CS313K: Logic, Sets, and Functions

J Strother Moore
Department of Computer Sciences
University of Texas at Austin

Lecture 20 – Chaps 6, 7 (7.1, 7.2, 7.3)

Which expression best captures the sentence "`x` and `y` are permutations of each other if every object occurs the same number of times each of them."? (I use `hm` for the familiar `how-many` function.)

A. `(perm x y)` $\longrightarrow$ `((hm e x)=(hm e y))`

B. `((hm e x)=(hm e y))` $\longrightarrow$ `(perm x y)`

C. `(`$\forall$ `e : ((hm e x)=(hm e y))` $\longrightarrow$ `(perm x y))`

D. `(`$\forall$ `e : ((hm e x)=(hm e y)))` $\longrightarrow$ `(perm x y)`

E. `((`$\forall$`e:(hm e x))=(`$\forall$`e:(hm e y)))` $\longrightarrow$ `(perm x y)`

Is this English statement: “`x` and `y` are permutations of each other if every object occurs the same number of times in each of them”

formalized by this:

γ: `(`$\forall$ `e : ((hm e x)=(hm e y))` $\longrightarrow$ `(perm x y))`

The English statement: "x and y are permutations of each other if every object occurs the same number of times in each of them" is valid.

So γ doesn't formalize it if I can show you an x and y that make γ false.

$$\gamma\texttt{: (}\forall\texttt{ e : ((hm e x)=(hm e y))} \longrightarrow \texttt{(perm x y))}$$

γ: `(`$\forall$ `e : ((hm e x)=(hm e y))` $\longrightarrow$ `(perm x y))`

Let

`x : '(0)`

`y : '(1)`

$$\gamma\text{:}\ (\forall\ \texttt{e}\ :\ \texttt{((hm e '(0))=(hm e '(1)))}\rightarrow\texttt{(perm '(0) '(1))})$$

γ: ($\forall$ e : ((hm e '(0))=(hm e '(1)))$\longrightarrow$nil)

γ: (∀ e : ((hm e '(0))=(hm e '(1)))→nil)

↔

[((hm e '(0))=(hm e '(1)))→nil)/{e ← 0}

∧

((hm e '(0))=(hm e '(1)))→nil)/{e ← 1}

∧

((hm e '(0))=(hm e '(1)))→nil)/{e ← 2}

∧

...]

γ: (∀ e : ((hm e '(0))=(hm e '(1)))→nil)

↔

[((hm 0 '(0))=(hm 0 '(1)))→nil

∧

((hm 1 '(0))=(hm 1 '(1)))→nil

∧

((hm 2 '(0))=(hm 2 '(1)))→nil

∧

...]

γ: (∀ e : ((hm e '(0))=(hm e '(1)))→nil)

↔

[((1=0)→nil)

∧

((hm 1 '(0))=(hm 1 '(1)))→nil

∧

((hm 2 '(0))=(hm 2 '(1)))→nil

∧

...]

γ: `(`$\forall$ `e : ((hm e '(0))=(hm e '(1)))`$\rightarrow$`nil)`

$\leftrightarrow$

`[nil`$\rightarrow$`nil`

$\wedge$

`((hm 1 '(0))=(hm 1 '(1)))`$\rightarrow$`nil`

$\wedge$

`((hm 2 '(0))=(hm 2 '(1)))`$\rightarrow$`nil`

$\wedge$

`...]`

γ: `(`$\forall$ `e : ((hm e '(0))=(hm e '(1)))`$\rightarrow$`nil)`

$\leftrightarrow$

`[t`

$\wedge$

`((hm 1 '(0))=(hm 1 '(1)))`$\rightarrow$`nil`

$\wedge$

`((hm 2 '(0))=(hm 2 '(1)))`$\rightarrow$`nil`

$\wedge$

`...]`

γ: (∀ e : ((hm e '(0))=(hm e '(1)))→nil)

↔

[t

∧

(0=1)→nil

∧

((hm 2 '(0))=(hm 2 '(1)))→nil

∧

...]

γ: ($\forall$ e : ((hm e '(0))=(hm e '(1)))$\rightarrow$nil)

$\leftrightarrow$

[t

$\wedge$

nil$\rightarrow$nil

$\wedge$

((hm 2 '(0))=(hm 2 '(1)))$\rightarrow$nil

$\wedge$

...]

γ: `(`$\forall$ `e : ((hm e '(0))=(hm e '(1)))`$\rightarrow$`nil)`

$\leftrightarrow$

`[t`

$\wedge$

`t`

$\wedge$

`((hm 2 '(0))=(hm 2 '(1)))`$\rightarrow$`nil`

$\wedge$

`...]`

γ: (∀ e : ((hm e '(0))=(hm e '(1)))$\longrightarrow$nil)

$\leftrightarrow$

[t

$\wedge$

t

$\wedge$

(0=0)$\longrightarrow$nil

$\wedge$

...]

γ: `(`$\forall$ `e : ((hm e '(0))=(hm e '(1)))`$\longrightarrow$`nil)`

$\leftrightarrow$

`[t`

$\wedge$

`t`

$\wedge$

`t`$\longrightarrow$`nil`

$\wedge$

`...]`

γ: (∀ e : ((hm e '(0))=(hm e '(1)))→nil)

↔

[t

∧

t

∧

nil

∧

...]

γ: `(`$\forall$ `e : ((hm e '(0))=(hm e '(1)))`$\rightarrow$`nil)`

$\leftrightarrow$

`nil`

So γ is not valid but the English statement is valid. Thus, γ doesn't have the same meaning the English statement.

Key Idea

Think of `(`$\forall$ `e : (`ϕ `e))` as a "big conjunct"

`((`ϕ `0)`

$\wedge$

`(`ϕ `1)`

$\wedge$

`(`ϕ `2)`

$\wedge$

$\ldots$)

Think of `(`$\exists$ `e : (`ϕ `e))` as a "big disjunct"

$$
\begin{array}{l}
\texttt{((}\phi\ \texttt{0)} \\
\ \vee \\
\ \texttt{(}\phi\ \texttt{1)} \\
\ \vee \\
\ \texttt{(}\phi\ \texttt{2)} \\
\ \vee \\
\ \ldots\texttt{)}
\end{array}
$$

γ: `(`$\forall$` e : [(hm e x)=(hm e y)]`$\rightarrow$` (perm x y))`

$\leftrightarrow$

`([(hm 0 x)=(hm 0 y)]`$\rightarrow$` (perm x y))`

$\wedge$

`([(hm 1 x)=(hm 1 y)]`$\rightarrow$` (perm x y))`

$\wedge$

`([(hm 2 x)=(hm 2 y)]`$\rightarrow$` (perm x y))`

$\wedge$

`...]`

γ: [$\forall$ e : ((hm e x)=(hm e y))]$\longrightarrow$ (perm x y))

$\longleftrightarrow$

[((hm 0 x)=(hm 0 y))

$\wedge$

((hm 1 x)=(hm 1 y))

$\wedge$

((hm 2 x)=(hm 2 y))

$\wedge$

...]

$\longrightarrow$ (perm x y)

Are These Two the Same?

$((p \wedge q) \rightarrow r)$

versus

$((p \rightarrow r) \wedge (q \rightarrow r))$?

No! Let `p` be `nil` and `q` be `t` and `r` be `nil`.