Copredication in Homotopy Type Theory

Hamidreza Bahramian[1]
Indiana University Bloomington

Presentation based on a joint work with Narges Nematollahi[2] and Amr Sabry[3]

[1] Department of Computer Science
[2] Department of Linguistics
[3] Department of Computer Science

Introduction

Copredication:

the phenomenon where two or more predicates with different requirements on their arguments are applied to a single argument

a) The lunch was delicious but took forever. (Asher, 2011)
b) The heavy book is easy to understand. (Gotham, 2014)
c) John picked up and mastered three books. (Asher 2011)

History

- Asher(2011), nouns as “dot objects”(Pustejovsky, 1995)
  - i.e., objects that can be viewed under different “aspects”
- Cooper (2011)
  - nouns as functions from “records” to “record types”
- Luo (2012), Chatzikyriakidis and Luo (2012, 2013, 2015)
  - common nouns as types
  - dot types + coercive subtyping
- Gotham(2014)
  - `book' denotes the set of composite objects physical+informational
  - criteria of individuation are combined during semantic composition

Montague semantics

(1) a. The lunch was delicious.

b. The lunch took forever.

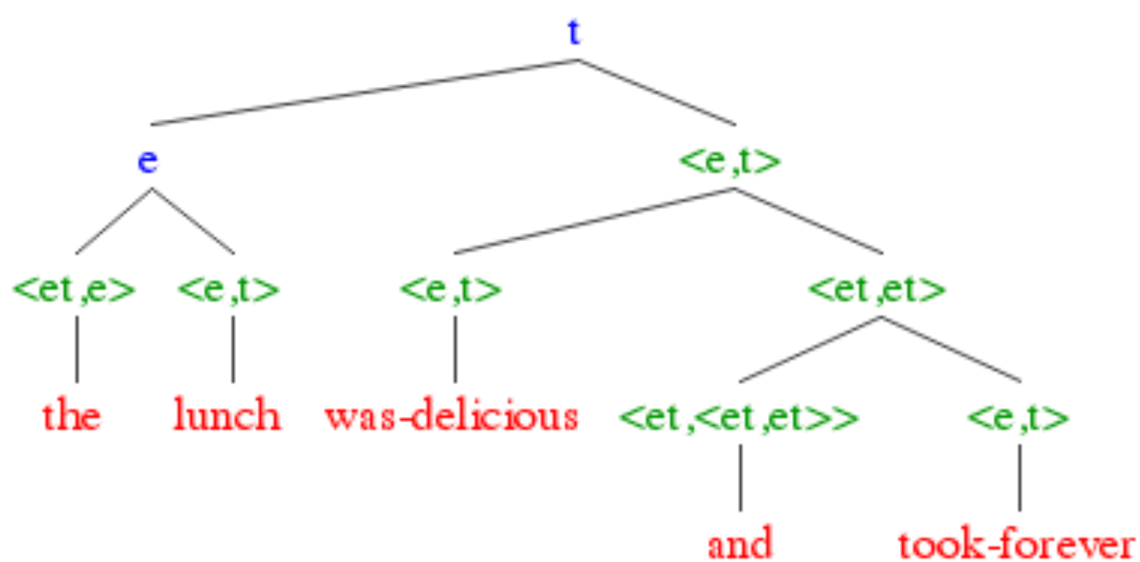
c. The lunch was delicious and took forever.

$$
\begin{array}{rcl}
[\![\text{was delicious}]\!] & = & \lambda x \in D_e . x \text{ was delicious} \\
[\![\text{took forever}]\!] & = & \lambda x \in D_e . x \text{ took forever} \\
[\![\text{and}]\!] & = & [\lambda f \in D_{\langle e,t\rangle} . \\
 & & [\lambda g \in D_{\langle e,t\rangle} . [\lambda x \in D_e . f(x) = g(x) = 1]]] \\
[\![\text{lunch}]\!] & = & \lambda x \in D_e . x \text{ is a lunch} \\
[\![\text{the}]\!] & = & \lambda f \in D_{<e,t>} \;\&\; \exists ! x \in D_e [f(x) = 1] . \\
 & & !y[f(y) = 1], \\
 & & \text{where } \exists ! x[f(x) = 1] \\
 & & \text{abbreviates "there is exactly} \\
 & & \text{one x such that f(x)=1 and } !y[\phi] \text{ returns} \\
 & & \text{"that unique y such that f(y)=1".}
\end{array}
$$

Functional Application (FA): if α is a branching node with β and γ as its daughters, then α is in the domain of $[\![\cdot]\!]$ if both β and γ are, and if $[\![\gamma]\!]$ is in the domain of $[\![\beta]\!]$. In this case, $[\![\alpha]\!]=[\![\beta]\!]([\![\gamma]\!])$ (Heim and Kratzer[1998]).

Montague semantics

(1) a. The lunch was delicious.
  b. The lunch took forever.
  c. The lunch was delicious and took forever.



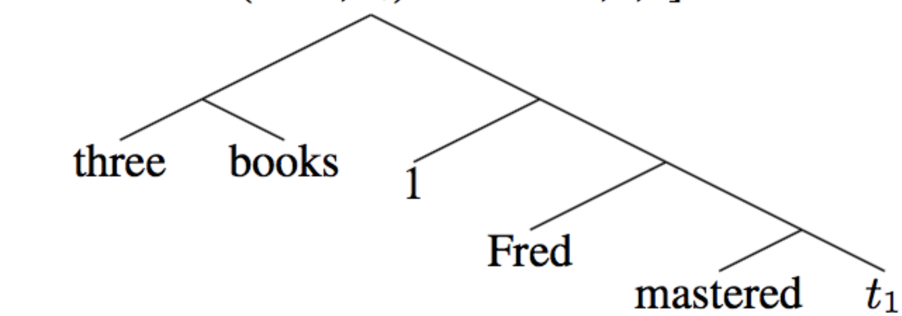
Montague semantics

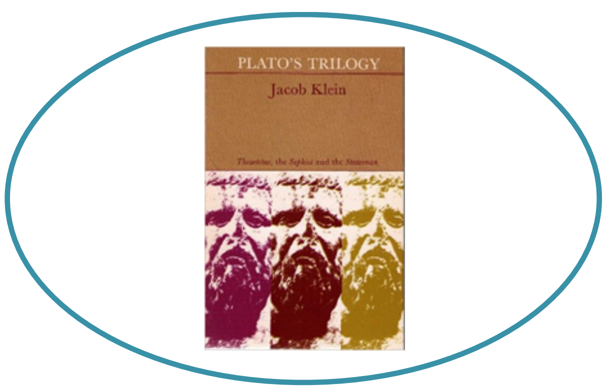
(2) a. Fred picked up three books.

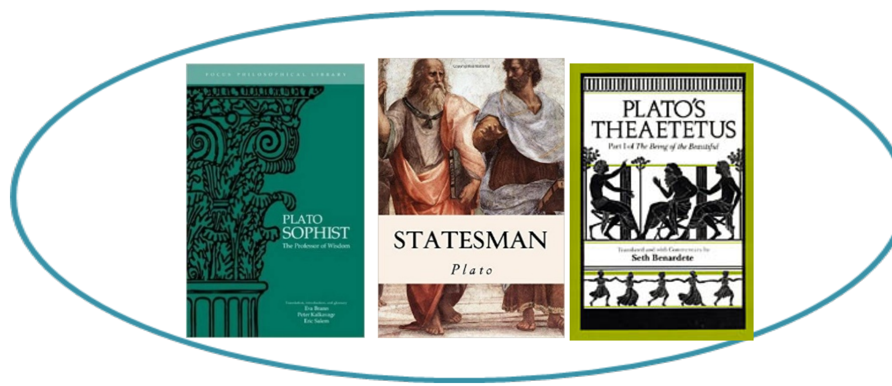
b. Fred mastered three books.

c. Fred picked up and mastered three books.

1 iff $\exists x_1, x_2, x_3 \in D_e$ [book(x_i)=1 and mastered (Fred,x_i)=1 for i=1,2,3]







In Luo's framework

(1) a. The lunch was delicious.

b. The lunch took forever.

c. The lunch was delicious and took forever.

A.B is only well-formed if A and B do not share common components, and both projections, one from A.B to A and the other from A.B to B, are coercions in the coercive subtyping framework.(Luo[2012])

a. $\text{Food.Event} <_c \text{Food}$

b. $\text{Food.Event} <_c \text{Event}$

c. $\text{Lunch} <_c \text{Food.Event} <_c \text{Food}$

d. $\text{Lunch} <_c \text{Food.Event} <_c \text{Event}$

In Luo's framework

(1) a. The lunch was delicious.
b. The lunch took forever.
c. The lunch was delicious and took forever.

a. $\text{Food.Event} <_c \text{Food}$
b. $\text{Food.Event} <_c \text{Event}$
c. $\text{Lunch} <_c \text{Food.Event} <_c \text{Food}$
d. $\text{Lunch} <_c \text{Food.Event} <_c \text{Event}$

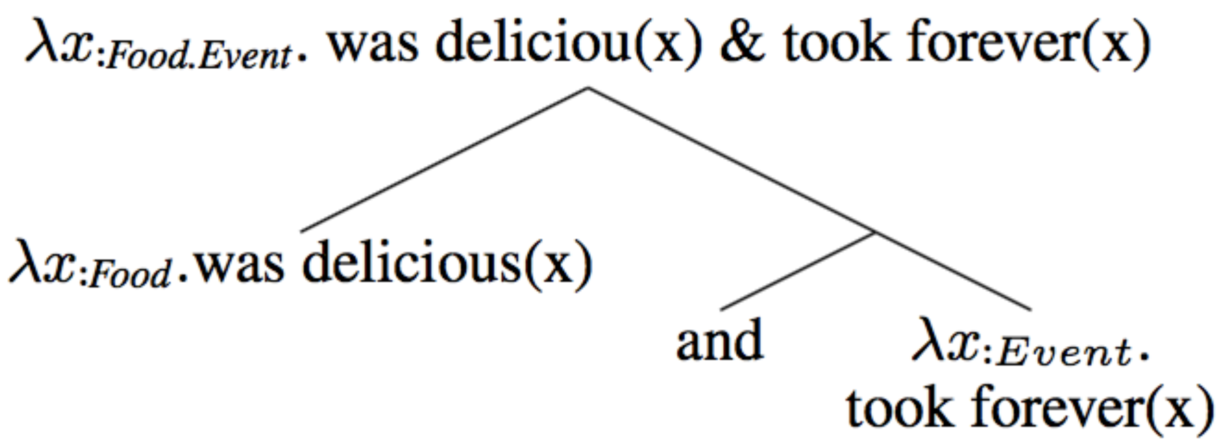
a. $\text{Food} \to \text{Prop} <_c \text{Food.Event} \to \text{Prop} <_c \text{Lunch} \to \text{Prop}$
b. $\text{Event} \to \text{Prop} <_c \text{Food.Event} \to \text{Prop} <_c \text{Lunch} \to \text{Prop}$

In Luo's framework

(1) a. The lunch was delicious.

b. The lunch took forever.

c. The lunch was delicious and took forever.

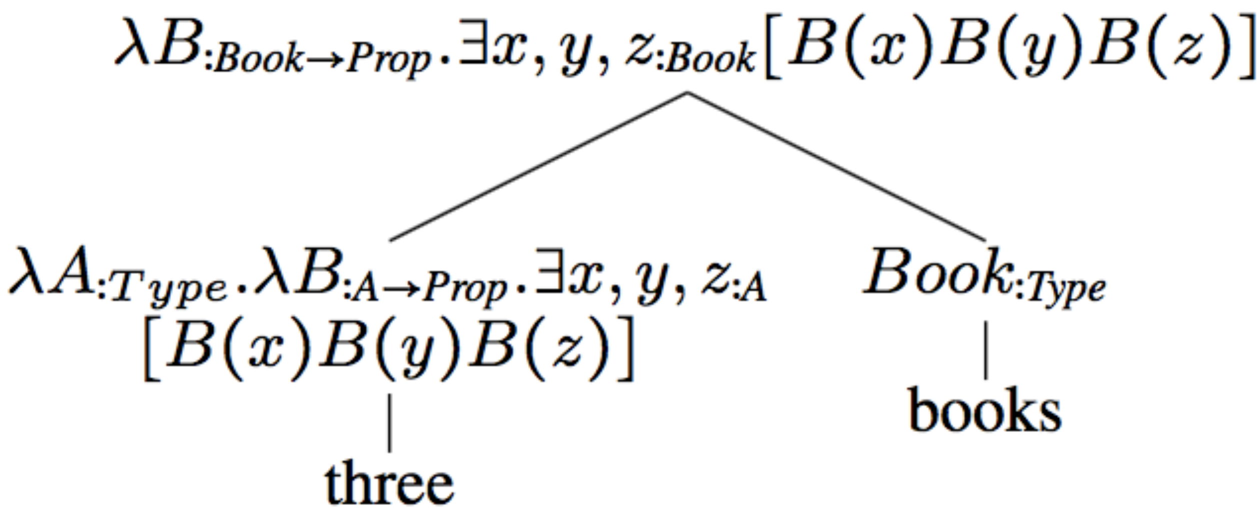


In Luo's framework

(2) a. Fred picked up three books.

b. Fred mastered three books.

c. Fred picked up and mastered three books.

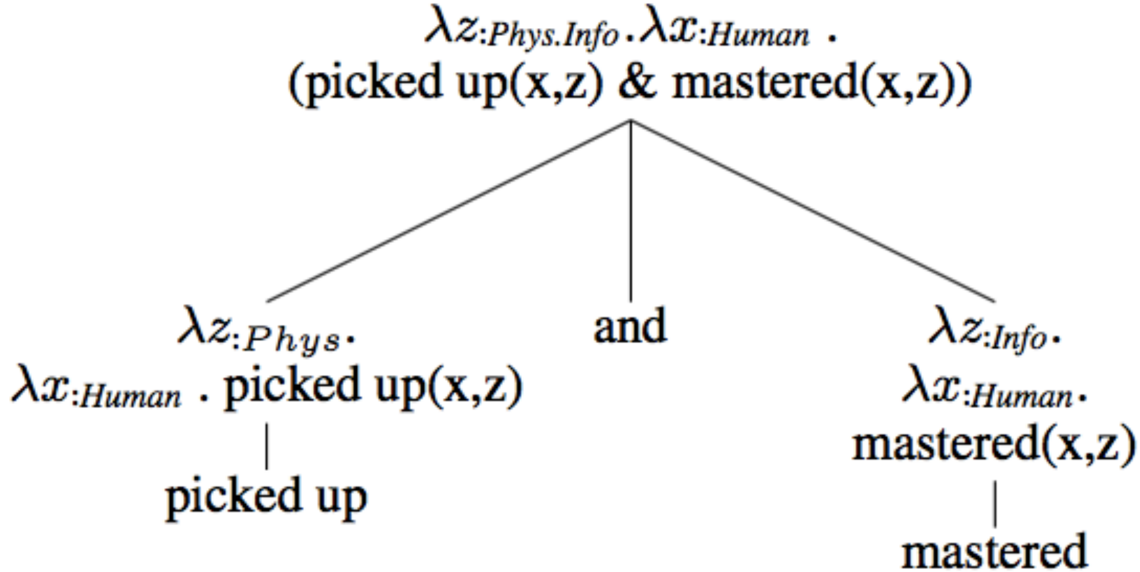


In Luo's framework

(2) a. Fred picked up three books.

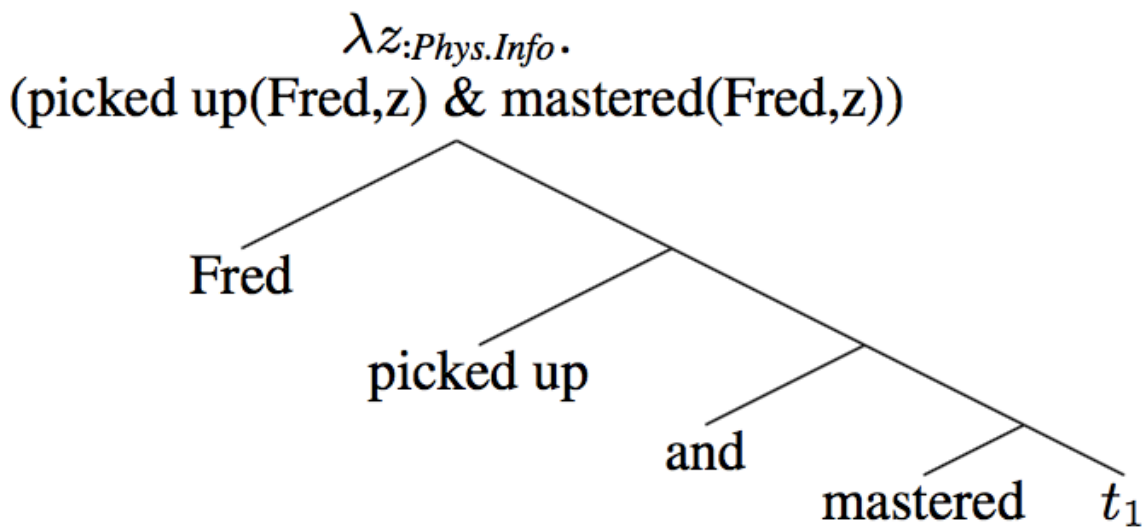
b. Fred mastered three books.

c. Fred picked up and mastered three books.



In Luo's framework

(2) a. Fred picked up three books.
b. Fred mastered three books.
c. Fred picked up and mastered three books.



In Luo's framework

(2) a. Fred picked up three books.

b. Fred mastered three books.

c. Fred picked up and mastered three books.

$\exists x, y, z_{:Book}$ [picked up(Fred,x) and mastered(Fred,x)
picked up(Fred,y) and mastered (Fred,y)
picked up(Fred,z) and mastered (Fred,z)]

$\lambda B_{:Book \to Prop}.$
$\exists x, y, z_{:Book}$ [B(x) B(y) B(z)]

three books

$\lambda z_{:Phys.Info}.$
(picked up(Fred,z) and
mastered(Fred,z))

Fred picked up and
mastered t_1

a. Book $<_{c_1}$ Physical

b. Book $<_{c_2}$ Informational

In Luo's framework

(2) a. Fred picked up three books.
b. Fred mastered three books.
c. Fred picked up and mastered three books.

$$\exists x, y, z_{:Book}[\text{picked up(Fred,x) \& mastered(Fred,x)} \\ \text{picked up(Fred,y) \& mastered (Fred,y)} \\ \text{picked up(Fred,z) \& mastered (Fred,z)}]$$

In Luo's framework

(2) a. Fred picked up three books.
b. Fred mastered three books.
c. Fred picked up and mastered three books.

$$\exists x, y, z_{:Book}[\text{picked up(Fred,x) \& mastered(Fred,x)} \\ \text{picked up(Fred,y) \& mastered (Fred,y)} \\ \text{picked up(Fred,z) \& mastered (Fred,z)}]$$

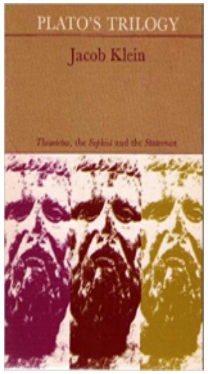
```
Variable PHY:forall x:Book, forall
y:Book, not(x=y:>Book)-> not(x=y:>Phy).
Variable INFO:forall x:Book, forall
y:Book, not(x=y:>Book)-> not(x=y:>Info).
```

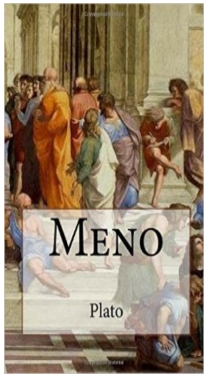
The problem

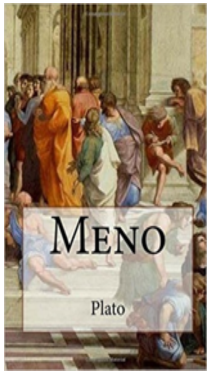
Fred mastered three books.

The problem

Five Books are heavy but easy to understand.(Gotham[2012])







Homotopy Type Theory

- *intensional* version of Martin-Löf’s type theory [ML75]
- a *proof-relevant* interpretation of equality
- *propositions-as-types* principle
- a full cumulative hierarchy of universes
- judgmental vs propositional equality
  - “=” vs “≡”

Homotopy Type Theory

- variables x, x′, . . .
- primitive constants c, c′, . . .
- defined constants f , f′, . . .
- $t ::= x \,|\, \lambda x.t \,|\, t(t') \,|\, c \,|\, f$

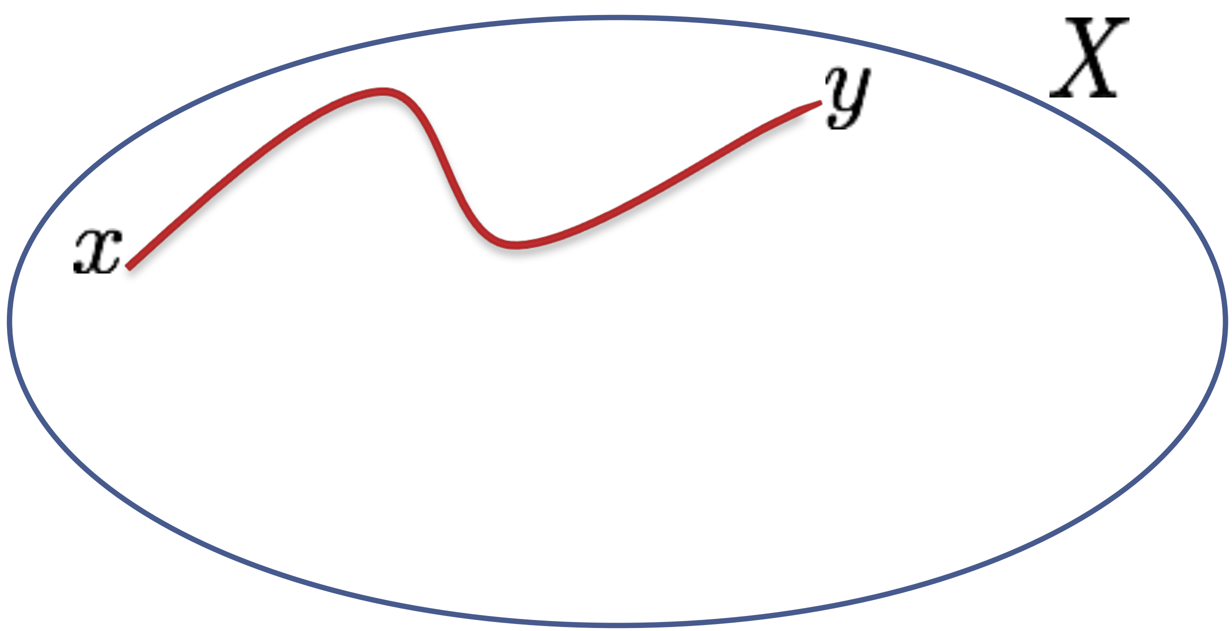
some primitive constants:

- a hierarchy of universes $\mathcal{U}_1, \mathcal{U}_2, \ldots$
- dependent function types $\Pi_{a:A} B$
- dependent pair types $\Sigma_{a:A} B$
- identity types $a =_A b$, $A =_{\mathcal{U}} B$

Homotopy Theory

$p : [0,1] \to X$

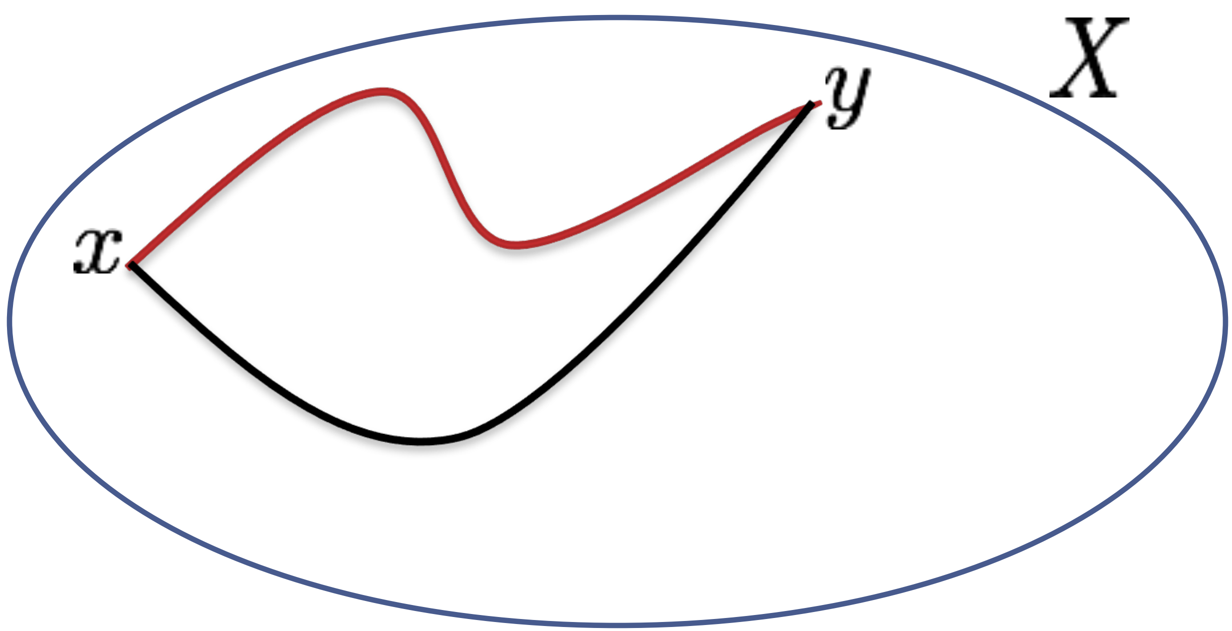
where $p(0) = x$ and $p(1) = y$



Homotopy Theory

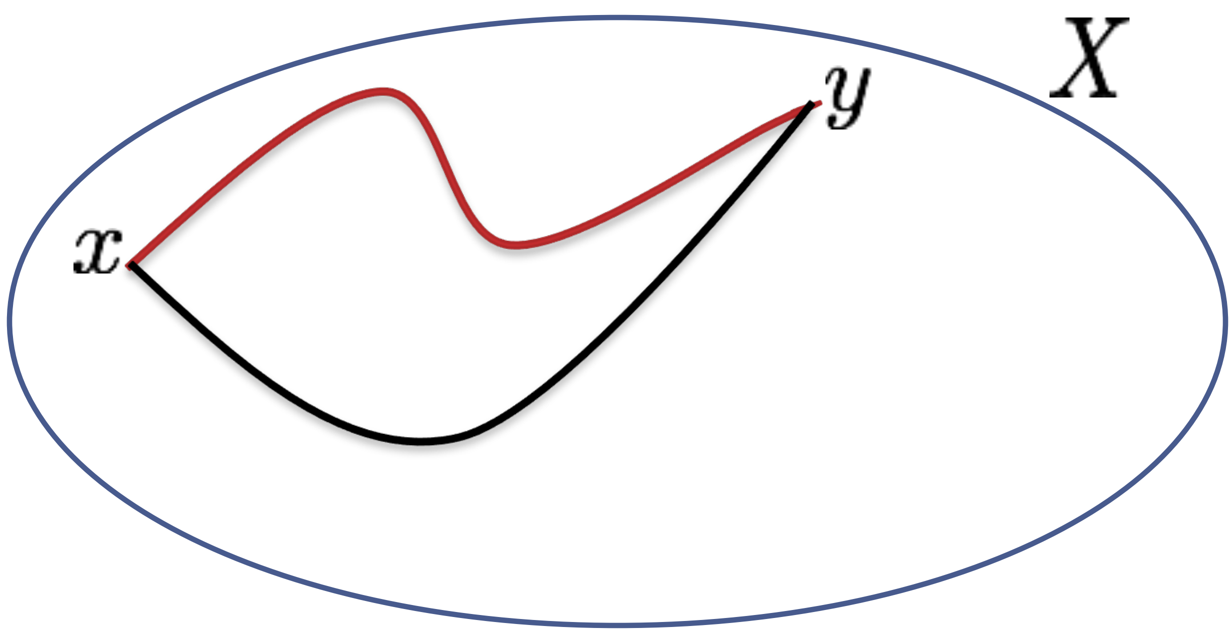
$p : [0, 1] \to X$

where $p(0) = x$ and $p(1) = y$



Homotopy Theory

$$x =_X y$$



Homotopy Theory

$$A = B$$

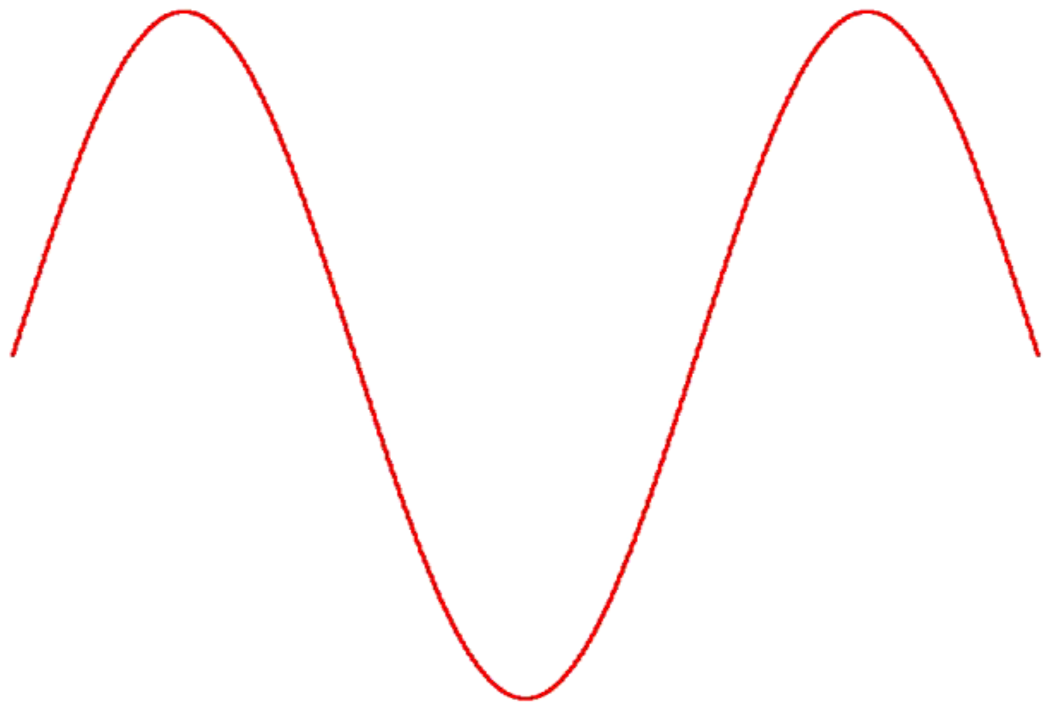
$$=?$$

Homotopy Theory

A homotopy between a pair of continuous maps
$f : X1 \to X2$ and $g : X1 \to X2$
is a continuous map $H : X1 \times [0, 1] \to X2$
satisfying $H(x, 0) = f(x)$ and $H(x, 1) = g(x)$.

If there is such a function H then we say f and g are homotopic, written $f \sim g$.

Homotopy Theory

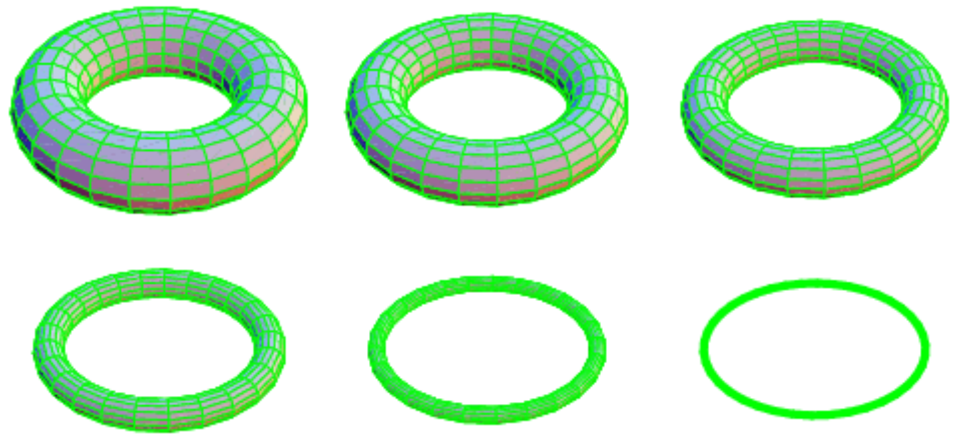


Johnson, Chris. " Path Homotopy Animation." Youtube. Youtube, 11 May 2009. Web. 9 July 2016.

Homotopy Theory

Two spaces X and Y are homotopy equivalent if there are maps $f : X \to Y$ and $f' : Y \to X$ such that $f' \circ f \sim idX$ and $f \circ f' \sim idY$.

Homotopy Theory



Rowland, Todd. "Homotopic." From *MathWorld--A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld.wolfram.com/Homotopic.html*

Homotopy Theory

$$A = B$$

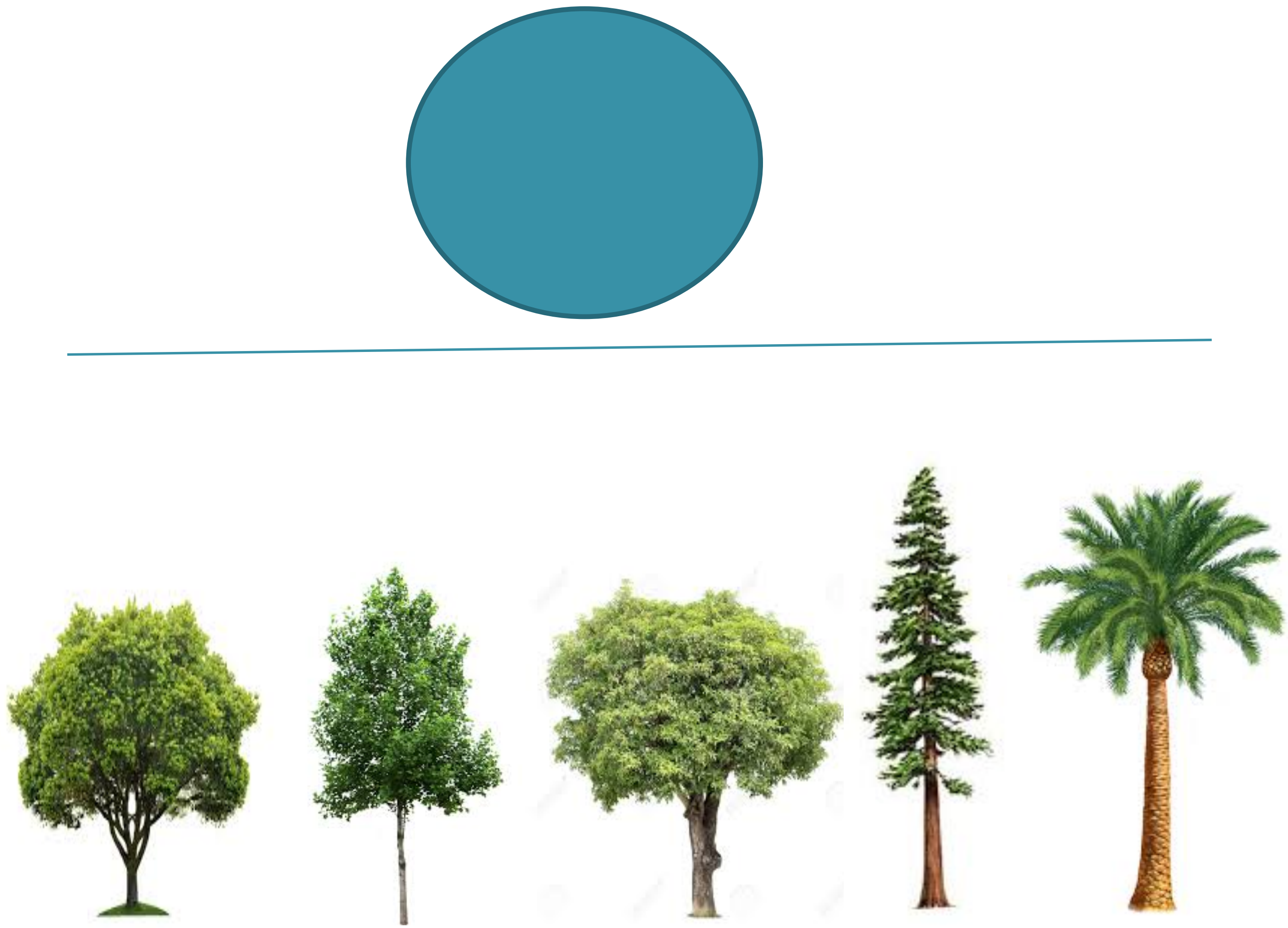
$$=?$$

Homotopy Theory

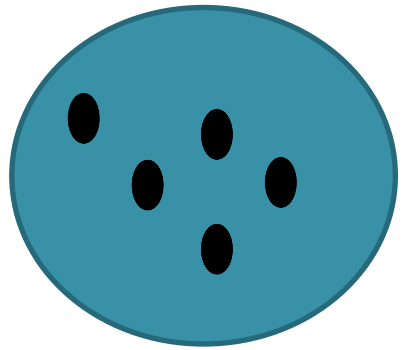
$$A = B$$



CN as identifications



CN as identifications



CN as identifications

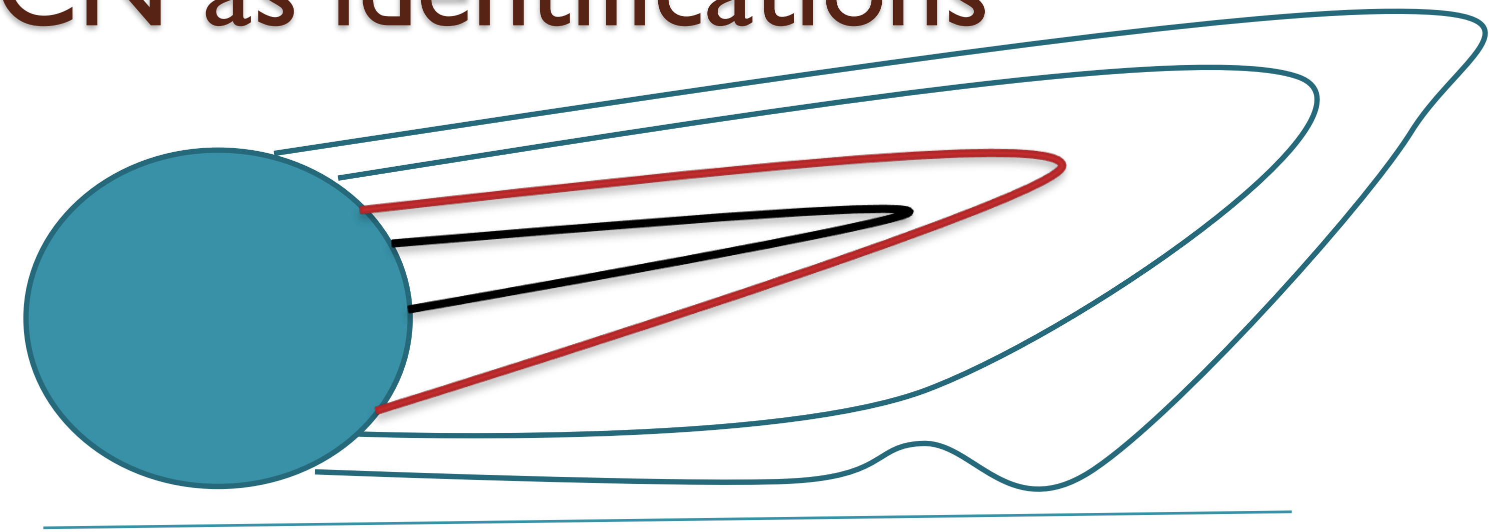
CN as identifications

Sameness

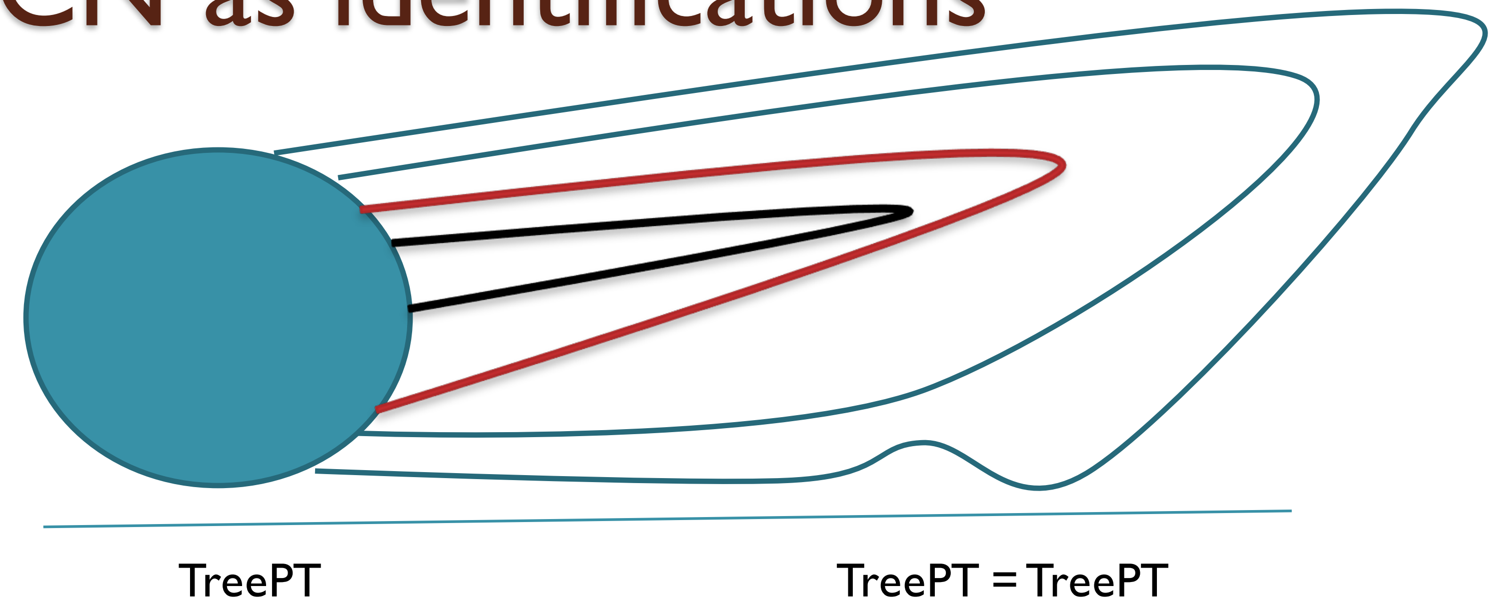
CN as identifications

Sameness Unlikeness

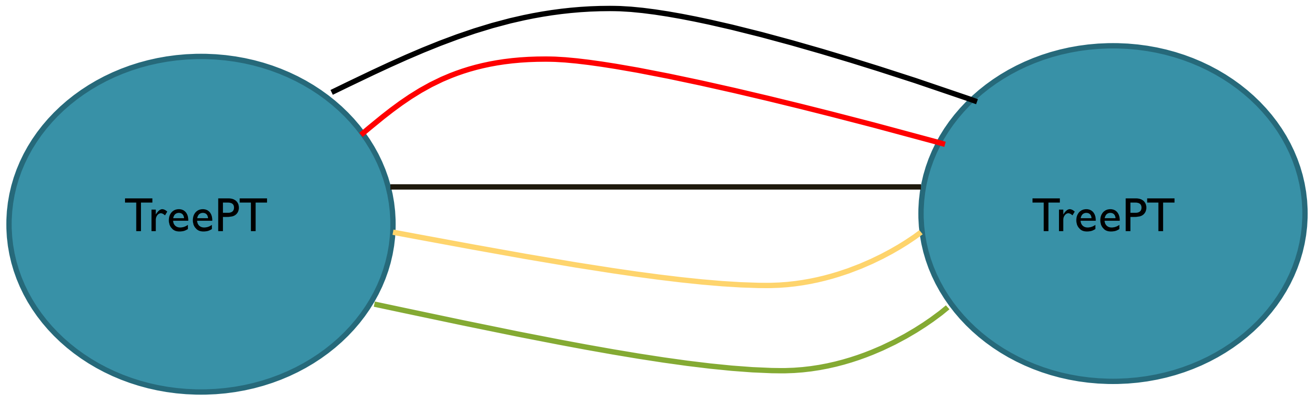
CN as identifications



CN as identifications

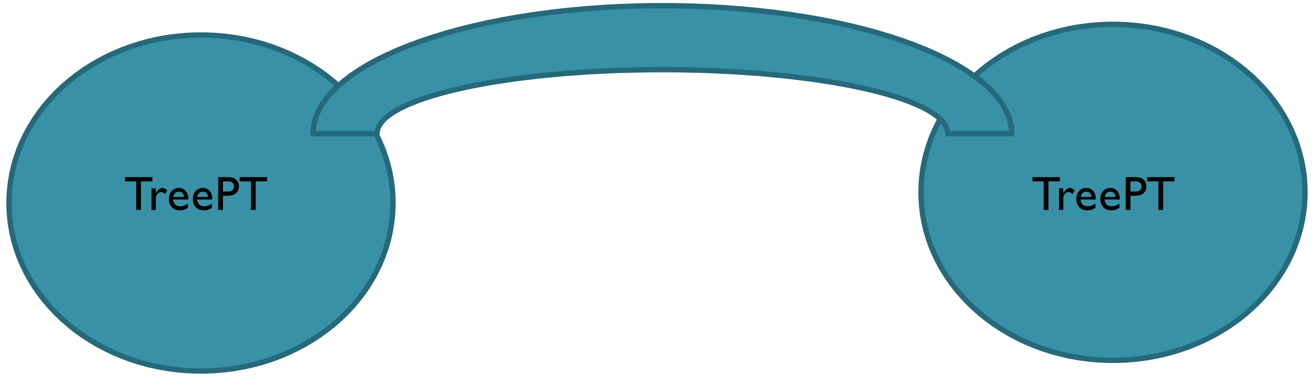


CN as identifications



TreePT = TreePT

CN as identifications

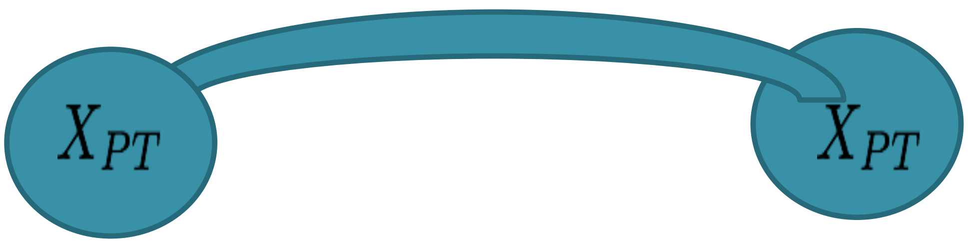


TreePT = TreePT

CN as identifications

when X has no aspects, X is :

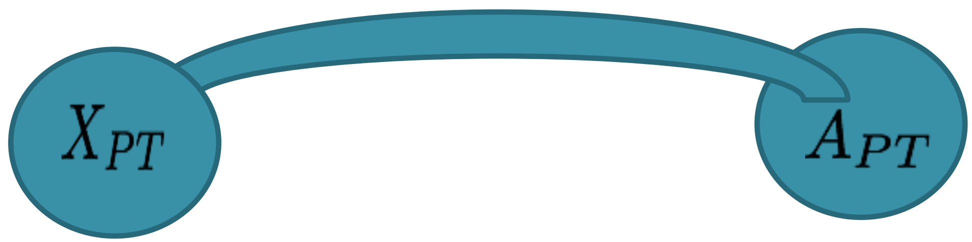
$$X_{PT} = X_{PT}$$



CN as identifications

when X has one aspect, namely A, X is:

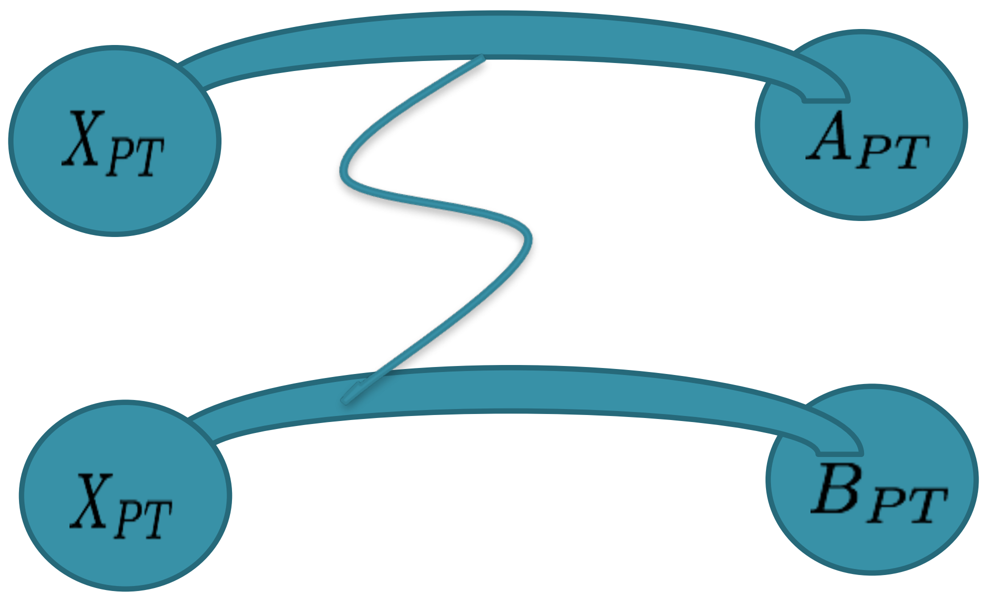
$$X_{PT} = A_{PT}$$



CN as identifications

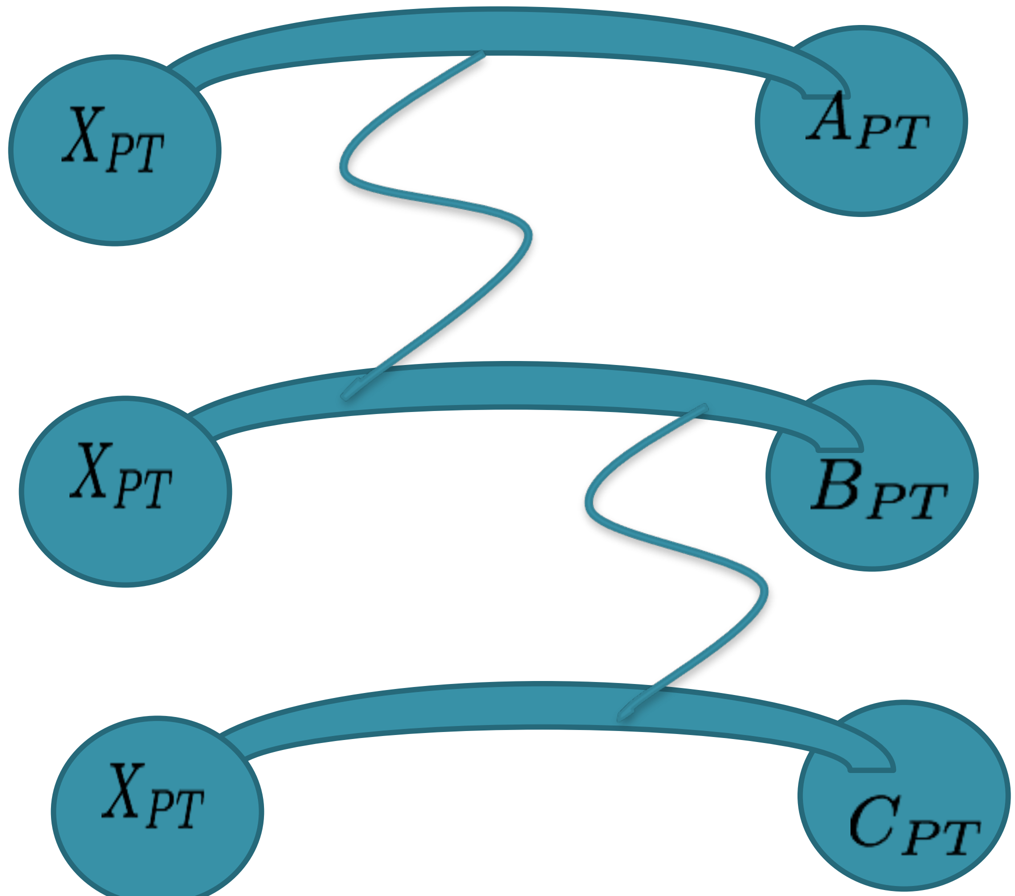
when X has two aspects, namely A and B, X is:

$$(X_{PT} = A_{PT}) = (X_{PT} = B_{PT})$$



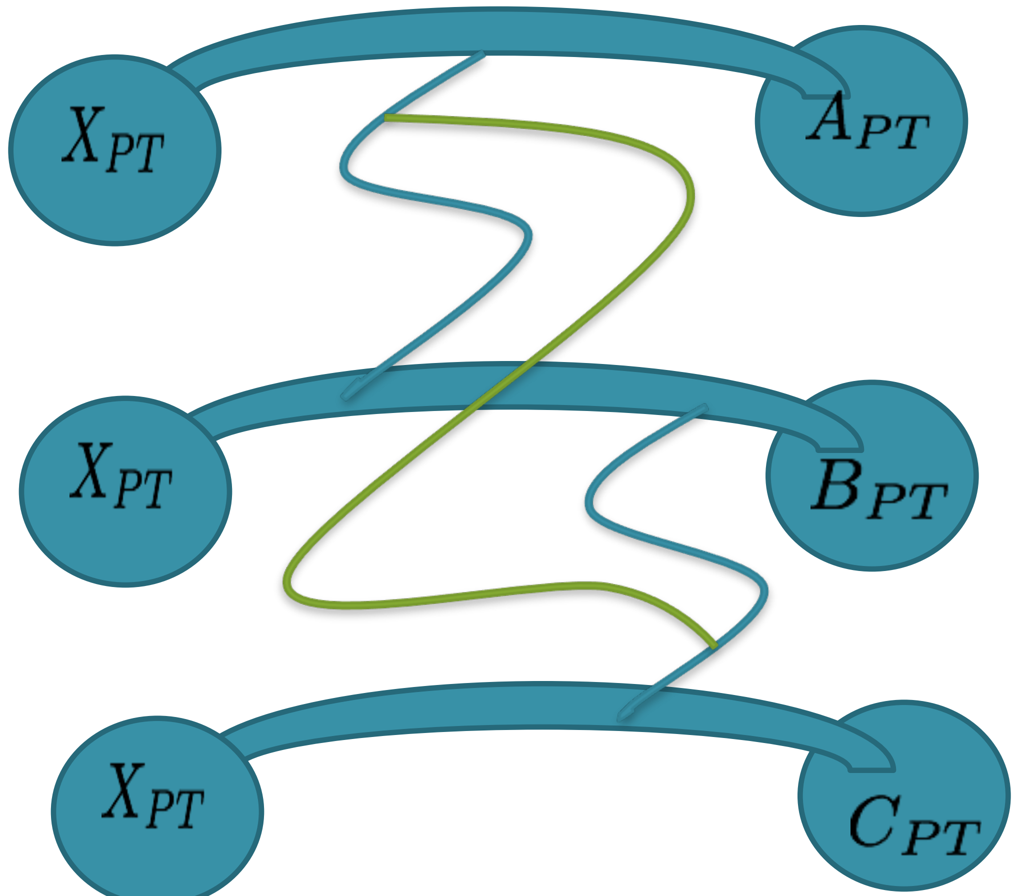
CN as identifications

when X has three aspects, namely A, B and C, X is:

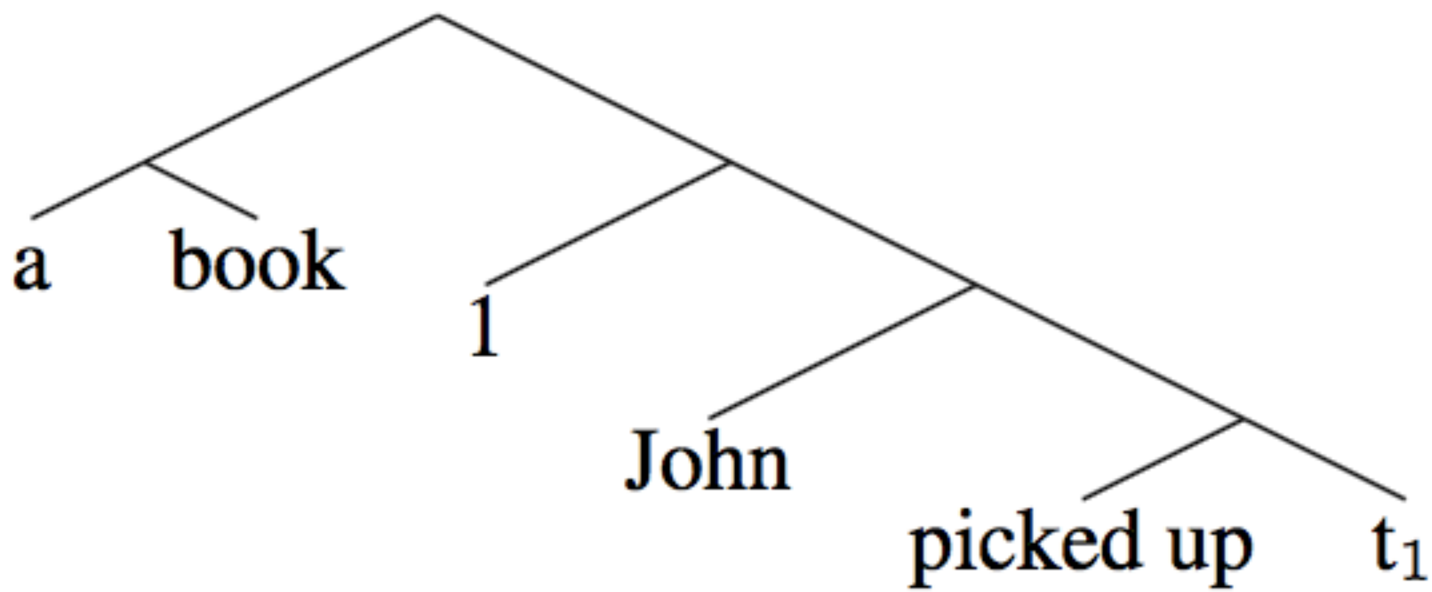


CN as identifications

when X has three aspects, namely A, B and C, X is:



Heim and Kratzer's framework



Heim and Kratzer's framework

$$
\begin{aligned}
&= [\![\text{a book}]\!]^{g}(\lambda b.[\![\text{John picked up t}_1]\!]^{g[1\to b]}) && \text{PA}\\
&= [\![\text{a book}]\!]^{g}(\lambda b.[\![\text{picked up } t_1]\!]^{g[1\to b]}([\![John]\!]^{g[1\to b]})) && \text{FA}\\
&= [\![\text{a book}]\!]^{g}(\lambda b.[\![\text{picked up } t_1]\!]^{g[1\to b]}(John)) && \text{Lex John}\\
&= [\![\text{a book}]\!]^{g}(\lambda b.[\![\text{picked up }]\!]^{g[1\to b]}([\![t_1]\!]^{g[1\to b]})(John)) && \text{FA}\\
&= [\![\text{a book}]\!]^{g}(\lambda b.[\![\text{picked up }]\!]^{g[1\to b]}(b)(John))
\end{aligned}
$$

Heim and Kratzer's framework

$$
\begin{array}{lr}
= [\![\text{a book}]\!]^g(\lambda b.[\![\text{John picked up t}_1]\!]^{g[1\to b]}) & \text{PA} \\
= [\![\text{a book}]\!]^g(\lambda b.[\![\text{picked up } t_1]\!]^{g[1\to b]}([\![John]\!]^{g[1\to b]})) & \text{FA} \\
= [\![\text{a book}]\!]^g(\lambda b.[\![\text{picked up } t_1]\!]^{g[1\to b]}(John)) & \text{Lex John} \\
= [\![\text{a book}]\!]^g(\lambda b.[\![\text{picked up }]\!]^{g[1\to b]}([\![t_1]\!]^{g[1\to b]})(John)) & \text{FA} \\
= [\![\text{a book}]\!]^g(\lambda b.[\![\text{picked up }]\!]^{g[1\to b]}(b)(John)) &
\end{array}
$$

Lemma 1. *Given an element b of type Book, the physical component of b exists and it is an inhabitant of type Physical.*

Heim and Kratzer's framework

Lemma 1. *Given an element b of type Book, the physical component of b exists and it is an inhabitant of type Physical.*

Proof. Let b is an element of *Book* then b is a path from:

$$(BookPrototype =_{\mathcal{U}_i} PhysicalPrototype)$$

to $(BookPrototype =_{\mathcal{U}_i} InformationalPrototype)$. Every path from a type A to a type B induces two functions, one from A to B and the other from B to A. So the path b induces a function f_b from $(BookPrototype =_{\mathcal{U}_i} InformationalPrototype)$ to $(BookPrototype =_{\mathcal{U}_i} PhysicalPrototype)$. By ANN $(BookPrototype =_{\mathcal{U}_i} InformationalPrototype)$ and $(BookPrototype =_{\mathcal{U}_i} PhysicalPrototype)$ are not empty and ANN gives us one member of each namely x and y, which are dependent on b. Then $f_b(x)$ is a path from *BookPrototype* to *PhysicalPrototype*. But $(f_b(x))^{-1}$ is a path from *PhysicalPrototype* to *BookPrototype*. The path $y \circ (f_b(x))^{-1}$ is a path from *PhysicalPrototype* to itself which means it is an element of the $Physical$ type. $y \circ (f_b(x))^{-1}$ is an inhabitant of type $Physical$ and it is determined by b therefore it can be considered as the physical component of b. □

Heim and Kratzer's framework

Lemma 1. *Given an element b of type Book, the physical component of b exists and it is an inhabitant of type Physical.*

$$\mathsf{book\text{-}is\text{-}physical} : (b : \mathsf{Book}) \rightarrow \mathsf{Physical}$$

$$\mathsf{book\text{-}is\text{-}physical}\ b =$$

$$\mathsf{PhysicalPrototype} =\langle\ \mathsf{inverse\text{-}p(coe!}\ (\mathsf{fst}\ b)$$
$$(\ (\mathsf{not\text{-}empty.bPoint}\ (\mathsf{snd}\ b))))\ \rangle$$
$$\mathsf{BookPrototype} =\langle$$
$$(\mathsf{not\text{-}empty.aPoint}\ (\mathsf{snd}\ b))\ \rangle$$
$$\mathsf{PhysicalPrototype}\ \blacksquare$$

Heim and Kratzer's framework

$$
\begin{array}{lr}
= [\![\text{a book}]\!]^g(\lambda b.[\![\text{John picked up t}_1]\!]^{g[1\to b]}) & \text{PA} \\
= [\![\text{a book}]\!]^g(\lambda b.[\![\text{picked up } t_1]\!]^{g[1\to b]}([\![John]\!]^{g[1\to b]})) & \text{FA} \\
= [\![\text{a book}]\!]^g(\lambda b.[\![\text{picked up } t_1]\!]^{g[1\to b]}(John)) & \text{Lex John} \\
= [\![\text{a book}]\!]^g(\lambda b.[\![\text{picked up }]\!]^{g[1\to b]}([\![t_1]\!]^{g[1\to b]})(John)) & \text{FA} \\
= [\![\text{a book}]\!]^g(\lambda b.[\![\text{picked up }]\!]^{g[1\to b]}(b)(John)) &
\end{array}
$$

Functional Application(revised): if α is a constituent with β and γ as its daughters, then if $[\![\beta]\!]$ is a function whose domain contains $[\![\gamma]\!]$ or an aspect of $[\![\gamma]\!]$, then $[\![\alpha]\!]=[\![\beta]\!]([\![\gamma]\!]_\beta)$

where $[\![\gamma]\!]_\beta = [\![\gamma]\!]$ if $[\![\gamma]\!]$ is in the domain of $[\![\beta]\!]$, and if $[\![\beta]\!]$ is a function whose domain contains an aspect $[\![\chi]\!]$ of $[\![\gamma]\!]$, $[\![\gamma]\!]_\beta$ is the aspect $[\![\chi]\!]$.

Heim and Kratzer's framework

$= [\![\text{a book}]\!]^g(\lambda b.[\![\text{John picked up t}_1]\!]^{g[1\to b]})$ PA
$= [\![\text{a book}]\!]^g(\lambda b.[\![\text{picked up } t_1]\!]^{g[1\to b]}([\![John]\!]^{g[1\to b]}))$ FA
$= [\![\text{a book}]\!]^g(\lambda b.[\![\text{picked up } t_1]\!]^{g[1\to b]}(John))$ Lex John
$= [\![\text{a book}]\!]^g(\lambda b.[\![\text{picked up }]\!]^{g[1\to b]}([\![t_1]\!]^{g[1\to b]})(John))$ FA
$= [\![\text{a book}]\!]^g(\lambda b.[\![\text{picked up }]\!]^{g[1\to b]}(b)(John))$
$= [\![\text{three book}]\!]^g(\lambda b.[\![\text{picked up }]\!]^{g[1\to b]}(\varphi_b)(John))$

FA

$= [\![\text{a book}]\!]^g(\lambda b.[\lambda h.1 \text{ iff } h \text{ picked up } \varphi_b](John))$ Lex picked up, λ-Conv.
$= [\![\text{a book}]\!]^g(\lambda b.1 \text{ iff John picked up } \varphi_b)$ λ-Conv.
$= 1$ iff there exists $x \in Book$
such that John picked up φ_x by (20) λ-Conv.

Heim and Kratzer's framework

(2) a. John picked up three books.

b. John mastered three books.

c. John picked up and mastered three books.

Heim and Kratzer's framework

(2) a. John picked up three books.
    b. John mastered three books.
    c. John picked up and mastered three books.

$[\![\text{three books}]\!] = [\![\text{three}]\!]\,([\![\text{books}]\!])$
$= \lambda\gamma \in D_{\langle Book, \mathcal{U}\rangle}.1$ iff there exist $x, y, z \in Book$
such that $\gamma(x), \gamma(y)$ and $\gamma(z)$ are inhabitable

Heim and Kratzer's framework

(2) a. John picked up three books.

b. John mastered three books.

c. John picked up and mastered three books.

$[\![(2a)]\!]^g = 1$ iff there exist $x, y, z \in Book$ such that John picked up φ_x, φ_y and φ_z

$[\![(2b)]\!]^g = 1$ iff there exist $x, y, z \in Book$ such that John mastered ρ_x, ρ_y and ρ_z

$[\![(2c)]\!]^g = 1$ iff there exist $x, y, z \in Book$ such that $John$ mastered ρ_x, ρ_y and ρ_z and $John$ picked up φ_x, φ_y and φ_z

Semantic computation using HoTT logic

Theorem 4. $(h : \mathit{Human})(a : \forall(P : \mathit{Book} \to \mathcal{U}_i) \to (\Sigma\, \mathit{Book}\, P))$
$(p : \mathit{Physical} \to \mathit{Human} \to \mathcal{U}_j) \to \Sigma\, \mathit{Physical}\, (\lambda x \to p\; x\; h)$

$$
\begin{aligned}
&\text{john-picked-up-a-book} : (j : \text{Human}) \\
&\qquad (a : \forall\, (P : \text{Book} \to \text{Type}\ \_) \\
&\qquad \to (\Sigma\ \text{Book}\ P)) \\
&\qquad (p : \text{Physical} \to \text{Human} \to \text{Type}\ \_) \\
&\qquad \to \Sigma\ \text{Physical}\ (\lambda\, x \to p\ x\ j) \\
&\text{john-picked-up-a-book}\ j\ a\ p = \\
&\qquad (\text{a-book-is-a-physical}\ a) \\
&\qquad (\text{p-fix-middle}\ j\ p)
\end{aligned}
$$

Future work

- Formal semantics of natural languages
  - “possible world”?
- HoTT
  - subtyping?
- Homotopy
  - Prefiguration as theorem?

References

[1] N. Asher. A type driven theory of predication with complex types. *Fundamenta Informaticae*, 84(2):151–183, 2008.

[2] N. Asher. *Lexical meaning in context: A web of words*. Cambridge University Press, 2011.

[3] N. Asher and J. Pustejovsky. A type composition logic for generative lexicon. *Journal of Cognitive Science*, 6:1–38, 2006.

[4] S. Awodey. Homotopy type theory. In *Logic and Its Applications*, pages 1–10. Springer, 2015.

[5] G. Brunerie, E. Cavallo, K.-B. Hou, N. Kraus, D. Licata, and C. Sattler. Development of homotopy type theory in agda. `https://github.com/HoTT/HoTT-Agda`, 2015.

[6] S. Chatzikyriakidis and Z. Luo. An account of natural language coordination in type theory with coercive subtyping. In *Constraint Solving and Language Processing*, pages 31–51. Springer, 2013.

[7] S. Chatzikyriakidis and Z. Luo. Individuation criteria, dot-types and copredication: A view from modern type theories. In *ACL anthology*, 2015.

[8] A. Church. A formulation of the simple theory of types. *The journal of symbolic logic*, 5(02):56–68, 1940.

[9] R. Cooper. Copredication, quantification and frames. In *Logical aspects of computational linguistics*, pages 64–79. Springer, 2011.

[10] M. G. H. Gotham. *Copredication, quantification and individuation*. PhD thesis, UCL (University College London), 2014.

[11] I. Heim and A. Kratzer. *Semantics in generative grammar*. Blackwell textbooks in linguistics. Blackwell publishers, Cambridge (Mass.), Oxford, 1998. ISBN 0-631-19712-5. URL `http://opac.inria.fr/record=b1080204`.

[12] Z. Luo. *Computation and reasoning: a type theory for computer science*. Oxford University Press, Inc., 1994.

[13] Z. Luo. Coercive subtyping. *Journal of Logic and Computation*, 9(1): 105–130, 1999.

[14] Z. Luo. Common nouns as types. In *Logical aspects of computational linguistics*, pages 173–185. Springer, 2012.

[15] Z. Luo. Formal semantics in modern type theories with coercive subtyping. *Linguistics and Philosophy*, 35(6):491–513, 2012.

[16] R. Montague. Formal philosophy. 1975.

[17] B. Nordström, K. Petersson, and J. M. Smith. *Programming in Martin-Löf's type theory*, volume 200. Oxford University Press Oxford, 1990.

[18] B. Partee. Compositionality. *Varieties of formal semantics*, 3:281–311, 1984.

[19] J. Pustejovsky. The generative lexicon. *Computational linguistics*, 17 (4):409–441, 1991.

[20] A. Ranta. Type-theoretical grammar. 1994.

[21] Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. `http://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.