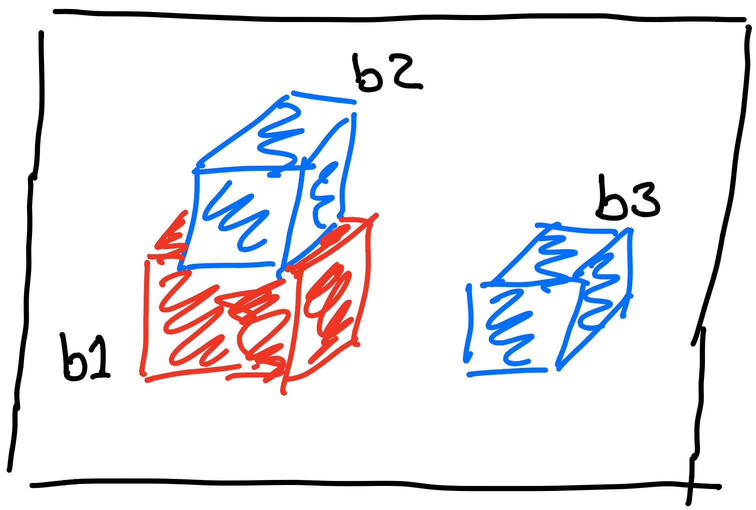
CS 245

An example world — the block world.

Block world consists of a table and some blocks.

The blocks may be colored.



Each block may be on the table or on another block.

Blocks have color.

The objects, blocks, may be represented by a set

$B = \{b1, b2, b3\}$

We use relations to describe the specifics of the world.

$$\{ b \in B \mid Blue(b) \} = \{ b2, b3 \}$$

$$\{ b \in B \mid Red(b) \} = \{ b1 \}$$

Notice that b1 is an object and Red(b1) a property about b1.

Blue(b1) is also a property — one that does not hold in the given world.

OnTable is also a property of the blocks.

In this examble we have

$OnTable(b1)$

$OnTable(b3)$

$\neg OnTable(b2)$

Notice that OnTable is a unary property.

We can write On as a binary property to describe relationships between the blocks

$$\{(b, b') \mid b, b' \in B \text{ and } On(b, b')\}$$

$$= \{(b2, b1)\}$$

On	b1	b2	b3
b1	0	0	0
b2	1	0	0
b3	0	0	0

Example properties of the block world

There does not exist a block that is on itself.

$$\neg \exists x (B(x) \wedge On(x, x))$$

$$\forall x (\neg (B(x)) \vee \neg On(x, x))$$

$$\forall x\,(\ B(x) \rightarrow \neg On(x,x)\ )$$

A box that is on the table is not on any box.

$$\forall x \left(\left(B(x) \wedge OnTable(x) \right) \rightarrow \neg \exists y \left(B(y) \wedge On(x, y) \right) \right)$$

These properties describe facts about all (or most) block worlds.

We can also describe properties of specific block worlds.

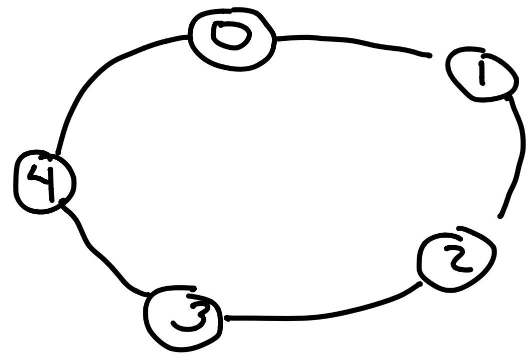
Each red box has a blue box that is on top of the red box.

$$\forall x\, ((B(x) \wedge Red(x)) \rightarrow \exists y\, (B(y) \wedge Blue(y) \wedge On(y, x)))$$

There is a box that is on top of a box that is on the table.

$$\exists x \, \exists y \, (B(x) \wedge B(y) \wedge OnTable(x) \wedge On(y, x))$$

Example: consider a finite graph of nodes and edges.



A graph is undirected if

$$\forall x \forall y \left(N(x) \wedge N(y) \wedge E(x,y) \rightarrow E(y,x) \right)$$

In using first-order logic to study a particular theory there is, typically, a non-empty domain of individuals.

We then add some designated individuals, relations and functions (from which others may be defined).

Together, a domain, and the designated individuals, relations and functions constitute a structure.

The natural numbers can be given by

$\mathbb{N}$ — the domain

0 — element 0

$=$ — equality relation

$'$ — unary successor function

$+$ — binary addition function

$\cdot$ — binary multiplication function

Variables ranging over the domain are used to state properties of all elements (or some elements) of the domain.

For all x, $x^2 \geq 0$.

For all x, $x+x = x \cdot x$.

For a given structure, some properties may hold while others do not hold.

Connectives are used to form compound propositions.

As in propositional logic we have

$\wedge$ and

$\vee$ or

$\rightarrow$ implication

$\leftrightarrow$ bi-implication

$\neg$ negation

We also use:

for all x

written

$\forall x$

and

there exists x

written

$\exists x$

For domain, D, an n-ary proposition function, R, is a mapping from D^n into $\{0, 1\}$.

For instance, given domain $\mathbb{N}$, the unary proposition function, even, gives

$$\text{even}(3) = 0$$

$$\text{even}(4) = 1$$

Consider:

for all x, even(x)

Here x is a bound variable, it is bound to the for all, or universal, quantifier.

Similarly in

there exists x, even(x)

x is bound to the existential quantifier, exists.

However in writing

$$even(x)$$

x is unbound, a free variable.

Similarly

$$divedes(x, y)$$

contains two free variables, x and y.

for all y, divedes(x,y)

contains one free variable, x,
and one bound variable, y.

there exists x, for all y, divides(x,y)

contains two bound variables,
x and y, and no free variables.

Universal quantification can be understood as a generalization of $\wedge$.

Existential quantification can be understood as a generalization of $\vee$.

For instance if the domain $D = \{a_1, a_2, a_3\}$

then

for all $x \in D$: $R(x)$

holds if and only if

$R(a_1)$ and $R(a_2)$ and $R(a_3)$.

While

there exists $x \in D$: $R(x)$

holds if and only if

$R(a_1)$ or $R(a_2)$ or $R(a_3)$

We may also restrict the range of quantification over a domain.

For example:

for all even x in $\mathbb{N}$,
$\quad div(2, x)$

Variables that appear in first-order logic formulae range over the individuals of the domain.

In second order logic, second order variable range over sub-sets of the domain.

In propositional logic we gave rules to define the well formed formulae of Form(L^P). For first-order logic the formulae will describe Form(L).

The basic elements of Form(L) are given without respect to any particular structure.

There are an unbounded number of individual symbols, these include

$a \quad b \quad c \quad a_1 \quad b_1 \quad c_2$

There are an unbounded number of n-ary relation symbols that include:

F G H R

There is a designated binary relation symbol for equality: $\approx$

Note that a particular logic, L, used to define Form(L) may not include the symbol $\approx$.

When a first order logic is given with $\approx$ it may be described as a 'first order logic with equality.'

There is an unbounded number of n-ary function symbols

$f \quad g \quad h$

There is an unbounded number of free variable symbols

$$u \quad v \quad w \quad u_1 \quad v_1 \quad w_2$$

There is an unbounded number of bound variable symbols

$x \quad y \quad z \quad x_1 \quad y_1 \quad z_2$

There is the unary connective: $\neg$

And the binary connectives: $\wedge$

$\vee$

$\rightarrow$

$\leftrightarrow$

And the quantifiers: $\forall$

$\exists$

with a quantifier symbol and a bound variable we can write a quantifier:

$\exists x$ existential quantifier

$\forall x$ universal quantifier

Finally we use the punctuation symbols:

( ) ,

For a given set of symbols we can form the expressions of L, the finite strings of symbols.

The set of terms of L, $Term(L)$ is given as follows:

[1] Each individual symbol, a, is a term. That is $a \in Term(L)$.
Each free variable, u, is a term.
That is $u \in Term(L)$.

[2] If $t_1, \ldots, t_n \in Term(L)$ and f is an n-ary function symbol then $f(t_1, \ldots, t_n) \in Term(L)$.

Example: the following are all in $Term(L)$

a

b

$f(a)$

u

$g(u, f(a))$

$f(g(u), f(a))$

A term that does not have a free variable symbol is 'closed.'

Examples: a

$f(a)$

$g(a, f(b))$

are closed terms.

u

$g(u, f(a))$

are not closed terms.

If $U, V_1, \ldots, V_n$ are expressions and $s_1, \ldots, s_n$ are symbols then

$$U(s_1, \ldots, s_n)$$

is an expression in which $s_1, \ldots, s_m$ appear.

If $U(s_1, \ldots, s_n)$ is an expression then $U(V_1, \ldots, V_n)$ is the expression formed by replacing each s_i in $U(s_1, \ldots, s_m)$ with V_i.

Example: Let $U(a, u)$ be the expression $F(a) \to G(a, u)$.

Then $U(u, a) = F(u) \to G(u, a)$.

Notice that to form $U(u, a)$ we replace, in $U(a, u)$, each occurrence of 'a' by 'u' and, simultaneously, each occurrence of 'u' by 'a'.

Def. Atom(L). An expression of L is in Atom(L) if it is of one of the two forms:

[1] $F(t_1, \ldots, t_n)$ where F is an n-ary relational symbol and for all $i \in [1..n]$: $t_i \in$ Term(L).

[2] $\approx(t_1, t_2)$ where $t_1, t_2 \in$ Term(L).

Note that $t_1 \approx t_2$ may be written for $\approx(t_1, t_2)$.

Def. Form(L) is the smallest class of expressions of L closed under the formation rules:

[1] Atom(L) $\subseteq$ Form(L)

[2] If $A \in$ Form(L) then $(\neg A) \in$ Form(L).

[3] If $A, B \in$ Form(L) then $(A * B) \in$ Form(L) where $*$ is one of $\wedge, \vee, \rightarrow, \leftrightarrow$.

[4] If $A(u) \in$ Form(L), x is a free variable not occurring in $A(u)$, then $\forall x\, A(x) \in$ Form(L) and $\exists x\, A(x) \in$ Form(L).

For $A \in Form(\mathcal{L})$, if A has no free variables then A is a sentence.

Examples: $F(a, b)$

$\forall y\, F(a, y)$

$\exists x \forall y\, F(x, y)$

are sentences.

while

$F(a, u)$

$\exists x\, (u, x)$

are not sentences.