Variables, pronouns, relative clauses, and movement

1 Notes on variables

(1) **Some math "sentences":**

a. $1 = 2 - 1$ — a sentence with no variables; not context-sensitive

b. $n = 2 - 1$ — a sentence with a variable; context-sensitive

c. $\forall n\ (2(n + 1) = 2n + 2)$ — a sentence with a variable; *not* context-sensitive

- We say (1b) contains a *free variable* because the truth of the sentence depends on the context. In particular, the sentence is true iff the variable "n" is interpreted as 1.
- The truth of sentence (1c), like (1a), does not depend on the context at all.

(2) **Some terminology, using (1c) as an example:**

$$\underset{binder}{\forall n} \underbrace{\left(2(\underset{bound}{n}+1) = 2\underset{bound}{n}+2\right)}_{scope}$$

- *Binders* control the interpretation of a particular variable within a certain part of its structure, which we call its *scope*. Here, $\forall$ *binds* the variable n in its scope.
- We call variables that are in the scope of a matching binder *bound variables*.

With this in mind, let's revisit (3) from PS2 (H&K p. 10):

(3) When is the following true? $\{x : \{y : y \text{ likes } x\} = \emptyset\} = \{x : \{x : x \text{ likes } x\} = \emptyset\}$

Let's call the mapping between free variables and their values *assignment*.

2 Pronouns

This free/bound terminology is useful for natural language as well:

(4) a. John likes Mary. — a sentence with no variables; not assignment-sensitive

b. John likes him. — a sentence with a variable; assignment-sensitive

c. Every boy likes himself. — a sentence with a variable; *not* assignment-sensitive

We'll formalize this by giving each pronoun a numerical *index*. We'll compute denotations relative to an *assignment function*, which is a function from the set of indices ($\mathbb{N}$) to D_e.

(5) **Pronouns Rule (to be replaced later):**
If α is a pronoun, g is a variable assignment, and $g(i)$ is defined, then $[\![\alpha_i]\!]^g = g(i)$.

(6) Suppose g is a function and $g(3) = \text{Sam} \in D_e$.

a. $[\![\text{him}_3]\!]^g = \text{Sam}$

b. $[\![\text{John likes him}_3]\!]^g = 1$ iff John likes Sam

Q: Does it matter what g returns for other values in (6)?

A: No. It might even be undefined for other values.

Q: Why did we use 3? Does the number matter?

A: The choice of number was arbitrary, but it is important whether or not we reuse numbers:

(7) a. He_2 thinks that he_2 is smart.

b. He_2 thinks that he_7 is smart.

Q: Does the assignment function affect other parts of the sentence?

A: No. "John" and "likes" are *constants*, meaning their values are the same no matter the assignment: for any assignment function f, $[\![\text{John}]\!]^f = \text{John}$.

Warning: There's a section of H&K (pp. 92–109) where they just use notation like $[\![\text{him}]\!]^{\text{John}} =$ John, which only accommodates one variable at a time, but then they introduce their actual notation on page 110, which we use here.

3 Rules with assignments

In order to work with assignment functions, we need to modify all our existing rules so that they pass assignment functions. These definitions are based on H&K p. 95:

(8) **Terminal Nodes (TN):** (unchanged)
If α is a terminal node, $[\![\alpha]\!]$ is specified in the lexicon.[1]

(9) **Non-branching Nodes (NN):**
If α is a non-branching node, and β is its daughter node, then, for any assignment g, $[\![\alpha]\!]^g = [\![\beta]\!]^g$.

(10) **Functional Application (FA):**
If α is a branching node, $\{\beta, \gamma\}$ is the set of α's daughters, then, for any assignment g, if $[\![\beta]\!]^g$ is a function whose domain contains $[\![\gamma]\!]^g$, then $[\![\alpha]\!]^g = [\![\beta]\!]^g([\![\gamma]\!]^g)$.

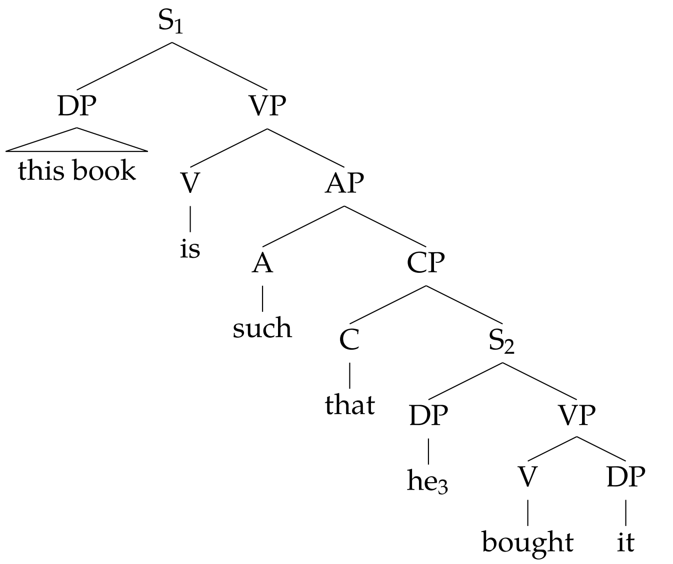
(11) **Predicate Modification (PM):**
If α is a branching node, $\{\beta, \gamma\}$ is the set of α's daughters, then, for any assignment g, if $[\![\beta]\!]^g$ and $[\![\gamma]\!]^g$ are both of type $\langle e, t\rangle$, then $[\![\alpha]\!]^g = \lambda x \in D_e \,.\, [\![\beta]\!]^g(x) = 1$ and $[\![\gamma]\!]^g = 1$.

[1] H&K proposes (p. 94) to still use $[\![\alpha]\!]$ without an assignment function superscript for *constants*, i.e. if $[\![\alpha]\!]^g$ is the same value for all assignment functions g.

4 *Such that*

The English expression *such that* allows us to construct some complex expressions.[2]

(12) ${}^?$ This book is such that he$_3$ bought it. ($g(3)$ = John)



Here, (12) does not seem assignment-dependent. But the Principle of Compositionality states that $[\![S_1]\!]$ be computed based on the meaning of $[\![S_2]\!]$, which contains a pronoun and *is* assignment-dependent.

Idea: *Such* binds *it*, doing the work of creating a *predicate* out of the assignment-dependent sentence "John bought it."

(13) ***Such* Rule (temporary):**[3]
$[\![\text{such}_i\ \gamma]\!]^g = \lambda x_e\ .\ [\![\gamma]\!]^{[i \mapsto x] || g}$

$[i \mapsto x] \mid\mid g$ is the *combination* of functions $[i \mapsto x]$ and g:

(14) **Definition: function combination**
$$f \mid\mid g \equiv \lambda x\ .\ \begin{cases} f(x) & \text{if } x \in \text{domain}(f) \\ g(x) & \text{otherwise} \end{cases}$$
Read "f or else g."

Let's compute $[\![S_1]\!]^g$ with the following global assignment function: $g = \begin{bmatrix} 3 \mapsto \text{John} \\ 11 \mapsto \text{Tama} \end{bmatrix}$.
Assume $[\![\text{that}]\!]$ = Id.

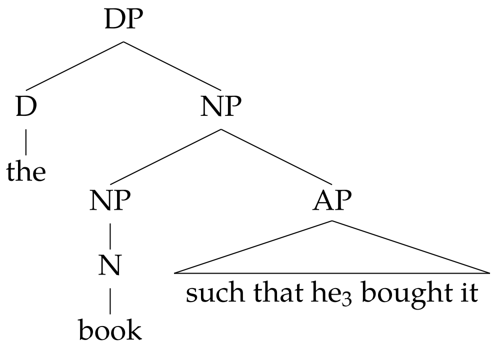
Warning: H&K uses $g^{x/i}$ notation for $[i \mapsto x] \mid\mid g$, but I think it's confusing so I don't use it.[4]

[2] Unfortunately, the use of *such that* sounds "unlyrical" (Quine, 1960, §23)... but we'll ignore that here.
[3] "Such" does not have a type. That's why it can only be interpreted using the *Such* Rule.
[4] For one, I've also seen very similar notation "$g(x/a)$" for a function that maps x to a, which is the reverse of

We can also use *such that* to construct (slightly awkward) *relative clauses*:

(15) $^{?}$ the book such that he$_3$ bought it

The semantics for *such* above works perfectly fine here.



"...the peculiar genius of the relative clause is that it creates from a sentence '...x...' a complex adjective summing up what that sentence says about x." — Quine (1960, §23)

(16) $[\![\text{the}]\!] = \lambda f : f \in D_{\langle e,t \rangle}$ and there is exactly one x such that $f(x) = 1$.
the unique y such that $f(y) = 1$

Binding multiple variables:

(17) $^{?}$ This book is such that he$_3$ bought it and then gave it to Sarah.

(18) $^{?}$ the book such that he$_3$ bought it and then gave it to Sarah

Binding no variables (vacuous binding):

(19) * This book is such that today is Monday.

(20) * the book such that today is Monday

The ungrammaticality of these examples shows that binding *no* variables is disallowed by the grammar. This is called *vacuous binding*.

5 Relative clauses and movement

Most relative clauses in English do not use *such that* and a pronoun, but instead involve a *gap*:

(21) **English relative clauses:**

a. the book he$_3$ bought ____

b. the book which he$_3$ bought ____

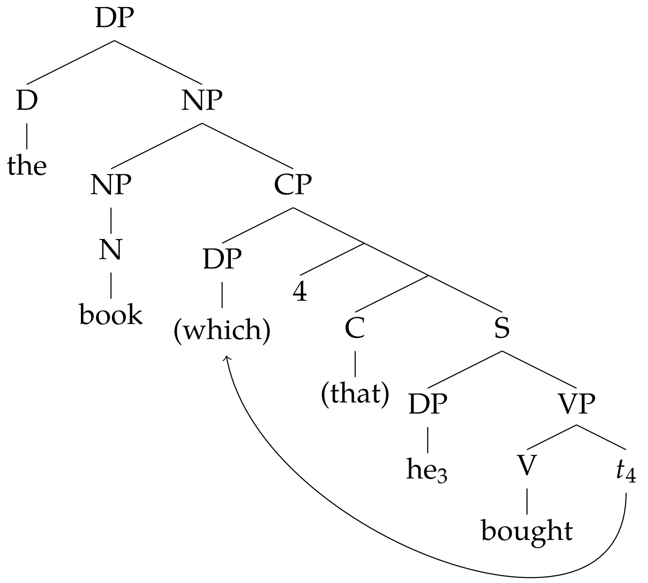
c. the book that he$_3$ bought ____

d. * the book which that he$_3$ bought ____

what H&K mean in their x/i.

Such gapped relative clauses are analyzed as the result of *movement* of the relative pronoun ("which") from the gap position to Spec,CP (Chomsky, 1977, and many others).[5]

Could traces be vacuous? Consider the meaning of ⟦S⟧ here if ⟦VP⟧ = ⟦bought⟧.



Instead:

(22) **The interpretation of movement:**
Pick an arbitrary index i.

a. The base position of movement is replaced with a *trace* with index i: t_i.

b. A *binder index i* is adjoined right under the target position of the movement chain.

(23) **Traces and Pronouns Rule (T&P):** replaces Pronouns Rule in (5)
If α is a pronoun or trace, g is a variable assignment, and $g(i)$ is defined, then $[\![\alpha_i]\!]^g = g(i)$.

(24) **Predicate Abstraction (PA):** (H&K p. 186 version) replaces *Such* Rule in (13)[6]
Let α be a branching node with daughters β and γ, where β dominates only a numerical index i. Then, for any assignment g, $[\![\alpha]\!]^g = \lambda x \,.\, [\![\gamma]\!]^{[i \mapsto x]||g}$.

Let's compute the relative clauses in (21) with global assignment $g = [3 \mapsto \text{John}]$. Assume ⟦that⟧ = Id and ⟦which⟧ = Id.

References

Chomsky, Noam. 1977. On *wh*-movement. In *Formal syntax*, ed. Peter Culicover, Thomas Wasow, and Adrian Akmajian, 71–132. New York: Academic Press.

Chomsky, Noam, and Howard Lasnik. 1977. Filters and control. *Linguistic Inquiry* 8:425–504.

Quine, Willard Van Orman. 1960. *Word and object*. Cambridge.

[5]Both the relative pronoun and complementizer "that" are then pronounced optionally. Following Chomsky and Lasnik (1977), we assume a "Doubly Filled COMP Filter" that states that both positions cannot be pronounced at the same time, explaining (21d).

[6]We can think of "such" as the pronunciation of a lexicalized binder index, not generated through movement.