First Order Logic

Next Chapter:

- Bounded Variable
- Free Variable
- Scope of a Quantifier

- By Deepak Poonia (IISc Bangalore)

Instructor:

Deepak Poonia
IISc Bangalore

GATE CSE AIR 53; AIR 67;
AIR 107; AIR 206; AIR 256

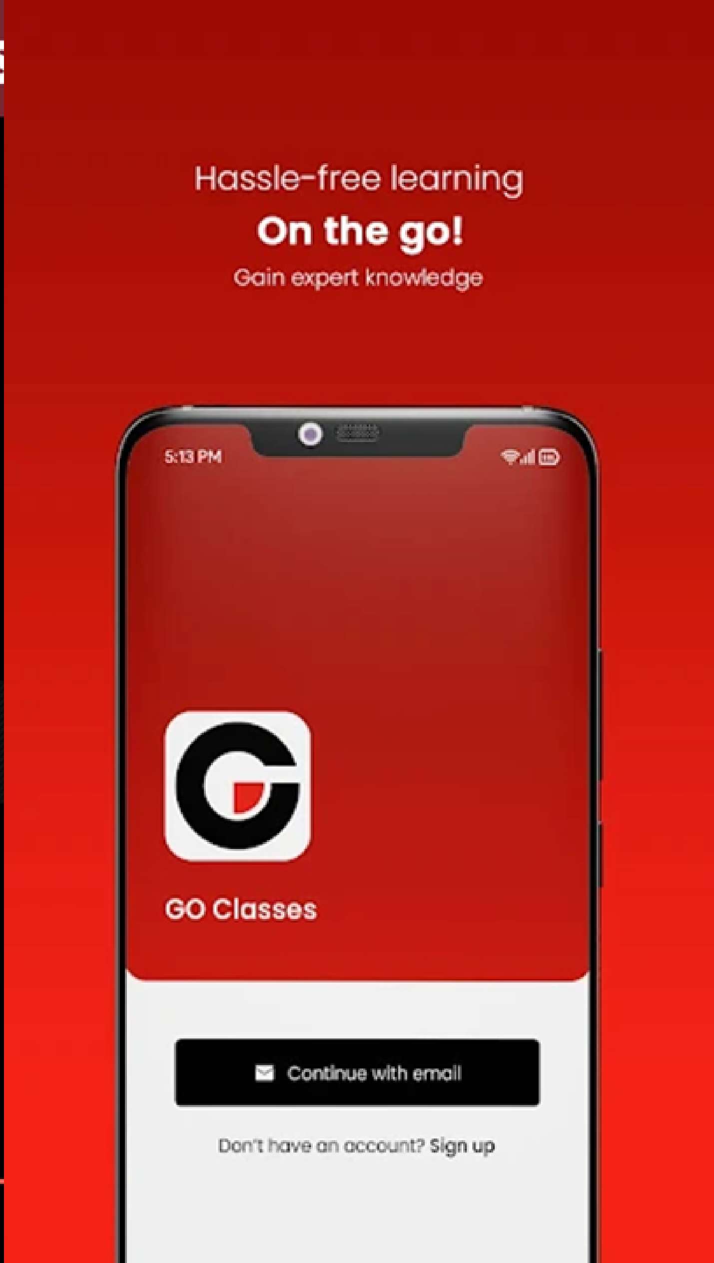
Discrete Mathematics Complete Course:

https://www.goclasses.in/courses/Discrete-Mathematics-Course

Download the GO Classes Android App:

https://play.google.com/store/apps/details?id=com.goclasses.courses

Search "GO Classes" on Play Store.



NOTE :

Complete Discrete Mathematics & Complete Engineering Mathematics Courses, by GO Classes, are **FREE** for ALL learners.

Visit here to watch : https://www.goclasses.in/s/store/

SignUp/Login on Goclasses website for free and start learning.

We are on Telegram. **Contact us** for any help.

Link in the Description!!

Join GO Classes **Doubt Discussion** Telegram Group :

Username:

@GATECSE_Goclasses

We are on Telegram. **Contact us** for any help.

Join GO Classes Telegram Channel, Username: @GOCLASSES_CSE

Join GO Classes **Doubt Discussion** Telegram Group :

Username: @GATECSE_Goclasses

(Any doubt related to Goclasses Courses can also be asked here.)

Join GATEOverflow Doubt Discussion Telegram Group :

Username: @GateOverflow_CSE

First Order Logic

Next Topic:

- Bounded Variable
- Free Variable

First Order Logic

Next Topic:

1. Bounded Variable

(or Quantified Variable)

Consider the following statement:

$\boxed{\text{Every Natural Number is Even.}}$

proposition

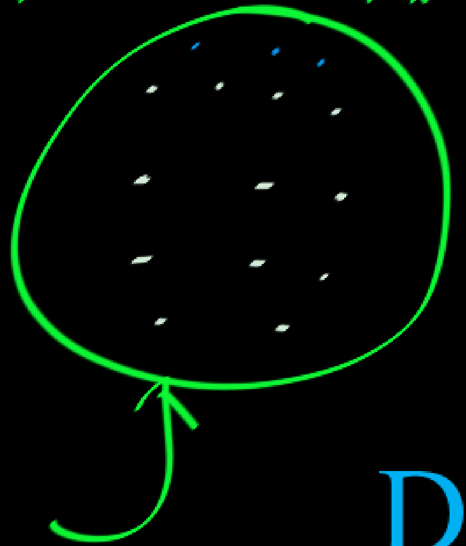
false

Does it have any Variable?

Domain: Set of all natural numbers.

E(x) = x is even.

Domain: $\mathbb{N}$



x

$\boxed{\forall x\, E(x)}$ = false → proposition

Does it have any Variable? Yes, x

Domain: Set of all natural numbers.

$E(x)$ = x is even.

$\forall x\, E(x)$ = Every Natural Number is Even.

Does it Really have any Variable?

Let’s see another example.

Consider: Domain: {a,b,c}

Predicate, N(x): x is Nice.

$\boxed{\forall x\, N(x)} = N(a) \wedge N(b) \wedge N(c)$

a variable

(Dummy variable)

No variable

Domain

a, b, c

x

Constant

(objects)

$$\boxed{\forall_{x}\, P(x)}$$

Proposition

Quantified Variable

OR

Dummy variable

Bounded Variable

Same

Consider: Domain: {a,b,c}

Predicate, N(x): x is Nice.

$\exists x\, N(x)$ = $N(a) \vee N(b) \vee N(c)$

meaning

variable (Dummy or Bounded or Quantified)

No variable

Domain: a, b, c

x

Constants

Consider: Domain: {a,b,c}

Predicate, N(x): x is Nice.

proposition

$\forall x\, N(x) = N(a) \wedge N(b) \wedge N(c)$

propositions

$\exists x\, N(x) = N(a) \vee N(b) \vee N(c)$

proposition

Domain: Set of all natural numbers.

$E(x) = x$ is even.

meaning

$\forall x\, E(x)$ = Every Natural Number is Even.

Does it Really have any Variable?

Yes; 'x' is a Dummy (Bounded) Variable.

Quantified Variable:
(or Bounded Variable or Dummy Variable)

$\forall x \; P(x)$ $\qquad$ $\exists x \, P(x)$

Quantifies variable (Bounded Var.)

Popular name

$\forall_x \, P(x)$

Bounded Var.

$\exists_y \, P(y)$

Bounded Var.

Domain: Set of all natural numbers. E(x) = x is even.

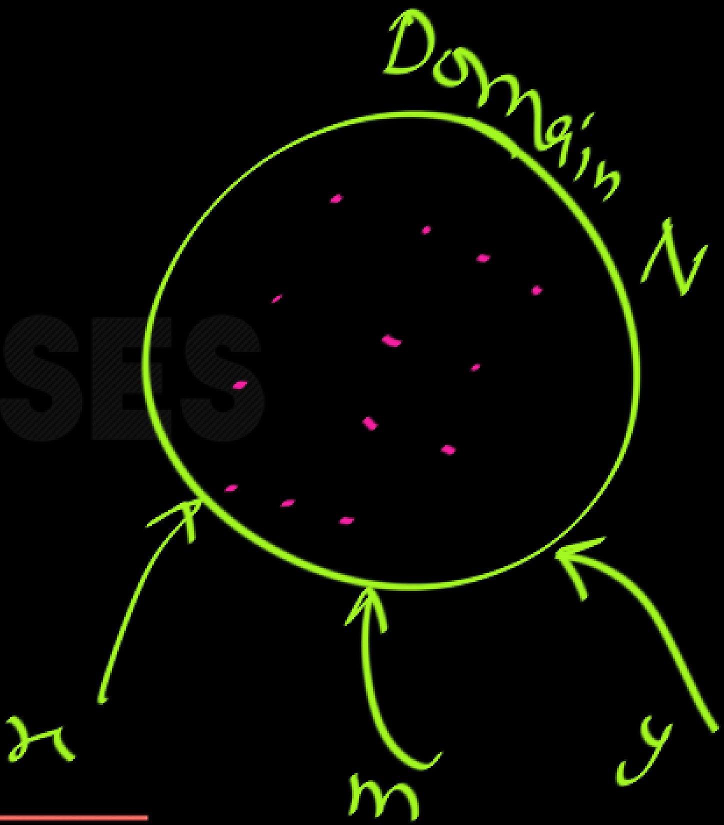
$\forall x \, E(x)$

same

$\forall m \, E(m)$

$\forall y \, E(y)$

Bounded Var



Domain: Set of all natural numbers. $E(x)$ = x is even.

$\forall x\, E(x)$: For every natural number x, x is even.

= Every natural number is even.

$\forall m\, E(m)$: For every natural number m, m is even.

= Every natural number is even.

$\forall y\, E(y)$: For every natural number y, y is even.

= Every natural number is even.

Domain: Set of all natural numbers. E(x) = x is even.

$\exists x\ E(x)$

$\exists m\ E(m)$

$\exists y\ E(y)$

Same

Domain: Set of all natural numbers. $E(x)$ = x is even.

$\exists x\, E(x)$: For some natural number x, x is even.

= Some natural number is even.

prop. True No variable

$\exists m\, E(m)$: For some natural number m, m is even.

= Some natural number is even.

$\exists y\, E(y)$: For some natural number y, y is even.

= Some natural number is even.

True

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

Boundes Var / Quantifiel var

NOTE :

There is No individual importance of Bounded Variable,
They are just placeholders.

$$\forall x\, P(x) \equiv \forall y\, P(y) \equiv \forall z\, P(z) \equiv \forall m\, P(m)$$

Dummy var. (placeholder)

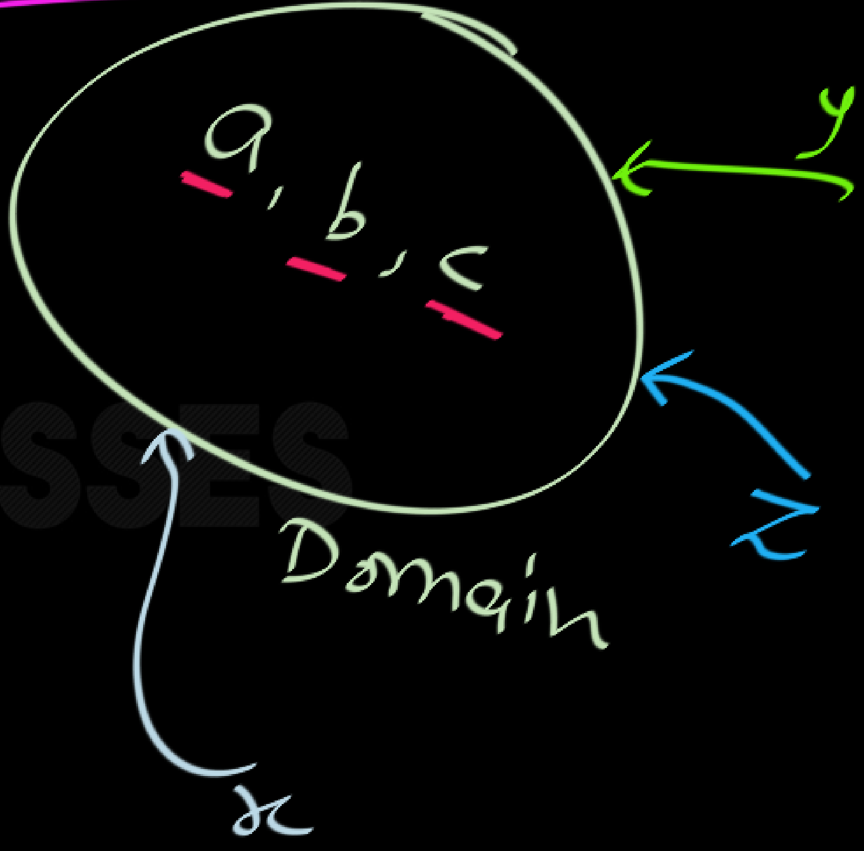
$N(x)$: x is Nice.

proposition

$$\boxed{N(a) \wedge N(b) \wedge N(c)}$$

$$\equiv \forall_{z} N(z)$$

$$\equiv \forall_{x} N(x) \equiv \forall_{y} N(y)$$



$\forall_x P(x)$ $\exists_y P(y)$

Quantified Variable / Bounded Var.

First Order Logic

Next Topic:

2. Free Variable

(Real Variable)

Domain: Set of all natural numbers.

E(x) = x is even. → Not a prop.
→ Predicate

free variable

N
1, 2, 3, 4 ...

x

E(2) : $\boxed{\text{2 is even.}}$ Proposition → True

E(7) : $\boxed{\text{7 is even}}$ false proposition

Domain: Set of all natural numbers. E(x) = x is even.

$\forall x\ E(x)$ → Bounded

$\exists x\ E(x)$ → Bounded

$E(x)$ → free

$\forall_3 E(3)$ Nonsense

$\exists_{100} E(100)$

$E(8)$: True

$E(101)$: False

N

1, 2, 3, ...

x

Domain: $\{a, b, c\}$; $N(x)$: x is Nice.

$N(x)$: $\boxed{x \text{ is nice.}}$ Not a proposition

free variable

Domain
a, b, c
Ram, John, Sonu

x

$N(a)$: Proposition
$N(b)$: "
$N(c)$: Prop.

Domain: $\{a, b, c\}$; $N(x)$: x is Nice.

$N(x)$: free var.

$\forall_x N(x)$: $N(a) \wedge N(b) \wedge N(c)$

Proposition

Domain

a, b, c

x

Bounded var.

Dummy var.

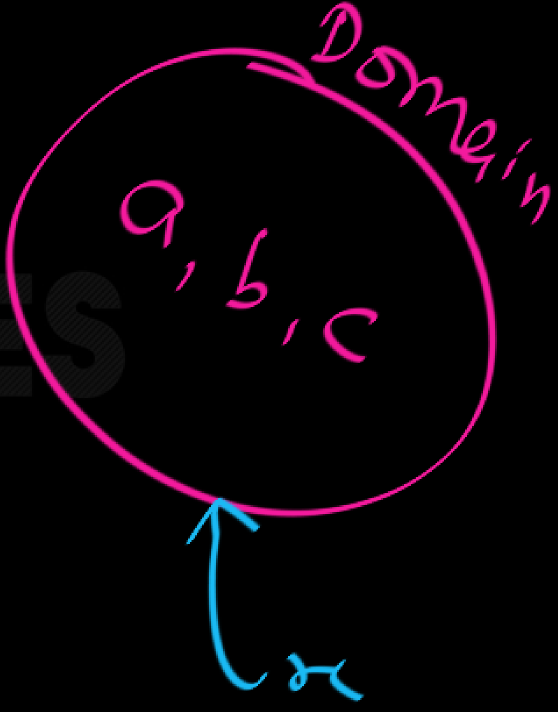
Domain: $\{a, b, c\}$; $N(x)$: x is Nice.

$N_{(x)}$: x is Nice

$x \rightarrow$ free Var.

$N_{(a)}$: prop.

$N_{(b)}$: prop.



Domain: $\{a, b, c\}$; $N(x)$: x is Nice.

$N(x)$: x is Nice

free var.

$\exists_x N(x) \equiv \boxed{N(a) \vee N(b) \vee N(c)}$

Bounded

prop.

Domain

a, b, c

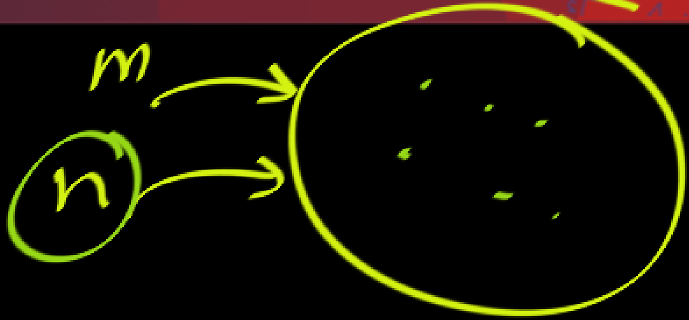
x

free variable : Not Bounded by Any Quantifier.

free to take any value from Domain.

Domain: Set of all integers.

$P(m,n)$: $m+n > 4$



$P(1,2)$: $\boxed{1+2>4} \rightarrow$ false

prop.

$P(9, -3)$: $\boxed{9-3>4} \rightarrow$ True.

prop.

$P(0,n)$: $\boxed{0+n>4}$: $\boxed{n>4}$ → Not a prop.

free var.

$\boxed{\forall n\, P(0,n)}$: $\boxed{\forall n\,(n>4)}$ → false

Bounded var.

$P(m,n)$

free var.

free var.

$\forall_n P(0,n)$ → prop.

$\equiv$ $\boxed{\text{every integer is } > 4.}$ → prop. → false

$P(0,n)$: $n > 4$: Not a prop.

→ free Var. (Real variable)

$\forall n \; P(o,n) \equiv$ Proposition

No free (Real) variable

Bounded var
(Dummy var)

NOTE:

If an expression contains NO Free Variable, then it is a Proposition.

How to create a Proposition/Statement from a Predicate?

Domain: Set of all integers.

P(m,n) : m+n > 4 : predicate

$\mathbb{Z}$: 1, 2, ... -1, -2, 0 ...

m

n

P(1,2) : $\boxed{1+2>4}$ prop.

P(9, -3) : $\boxed{9+(-3)>4}$ prop.

P(0,n) : $\boxed{n>4}$ Not a prop.

free

P(m,n) : $\boxed{m+n>4}$ → Not a prop.

free

free

$P(0, n)$: $n > 4$: $S(n)$ → free variable

Not a prop.

$\forall_n S(n)$: $\boxed{\text{every integer is } > 4.}$ false prop.

Bounded

No free var. So prop.

free

$P(o, n)$: $n > 4$; $S(n)$ → free variable

Not a prop.

$\exists n\, S(n)$: Some integer is > 4.

True Prop.

Bounded

No free var. → Prop.

Predicate $\xrightarrow{\text{make sure No free variable}}$ Proposition

① Replace free var with value from Domain

OR

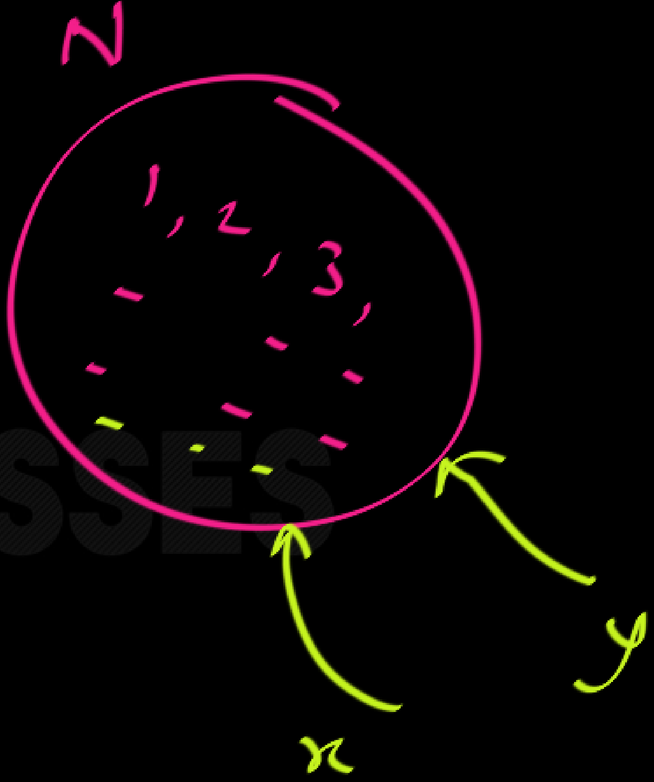
Quantify it.

Example: Domain: Set of all Natural Numbers

S : $\boxed{\forall x \, (x > y)}$

Bounded

free var.



Not a prop because there is a free var.

Example: Domain: Set of all Natural Numbers

$\boxed{\forall x\,(x > y)}$: $S(y)$

free

$y = 10$

$\boxed{\forall x\,(x > 10)}$: false

proposition

No free var.

N

1, 2, 3, ...

x

y

Example: Domain: Set of all Natural Numbers

$S(y)$

$\boxed{\forall x\,(x > y)}$

Quantify y.

~~prop.~~

N

$1, 2, 3, \ldots$

x y

Domain: N

$$\boxed{\forall_x (x > y)} = S(y)$$

Every Natural no. is $> y$.

$$\equiv S(y)$$

Domain: N = 1, 2, 3 ← y

$$\boxed{\forall_x (x > y)} = S(y) = \boxed{\text{every Natural no. is} > y.}$$

$$\forall_y S(y) : S(1) \wedge S(2) \wedge S(3) \wedge \ldots\ldots$$

$S(1) \to$ false, $S(2) \to$ false, $S(3) \to$ false

$\forall_y S(y) \to$ false

No free var ⇒ proposition.

$1 > 1$: false

$1 \geq 1$: True

Domain: $N = \{1, 2, 3, \ldots\}$ ← y

free

$\boxed{\forall_x (x > y)} = S(y) = \boxed{\text{every Natural no. is} > y.}$

$\exists_y S(y) : S(1) \lor S(2) \lor S(3) \lor \ldots\ldots\ldots$

false false false = false

Bounded

No free var ⟶ Proposition

$\boxed{\text{No free Variable}} \longrightarrow$ Proposition

How to create a Proposition/Statement from a Predicate?

Ans: Keep No Variable Free.

So,

Replace ALL free variables by some value from the domain

OR

Quantify Free Variables.

How to create a Proposition/Statement from a Predicate?

A. Every variable must be either bounded OR replaced with some value from the domain.
B. Every variable must be either free OR replaced with some value from the domain.
C. There should be No Free variable
D. There should be No Bounded variable

How to create a Proposition/Statement from a Predicate?

A. Every variable must be either bounded OR replaced with some value from the domain.

B. Every variable must be either free OR replaced with some value from the domain.

C. There should be No Free variable

D. There should be No Bounded variable

Domain: $\mathbb{Z}$

free

$\boxed{\text{Even}(x) : x \text{ is even.}}$ Not a prop.

Even(−9) : false — prop.

$\exists_x \text{Even}(x)$: True — prop.

Bounded var

Not free

Bounded

$\forall_x \text{Even}(x)$

prop.

false

NOTE 1:

If there is a Free variable in any expression, then that expression is NOT a proposition/statement.

$\boxed{\text{No free variable} \longleftrightarrow \text{prop.}}$

Domain: Set of all Integers.

Convert the following into a Proposition.

$S(y) : \exists x \, (x > y)$

Domain: Set of all Integers.

Convert the following into a Proposition.

$S(y) : \boxed{\exists x\,(x > y)} = S(y)$

free

Bounded

free

NOT a proposition

$\mathbb{Z}$

x

y

$S(y) : \boxed{\exists x\,(x > y)}$

Domain: Set of all Integers.

Convert the following into a Proposition.

$S(y) : \exists x\ (x > y)$ → Proposition

Replace free var y with a value from Domain Z.

$S(y) : \boxed{\exists_x (x > y)}$ True

$S(-10) : \boxed{\exists_x (x > -10)}$ True

→ No free variable → Proposition

Bounded

FREE

$S(y)$: $\boxed{\exists x\,(x > y)}$

Quantifies var.

Free var.

Quantify free var. y

Proposition ✓

$S(y) : \boxed{\exists x (x > y)}$

$\boxed{\forall y \, S(y)}$ = True / false ?

prop.

Bounded

No free var. → proposition

$S(y)$: $\boxed{\exists x \, (x > y)}$

prop.

$\boxed{\forall y \, S(y)}$ = True / false ?

Bounded

No free Var. ⟶ proposition

$S(y)$: $\boxed{\exists x\,(x > y)}$ → Some integer is $> y$.

$\boxed{\forall y\, S(y)}$: $S(0) \wedge S(1) \wedge S(2) \ldots \wedge S(-1) \wedge S(-2) \ldots$ = True

T T T T

Bounded

No free var. → proposition

$S(y)$: Some integer is $> y$.

$S(0)$: " " " > 0. True

$S(-10)$: " " " > -10. True

$S(10)$: " " " > 10. True

FREE

$S(y)$: $\boxed{\exists x\,(x > y)}$ → NOT a prop.

$\boxed{\exists y\, S(y)}$ = True / false ?

Bounded var.

No free Var. → proposition

FREE

$S(y)$: $\boxed{\exists x (x > y)}$ → NOT a prop.

$\boxed{\exists y\, S(y)}$ = True / false ?

Bounded var.

No free var. → proposition

FREE

$S(y)$: $\boxed{\exists x (x > y)}$ → NOT a prop.

Some integer is $> y$.

$S(y)$

True

$\boxed{\exists y\, S(y)} = S(0) \vee S(1) \vee S(2) \vee \ldots \vee S(-1) \vee S(-2) \vee \ldots$ = True

True

Bounded Var.

No free Var. → proposition

Domain: Set of all Integers.

Convert the following into a Proposition.

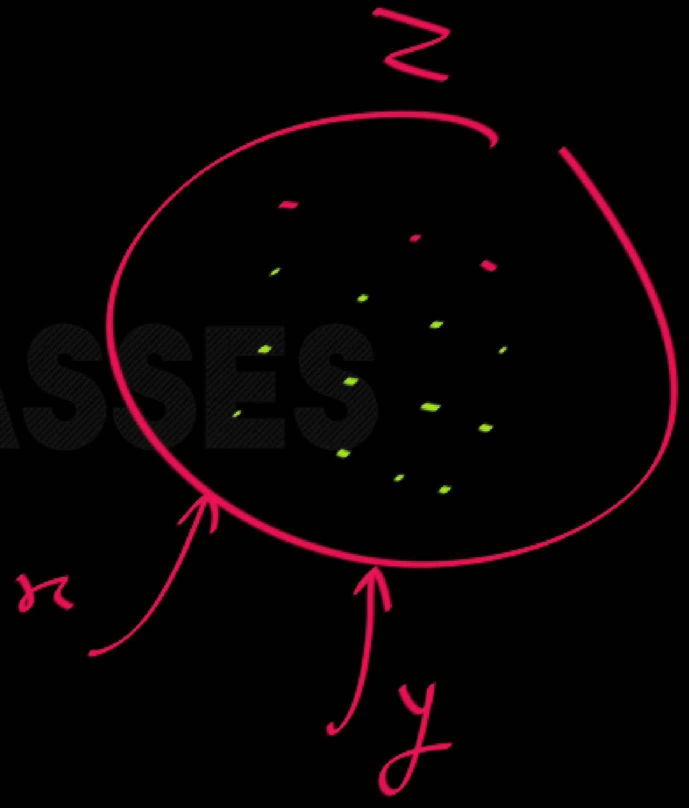
$S(y) : \boxed{\exists x\ (x < y)} = S(y)$

Bounded var.

free

Conversion

Proposition?



Domain: Set of all Integers.

Convert the following into a Proposition.

$S(y) : \boxed{\exists x\,(x < y)}$ = $\boxed{\text{Some integer is } < y.}$ = $S(y)$

y : Free var.

Replace free var y with some value from Domain

proposition

Domain: Set of all Integers.

Convert the following into a Proposition.

$S(y) : \exists x\ (x < y)$

No Free var. → prop.

$S(0) : \boxed{\exists x\ (x < 0)}$: True ✓

$S(10) : \boxed{\exists x\ (x < 10)}$: True ✓ → Proposition

$S(-10) : \exists x\ (x < -10)$: True ✓

Domain: Set of all Integers.

Convert the following into a Proposition.

S(y) : $\boxed{\exists x\ (x < y)}$ free y

Quantify the free var. y using $\forall y$ OR $\exists y$

Proposition

Domain: Set of all Integers.

Convert the following into a Proposition.

$S(y) : \exists x \, (x < y)$

$\boxed{\exists_y S(y)}$: True/f ?

Bounded

No free var. → Proposition

Domain: Set of all Integers.

Convert the following into a Proposition.

$S(y)$: $\boxed{\exists x\ (x < y)}$: $\boxed{\text{Some integer is } < y.}$

$\exists_y S(y)$ = True

$\rightarrow S(0) \vee S(1) \vee S(2) \vee \cdots$

$\vee S(-1) \vee S(-2) \vee \cdots$

True

True

Domain: Set of all Integers.

Convert the following into a Proposition.

$S(y) : \exists x \, (x < y)$

$\boxed{\forall y \, S(y)}$: True / false]

No free var. → prop.

Bounded

Domain: Set of all Integers.

Convert the following into a Proposition.

$S(y) : \boxed{\exists x\,(x < y)}$; Some int is $< y$.

$\boxed{\forall y\, S(y)}$: $S(0) \wedge S(1) \wedge S(2) \wedge \ldots \wedge S(-1) \wedge S(-2) \wedge \ldots$

T T T T T

True

Domain: Set of all Integers.

Convert the following into a Proposition.

$S(y) : \boxed{\forall x\,(x < y)}$ — every integer is < y. — free

free var y

Bounded

Not a prop

Domain: Set of all Integers.

Convert the following into a Proposition.

$S(y) : \forall x\,(x < y)$

No free var. → prop.

$S(10) : \boxed{\forall x\,(x < 10)}$: False

$S(-10) : \boxed{\forall x\,(x < -10)}$: False → prop.

Domain: Set of all Integers.

Convert the following into a Proposition.

$S(y)$: $\boxed{\forall x\,(x < y)}$: $\boxed{\text{every int is} < y.}$

$\boxed{\forall y\, S(y)}$: $\underbrace{S(0) \wedge S(1) \wedge S(2) \wedge \cdots \wedge S(-1) \wedge S(-2) \wedge \cdots}_{\text{false}}$

F F

Bounded

No free Var → prop.

Domain: Set of all Integers.

Convert the following into a Proposition.

$S(y)$: $\boxed{\forall x\ (x < y)}$: every int is < y.

$\exists y\ S(y)$: false

Bound vs. Free Variables

GO Classes

A **bound variable** is a variable that is subject to a quantifier. A variable that is not bound is called **free**.

$S(y)$

Example: in the expression $\boxed{\forall x P(x, y)}$, x is bound, y is free. Due to the free variable, $\forall x P(x, y)$ is not a proposition, but a propositional function of y. <u>A proposition can only contain bound variables, no free variables.</u>

- A variable is free if it is not referred to by any quantifier.

- A variable is bound if it is referred to by a quantifier.

Bounded Variable = Quantified Variable

Free Variable = Non-Quantified Variable

MOST Important NOTE:

MOST Important NOTE:

expression

If "A" doesn't have any Free Variable x then

$$\forall_x A \equiv A$$

$$\exists_x A \equiv A$$

MOST Important NOTE:

If "A" doesn't have any Free Variable x then

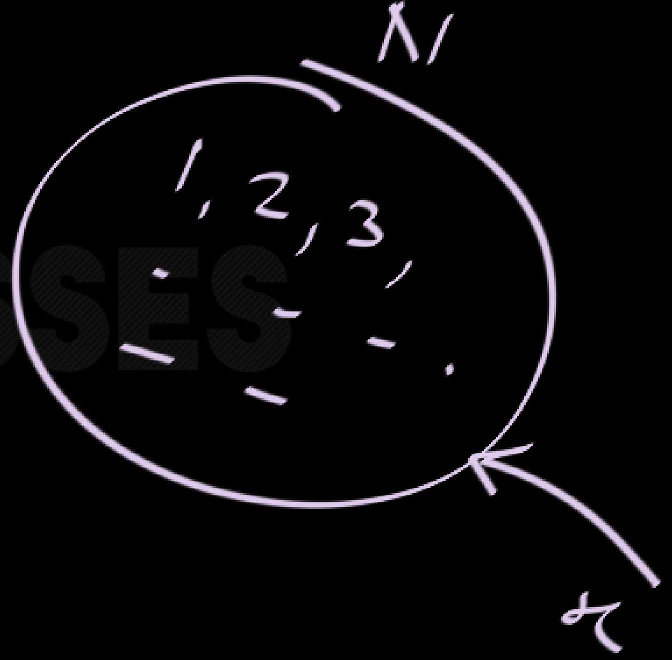
$$\forall_x A \equiv A$$

$$\exists_x A \equiv A$$

Eg: Domain: N

A : $\boxed{2+2=5}$ → No free x

→ false



$\forall_x A = ?$

$\exists_x A = ?$

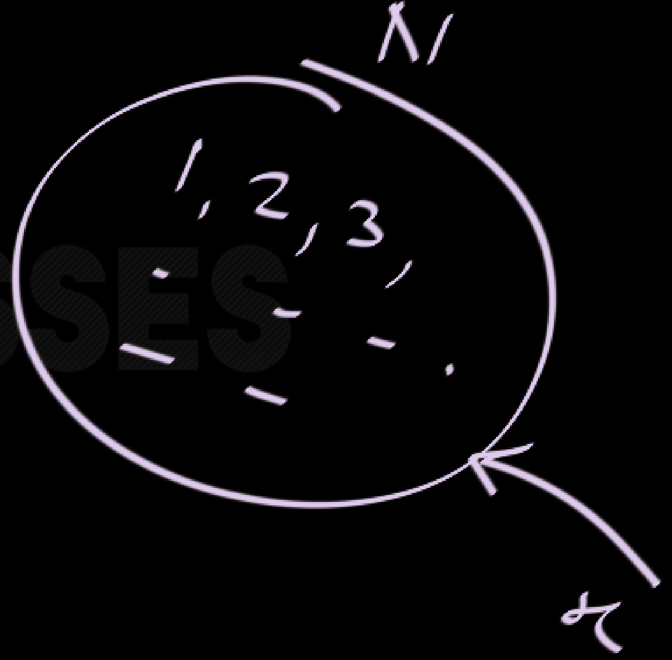
Eg: Domain: N

$A : \boxed{2+2=5}$ → No free x

→ false

$\boxed{\forall_x A = \text{false}}$

$\forall_x A \equiv A = \text{false}$



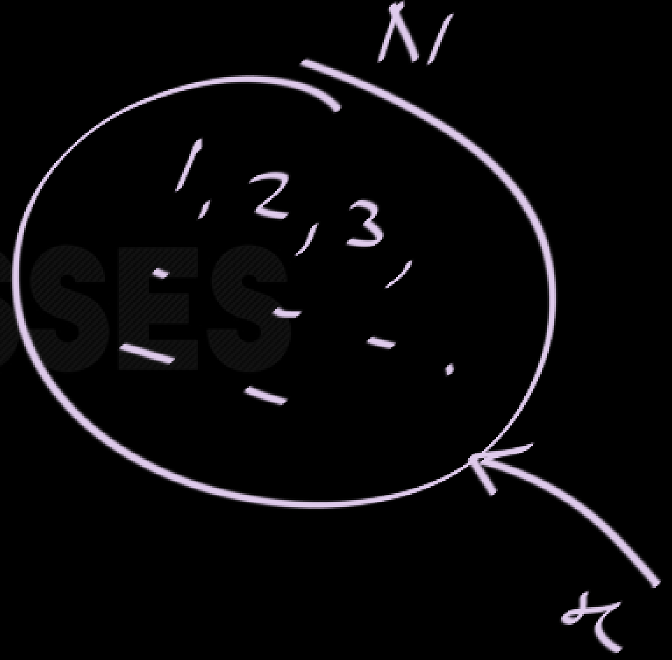
Eg: Domain: N

A : $\boxed{2+2=5}$ No free x

false

$\boxed{\exists_x A = \text{false}}$ ✓

$\exists_x A \equiv A = \text{false}$ ✓



Eg: Domain: $\mathbb{N}$

A: $\boxed{2 + 2 = 4}$ → No free x

→ True

$\underbrace{\forall x\, A}$ $\equiv A =$ True

$\underbrace{\exists x\, A}$ $\equiv A =$ True

$\forall x\, A =$ True

$\exists x\, A =$ True

$\mathbb{N}$

1, 2, 3, ...

If "A" with No free Var x

$$\boxed{\begin{aligned} \forall_x A &\equiv A \\ \exists_x A &= A \end{aligned}}$$

Domain : N

$P(x)$: x is prime.

$\boxed{\forall_x P(x)}$: A

Proposition

false

No free var x

$$\boxed{\begin{array}{l} \forall_x A \equiv A \\ \exists_x A \equiv A \end{array}}$$

N

1, 2, 3 ...

x

Domain: N

P(x) : x is prime.

$\boxed{\forall_x P(x)}$: A → false

Proposition

No free var x

$\forall_x A \equiv \forall_x (\text{false}) = \text{false} = A$

$\exists_x A \equiv \exists_x (\text{false}) = \text{false} = A$

N

1, 2, 3 ...

x

Domain: $\mathbb{N}$

$P(x)$: x is prime.

$\boxed{\forall_x P(x)}$: A → Proposition → false

No free var x

$$\boxed{\begin{aligned} \forall_x A &\equiv A \\ \exists_x A &\equiv A \end{aligned}}$$

$\forall_x (A) \equiv \forall_x (\boxed{\forall_x P(x)})$ false

$\equiv \forall_x(\text{false}) = \text{false}$

$\mathbb{N}$

1, 2, 3 ... x

Domain: $\mathbb{N}$

$P(x)$: x is prime.

$\boxed{\exists_x P(x)}$: A → Proposition → True

No free x

$$\boxed{\begin{array}{c} \forall_x A \equiv A \\ \exists_x A \equiv A \end{array}}$$

$\forall_x A \equiv$ True

$\exists_x A \equiv A \equiv$ True

Domain: $\mathbb{N}$

$P(x)$: x is prime.

$\boxed{\exists_x P(x)}$: A

Any Expression "A" with No free Var. x

then

$$\boxed{\begin{array}{l} \forall_x A = A \\ \exists_x A = A \end{array}}$$

Any Expression "A" with No free Var. y

then

$$\forall_y A = A$$

$$\exists_y A = A$$

Any Expression "A" with No free Var. z

then

$$\forall_z A = A$$

$$\exists_z A = A$$

Domain: $\mathbb{N}$

$P(x)$: $\boxed{x \text{ is prime.}}$ → free var. x

$A(x)$

$\forall_x A(x) \neq A(x)$

Prop. — false

Not a prop.

Domain: $\mathbb{Z}$

free var x

$E(x)$: $\boxed{x \text{ is even.}}$: $A(x)$

$\exists x\, A(x) \not\equiv A(x)$

prop. → True

Not even a prop.

B → $\boxed{\text{No free var. } x}$

B : $\boxed{\exists x\, A(x)}$: A proposition

→ No free var x

$$\boxed{\begin{array}{l} \forall x\, B \equiv B \\ \\ \exists x\, B \equiv B \end{array}}$$

Any Expression "A" with No free variable m.

then

$$\forall_m A \equiv A$$

$$\exists_m A \equiv A$$

Next Topic:

Scope of a Quantifier

(Scope of a Quantified Variable)

Scope of a Quantifier:

Area

The part of a logical expression to which a quantifier is applied is called the scope of this quantifier.

Area

$$\forall x \underbrace{\left[\ \ldots\ x\ \ldots\ x\ \ldots\ \right]}_{\text{Scope of } \forall x}$$

$\exists x \, [\text{- - } x \text{ - - - - } x \text{ - - - -}]$ → Scope of $\exists x$

$\forall x \, (\text{- } x \text{ - - } x \text{ - - - } x \text{ -})$ → Scope of $\forall x$

Scope of $\forall x$

$$\forall x[\overbrace{Hx \wedge Px}]$$

$$\forall x[Hx] \wedge Px$$

Scope of $\forall x$

The scope of a quantifier in a formula P is the quantifier itself and the subformula that immediately follows the quantifier.

$$\exists y[Py \vee Hx]$$

$$\exists y[Fy \wedge \forall x[Fx \rightarrow Gxy]]$$

$$\exists y[Fy \wedge \forall x[Fx]] \wedge Ly$$

The scope of a quantifier in a formula P is the quantifier itself and the subformula that immediately follows the quantifier.

$$\underbrace{\exists y[Py \vee Hx]}_{\text{scope}}$$

free

this y is Bounded by $\exists y$.

$$\exists y[Fy \wedge \forall x[Fx \rightarrow Gxy]]$$

scope

scope

Scope

$$\exists y[Fy \wedge \forall x[Fx]] \wedge Ly$$

free

$\forall x \underbrace{(\qquad\qquad)}$ → Scope of $\forall x$

$\underline{(\forall x)}\ \underbrace{(\qquad)}$

Scope

$\forall x, \underline{\qquad}$ → Scope of $\forall x$

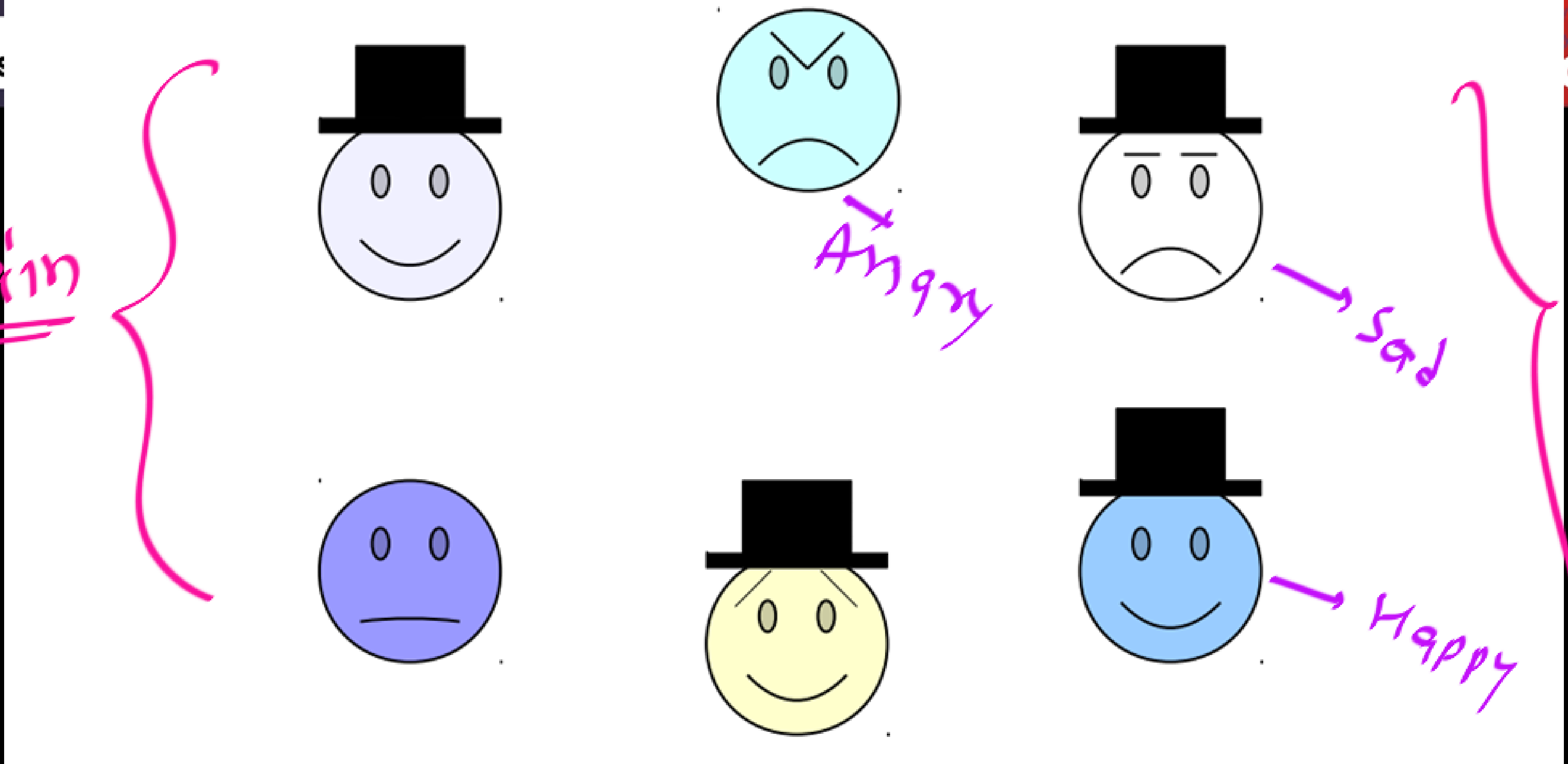
$\exists x \; (\ldots x \ldots x \ldots) \wedge P(x)$

free

Scope (Area) of $\exists x$

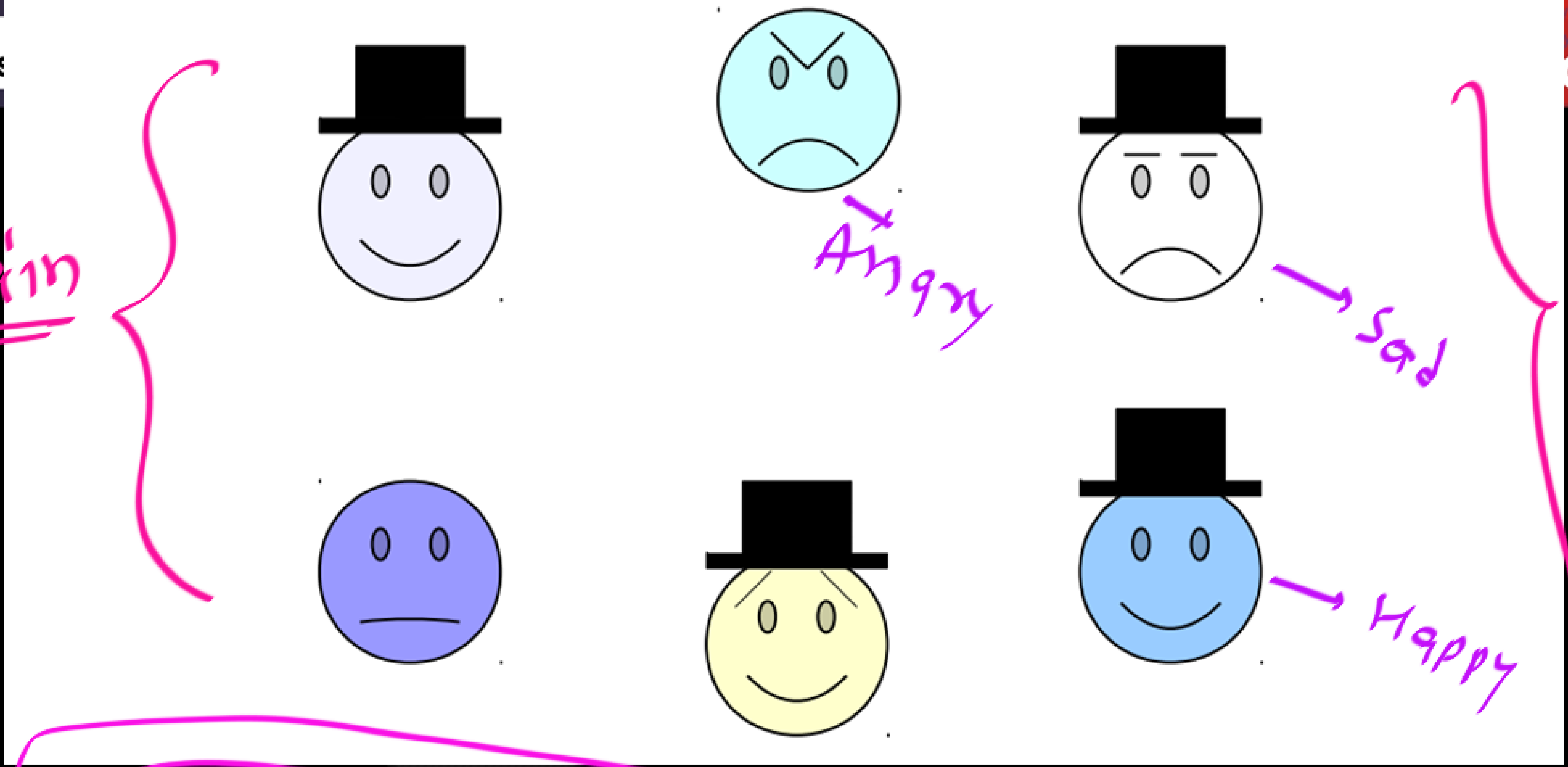
Scope Plays a Very Important Role, in the Meaning(interpretation) of an Expression:

Domain



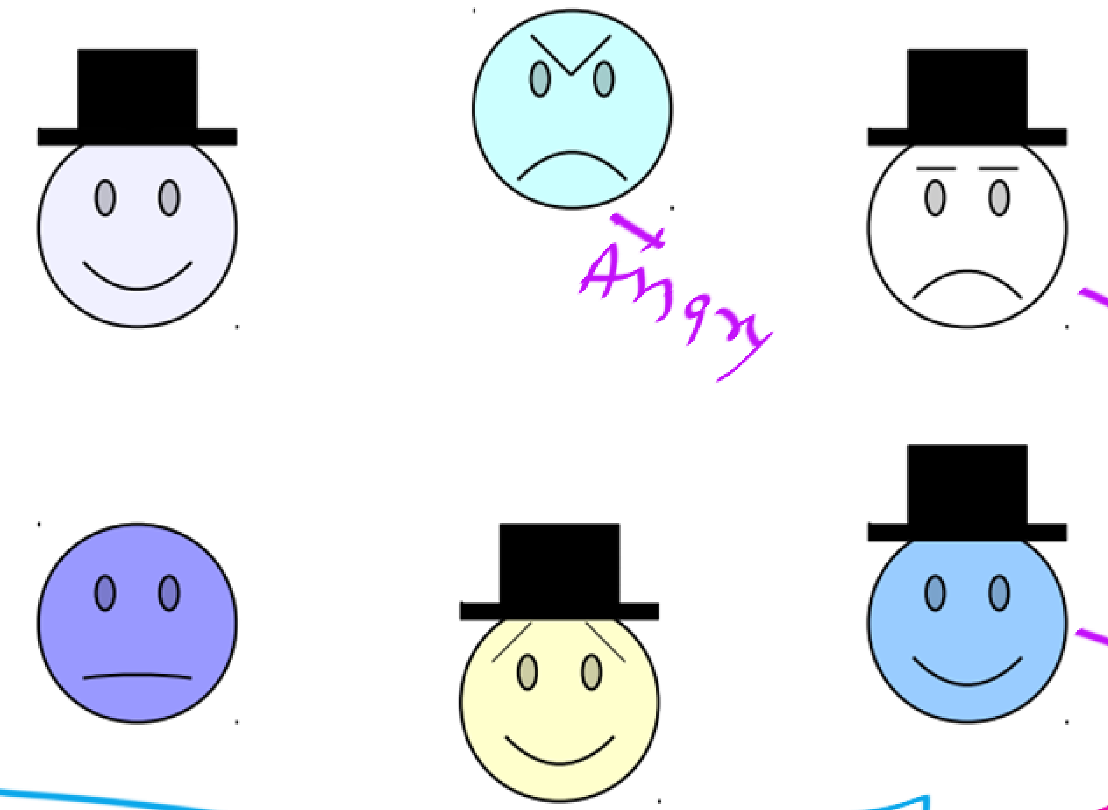
$\forall x\,(\text{Smile}(x) \rightarrow \text{wearHat}(x)) =$ True or false?

Domain



$$\forall x \left(\underbrace{Smile(x) \rightarrow WearHat(x)}_{\text{Scope}} \right)$$

Domain



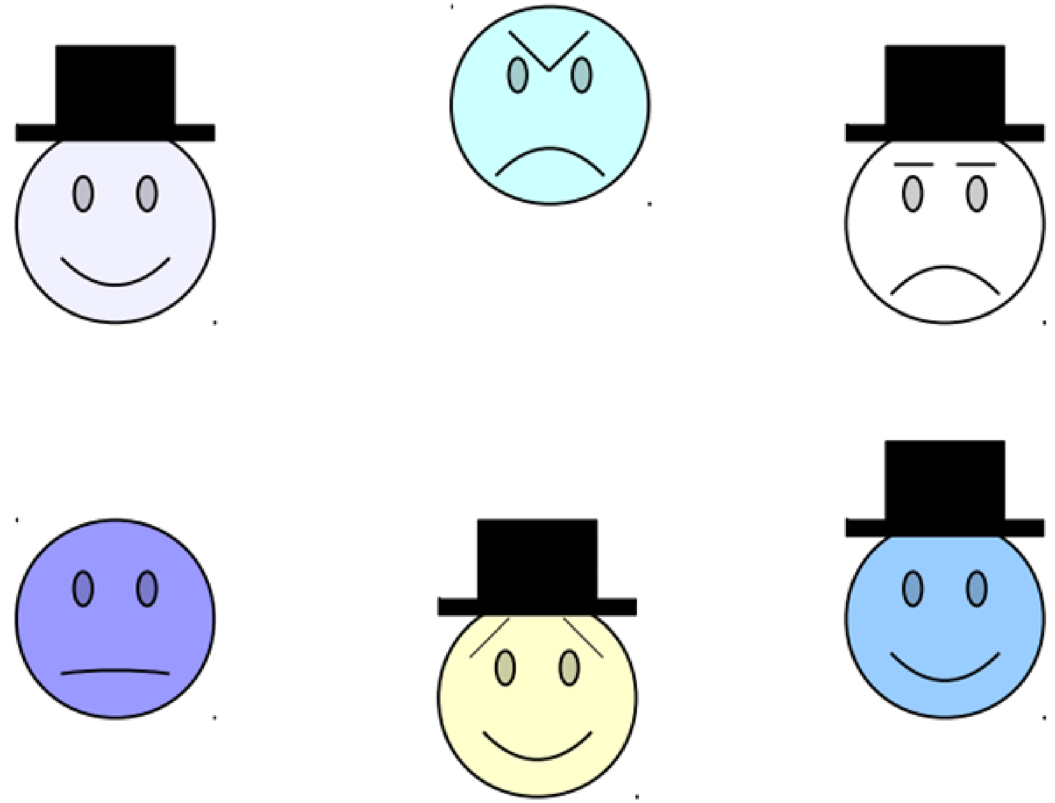
True

True

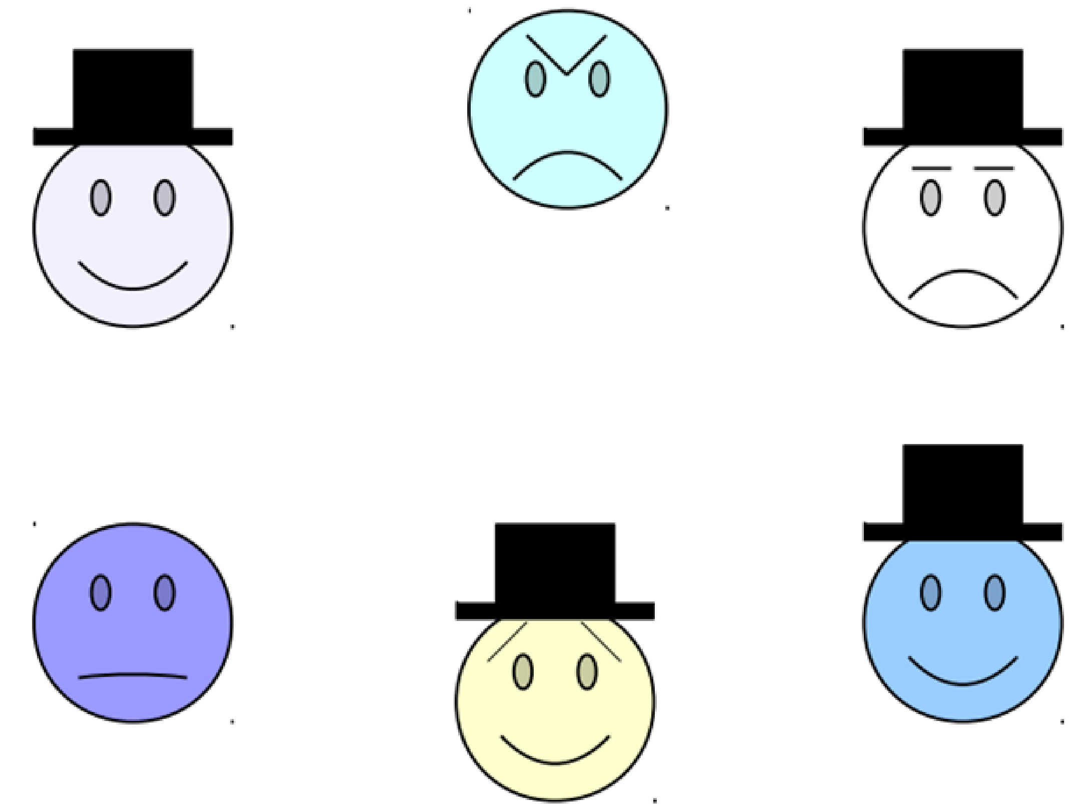
$$\forall x \, (\text{Smile}(x) \rightarrow \text{wearHat}(x))$$

Every smiling person is wearing Hat.

GO Classes



$$\boxed{\forall x\,(\text{Smile}(x)) \rightarrow \forall x\,(\text{WearHat}(x))}$$: True/false?



Scope

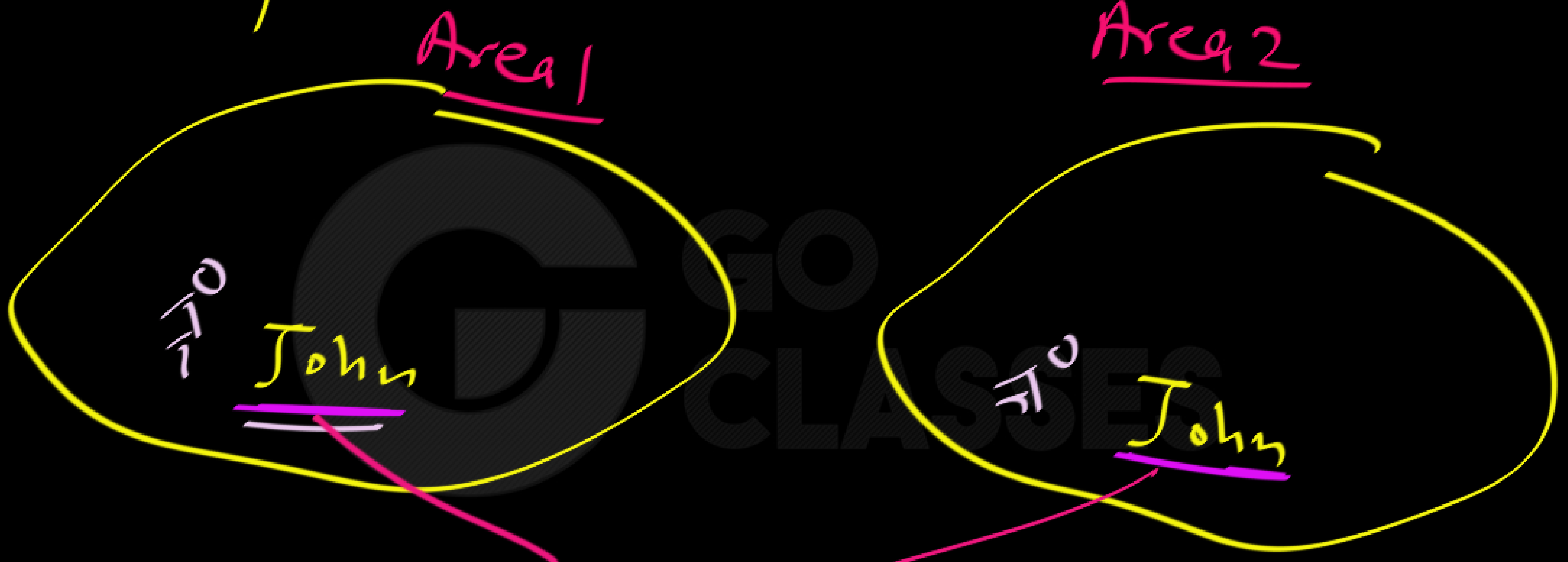
$$\forall x\,(\text{Smile}(x)) \rightarrow \forall x\,(\text{WearHat}(x))$$

Scope of $\forall x$

Scope

Scope of 2^{nd} $\forall x$

Analogy :



Nothing to do with each other.

$$\forall_x \left(P(x) \right) \longrightarrow \forall_x \left(Q(x) \right)$$

Scope

Scope

Nothing to do with each other

$\forall x \left(P(x) \rightarrow Q(x) \right)$

Scope of $\forall x$

Same

a, b, c

$$\forall x \left(P(x) \rightarrow Q(x) \right)$$

Scope

$a : \quad P(a) \rightarrow Q(a)$

$b : \quad P(b) \rightarrow Q(b)$

$c : \quad P(c) \rightarrow Q(c)$

a, b, c

$$\underbrace{\left(\forall_x \left(P(x)\right)\right)}_{\alpha} \rightarrow \underbrace{\left(\forall_x \left(Q(x)\right)\right)}_{\beta}$$

Scope

$\alpha \rightarrow \beta$

$a : P(a) \wedge$

$b : P(b) \wedge$

$c : P(c)$

$a : Q(a) \wedge$

$b : Q(b) \wedge$

$c : Q(c)$

Domain

a, b, c

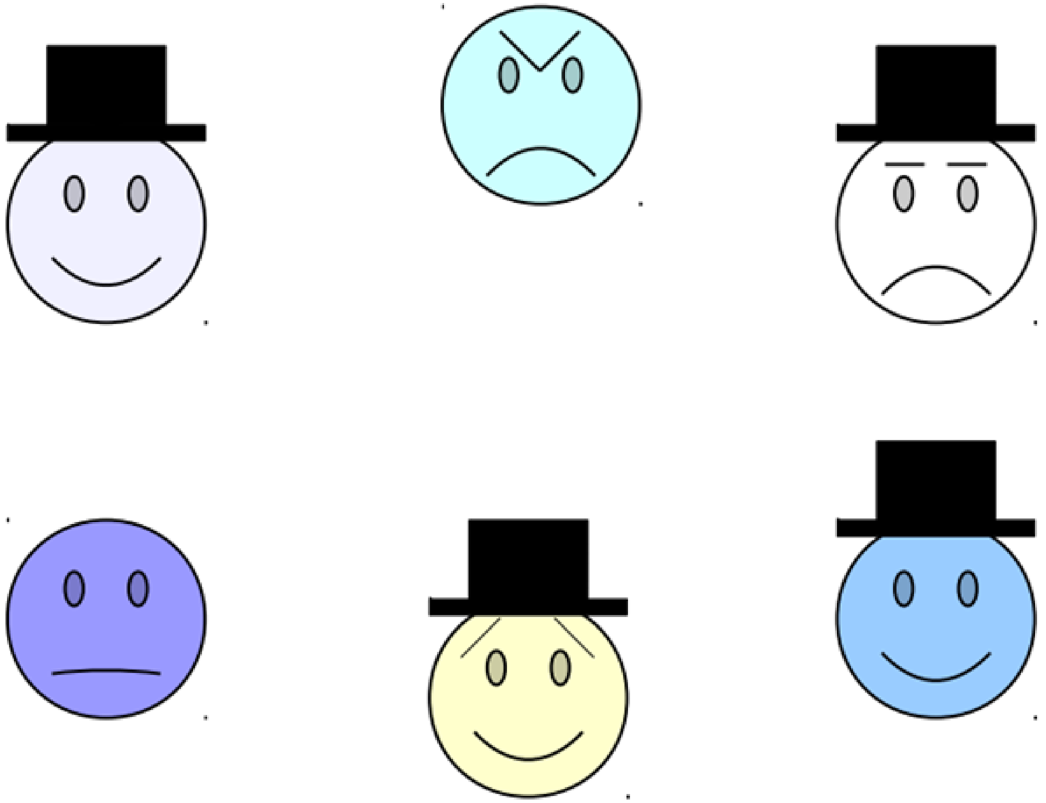
$$\forall x \left(P(x) \rightarrow Q(x) \right)$$

a, b, c

Domain

$$\underbrace{\forall x \left(P(x) \right)}_{\alpha} \longrightarrow \underbrace{\forall x \left(Q(x) \right)}_{\beta}$$

false

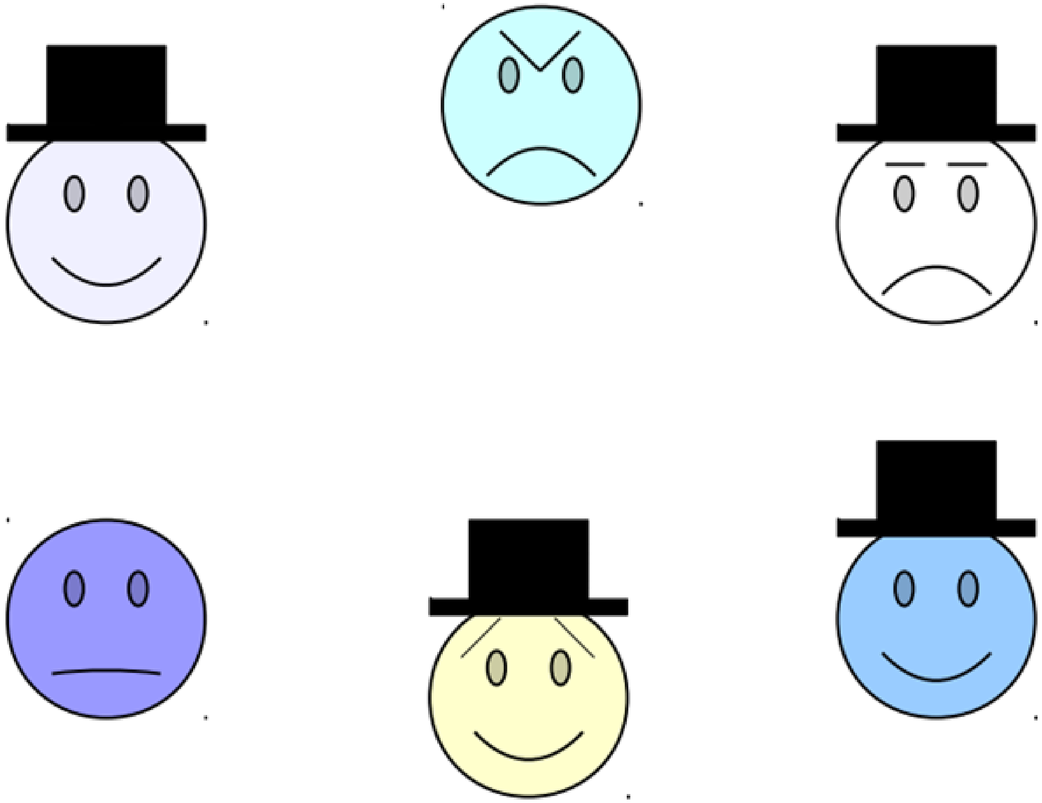


false

β

α

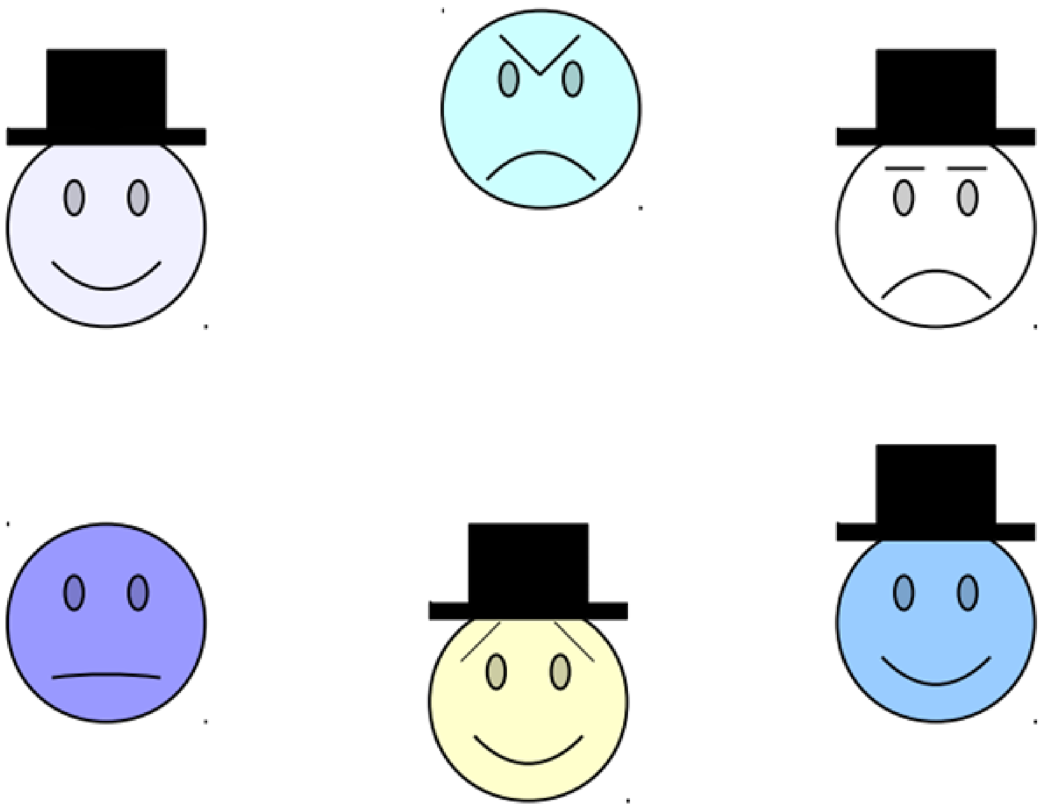
$$\forall x\,(\text{Smile}(x)) \rightarrow \forall x\,(\text{WearHat}(x))$$



$$\underbrace{\forall x\,(\text{Smile}(x))}_{F} \rightarrow \underbrace{\forall x\,(\text{WearHat}(x))}_{F} \equiv \text{True}$$

GO CLASSES

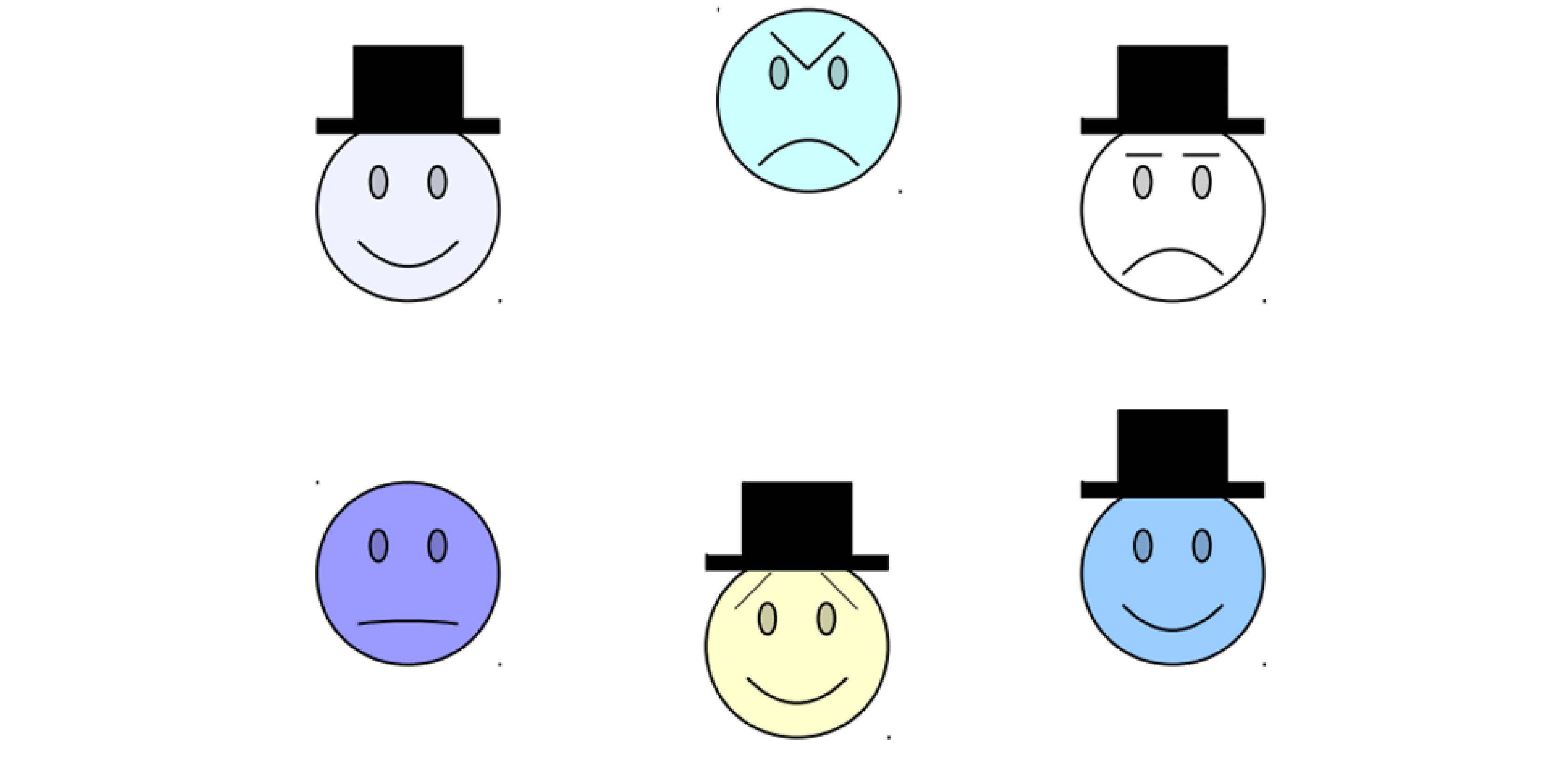
GO Classes



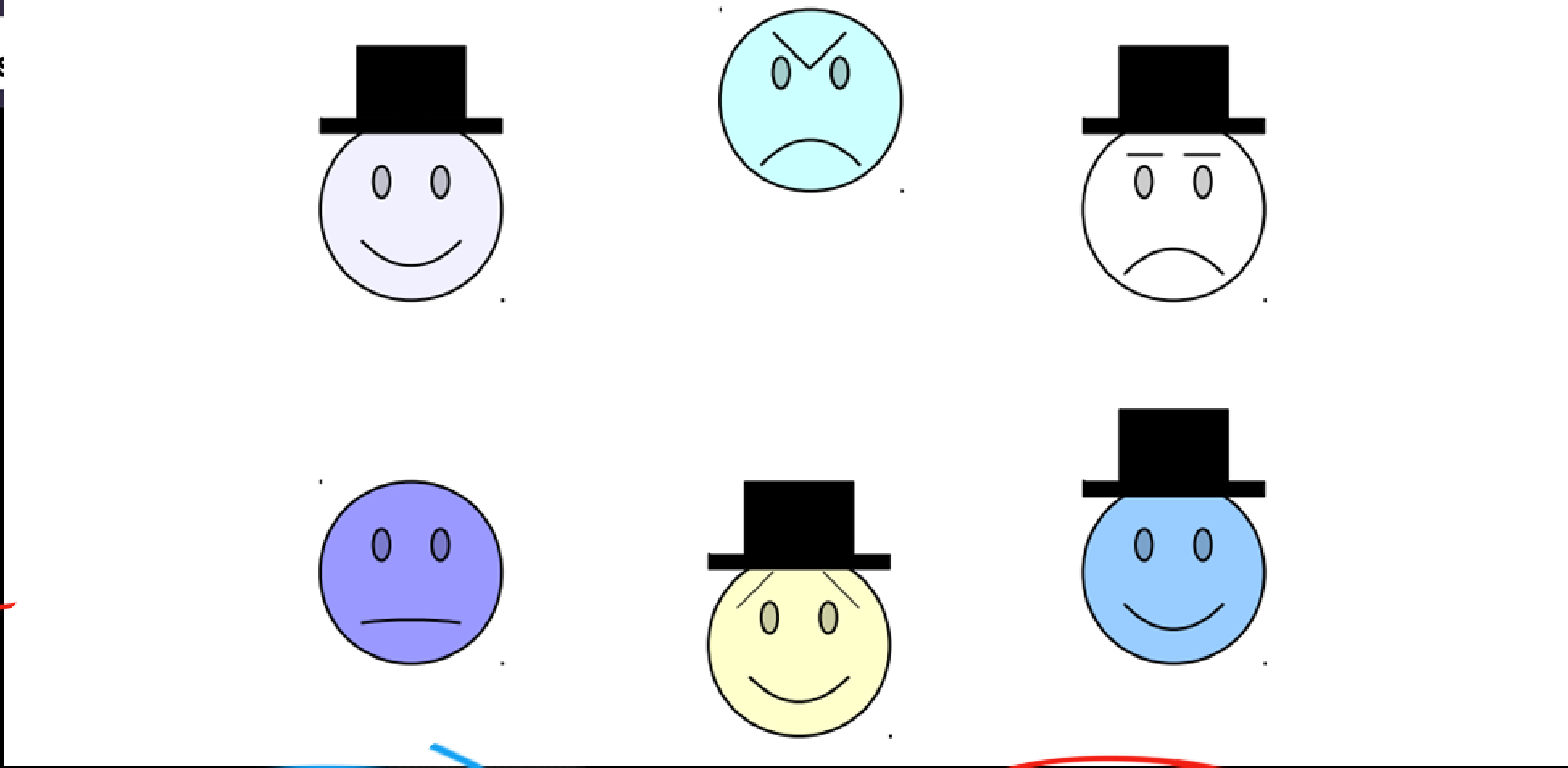
$\alpha = F$

$\beta = F$

$\forall x\, (\text{Smile}(x)) \rightarrow \forall x\, (\text{WearHat}(x))$



$$\forall x\,(\text{Smile}(x)) \rightarrow \forall y\,(\text{WearHat}(y))$$



false

α

$$\forall x\,(\text{Smile}(x)) \rightarrow \forall y\,(\text{WearHat}(y))$$

β

false

: True

$$\forall_x P(x) \equiv \forall_y P(y) \equiv \forall_z P(z)$$

Bounded Var

(Dummy Var)

$$\equiv \forall_m P(m)$$

① $\forall x\,(P(x)) \rightarrow \forall x\,(Q(x))$

$\equiv$ ② $\forall x\,(P(x)) \rightarrow \forall y\,(Q(y))$

Same

② is less Confusing.

$① \equiv ②$

Domain: {a,b,c}

$\exists_x \big(P(x) \wedge Q(x) \big)$

Scope

Domain

a, b, c

a : $\boxed{P(a) \wedge Q(a)}$ ✓

b : $\boxed{P(b) \wedge Q(b)}$ ✓

c : $\boxed{P(c) \wedge Q(c)}$

Domain: {a,b,c}

$\exists x\,(P(x) \wedge Q(x))$

Same

Domain

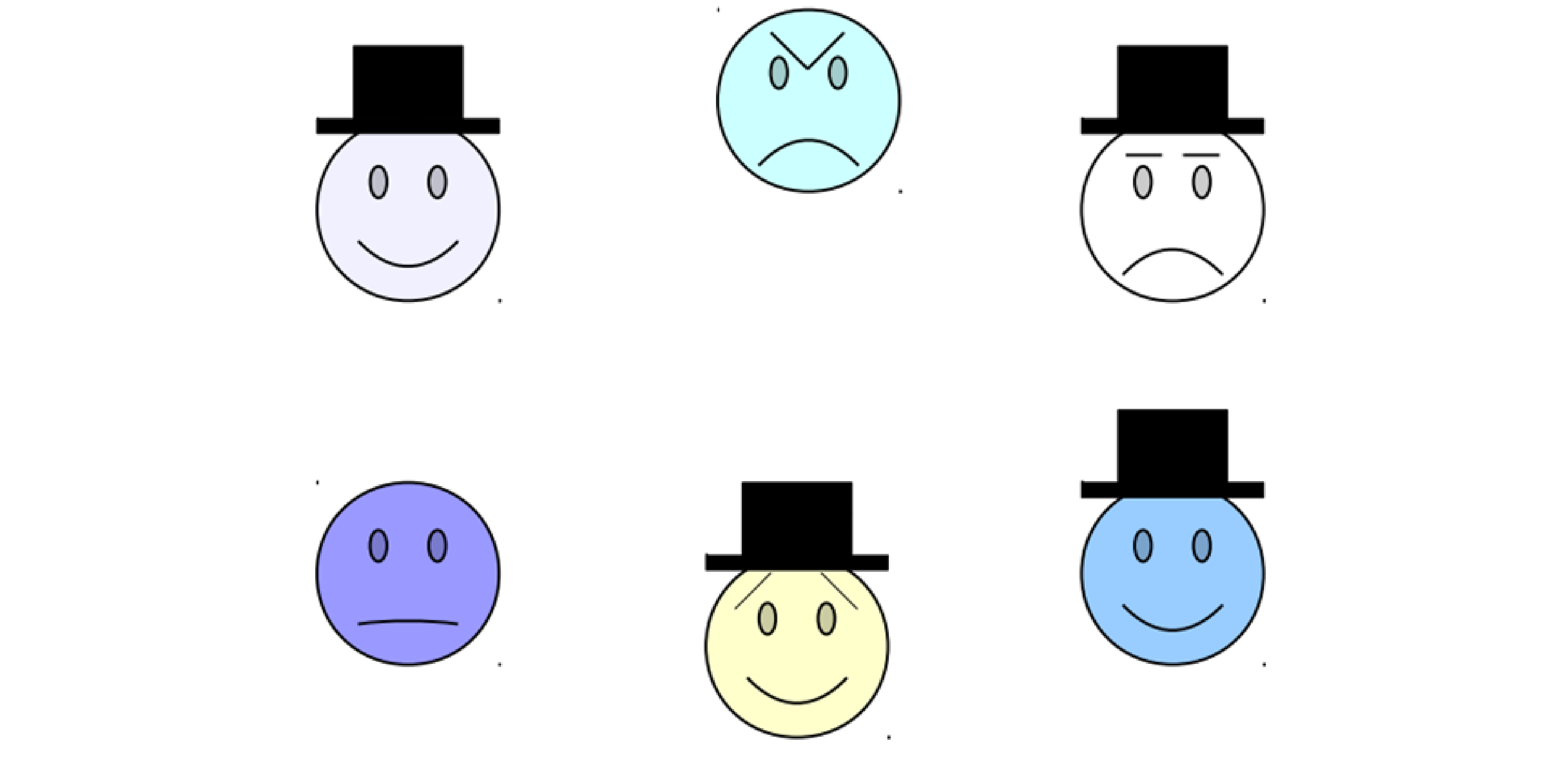
a, b, c

$P(a) \wedge Q(b)$

$P(b) \wedge Q(c)$

$P(a) \wedge Q(c)$

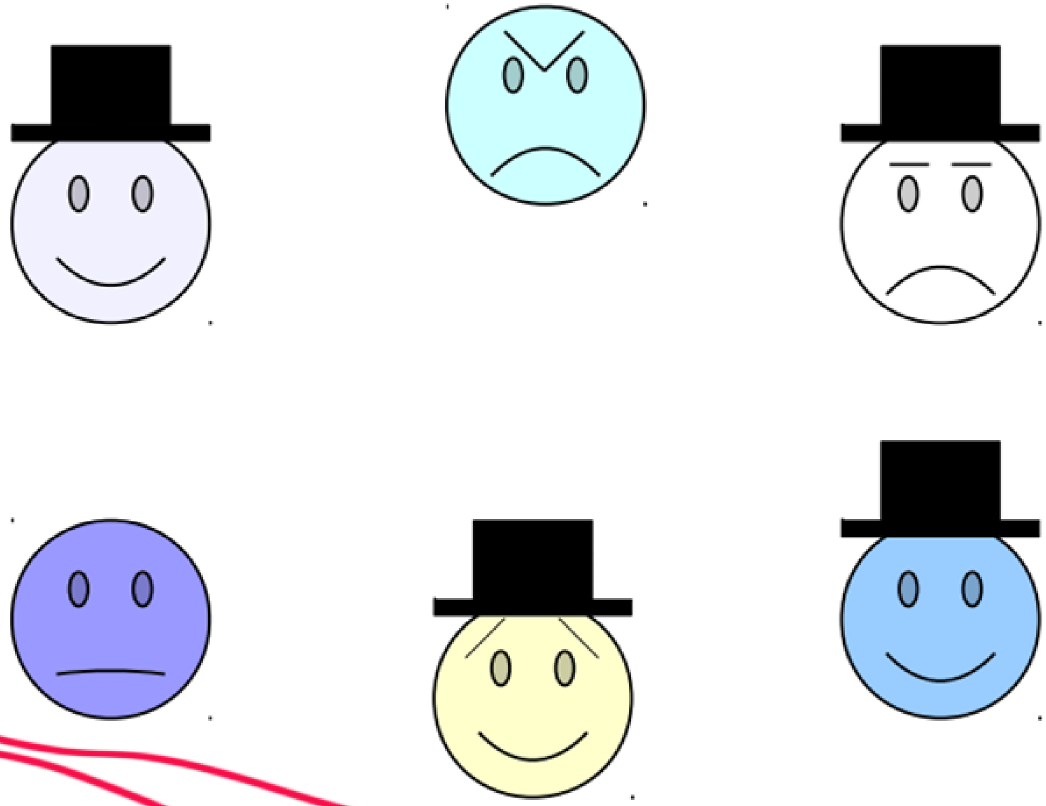
WRONG



$$\exists x\,(\text{Smile}(x) \land \text{WearHat}(x))$$

GO CLASSES

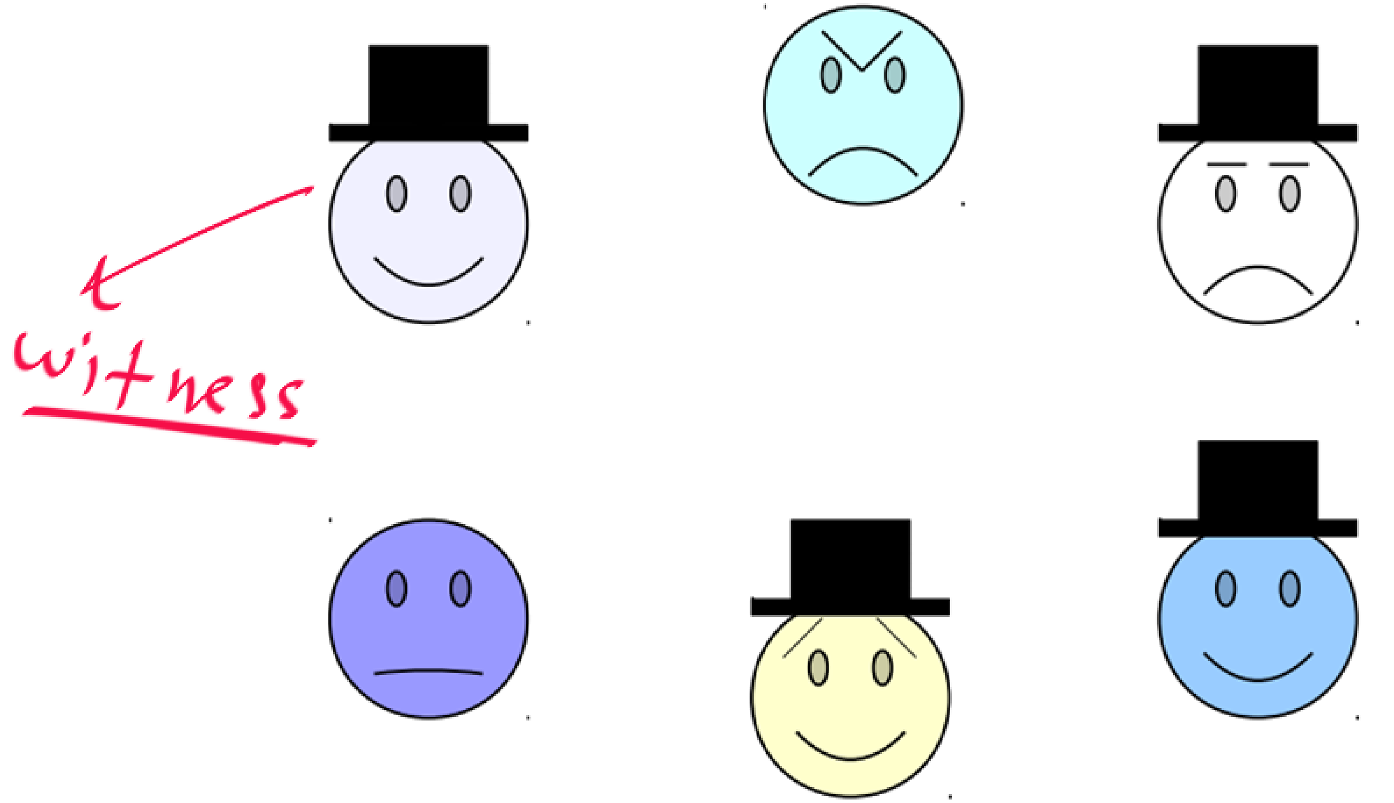
GO Classes



$$\exists x\,(\underbrace{Smile(x) \wedge WearHat(x)})$$

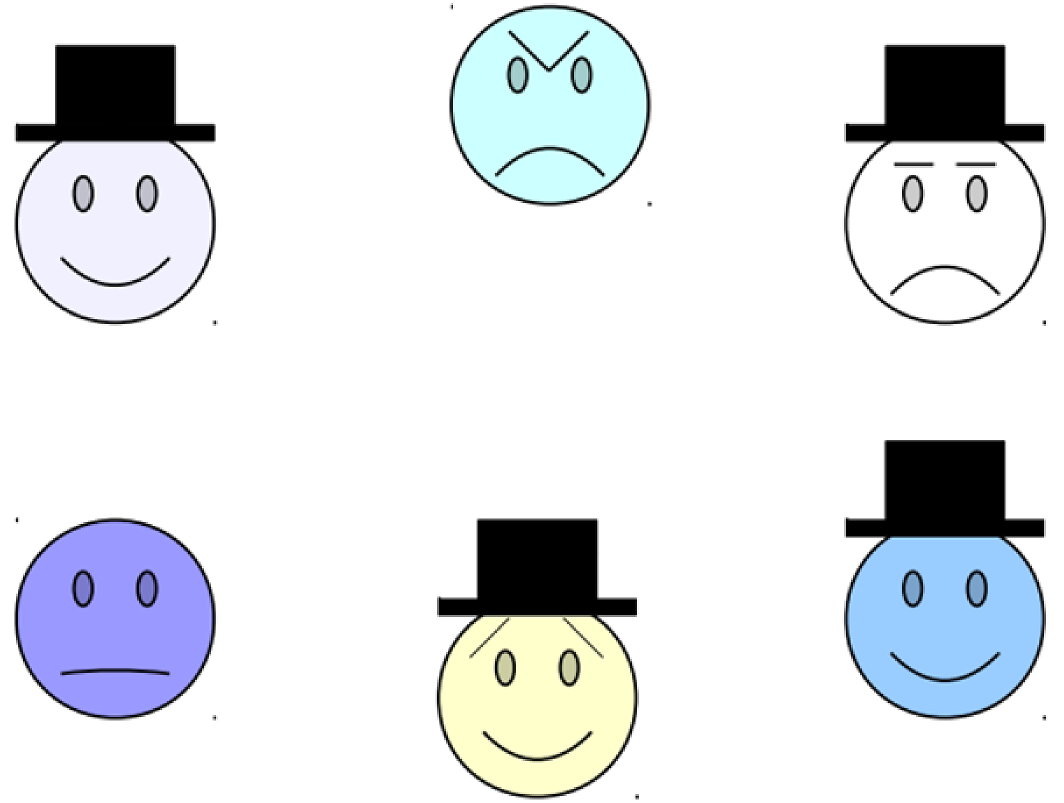
Scope

GO Classes



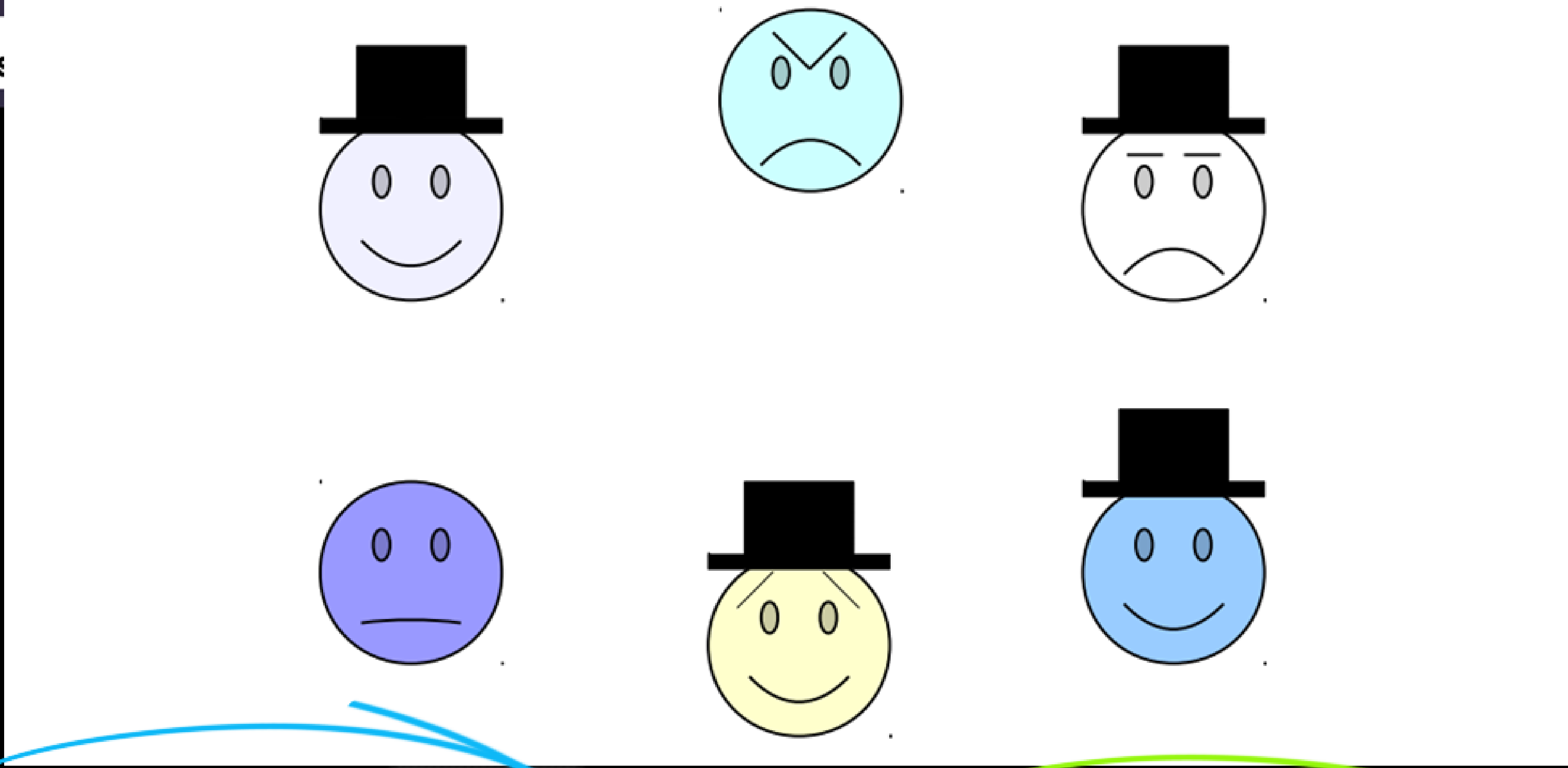
Box

$$\exists x \left(\boxed{Smile(x) \wedge WearHat(x)} \right) = True$$



$$\exists x\,(\mathrm{Smile}(x)) \land \exists x\,(\mathrm{wearHat}(x))$$

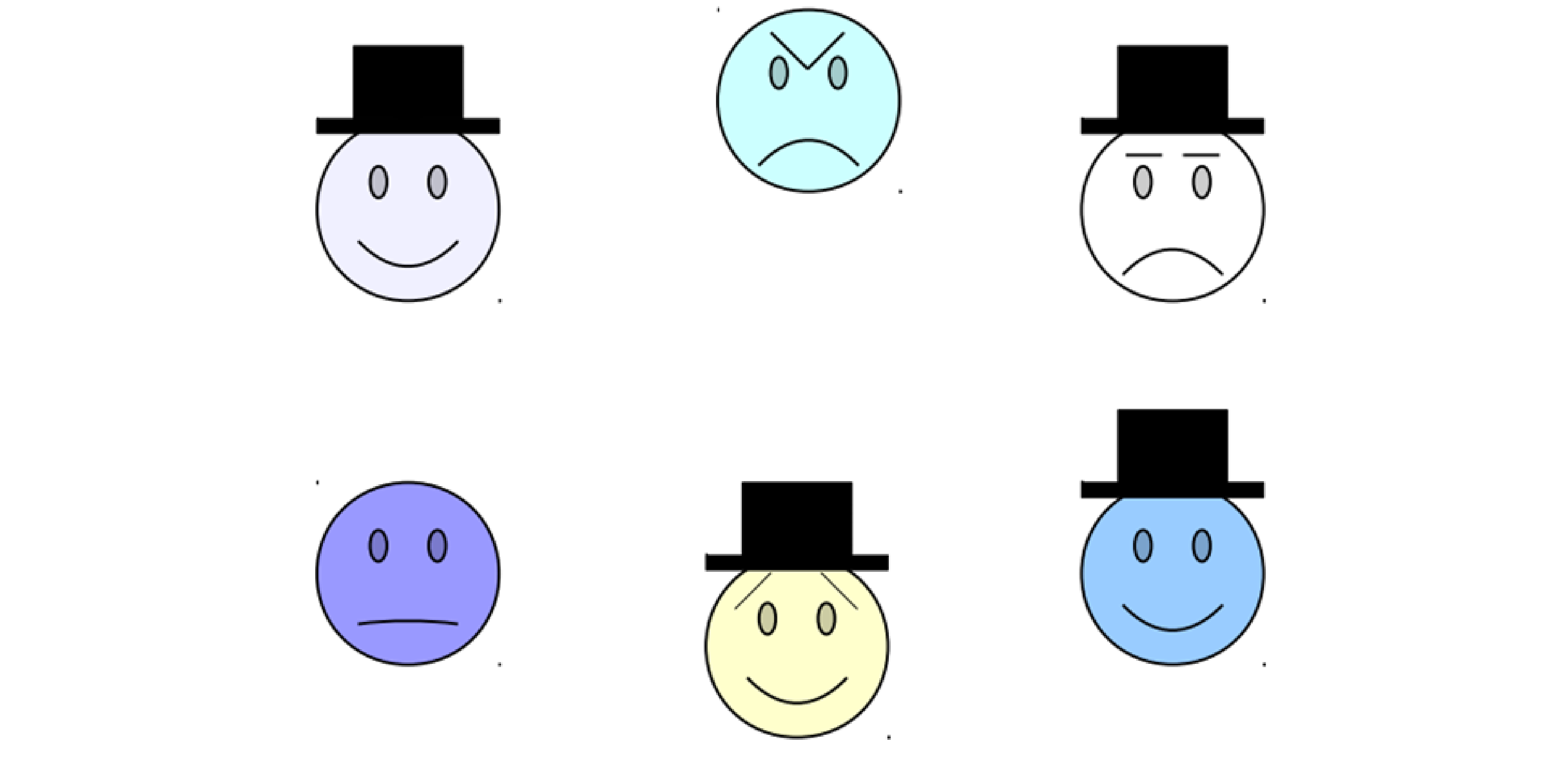
GO Classes



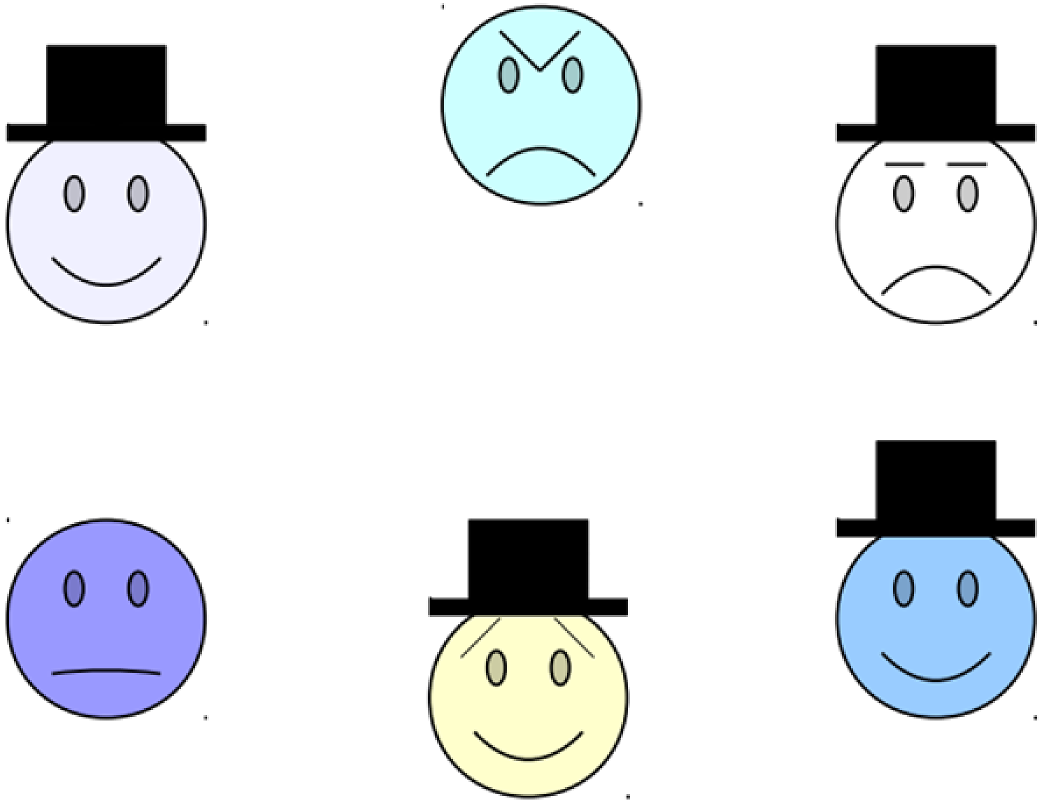
True

True

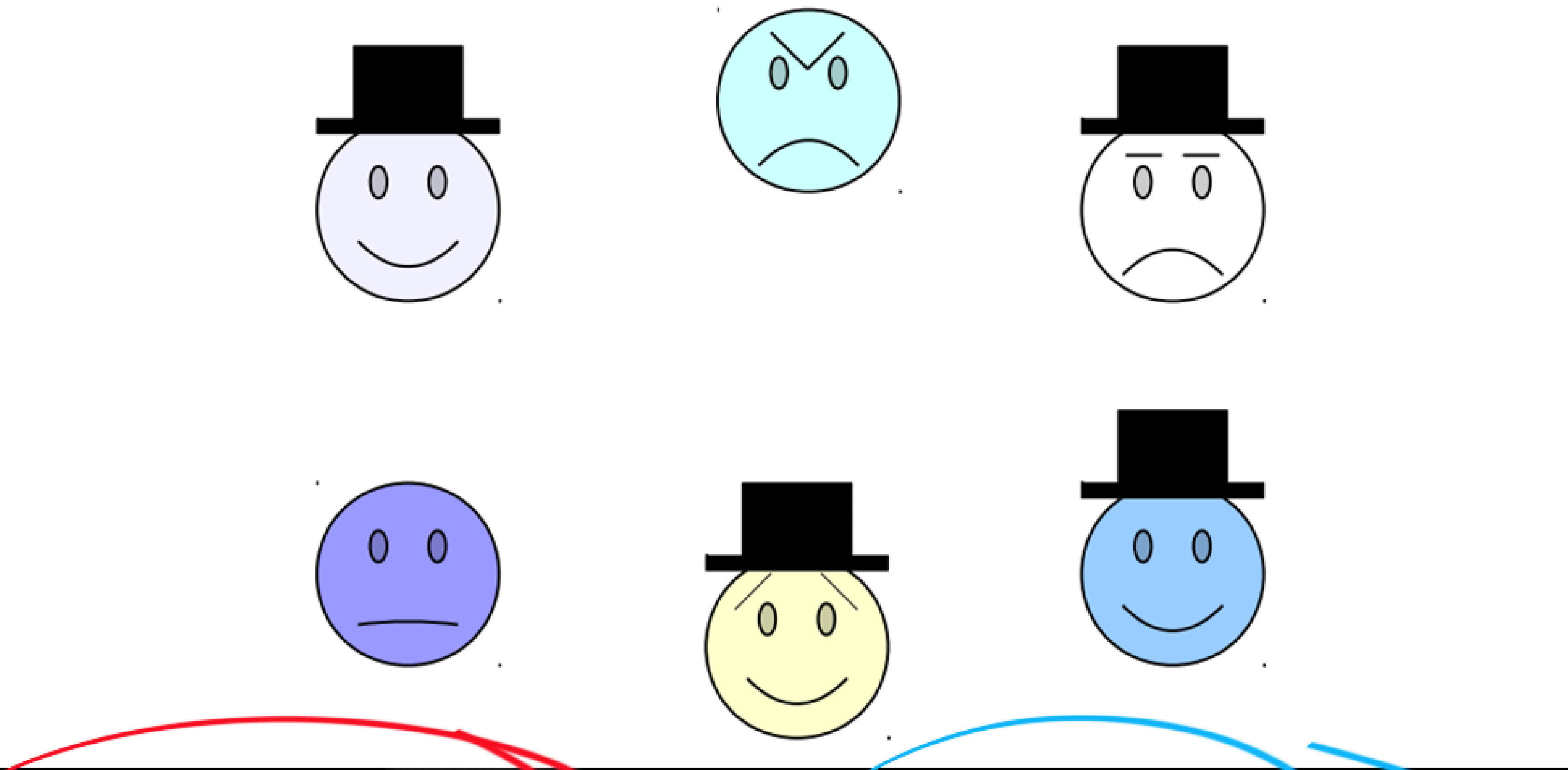
α = $\exists x\,(\text{Smile}(x)) \wedge \exists x\,(\text{wearHat}(x))$ β



$$\underbrace{\exists x\,(\text{Smile}(x))}_{\text{True}} \wedge \underbrace{\exists x\,(\text{wearHat}(x))}_{\text{True}} = \text{True}$$



$$\exists x\,(\text{Smile}(x)) \wedge \exists y\,(\text{wearHat}(y))$$



True

↑

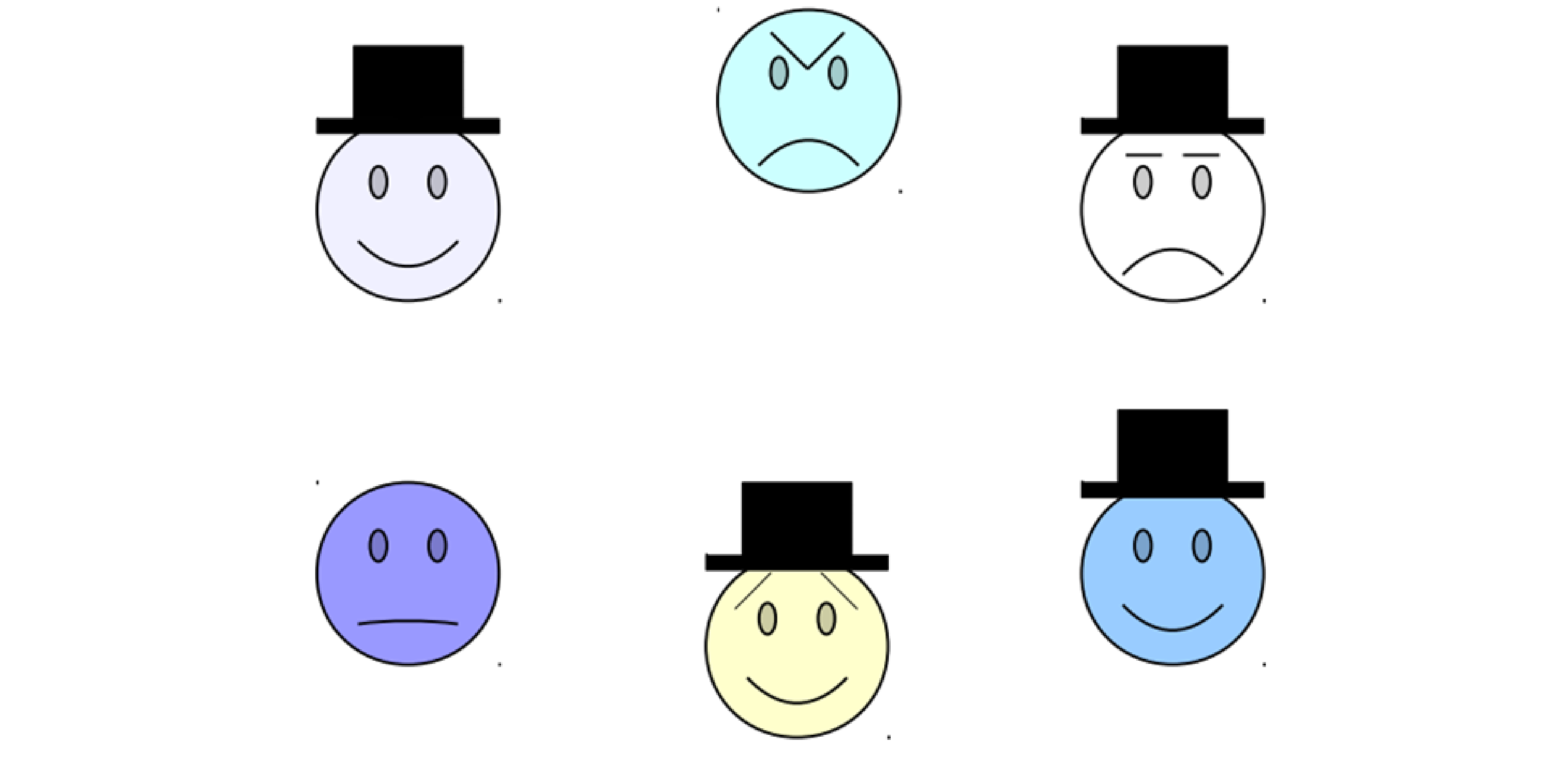
α

True

↑

β

$\exists x\,(Smile(x)) \wedge \exists y\,(wearHat(y))$

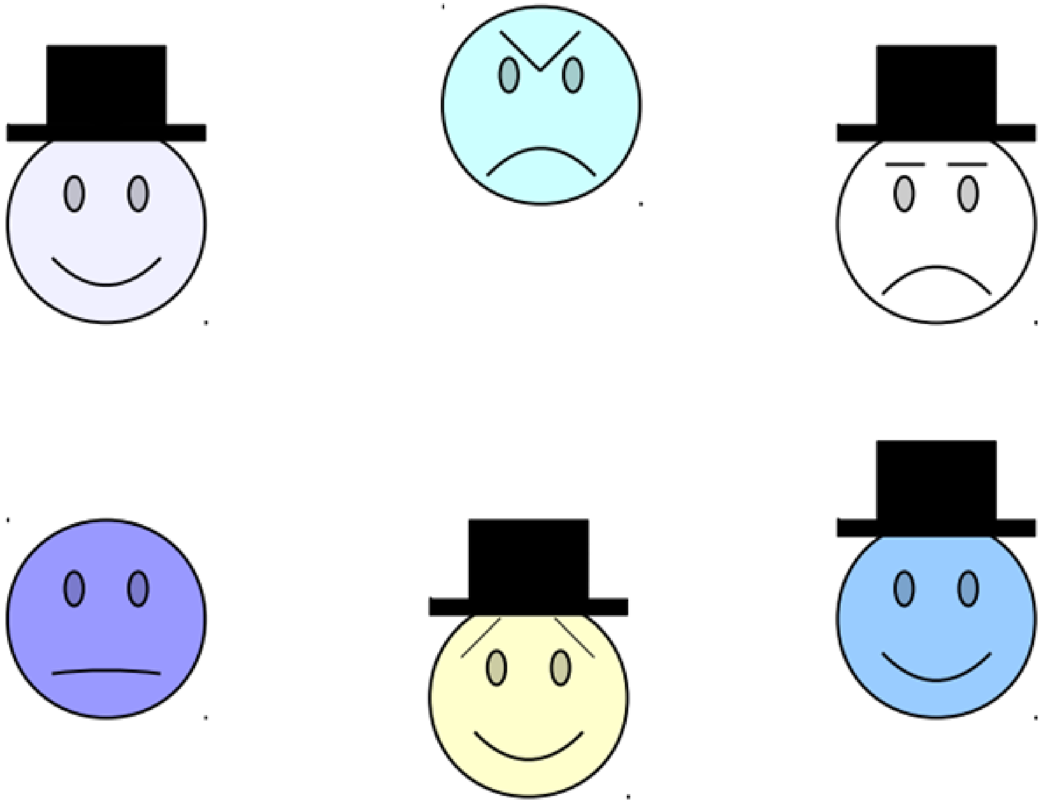


$$\exists x\,(Smile(x)) \wedge \exists y\,(wearHat(y)) = \underline{\underline{True}}$$

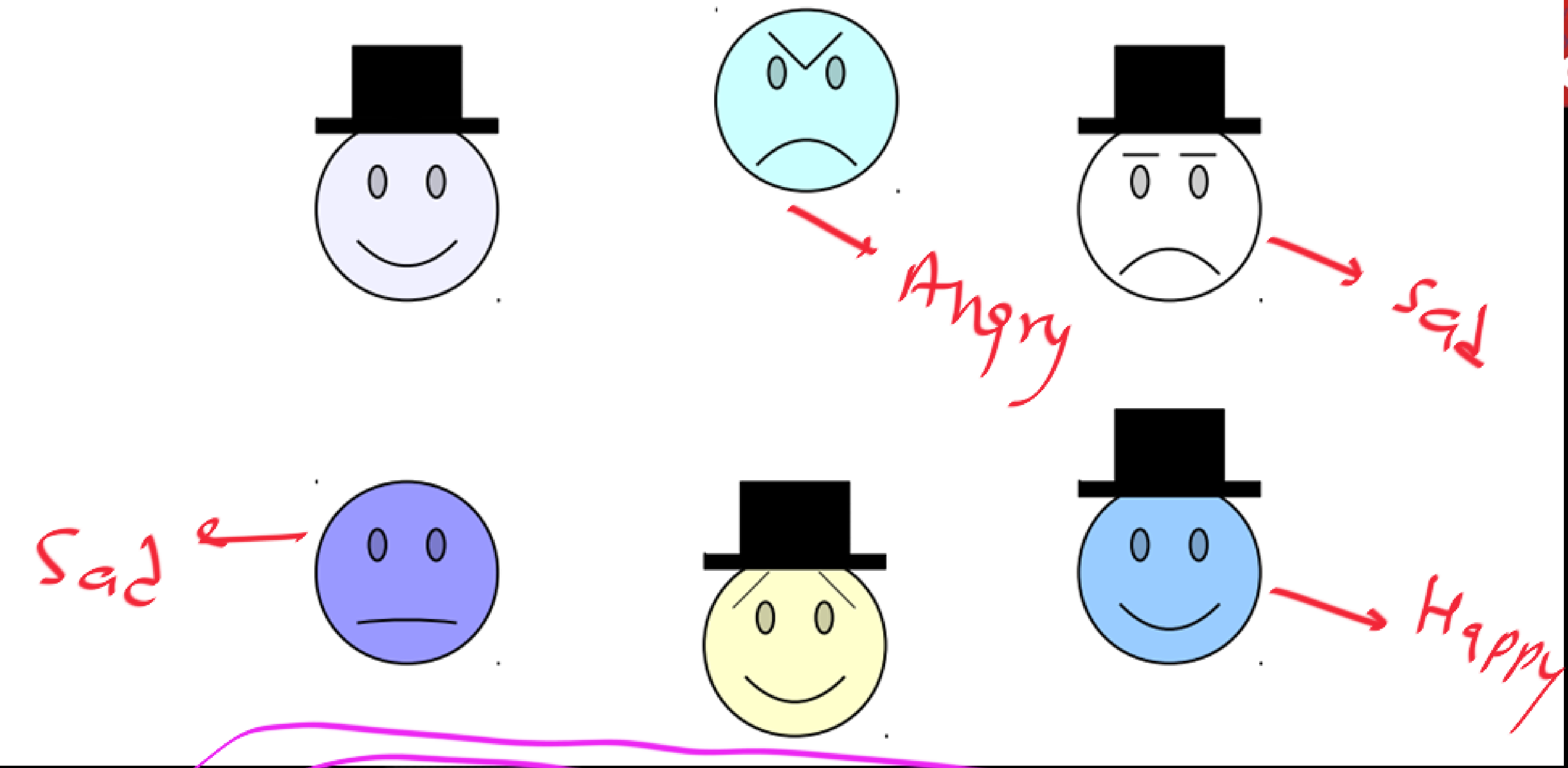
$$\exists x\,(P(x)) \wedge \exists x\,(Q(x))$$

$$\equiv \exists x\,(P(x)) \wedge \exists y\,(Q(y))$$

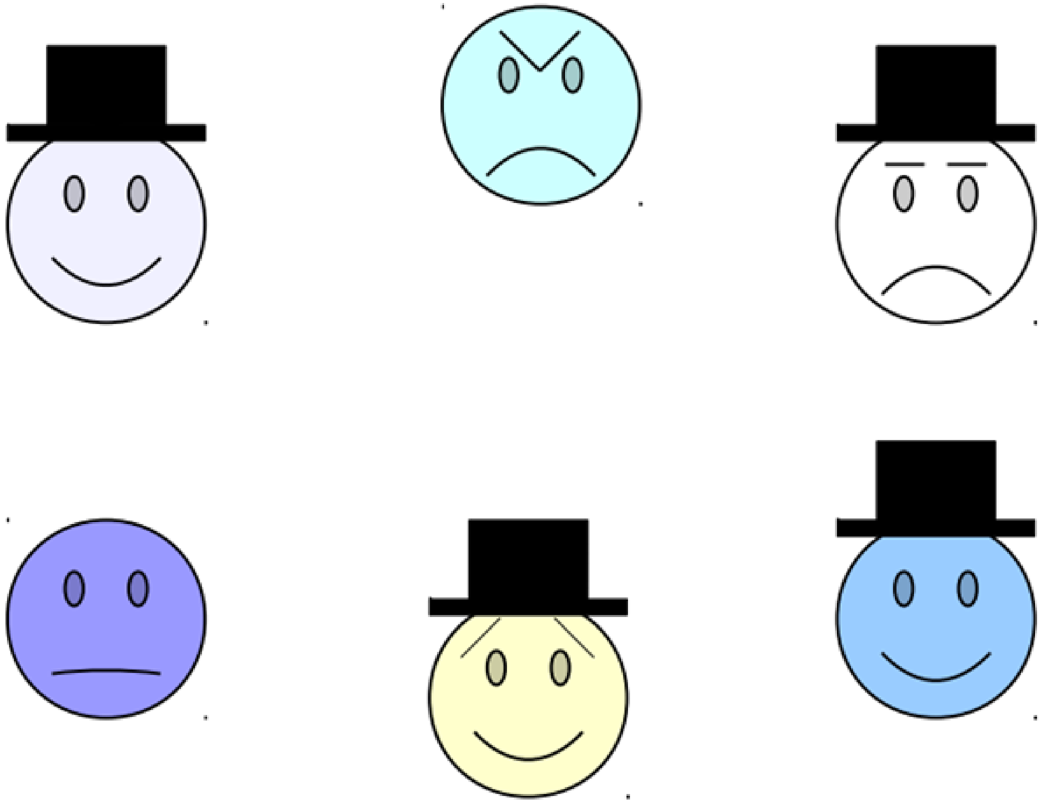
Same



$$\exists x\left(\text{Angry}(x) \wedge \text{wearHat}(x)\right)$$



$$\exists x\,(\text{Angry}(x) \land \text{wearHat}(x)) : \text{True}$$



$$\exists x\,(\text{Angry}(x)) \wedge (\exists x\,(\text{wearHat}(x)))$$

GO Classes

witness for β

witness for α

True

True

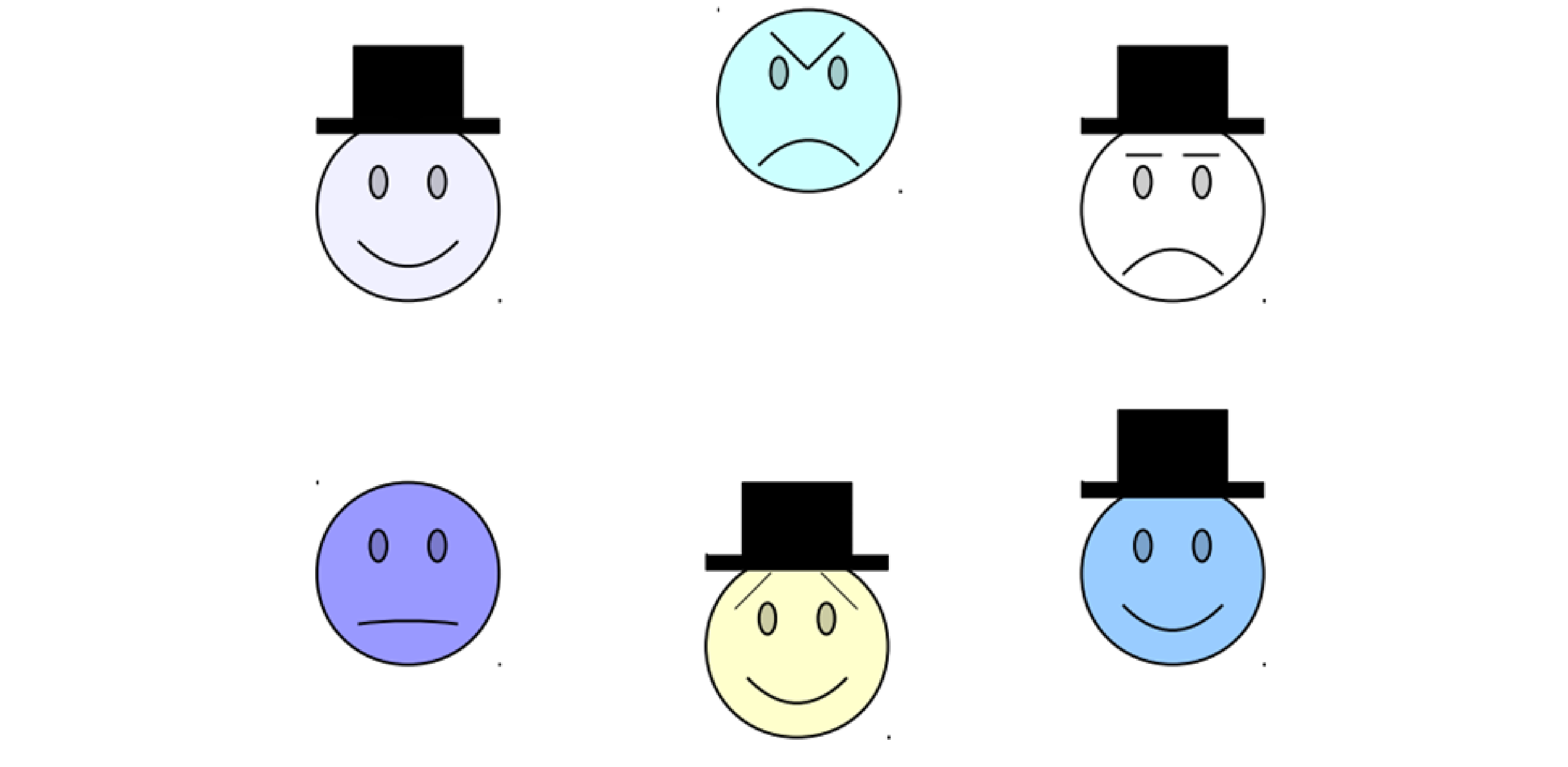
$\Uparrow$

$\uparrow$

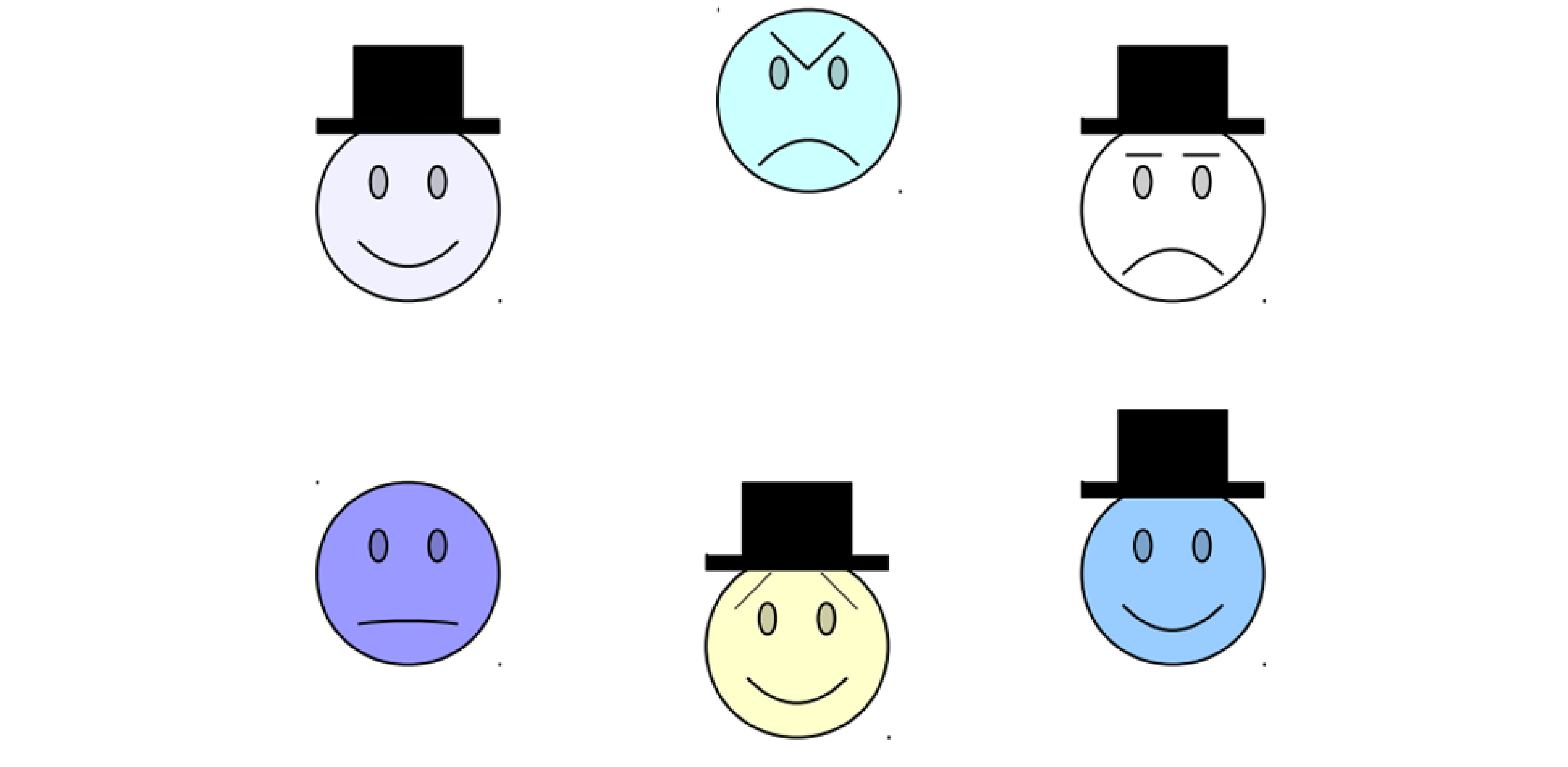
α

β

$$\exists x(\text{Angry}(x)) \wedge (\exists x(\text{wearHat}(x)))$$

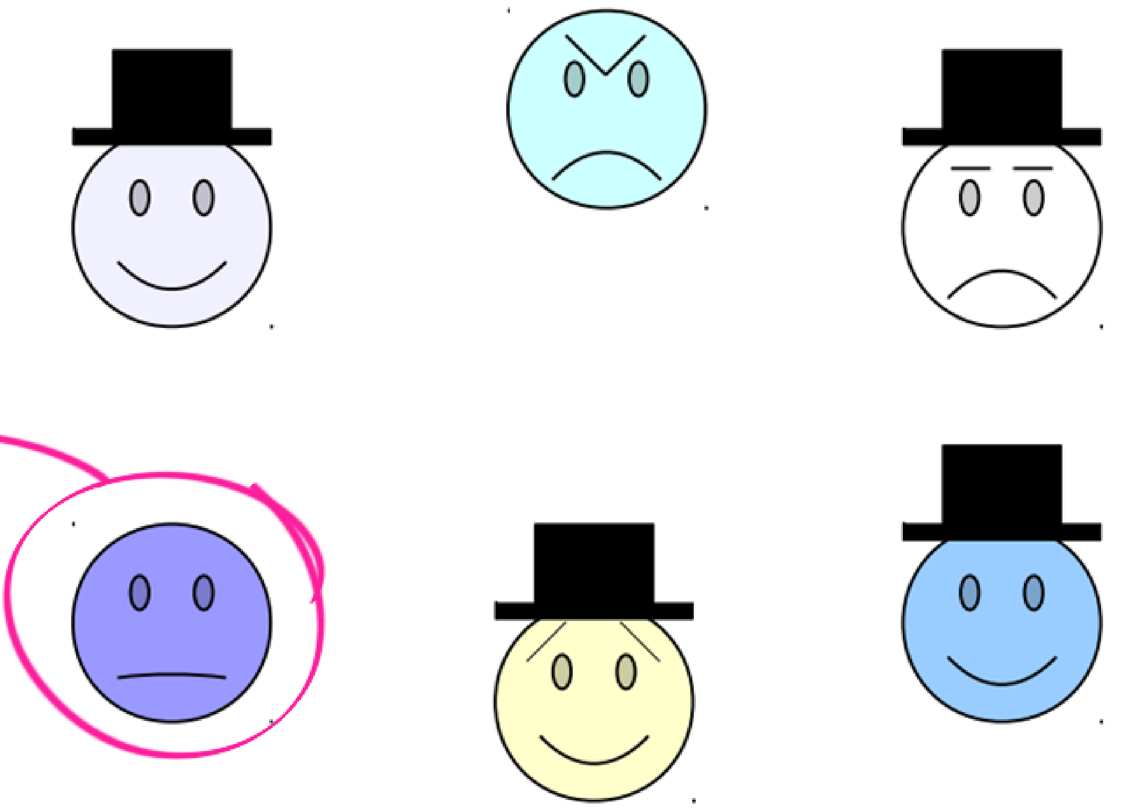


$$\underbrace{\exists x(\text{Angry}(x))}_{T} \land \left(\underbrace{\exists x(\text{wearHat}(x))}_{T}\right) : \text{True}$$



$$\forall x\left(Sad(x) \rightarrow WearHat(x)\right)$$

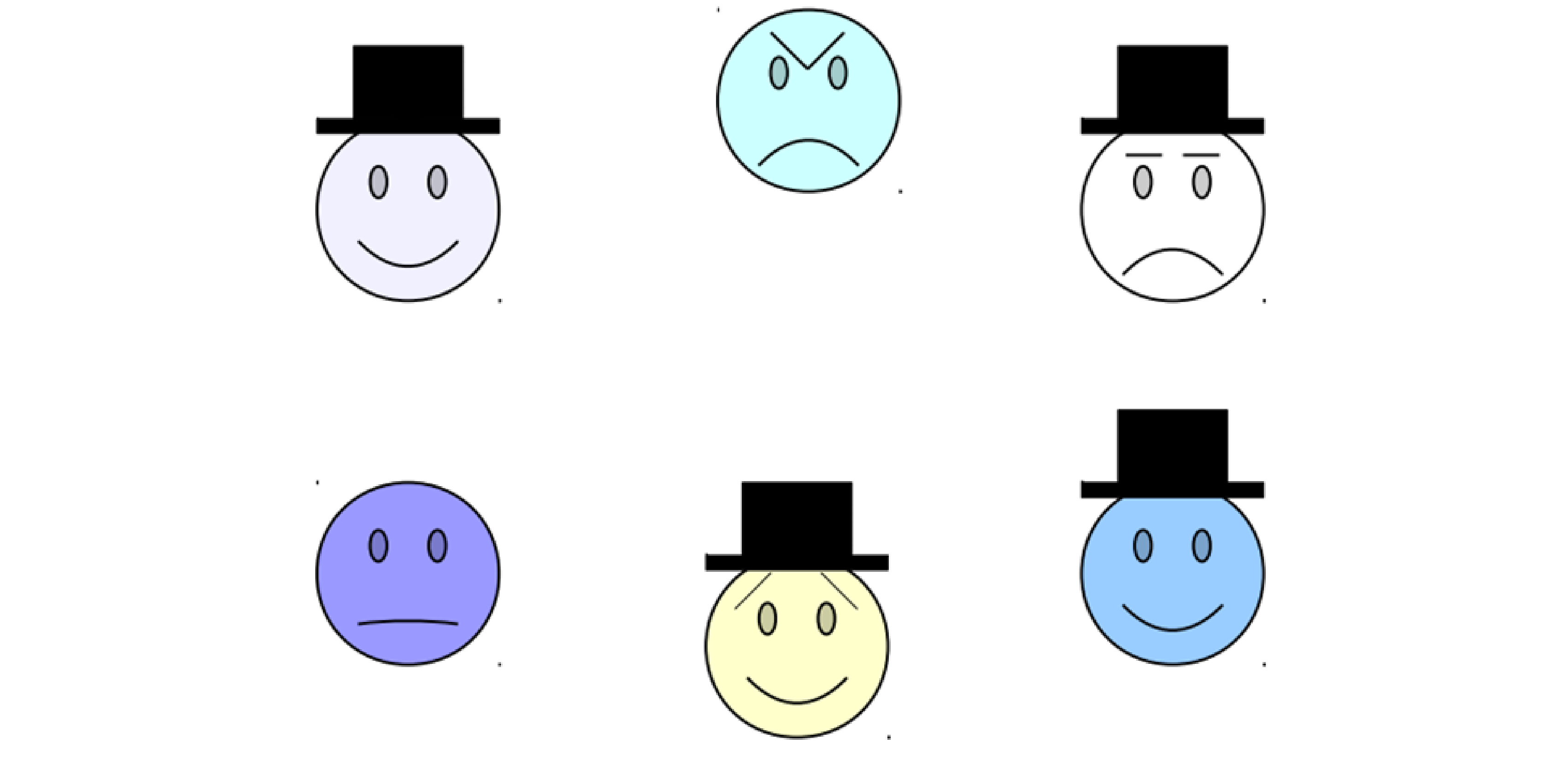
Counter Example



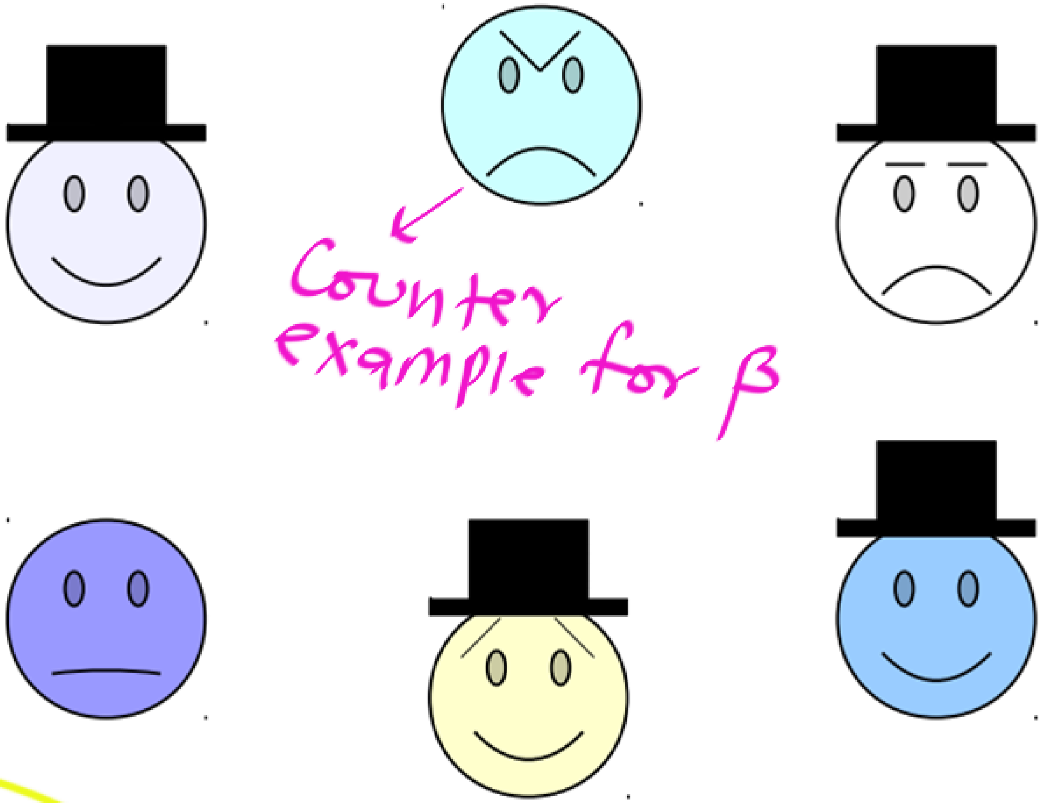
False

$$\forall x \left(Sad(x) \longrightarrow WearHat(x) \right)$$

Every sad person is wearing hat.



$$\forall x\ (Sad(x)) \rightarrow \forall x\ (WearHat(x))$$



Counter example for β

False

α

Counter example for α

$$\forall_x (Sad(x)) \rightarrow \forall_x (WearHat(x))$$

β = False

: True