Creating Domain D

Domain: The collection of objects assumed for a symbolization in FOL, or that gives the range of the quantifiers in an interpretation.

It represents the set of all elements that can be quantified over or that the variables in the logical system can refer to within a given model.

When formally defining a model in FOL, it is written as:

$$M := \langle D, I \rangle$$

where D is the domain of discourse or 'universe', a non-empty set of objects, and I is the interpretation function which maps constants, functions, and predicates to elements, functions, and relations over D.

Expanding Domain D

Domain expansion: Domain Expansion (or Domain Extension) refers to the process of enlaring the domain D to include additional elements. A new interpretation may need to be defined for predicates, functions, and constants, extending their meaning to the new elements in the domain.

In formal terms, if $M := \langle D, I \rangle$ is a model, and we want to extend the domain to a new domain D' (where $D \subset D'$), we would create an extended model:

$$M' := \langle D', I' \rangle$$

where I' is an extension of I that interprets the additional elements in D' as well.

This approach allows FOL statements to be evaluated within a larger context without altering the original structure's properties within D.

Domain Before Expansion $D = \{\}$

Domain After Expansion $D\text{'} = \{Corwin, Benedict\}$

Built FOL Syntax Tree - Level 1

∀x((N(x) ∨ ¬N(x))) Main Logical Operator: Quantifier Precedence: 4

Built FOL Syntax Tree - Level 2

(N(x) ∨ ¬N(x)) Main Logical Operator: ∨ Precedence: 7

Built FOL Syntax Tree - Level 3

Term 1: N(x) Main Logical Operator: Predicate Precedence: 2

Term 2: ¬N(x) Main Logical Operator: ¬ Precedence: 3

Built FOL Syntax Tree - Level 4

N(x) Main Logical Operator: Predicate Precedence: 2

Evaluated AST - Level 4

Object (d) satisfies N in interpretation I iff (x) is true in I [(d)/(x)]

To evaluate predicate N with terms (x), resolve terms under I: (I(x)) ↦ (x)

Then, find N(x) by determining if (x) ∈ I(N) = the set of tuples from the domain of I that satisfy the predicate N under I.

I(N) = {}

(x) ∈ {} = False

Evaluated AST - Level 3

Object (d) satisfies N in interpretation I iff (x) is true in I [(d)/(x)]

To evaluate predicate N with terms (x), resolve terms under I: (I(x)) ↦ (x)

Then, find N(x) by determining if (x) ∈ I(N) = the set of tuples from the domain of I that satisfy the predicate N under I.

I(N) = {}

(x) ∈ {} = False

¬N(x) is true in interpretation I iff N(x) is false in I

¬(N(x)) = True

Evaluated AST - Level 2

N(x) ∨ ¬N(x) is true in interpretation I iff either N(x) is true or ¬N(x) is true in I

(N(x) ∨ ¬N(x)) = True

Evaluated AST - Level 1

Checking if $Corwin$ **is in Domain** $D \quad \Vdash_M Corwin \in D$

Checking if $Benedict$ **is in Domain** $D \quad \Vdash_M Benedict \in D$

∀x((N(x) ∨ ¬N(x))) is true in I iff every object in I's domain ({Corwin, Benedict}) satisfies (N(x) ∨ ¬N(x))

(N(x) ∨ ¬N(x)) ↔ True for all objects in I's domain

∀x((N(x) ∨ ¬N(x))) ↔ True

Final Result

Formula ∀x(N(x) ∨ ¬N(x)) is True in M.