Relational Logic Semantics

Michael Genesereth Autumn 2002

Propositional Logic Semantics

A Propositional logic *interpretation* is an association between the propositional constants in a propositional language and the truth values T or F.

Relational Logic Semantics

The *big question*: what is a relational logic interpretation? There are no propositional constants, just object constants, function constants, and relation constants. To what do they refer?

Outline

→ Modeling the World
- Objects, Functions, Relations
- Data
- Models

Semantics of Relational Logic
- Atomic Sentences
- Logical Sentences
- Quantified Sentences

General Remarks
- Ontological Promiscuity
- Role of Logic

Objects

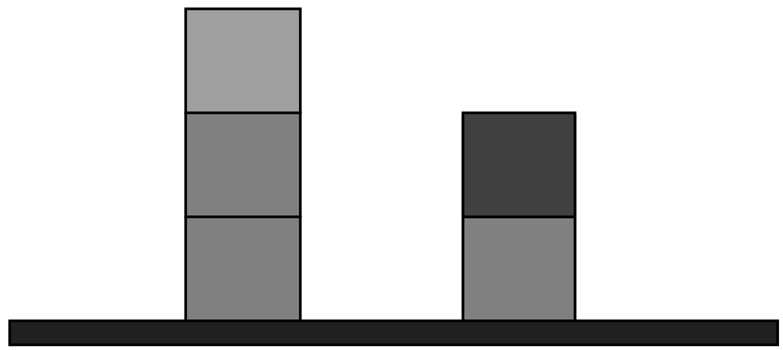
An object is an entity presumed or hypothesized to exist in the world we are discussing.

Primitive: *a quark*
Composite: *an engine*, *this class*

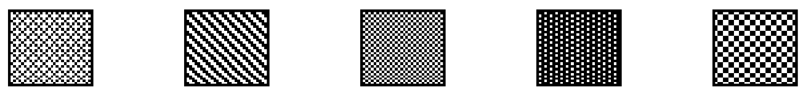
Real: *Sun*, *Mike*
Imaginary: *a unicorn*, *Sherlock Holmes*

Physical: *Earth*, *Moon*, *Sun*
Abstract: *Justice*

Blocks World

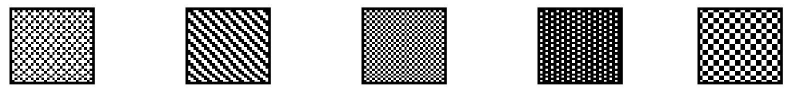


Universe of Discourse

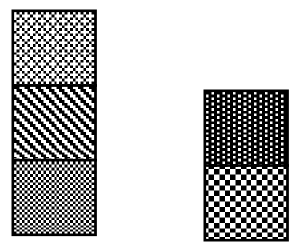


Different Universes

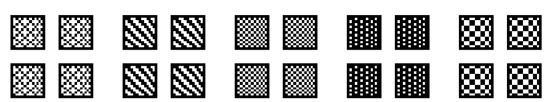
Blocks:



Stacks:



Fragments:



Relations

An *n*-ary *relation* is a property that holds of various combinations of *n* objects.

clear - true of a block iff it has no blocks above it.
table - true of a block iff it is resting on the table.

on - true of 2 blocks iff one is immediately on the other.
above - true of 2 blocks iff one is anywhere above the other.
below - true of 2 blocks iff one is anywhere below the other.

stack - true of 3 blocks iff they form a stack of 3 blocks.

Functions

An *n*-ary function is a relation associating each combination of *n* objects in a universe of discourse (called the *arguments*) with a single object (called the *value*).

Numerical Examples:

Unary: *sqrt*, *log*

Binary: +, -, *, /

People Examples:

Unary: *mother*, *father*

Functions

Functions are *total* and *single-valued* - one and exactly one value for each combination of arguments.

Partial - not defined for some combination of arguments

Multivalued - more than one value for some argument combination

NB: We ignore partial and multi-valued functions.

Ingredients for a Model

Universe of Discourse - a set of object constants, one for each object under discussion.

Functional Basis Set - a set of function constants, one for each function under discussion.

Relational Basis Set - a set of relation constants, one for each relation under discussion.

Note that, in some cases, we need more object constants than we can form from 26 letters and 10 digits. We can solve this problem by extending our alphabet. However, this won't be necessary in this course.

Data

A *datum* is a ground, atomic sentence in which all arguments are object constants.

$$on(a,b)$$

Intuitively, a datum is true if and only if the relation holds of the arguments. It can also be viewed as an instance of a relation.

Models

A *model* is an arbitrary set of data.

$$\{clear(a), clear(d), table(c), table(d),$$
$$on(a,b), on(b,c), on(d,e), stack(a,b,c)\}$$

Intuitively, a model is the set of *all* data that are true in the world being considered. If a datum is *not* included in a model, it is assumed to be false in that model.

Outline

Modeling the World
- Objects, Functions, Relations
- Data
- Models

→ Semantics of Relational Logic
- Atomic Sentences
- Logical Sentences
- Quantified Sentences

General Remarks
- Ontological Promiscuity
- Role of Logic

Atomic Sentences

A *ground atomic* sentence φ is true in a model Γ (written $\models_\Gamma \varphi$) if and only if φ is a member of Γ.

Model:

$$\{clear(a), clear(d), table(c), table(d), on(a,b), on(b,c), on(d,e), stack(a,b,c)\}$$

True:

$clear(a)$

$clear(d)$

Not True:

$clear(b)$

$clear(c)$

$clear(e)$

Logical Sentences

A *negation* is true if and only if its target is false.

A *conjunction* is true if and only if every conjunct is true.

A *disjunction* is true if and only if some disjunct is true.

An *implication* is true if and only if the antecedent is false or the consequent is true.

A *reduction* is true if and only if the antecedent is false or the consequent is true.

An *equivalence* is true if and only if the arguments are either both false or both true.

Instances

An *instance* of a sentence relative to a model is a sentence obtained by consistently substituting an object constant from the model's universe of discourse for each *free variable* in the sentence.

$$p(x,y) \wedge q(x,b,z) \quad \rightarrow \quad p(a,a) \wedge q(a,b,b)$$

Note that we do *not* substitute for *bound* variables (until later).

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

Quantified Sentences

A *universally quantified* sentence is true if and only if every instance of the scope is true. An *existentially quantified* sentence is true if and only if some instance of the scope is true.

Model:

$$\{clear(a), clear(d), table(c), table(d),$$
$$on(a,b), on(b,c), on(d,e), stack(a,b,c)\}$$

True:

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

$\exists x.clear(x)$

Not True:

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

$\exists x.(table(x) \wedge clear(x))$

Open Sentences

The preceding definitions apply to closed sentences, i.e. those with no free variables.

An *open* sentence is true in a model if and only if it satisfies every instance of the sentence relative to the model.

True:

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

Not True:

$$on(x,y)$$

This just formalizes the notion that free variables are universally quantified.

Functions and Models

A functional relation can be notated the same as any other relation.

$$\{boss(art,art), boss(joe,art)\}$$

However, to make explicit the functional character of functional relations, we often write as equations.

$$\{boss(art)=art, boss(joe)=art\}$$

Instances Again

An instance of a sentence relative to a model is a sentence obtained by (1) consistently substituting an object constant from the model for each *free variable* in the sentence (2) replacing all ground *functional terms* by their values in the model.

Model:

$$\{boss(art)=art, boss(joe)=art\}$$

Example:

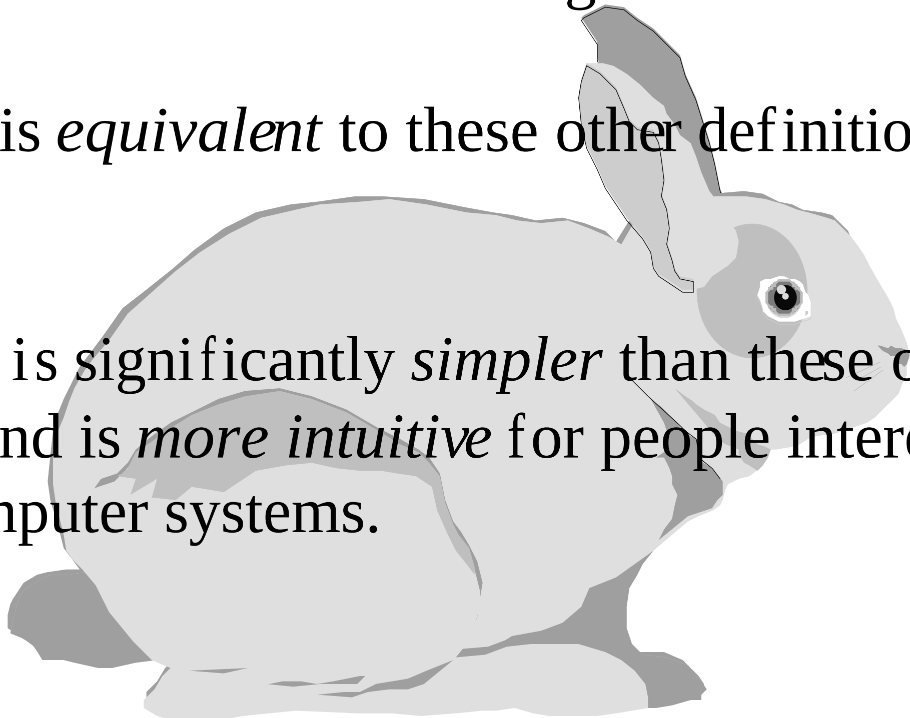
$$p(x,boss(x)) \rightarrow \quad p(joe,boss(joe)) \quad \rightarrow \quad p(joe,art)$$

Note

The definition of semantics given here is *not* the same as that given in the notes or in standard logic textbooks.

However, it is *equivalent* to these other definitions in terms of its results.

Moreover, it is significantly *simpler* than these other definitions and is *more intuitive* for people interested in building computer systems.



Definitions

A sentence is *valid* if and only if *every* model satisfies it. A sentence is *unsatisfiable* if and only if *no* model satisfies it. A sentence is *contingent* if and only there is some model that makes it true and some model that makes it false.

A set of premises Δ *logically entails* a conclusion φ (written $\Delta \models \varphi$) if and only if every model that satisfies the premises also satisfies the conclusion (i.e. $\models_\Gamma \Delta$ implies $\models_\Gamma \varphi$, for all Γ).

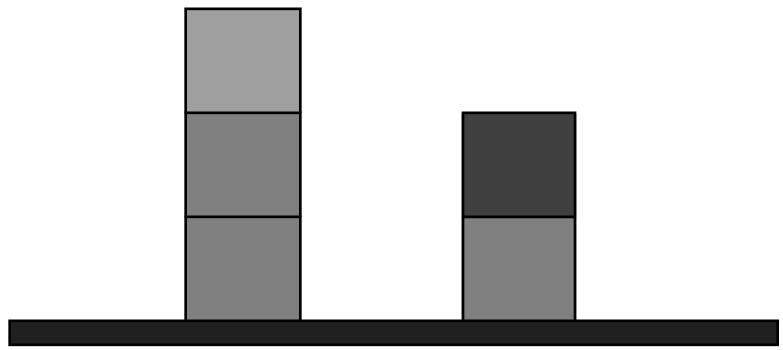
Outline

Modeling the World
- Objects, Functions, Relations
- Data
- Models

Semantics of Relational Logic
- Atomic Sentences
- Logical Sentences
- Quantified Sentences

→ General Remarks
- Ontological Promiscuity
- Role of Logic

Blocks World



Different Models

There is only one world, so how can there be more than one model?

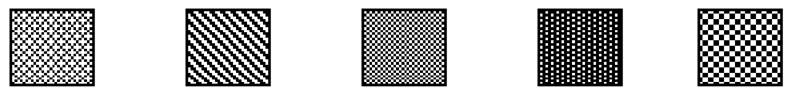
Different people can have different beliefs.

Different models correspond to different times.

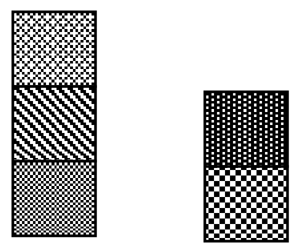
Different models correspond to different places.

Ontological Promiscuity

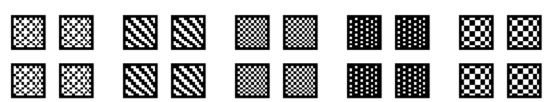
Blocks:



Stacks:



Fragments:



Interpretation

Universe of Discourse:

$$\{a,b,c,d,e\}$$

Relational Basis Set:

$$\{on_2, red_1, green_1, blue_1\}$$

Model:

$$\{on(a,b),\ on(b,c),\ on(d,e),$$
$$red(a), red(d), blue(b), green(c), green(e)\}$$

Reification

$on(a,b)$		$on(a,b)$
$on(b,c)$		$on(b,c)$
$on(d,e)$		$on(d,e)$
$red(a)$	$\longrightarrow$	$color(a,red)$
$red(d)$		$color(d,red)$
$blue(b)$		$color(b,blue)$
$green(c)$		$color(c,green)$
$green(e)$		$color(e,green)$

Universal Relation

$on(a,b)$		$property(on,a,b)$
$on(b,c)$		$property(on,b,c)$
$on(d,e)$		$property(on,d,e)$
$color(a,red)$		$property(color,a,red)$
$color(d,red)$	$\longrightarrow$	$property(color,d,red)$
$color(b,blue)$		$property(color,b,blue)$
$color(c,green)$		$property(color,c,green)$
$color(e,green)$		$property(color,e,green)$

Role of Logic

Incomplete Information

Block *a* is on block *b* *or* it is on block *c*.

Block *a* is *not* on block *b*.

Integrity

A block may not be *on* itself.

A block may be *on* only one block at a time.

Definitions

One block is *under* another iff the second is *on* the first.

A block is *clear* iff there is no block *on* it.

A block is on the *table* iff there is no block *under* it.