Cooperating Intelligent Systems

First-order predicate logic

Chapter 8, AIMA

Why first order logic (FOL)?

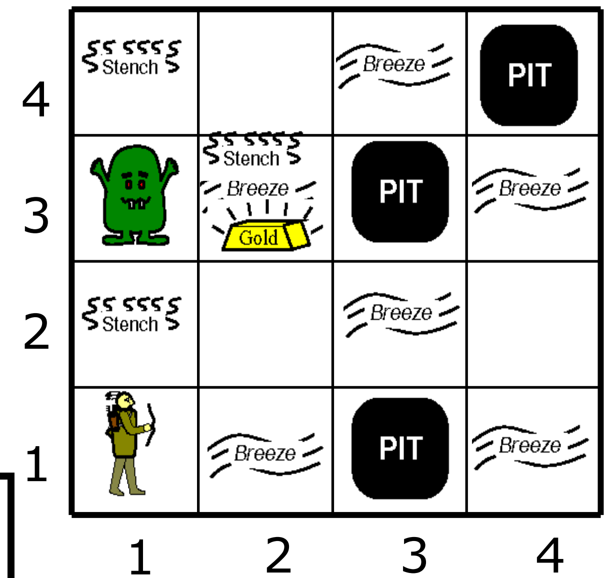
- **Logic is a language** we use to express knowledge in rigorous manner
  - consists of syntax and semantics
- **Propositional (boolean) logic** is too limited for a lot of (even simple) domains
  - complex environments cannot be described in a sufficiently natural and concise way
- **First order logic (predicate calculus)** can express a lot more of common-sense knowledge in a reasonable manner

Limitations of propositional logic

Wumpus in (3,1) ⇒ Stench in (3,2)

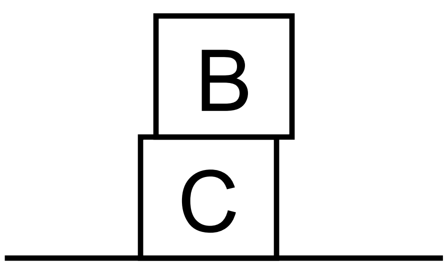
$W_{31} \Rightarrow S_{32}$

Propositional logic needs to express this for every square in the Wumpus world.



A = John has a bike ∧ B = John has a car

Propositional logic cannot express that these two statements are about the same person.



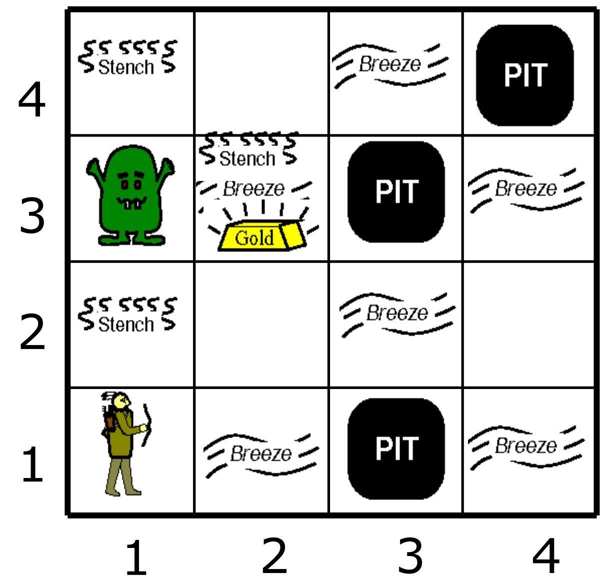
Block B is on top of C ⇒ ¬(C is free to be moved)

If we have more blocks, we need *a lot* of statements like this.

What we want:

”If there is a Wumpus in square x, then there will be a stench in all *neighboring* squares.”

Say it once and for all.

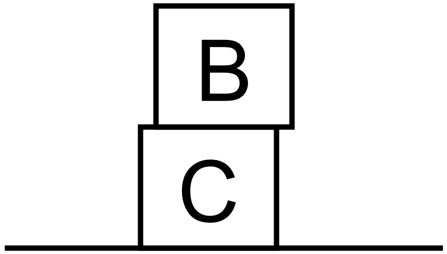


”John has a bike and a car.”

...

”People with mutliple vehicles watch weather forecasts more often.”

”We cannot move an object if there is something on top of it.”



First-order logic (FOL)

- **Logical symbols** (always the same meaning)
  - logical connectives: *and, or, implication,* etc.
  - quantifiers: *for all* (∀) and *there exists* (∃)
  - an infinite set of variables: *x, y, z, …*
  - equality symbol and truth constants: *=, T, F*

- **Non-logical symbols** (depend on interpretation)
  - constants (objects): man, woman, house, car, conflict, slawek, stefan, denni, halmstaduniversity, ...
  - predicates (relations between objects): red, green, nice, larger, above, below, schedule, itinerary, ...
  - functions: fatherOf, brotherOf, beginningOf, birthday, employer, flightNumber, slideTitle, man, woman, …

constants are actually a special case of functions

First-order logic (FOL)
Syntax

Constants

A, 125, Q, John, KingJohn, TheCrown, EiffelTower, D215, Wumpus, HH, TravelAgent,...

Relations/predicates (of various arities)

Unary predicates (properties): $Orange^1$, $Nice^1$, $Rich^1$, ...

N-ary relations: $Parent^2$, $Brother^2$, $Married^2$, $Before^2$, ...

Function constants (of various arities)

$FatherOf^1$(KingJohn), $LeftLegOf^1$(John), $NeighborOf^1$(HH),

$DistanceBetween^2$(A,B), $Times^2$(2,4), $Price^2$(Fruit,Weight),

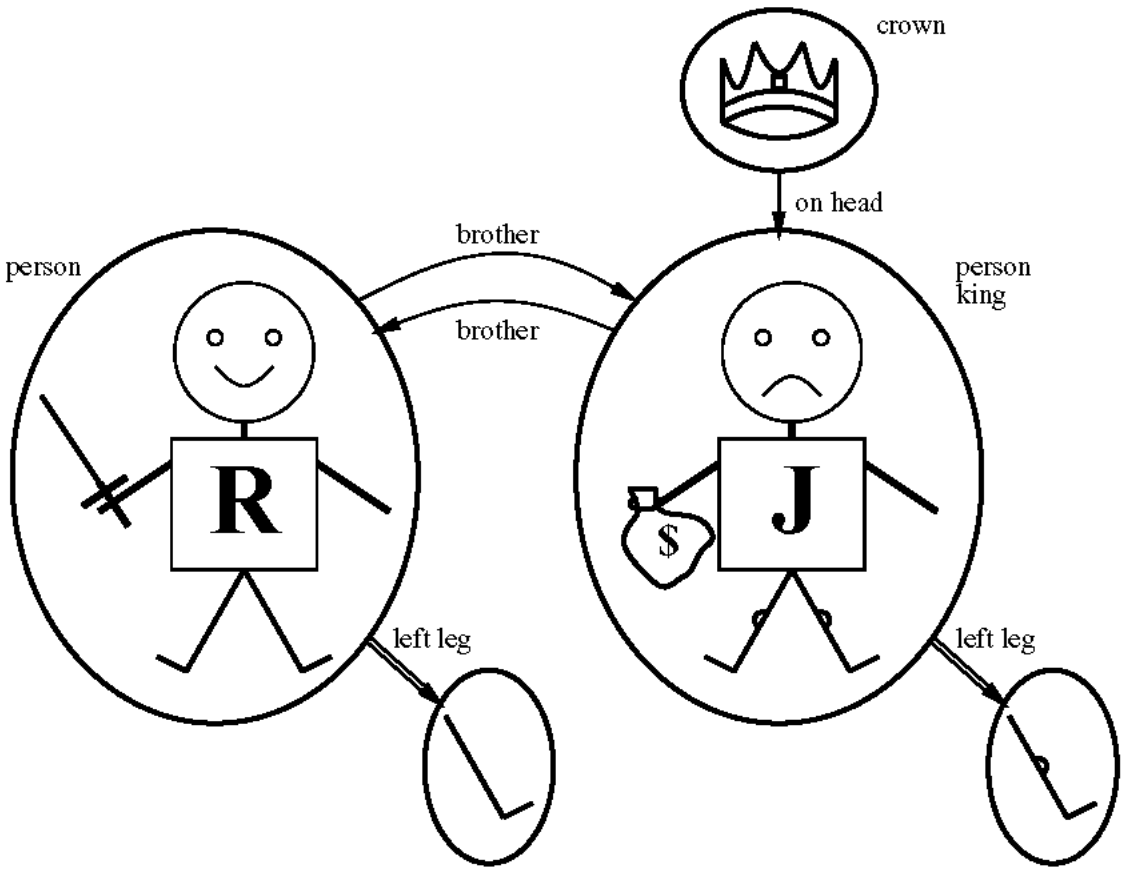
$Itinerary^3$(DepartureAirport, ArrivalAirport, DepartureTime),

$KingJohn^0$(), A^0(), 125^0(), HH^0(), $Agent^0$(), ...

The superscript denotes the ”arity” = the number of arguments

R = RichardTheLionheart
J = KingJohn
C = Crown

Object constants



<u>Function constants</u>

LeftLegOf(R)

LeftLegOf(J)

<u>Relations (predicates)</u>

Person(R)
Person(J)
King(J)
Crown(C)

Unary

Brother(J,R)
Brother(R,J)
OnHead(C,J)

Binary

First-order logic (FOL)
Syntax

Term

1. An object constant is a term
2. A complete function constant is a term (complete = all arguments are provided and each one of them is a term)
3. A *variable* is a term.

Intuitively, a term corresponds to a well-defined object in the world.

First-order logic (FOL)

Syntax

Well-Formed Formula (wff)

1. A complete predicate symbol is a wff (complete = all arguments are provided and each one of them is a term)
2. An equality between two terms is a wff
3. Negation of a wff is a wff
4. Two wffs connected by a connective is a wff
5. Quantifier ($\forall$ or $\exists$ with a variable) followed by a wff is a wff.

First-order logic (FOL)
Syntax

Variables and quantifiers

Variables refer to unspecified objects in the domain. We will denote them by lower case letters (at the end of the alphabet)

x, y, z, ...

Quantifiers constrain the meaning of a variable in a sentence. There are two quantifiers:

"For all" (∀) and "There exists" (∃)

Universal quantifier Existential quantifier

First-order logic (FOL)
Syntax

Variables in wff

1. Variable is said to be *free* in a wff if it occurs in this wff and there is no quantifier *binding* this variable

   Brother(x,y) ∧ King(x) ∧ Mother(x,y) ⇒ Woman(x)

2. Variable is said to be *bound* in a wff if it occurs in this wff and it is not free

   $\forall_x \forall_y$ Mother(x,y) ⇒ Woman(x)

   $\forall_y \exists_x$ Mother(x,y)

First-order logic (FOL)
Syntax
Sentence

A well formed formula without any free variables is called a sentence

- Atomic sentence
  A complete predicate symbol (relation)
  Brother(RichardTheLionheart,KingJohn), Dead(Mozart), Married(CarlXVIGustaf,Silvia), Orange(Block(C)),...
- Complex sentence
  Formed by sentences and connectives
  Dead(Mozart) $\wedge$ Composer(Mozart),
  $\neg$King(RichardTheLionheart) $\Rightarrow$ King(KingJohn),
  King(CarlXVIGustaf) $\wedge$ Married(CarlXVIGustaf,Silvia) $\Rightarrow$ Queen(Silvia)

First-order logic (FOL)
Syntax
Sentence

A well formed formula without any free variables is called a sentence

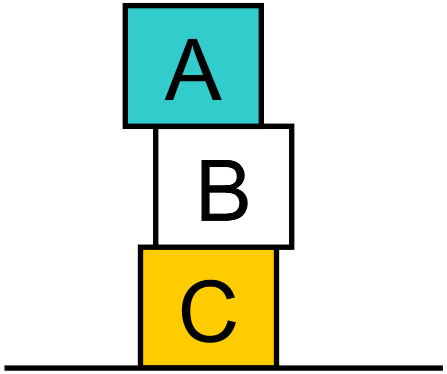
- Atomic sentence
  A complete predicate symbol (relation)
  Brother(RichardTheLionheart,KingJohn), Dead(Mozart), Married(CarlXVIGustaf,Silvia), Orange(Block(C)),...
- Complex sentence
  Formed by sentences and connectives
  Dead(Mozart) $\wedge$ Composer(Mozart),
  $\neg$King(RichardTheLionheart) $\Rightarrow$ King(KingJohn),
  King(CarlXVIGustaf) $\wedge$ Married(CarlXVIGustaf,Silvia) $\Rightarrow$
  $\Rightarrow$ Queen(Silvia)

First-order logic (FOL)
Semantics

- Semantics assigns truth values to sentences
  - terms and wffs that are not sentences do not, in general, have any truth values: King(X)
- The truth value of atomic sentences comes from the model/interpretation
  - just like in propositional logic: King(Richard)
- The truth value of complex sentences is determined by truth tables
  - Quantifiers take into account *domain of discourse:* $\forall_X \exists_Y \; X = Y * Y$

Example: Block world

On(A,B)
On(B,C)
On(B,A)
On(C,B)
On(C,A)
On(A,C)
Clear(A)
Clear(B)
Clear(C)
Under(B,A)
Under(C,B)
Under(A,B)
Under(C,A)
Under(B,A) $\wedge$ Over(B,C)
Under(C,B) $\vee$ Under(C,A)



First-order logic (FOL)
Syntax

Variables and quantifiers

(∀ ”For all...”)

$\forall x\ King(x) \Rightarrow Person(x)$	*”All kings are persons”*
$\forall x,y\ Brother(x,y) \Rightarrow Sibling(x,y)$	*”All brothers are siblings”*
$\forall x,y\ Son(x,y) \wedge King(y) \Rightarrow Prince(x)$	*”All sons of kings are princes”*
$\forall x\ AIstudent(x) \Rightarrow Overworked(x)$	*”All AI students are overworked”*

$\forall_{\text{<variables>}}$ <wff>

Everyone at Berkeley is smart:

$$\forall_x \quad At(x, Berkeley) \Rightarrow Smart(x)$$

$\forall_x P$ is equivalent to the *conjunction* of *instantiations* of P

$$\begin{aligned} & At(KingJohn, Berkeley) \Rightarrow Smart(KingJohn) \\ \wedge \; & At(Richard, Berkeley) \Rightarrow Smart(Richard) \\ \wedge \; & At(Berkeley, Berkeley) \Rightarrow Smart(Berkeley) \\ \wedge \; & \ldots \end{aligned}$$

Typically, $\Rightarrow$ is the main connective with $\forall$

Common mistake: using $\wedge$ as the main connective:

$$\forall_x \quad At(x, Berkeley) \wedge Smart(x)$$

„Everybody is at Berkeley and everybody is smart”

Slide from S. Russel @ Berkeley

First-order logic (FOL)
Syntax

Variables and quantifiers

(∃ ”There exists...”)

$\exists x$ King(x) $\wedge$ Person(x)	*”There is a king who is a person / There is a person who is a king”*
$\exists x$ Loves(x,KingJohn)	*”There is someone who loves King John”*
$\exists x$ $\neg$Loves(x,KingJohn)	*”There is someone who does not love King John”*
$\exists x$ AIstudent(x) $\wedge$ Overworked(x)	*”There is an AI student that is overworked”*

$\exists_{\text{<variables>}}$ <wff>

Someone at Stanford is smart:

$$\exists_x \quad At(x, Stanford) \land Smart(x)$$

$\exists_x P$ is equivalent to the *disjunction* of *instantiations* of P

$$
\begin{aligned}
& At(KingJohn, Stanford) \land Smart(KingJohn) \\
\lor\ & At(Richard, Stanford) \land Smart(Richard) \\
\lor\ & At(Berkeley, Stanford) \land Smart(Berkeley) \\
\lor\ & \ldots
\end{aligned}
$$

Typically, $\land$ is the main connective with $\exists$

Common mistake: using $\Rightarrow$ as the main connective:

$$\exists_x \quad At(x, Stanford) \Rightarrow Smart(x)$$

This is true whenever there is somebody not at Stanford

Slide from S. Russel @ Berkeley

First-order logic (FOL)
Syntax

Nested quantifiers

$\forall x\ \exists y\ Loves(x,y)$	*"Everybody loves somebody"*
$\exists y\ \forall x\ Loves(x,y)$	*"Someone is loved by everyone"*
$\forall x\ \exists y\ Loves(y,x)$	*"Everyone is loved by someone"*
$\exists y\ \forall x\ Loves(y,x)$	*"Someone loves everyone"*
$\forall x\ \exists y\ Loves(x,y) \wedge (y \neq x)$	*"Everybody loves somebody else"*

First-order logic (FOL)
Syntax

Nested quantifiers

$\forall x\ \exists y\ Loves(x,y) \neq \exists y\ \forall x\ Loves(x,y)$

"Everybody loves somebody" ≠ "Someone is loved by everyone"

$\forall x\ \exists y\ Loves(y,x) \neq \exists y\ \forall x\ Loves(y,x)$

"Everyone is loved by someone" ≠ "Someone loves everyone"

The order of ∀ and ∃ matters!

Quantifier duality

DeMorgan's rules

$$\forall x\ \neg P(x) \equiv \neg\exists x\ P(x)$$

$$\neg\forall x\ P(x) \equiv \exists x\ \neg P(x)$$

$$\forall x\ P(x) \equiv \neg\exists x\ \neg P(x)$$

$$\exists x\ P(x) \equiv \neg\forall x\ \neg P(x)$$

Ponder these for a while...

Family fun

Family axioms:

”A mother is a female parent”

”A husband is a male spouse”

”You’re either male or female”

”A child’s parent is the parent of the child” (sic!)

”My grandparents are the parents of my parents”

”Siblings are two children who share the same parents”

”A first cousin is a child of the siblings of my parents”

...etc.

Family theorems:

Sibling is reflexive

Write these in FOL

Family fun

Family axioms:

$\forall_{m,c}$ (m = Mother(c)) $\Leftrightarrow$ (Female(m) $\wedge$ Parent(m,c))

or $\forall$c Female(Mother(c)) $\wedge$ Parent(Mother(c),c)

$\forall_{w,h}$ Husband(h,w) $\Leftrightarrow$ Male(h) $\wedge$ Spouse(h,w)

$\forall_{x}$ Male(x) $\Leftrightarrow$ $\neg$Female(x)

$\forall_{p,c}$ Parent(p,c) $\Leftrightarrow$ Child(c,p)

$\forall_{g,c}$ Grandparent(g,c) $\Leftrightarrow$ $\exists_{p}$ (Parent(g,p) $\wedge$ Parent(p,c))

$\forall_{x,y}$ Sibling(x,y)$\Leftrightarrow$($\exists_{p}$ (Parent(p,x)$\wedge$Parent(p,y)))$\wedge$(x $\neq$ y)

$\forall_{x,y}$ FirstCousin(x,y) $\Leftrightarrow$ $\exists_{p,s}$ (Parent(p,x) $\wedge$ Sibling(p,s) $\wedge$ Parent(s,y))

© 1998, Jon A. Davis

Family theorems:

$\forall$x,y Sibling(x,y) $\Leftrightarrow$ Sibling(y,x)

Spouse(Gomez,Morticia)
Parent(Morticia,Wednesday)
Sibling(Pugsley,Wednesday)
Sister(Ophelia,Morticia)
FirstCousin(Gomez,Itt)
$\exists$p (Parent(p,Morticia) $\wedge$ Sibling(p,Fester))

Family fun

Family axioms:

”A mother is a female parent”

”A husband is a male spouse”

”You’re either male or female”

”A child’s parent is the parent of the child” (sic!)

”My grandparents are the parents of my parents”

”Siblings are two children who share the same parents”

”A first cousin is a child of the siblings of my parents”

...etc.

Family theorems:

Sibling is reflexive

Matematical fun

- ”The square of every negative integer is positive”
    a) $\forall x\ [\text{Integer}(x) \wedge (x > 0) \Rightarrow (x^2 > 0)]$
    b) $\forall x\ [\text{Integer}(x) \wedge (x < 0) \Rightarrow (x^2 > 0)]$
    c) $\forall x\ [\text{Integer}(x) \wedge (x \leq 0) \Rightarrow (x^2 > 0)]$
    d) $\forall x\ [\text{Integer}(x) \wedge (x < 0) \wedge (x^2 > 0)]$

a) ”Not every integer is positive”
    a) $\forall x\ [\neg\text{Integer}(x) \Rightarrow (x > 0)]$
    b) $\forall x\ [\text{Integer}(x) \Rightarrow (x \leq 0)]$
    c) $\forall x\ [\text{Integer}(x) \Rightarrow \neg(x > 0)]$
    d) $\neg\forall x\ [\text{Integer}(x) \Rightarrow (x > 0)]$

Borrowed from http://people.hofstra.edu/faculty/Stefan_Waner/RealWorld/logic/logic7.html

Matematical fun

- ”The square of every negative integer is positive”
  a) $\forall x\ [\text{Integer}(x) \wedge (x > 0) \Rightarrow (x^2 > 0)]$
  b) $\forall x\ [\text{Integer}(x) \wedge (x < 0) \Rightarrow (x^2 > 0)]$
  c) $\forall x\ [\text{Integer}(x) \wedge (x \leq 0) \Rightarrow (x^2 > 0)]$
  d) $\forall x\ [\text{Integer}(x) \wedge (x < 0) \wedge (x^2 > 0)]$

a) ”Not every integer is positive”
  a) $\forall x\ [\neg\text{Integer}(x) \Rightarrow (x > 0)]$
  b) $\forall x\ [\text{Integer}(x) \Rightarrow (x \leq 0)]$
  c) $\forall x\ [\text{Integer}(x) \Rightarrow \neg(x > 0)]$
  d) $\neg\forall x\ [\text{Integer}(x) \Rightarrow (x > 0)]$

Borrowed from http://people.hofstra.edu/faculty/Stefan_Waner/RealWorld/logic/logic7.html

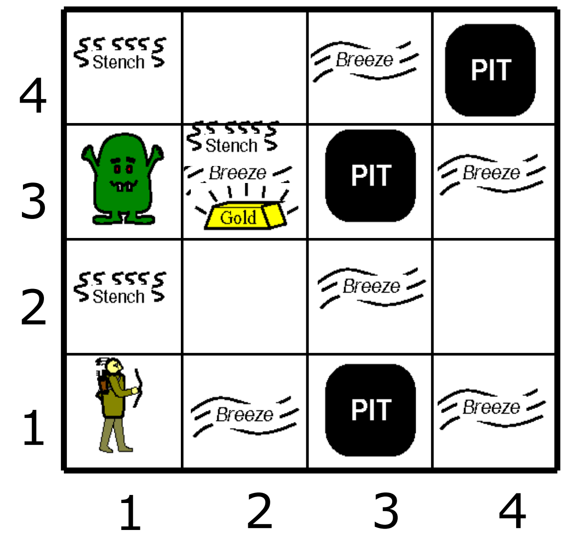
The Wumpus world revisited

Object constants:

Square **s** = [x,y], Agent, Time (t),
Percept **p** = [p_1,p_2,p_3,p_4,p_5], Gold

Predicates:

Pit(**s**), Breezy(**s**), EvilSmelling(**s**), Wumpus(**s**), Safe(**s**), Breeze(**p**,t), Stench(**p**,t), Glitter(**p**,t), Wall(**p**,t), Scream(**p**,t), Adjacent(**s**,**r**), At(Agent,**s**,t), Hold(Gold,t)



(There are other possible representations)

∀ x,y,z,w Adjacent([x,y],[z,w]) ⇔ ([z,w] ∈ {[x+1,y],[x-1,y],[x,y+1],[x,y-1]})

∀**s** Breezy(**s**) ⇔ ∃**r** (Adjacent(**r**,**s**) ∧ Pit(**r**))

∀**s** EvilSmelling(**s**) ⇔ ∃**r** (Adjacent(**r**,**s**) ∧ Wumpus(**r**))

∀**s** (¬EvilSmelling(**s**) ∧ ¬Breezy(**s**)) ⇔ ∀**r** (Adjacent(**r**,**s**) ∧ Safe(**r**))

∀**s**,t (At(Agent,**s**,t) ∧ Breeze(**p**,t)) ⇒ Breezy(**s**)

∀**s**,t (At(Agent,**s**,t) ∧ Stench(**p**,t)) ⇒ EvilSmelling(**s**)

Compare to the 275 rules in boolean KB!

Puzzles with nested quantifiers

- Are both these statements true?

$$\forall x \exists y \;\; x^2 < y$$ TRUE

$$\exists y \forall x \;\; x^2 < y$$ FALSE

Puzzles with nested quantifiers

- Are both these statements true?

$$\forall x \exists y \ \ x + y = 0$$ TRUE

$$\exists y \forall x \ \ x + y = 0$$ FALSE

Translations...

Translate the following sentences to a first order logic expression

1. The product of two negative integers is positive

2. The difference of two negative integers is not necessarily negative

Examples borrowed from Aaron Blomfield @ University of Virginia & Kenneth Rosen, *Discrete Math and Its Applications*, 5th edition. McGraw Hill, 2003

Translations...

Translate the following sentences to a first order logic expression

1. The product of two negative integers is positive

$$\forall x \forall y \ (x < 0) \wedge (y < 0) \wedge Integer(x) \wedge Integer(y) \Rightarrow x \cdot y > 0$$

2. The difference of two negative integers is not necessarily negative

Examples borrowed from Aaron Blomfield @ University of Virginia & Kenneth Rosen, *Discrete Math and Its Applications*, 5th edition. McGraw Hill, 2003

Translations...

Translate the following sentences to a first order logic expression

1. The product of two negative integers is positive

$$\forall x \forall y \ \ (x < 0) \wedge (y < 0) \wedge Integer(x) \wedge Integer(y) \Rightarrow x \cdot y > 0$$

2. The difference of two negative integers is not necessarily negative

$$\exists x \exists y \ \ (x < 0) \wedge (y < 0) \wedge Integer(x) \wedge Integer(y) \wedge (x - y > 0)$$

Examples borrowed from Aaron Blomfield @ University of Virginia & Kenneth Rosen, *Discrete Math and Its Applications*, 5th edition. McGraw Hill, 2003

Translations...

Translate the following sentences to a first order logic expression

1. The product of two negative integers is positive

Why not ∧ ?

$$\forall x \forall y \;\; (x < 0) \wedge (y < 0) \wedge Integer(x) \wedge Integer(y) \Rightarrow x \cdot y > 0$$

2. The difference of two negative integers is not necessarily negative

Why not ⇒ ?

$$\exists x \exists y \;\; (x < 0) \wedge (y < 0) \wedge Integer(x) \wedge Integer(y) \wedge (x - y > 0)$$

Examples borrowed from Aaron Blomfield @ University of Virginia & Kenneth Rosen, *Discrete Math and Its Applications*, 5th edition. McGraw Hill, 2003

Translations...

Translate the following sentences to a first order logic expression

1. The product of two negative integers is positive

Can we write ∀y∀x ?

Why not ∧ ?

$$\forall x \forall y \ \ (x < 0) \wedge (y < 0) \wedge Integer(x) \wedge Integer(y) \Rightarrow x \cdot y > 0$$

2. The difference of two negative integers is not necessarily negative

Why not ⇒ ?

$$\exists x \exists y \ \ (x < 0) \wedge (y < 0) \wedge Integer(x) \wedge Integer(y) \wedge (x - y > 0)$$

Can we write ∃y∃x ?

Examples borrowed from Aaron Blomfield @ University of Virginia & Kenneth Rosen, *Discrete Math and Its Applications*, 5th edition. McGraw Hill, 2003

Translations...

Translate the following sentences to a first order logic expression

1. There is a student at HH who has taken every mathematics course offered at HH.

2. Every salesman has at least one apple

Translations...

Translate the following sentences to a first order logic expression

1. There is a student at HH who has taken every mathematics course offered at HH.

$$\exists x \;\; StudentAtHH(x) \wedge \forall y \left[MathematicsCourseAtHH(y) \Rightarrow Taken(x,y) \right]$$

2. Every salesman has at least one apple

Translations...

Translate the following sentences to a first order logic expression

1. There is a student at HH who has taken every mathematics course offered at HH.

$$\exists x \forall y \;\; StudentAtHH(x) \wedge \left[MathematicsCourseAtHH(y) \Rightarrow Taken(x, y) \right]$$

2. Every salesman has at least one apple

Translations...

Translate the following sentences to a first order logic expression

1. There is a student at HH who has taken every mathematics course offered at HH.

$$\exists x \forall y \;\; StudentAtHH(x) \wedge \left[MathematicsCourseAtHH(y) \Rightarrow Taken(x, y) \right]$$

2. Every salesman has at least one apple

$$\forall x \;\; Salesman(x) \Rightarrow \exists y \;\; Has(x, y) \wedge Apple(y)$$

Translations...

Translate the following sentences to a first order logic expression

1. There is a student at HH who has taken every mathematics course offered at HH.

$$\exists x \forall y \;\; StudentAtHH(x) \wedge \left[MathematicsCourseAtHH(y) \Rightarrow Taken(x, y) \right]$$

2. Every salesman has at least one apple

$$\forall x \exists y \;\; Salesman(x) \Rightarrow Has(x, y) \wedge Apple(y)$$

Translations...

Translate the following sentences to a first order logic expression

1. There is a student at HH who has taken every mathematics course offered at HH.

$$\exists x \forall y \;\; StudentAtHH(x) \wedge \left[MathematicsCourseAtHH(y) \Rightarrow Taken(x,y) \right]$$

2. Every salesman has at least one apple

$$\exists y \forall x \;\; Salesman(x) \Rightarrow Has(x,y) \wedge Apple(y)$$

Translations...

Translate the following sentences to a first order logic expression

1. There is a student at HH who has taken every mathematics course offered at HH.

$$\exists x \forall y \;\; StudentAtHH(x) \wedge \left[MathematicsCourseAtHH(y) \Rightarrow Taken(x,y) \right]$$

2. Every salesman has at least one apple

$$\forall x \exists y \;\; Salesman(x) \Rightarrow Has(x, Apple(y))$$