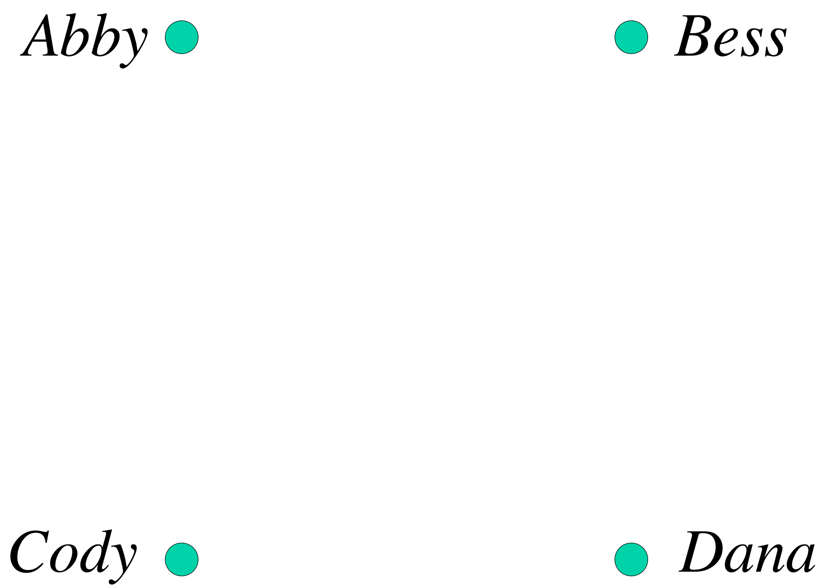
Everybody likes somebody.

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

There is somebody whom everybody likes.

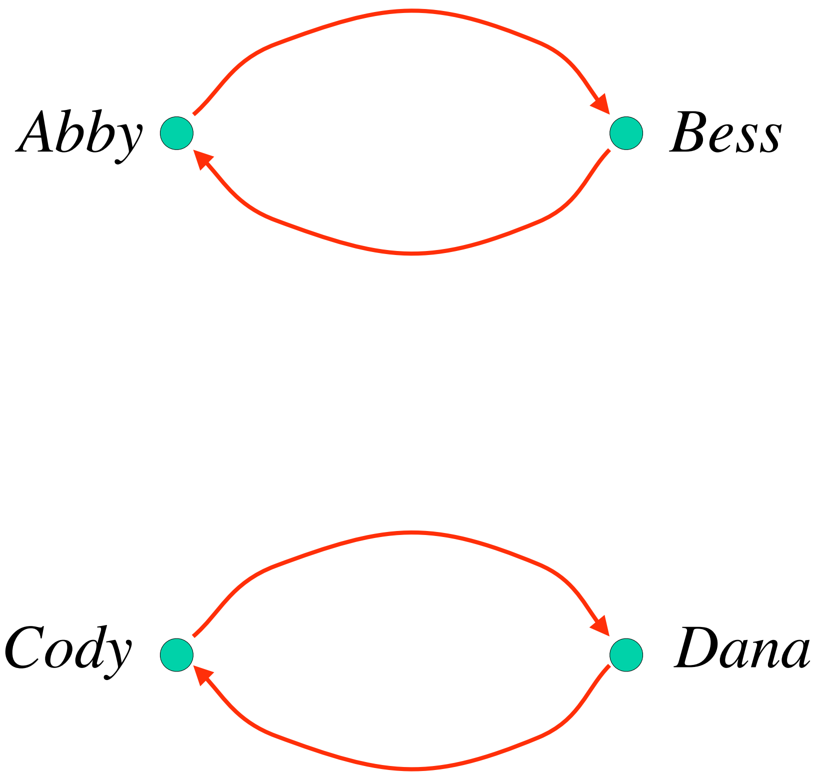
$$\exists y.\forall x.likes(x,y)$$

Example



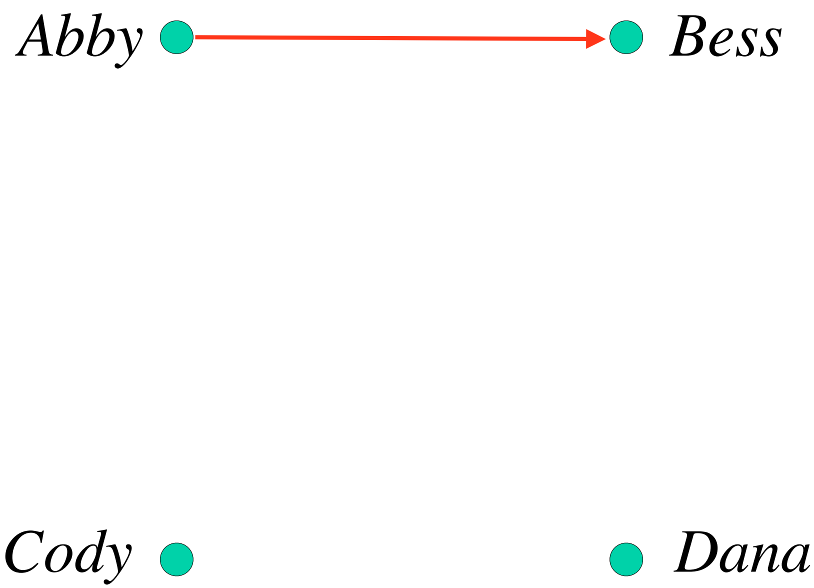
$$\forall x.\exists y.likes(x,y)$$

Everybody Likes Somebody



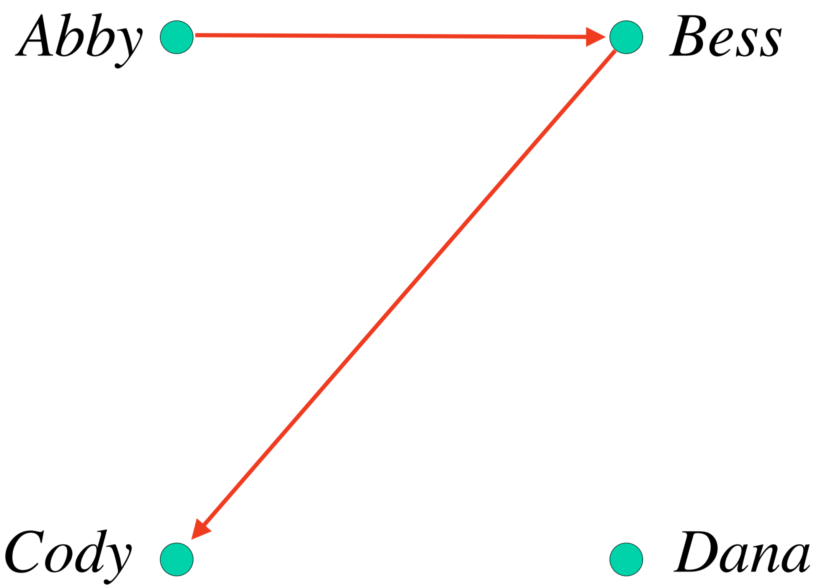
$$\forall x . \exists y . likes(x,y)$$

Everybody Likes Somebody



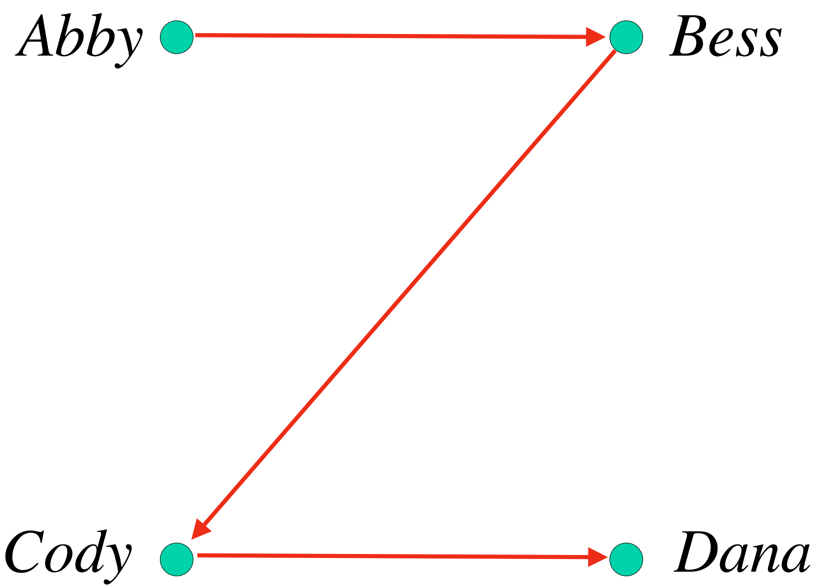
$$\forall x . \exists y . likes(x,y)$$

Everybody Likes Somebody



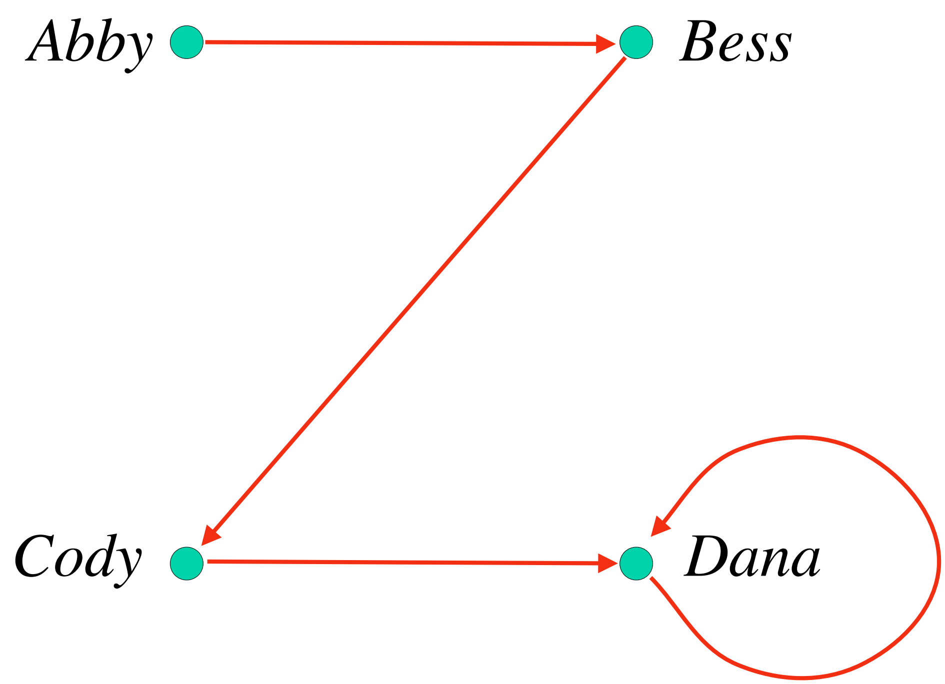
$$\forall x.\exists y.likes(x,y)$$

Everybody Likes Somebody



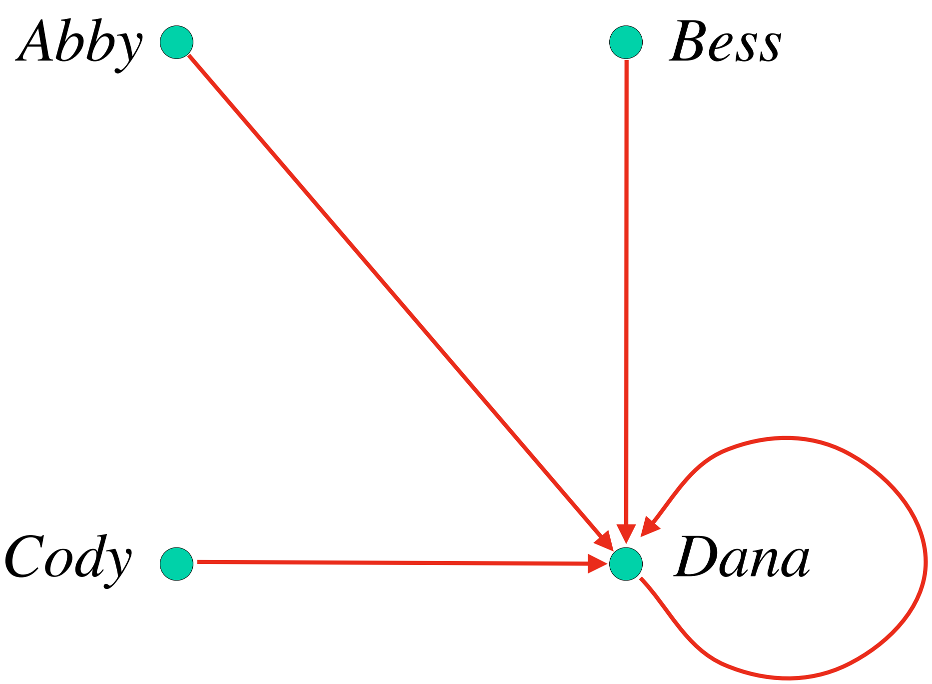
$$\forall x.\exists y.likes(x,y)$$

Everybody Likes Somebody



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

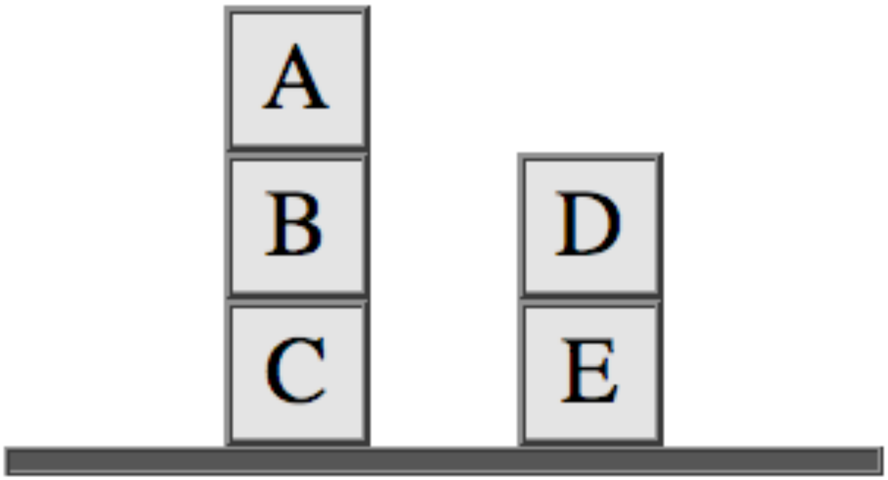
There is Somebody Whom Everyone Likes



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

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:

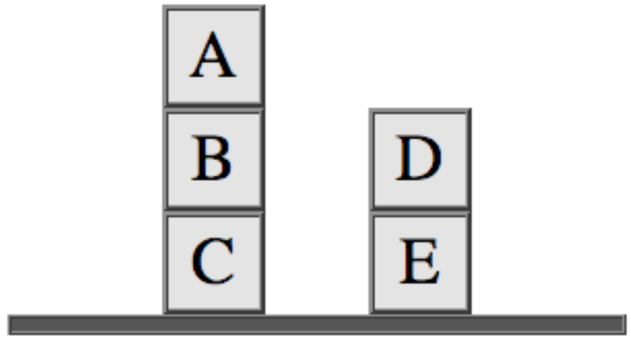
on - pairs of blocks in which first is on the second.

above - pairs in which first block is above the second.

Ternary Relation Constant:

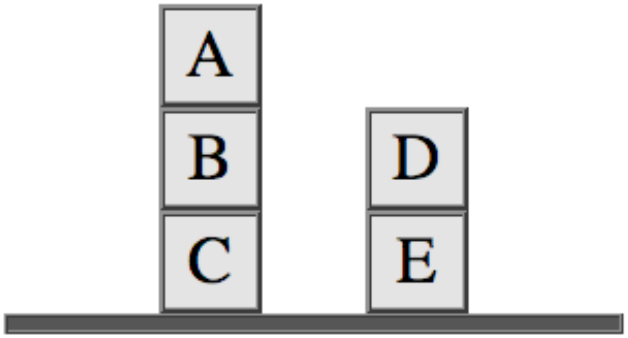
stack - triples of blocks arranged in a stack.

Ground Data



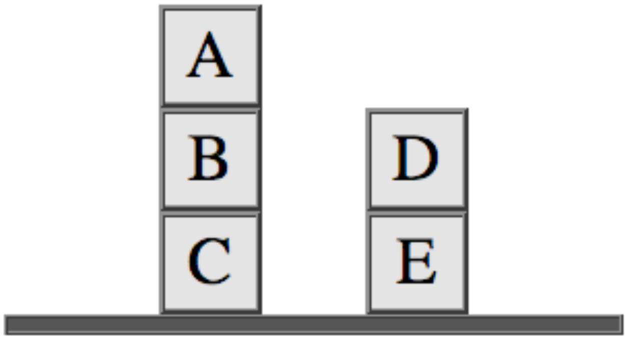
$\mathbf{clear(a)}$	$\neg table(a)$
$\neg clear(b)$	$\neg table(b)$
$\neg clear(c)$	$\mathbf{table(c)}$
$\mathbf{clear(d)}$	$\neg table(d)$
$\neg clear(e)$	$\mathbf{table(e)}$

Ground Data



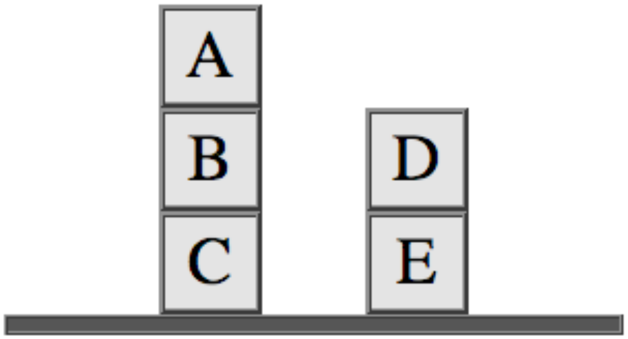
$\neg on(a,a)$	$\neg on(b,a)$	$\neg on(c,a)$	$\neg on(d,a)$	$\neg on(e,a)$
$\boldsymbol{on(a,b)}$	$\neg on(b,b)$	$\neg on(c,b)$	$\neg on(d,b)$	$\neg on(e,b)$
$\neg on(a,c)$	$\boldsymbol{on(b,c)}$	$\neg on(c,c)$	$\neg on(d,c)$	$\neg on(e,c)$
$\neg on(a,d)$	$\neg on(b,d)$	$\neg on(c,d)$	$\neg on(d,d)$	$\neg on(e,d)$
$\neg on(a,e)$	$\neg on(b,e)$	$\neg on(c,e)$	$\boldsymbol{on(d,e)}$	$\neg on(e,e)$

Ground Data



$\neg above(a,a)$	$\neg above(b,a)$	$\neg above(c,a)$	$\neg above(d,a)$	$\neg above(e,a)$
$\mathbf{above(a,b)}$	$\neg above(b,b)$	$\neg above(c,b)$	$\neg above(d,b)$	$\neg above(e,b)$
$\mathbf{above(a,c)}$	$\mathbf{above(b,c)}$	$\neg above(c,c)$	$\neg above(d,c)$	$\neg above(e,c)$
$\neg above(a,d)$	$\neg above(b,d)$	$\neg above(c,d)$	$\neg above(d,d)$	$\neg above(e,d)$
$\neg above(a,e)$	$\neg above(b,e)$	$\neg above(c,e)$	$\mathbf{above(d,e)}$	$\neg above(e,e)$

Constraints

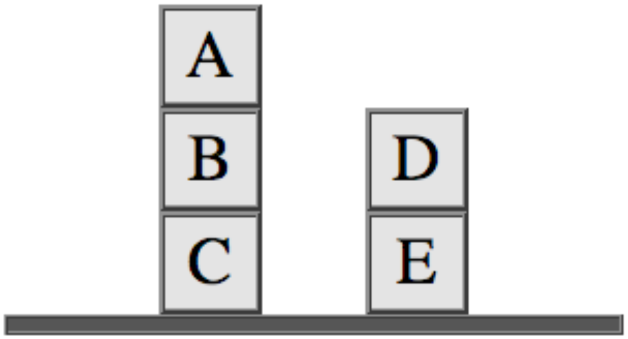


Constraints on the *on* relation:

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

$$\forall x.\forall y.(on(x,y) \Rightarrow \neg on(y,x))$$

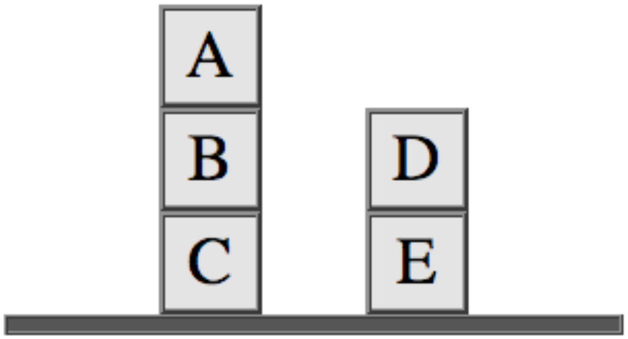
Definitions



Definition of *clear*:

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

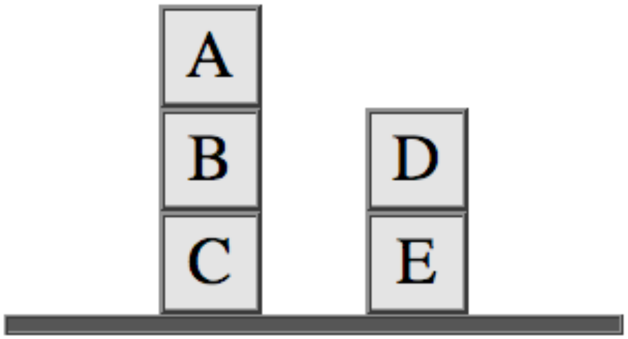
Definitions



Definition of *table*:

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

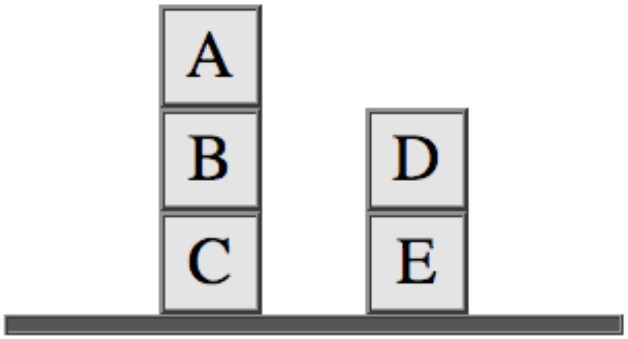
Definitions



Definition of *stack*:

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

Definitions



Definition of *above*:

$$\forall x.\forall z.(above(x,z) \Leftrightarrow on(x,z) \lor \exists y.(on(x,y) \land above(y,z)))$$

Exercise 8.3

Introduction to Logic

Tools for Thought

Exercise 8.3 - Consistency

Consider a version of the Blocks World with just three blocks - a, b, and c. The *on* relation is axiomatized below.

$\neg on(a,a)$	$on(a,b)$	$\neg on(a,c)$
$\neg on(b,a)$	$\neg on(b,b)$	$on(b,c)$
$\neg on(c,a)$	$\neg on(c,b)$	$\neg on(c,c)$

Let's suppose that the *above* relation is defined as follows. This is *almost* the same as in Section 7.7 except that we have replaced an occurrence of *on* with *above*.

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

A sentence ϕ is consistent with a set Δ of sentences if and only if there is a truth assignment that satisfies all of the sentences in $\Delta \cup \{\phi\}$. Say whether each of the following sentences is consistent with the sentences about *on* and *above* shown above. Be careful. It's tricky.

a. $above(a,c)$

b. $above(a,a)$

c. $above(c,a)$

Show Answers Reset Answers

Example - Minefinder

Course Website

Introduction to Logic

Tools for Thought

Help · Lessons · Chapters · Dictionary · Tools

Lesson 7 - Relational Logic

- Lesson 7.1 - Introduction
- Lesson 7.2 - Syntax
- Lesson 7.3 - Semantics
- Lesson 7.4 - Evaluation
- Lesson 7.5 - Satisfaction
- Lesson 7.6 - Sorority World
- Lesson 7.7 - Blocks World
- Lesson 7.8 - Modular Arithmetic
- Exercise 7.1
- Exercise 7.2
- Exercise 7.3
- Exercise 7.4
- Extra - Sorority Life
- Extra - Minefinder
- Extra - Minefield
- Extra - Logicians
- Puzzle - Cards

Use the arrow keys to navigate.

Preface	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Postface

Press the escape key to toggle all / one.

Not Secure — intrologic.stanford.edu

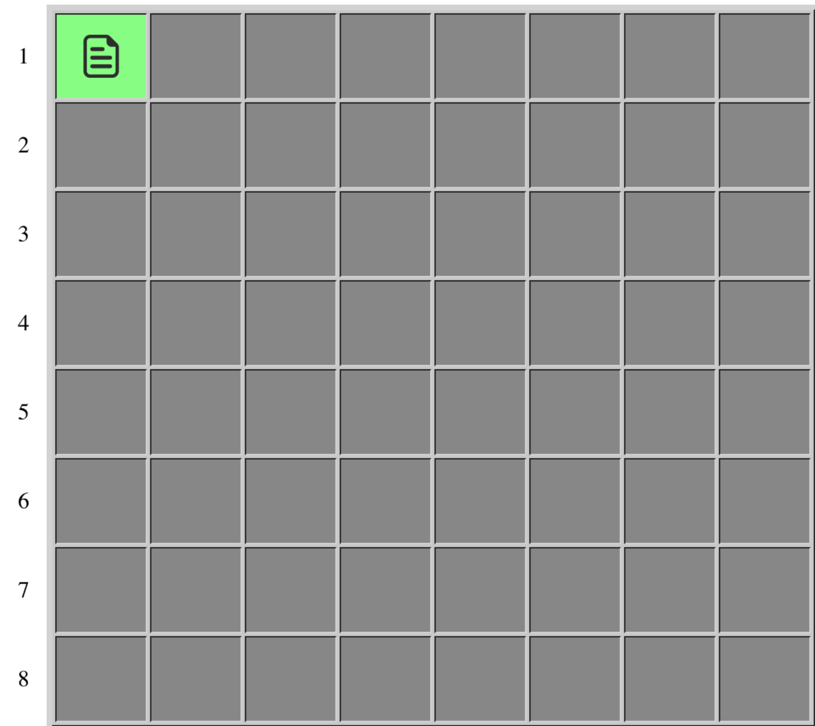
Introduction to Logic

Tools for Thought

Minefinder

$\neg\exists y.mine(1,y)$

	1	2	3	4	5	6	7	8



Show Answer | Show Instructions | Reset Game

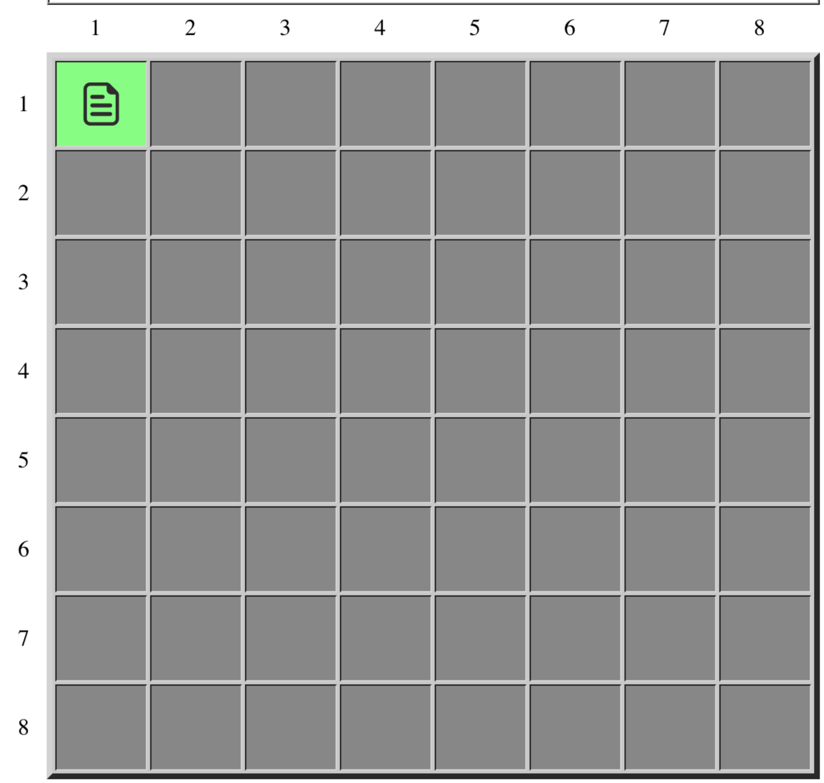
Not Secure — intrologic.stanford.edu

Introduction to Logic

Tools for Thought

Minefield

$\neg\exists y.mine(1,y)$



Show Answer | Show Instructions | Reset Game