RELATIVES, VARIABLES, BINDING

ENGLISH SEMANTICS • LECTURE 4

Moreno Mitrović

The Saarland Lectures on Formal Semantics

RELATIVES

What do relative clauses mean?

(1) The house which is empty is available.

What do relative clauses mean?

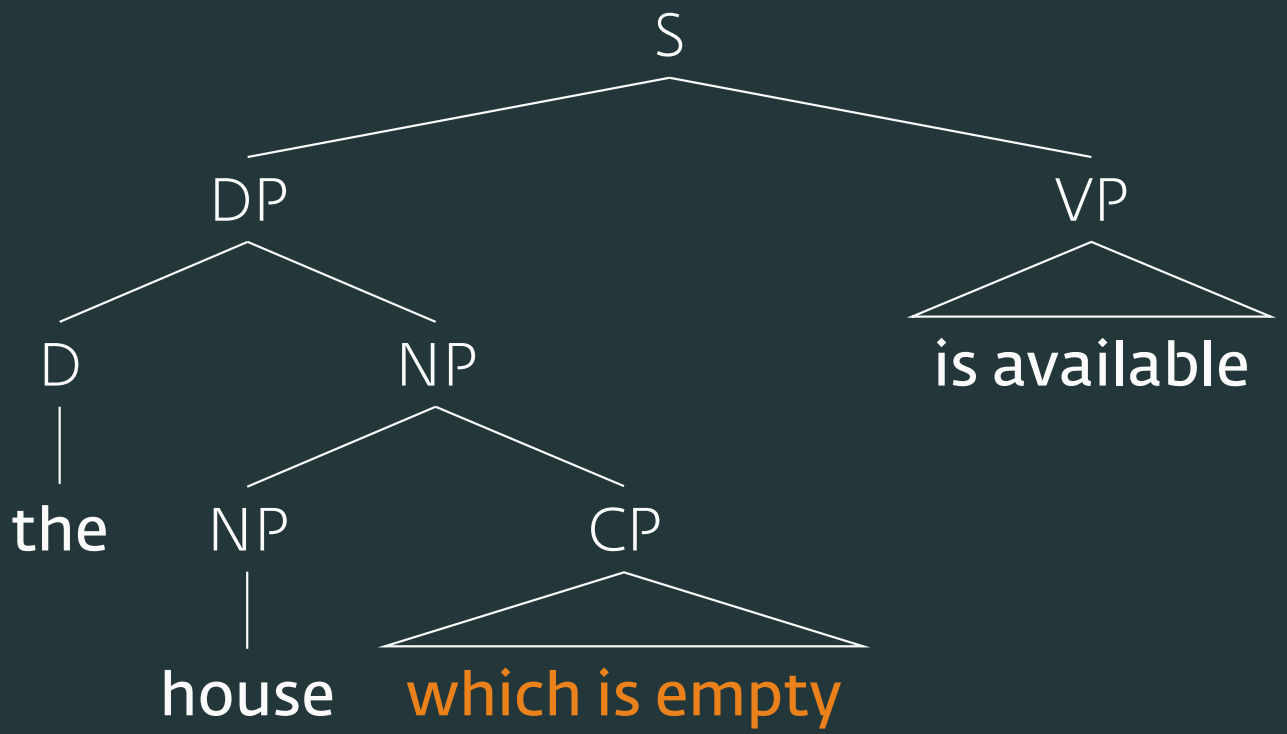
(1) **The house which is empty is available.**

(2) **The empty house is available.**

- Relative clauses (or relatives, for short) are **predicates**.
- But relatives have quite a lot of syntactic structure – what do we do about that?

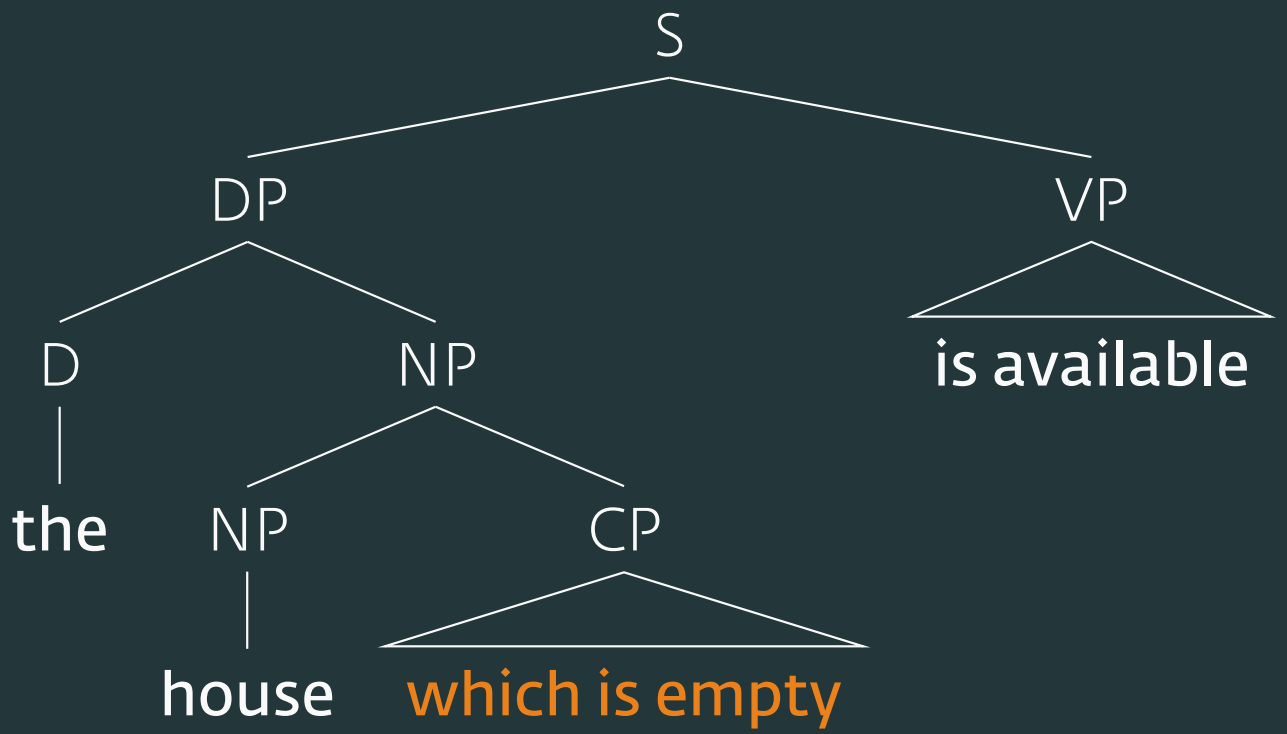
THE SYNTAX OF RELATIVES

(3)



THE SYNTAX OF RELATIVES

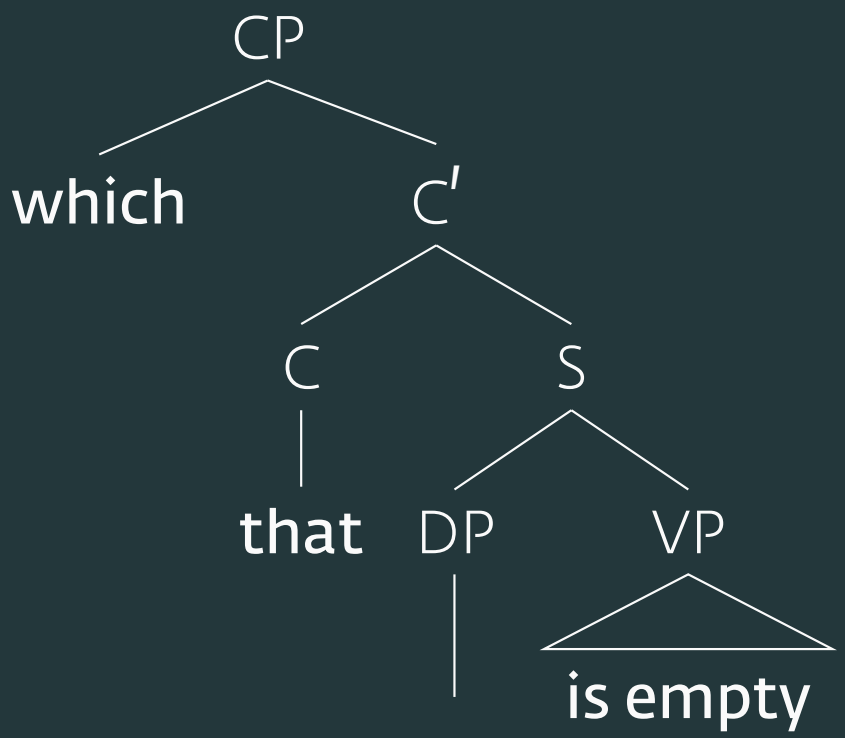
(3)



- Let's calculate the meaning of the NP (and then DP, as we now can) with what we intuitively know.

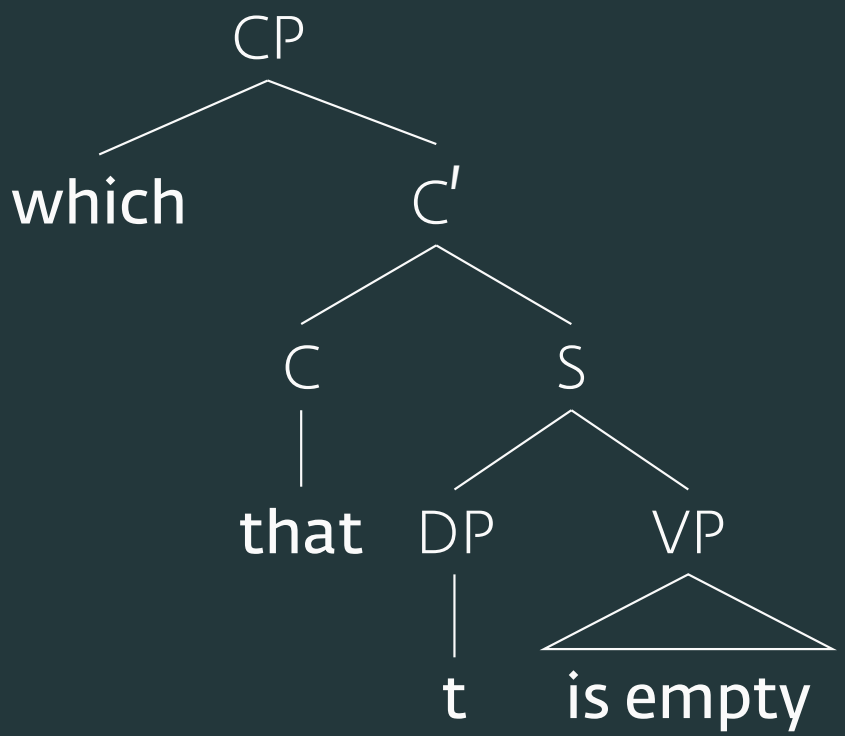
THE SYNTAX OF RELATIVES: A MORE DETAILED VIEW

(4)



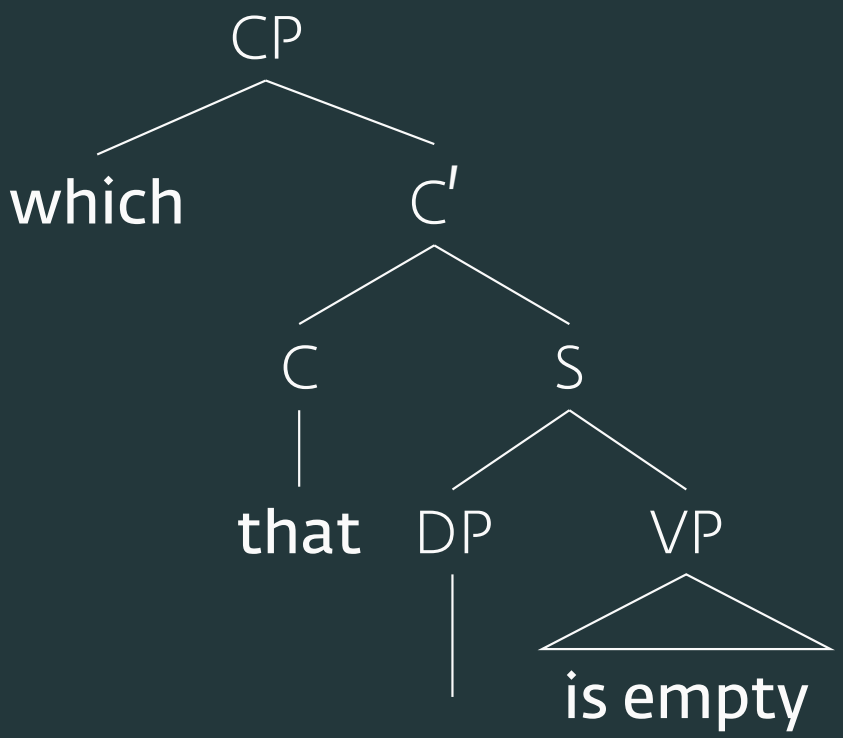
THE SYNTAX OF RELATIVES: A MORE DETAILED VIEW

(4)



THE SYNTAX OF RELATIVES: A MORE DETAILED VIEW

(4)



- It looks like ⟦VP⟧ gets passed onto ⟦CP⟧.
- Does not work in general – what about object-relatives?

The syntax of relatives: a more detailed view

(5) The house, which is empty, is available.

(6) The house, which John abandoned, is available.

THE SYNTAX OF RELATIVES: A MORE DETAILED VIEW

(5) **The house, which is empty, is available.**

(6) **The house, which John abandoned, is available.**

- There are silent elements, **traces**, which tell us whether **the house** is the subject or the object of the relative.
- This is semantically clearly important.
  - A subject-relative should denote a characteristic function of a set of empty things.
  - An object-relative should denote a characteristic function of things John abandoned.

THE MEANING OF TRACES

- We would like **t** to have the same extension as the coreferential **DP**.

THE MEANING OF TRACES

- We would like **t** to have the same extension as the coreferential **DP**.
- We will treat them as variables.

VARIABLES

TRACES AS VARIABLES

- Variables will get us the right semantics for traces.

TRACES AS VARIABLES

- Variables will get us the right semantics for traces.
- Variables denote individuals but **relative to the choice of an assignment of a value**.

TRACES AS VARIABLES

- Variables will get us the right semantics for traces.
- Variables denote individuals but **relative to the choice of an assignment of a value**.
- What is a value assignment?

Assignment (preliminary def.)

An **assignment** is an **individual**.

TRACES AS VARIABLES

- Variables will get us the right semantics for traces.
- Variables denote individuals but **relative to the choice of an assignment of a value**.
- What is a value assignment?

Assignment (preliminary def.)

An **assignment** is an **individual**.

Example

The denotation of **t** under the assignment **Texas** is **Texas** .

TRACES AS VARIABLES

- Variables will get us the right semantics for traces.
- Variables denote individuals but **relative to the choice of an assignment of a value**.
- What is a value assignment?

Assignment (preliminary def.)

An **assignment** is an **individual**.

Example

The denotation of **t** under the assignment **Texas** is **Texas** .

(7) $[\![\mathbf{t}]\!]^{\text{Texas}} = \text{Texas}$

TRACES AS VARIABLES

(8) "$[\![\alpha]\!]^a$" is read as **the denotation of α under a.**

- If α is a trace, then, for any assignment a, $[\![\alpha]\!]^a = a$.

TRACES AS VARIABLES

(8) "$[\![\alpha]\!]^a$" is read as **the denotation of α under a.**

- If α is a trace, then, for any assignment a, $[\![\alpha]\!]^a = a$.
- We now relativise denotations of, say VPs, to their assignments. [BLACKBOARD]

- In the system we've developed so fat, some denotations have assignment, and some don't (i.e., we have assignment-dependent and assignment-independent denotations).
- We need a way to not have assignments complicate the system we have.

(9) a. For any tree α, α is in the domain of $[\![\ ]\!]$ iff for all assignments a and b, $[\![\alpha]\!]^a = [\![\alpha]\!]^b$.

b. If α is in the domain of $[\![\ ]\!]$, then for all assignments a, $[\![\alpha]\!] = [\![\alpha]\!]^a$.

(9) a. For any tree α, α is in the domain of $[\![\ \ ]\!]$ iff for all assignments a and b, $[\![\alpha]\!]^a = [\![\alpha]\!]^b$.

b. If α is in the domain of $[\![\ \ ]\!]$, then for all assignments a, $[\![\alpha]\!] = [\![\alpha]\!]^a$.

- In this way, we now minimally supplement our core principles.

(10) **Lexical Terminals**

If α is a terminal node occupied by a lexical item, then $[\![\alpha]\!]$ is specified in the lexicon.

(11) **Non-branching nodes**

If α is a non-branching node and β its daughter, then, for any assignment a, $[\![\alpha]\!]^a = [\![\beta]\!]^a$.

(12) **Functional Application**

If α is a branching node and $\{\beta, \gamma\}$ its daughters, then, for any assignment a, if $[\![\beta]\!]^a$ is a function whose domain contains $[\![\gamma]\!]^a$, then $[\![\alpha]\!]^a = [\![\beta]\!]^a([\![\gamma]\!]^a)$.

(13) **Predicate Modification**

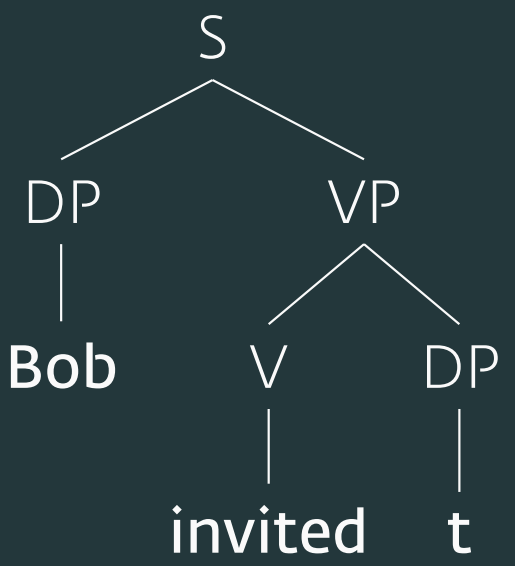
If α is a branching node and $\{\beta, \gamma\}$ its daughters, then, for any assignment a, if $[\![\beta]\!]^a$ and $[\![\gamma]\!]^a$ are both functions of type $\langle \mathbf{e,t} \rangle$, then $[\![\alpha]\!]^a = \lambda x \in D\,.\,[\![\beta]\!]^a(x) = [\![\gamma]\!]^a(x) = 1$.

EXERCISE

Consider a state of affairs with the following properties:

- $D = \{b, j, m, s$
- s invites b and m
- b invites b, j and m
- no other invitations take place.
- $[\![\textbf{Bob}]\!] = b$

(i)



a. Show that $[\![(\text{i})]\!]^m = 1$

b. Which assignments make (i) true? (List them.)

c. Show that $[\![(\text{i})]\!]$ is undefined.: that is, that the tree in (i) is not in the domain of $[\![\ \ ]\!]$.

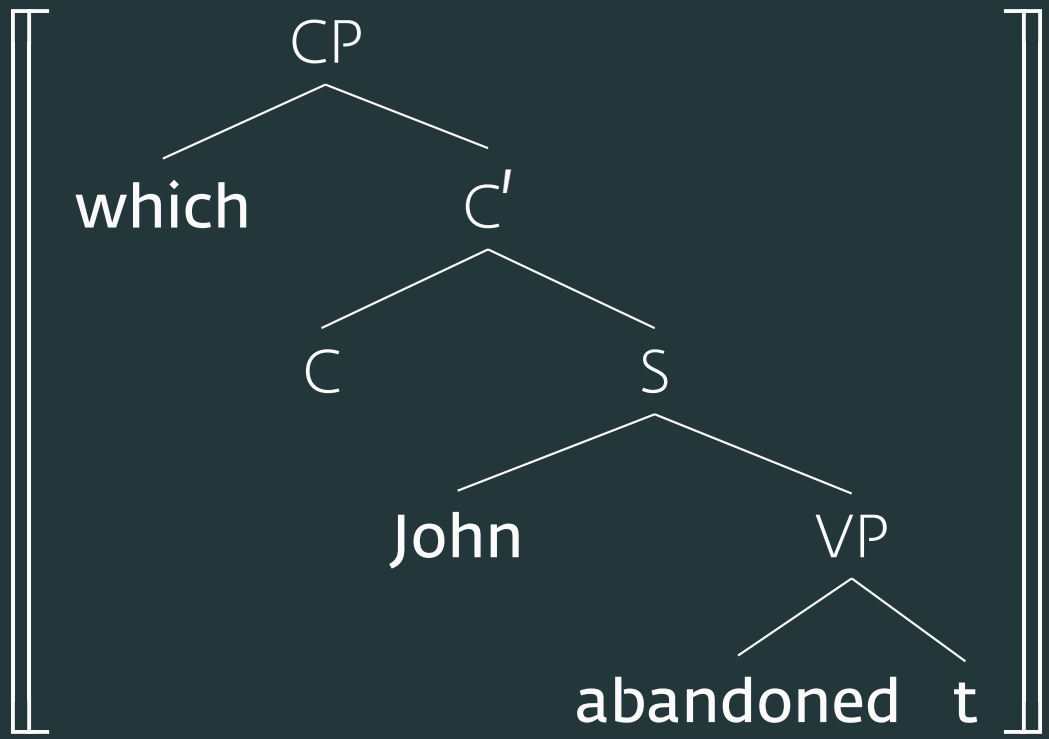
PREDICATE ABSTRACTION

(14) **Predicate Abstraction**
If $[\![\text{"}\alpha]\!]^a$ is a branching node whose daughters are **a relative pronoun** and β, then $[\![\alpha]\!] = \lambda x \in D\,.\,[\![\beta]\!]^x$

- This is identical to λ-abstraction, but note the x in assignment (and not argument) position.

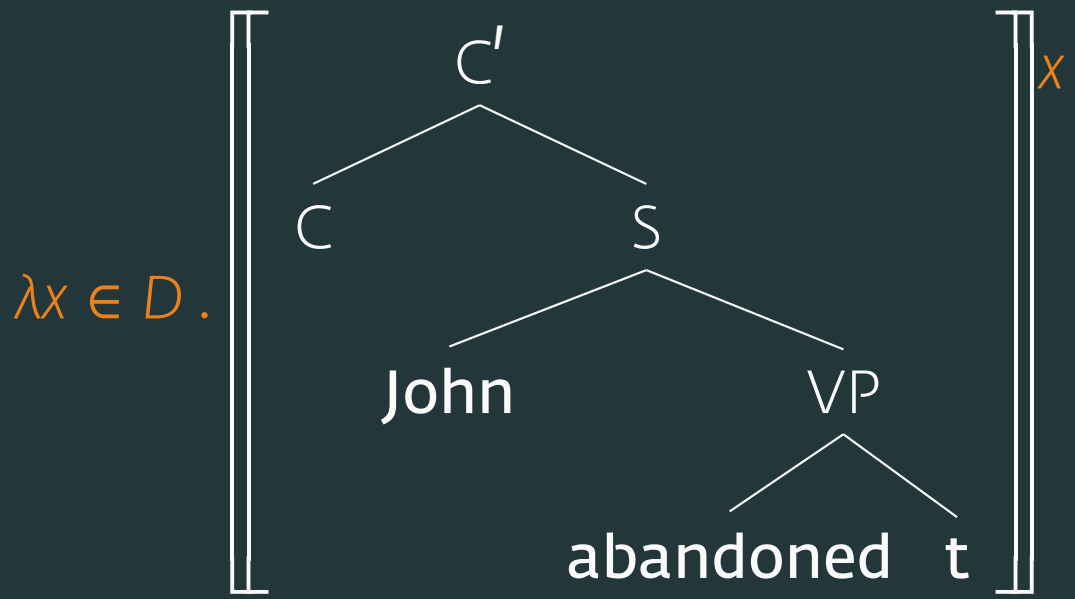
TRYING OUT PREDICATE ABSTRACTION

(Let's try calculating top→bottom first.)



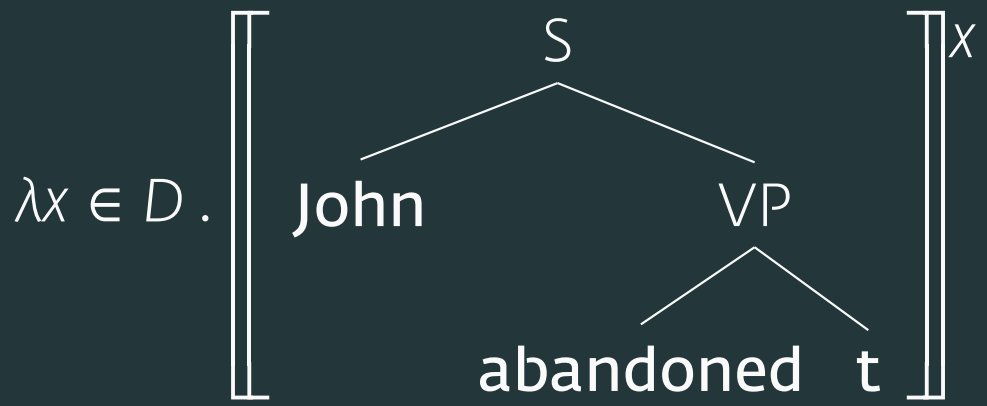
TRYING OUT PREDICATE ABSTRACTION

→ by PA of **which** and C′



TRYING OUT PREDICATE ABSTRACTION

→ by vacuity of C



TRYING OUT PREDICATE ABSTRACTION

→ by FA of **John**

$$\lambda x \in D\,.\,\Big[\!\Big[\; \overset{\text{VP}}{\underset{\text{abandoned}\quad \text{t}}{\wedge}} \;\Big]\!\Big]^{x}\left([\![\textbf{John}]\!]^{x}\right)$$

TRYING OUT PREDICATE ABSTRACTION

→ by (9) applied to **John**

$$\lambda x \in D\,.\,\Big[\!\Big[\; \begin{array}{c} \text{VP} \\ \text{abandoned} \quad \text{t} \end{array} \;\Big]\!\Big]^{x} \Big(\llbracket \textbf{John} \rrbracket\Big)$$

TRYING OUT PREDICATE ABSTRACTION

→ by lexical entry for **John**

$$\lambda x \in D\,.\,\Big[\!\Big[\; \begin{array}{c} \text{VP} \\ \diagup\quad\diagdown \\ \textbf{abandoned}\quad \textbf{t} \end{array} \;\Big]\!\Big]^{x}\,(\text{John})$$

TRYING OUT PREDICATE ABSTRACTION

→ by FA

$$\lambda x \in D \,.\, [\![\textbf{abandoned}]\!]^x\Big([\![\textbf{t}]\!]^x\Big)\Big(\text{John}\Big)$$

TRYING OUT PREDICATE ABSTRACTION

→ by (9) applied to **abandoned**

$$\lambda x \in D \,.\, [\![\textbf{abandoned}]\!]\Big(\text{John}\Big)\Big([\![\textbf{t}]\!]^x\Big)$$

TRYING OUT PREDICATE ABSTRACTION

→ by (9) applied to **t**

$$\lambda x \in D \,.\, [\![\textbf{abandoned}]\!]\big(\text{John}\big)\big(x\big)$$

TRYING OUT PREDICATE ABSTRACTION

→ by lexical entry for **abandon**

$$\lambda x \in D\,.\,[\lambda y \in D\,.\,[\lambda z \in D\,.\,z \text{ abandoned } y]]\big(\text{John}\big)\big(x\big)$$

TRYING OUT PREDICATE ABSTRACTION

$\rightarrow$ by FA_1

$$\lambda x \in D\,.\,[\lambda z \in D\,.\,z\,\text{abandoned}\,x]\big(\text{John}\big)$$

TRYING OUT PREDICATE ABSTRACTION

$\rightarrow$ by FA_2

$$\lambda x \in D \,.\, \text{John abandoned } x$$

Done.

- We had to fiddle with the order of λ-arguments since we've learnt to do β-reduce outside→in.
- Let's try it again, this time in a bottom→top fashion and you'll see that, this way, we never run into an argument-ordering problem.

A VERY QUICK TEST-QUESTION

- Are the two equivalent?

(15) $[\![\text{whom John abandoned t}]\!]^{\text{Charles}}$

(16) $[\![\text{whom John abandoned t}]\!](\text{Charles})$

MULTIPLE VARIABLES

MULTIPLE VARIABLES

'SUCH THAT'

'such that'

- We've used the term **'such that'** a lot in our semantic talk. It turns out it's even more helpful when talking about the semantics of relatives.

'such that'

"The reason for permuting word order in forming relative clauses is to bring the relative pronoun out to the beginning or near it. The task can be exacting in complex cases, and is sometimes avoided by recourse to an alternative construction, the unlyrical "**such that**". This construction demands none of the tricks of word order demanded by "**which**": the responsibility of standing in a singular-term position within the clause is delegated to "**it**", and the responsibility of signalling the beginning of the clause is discharged by "**such that**". Thus, "which I bought" becomes "such that I bought it."

RECAP OF TWO PRINCIPLES

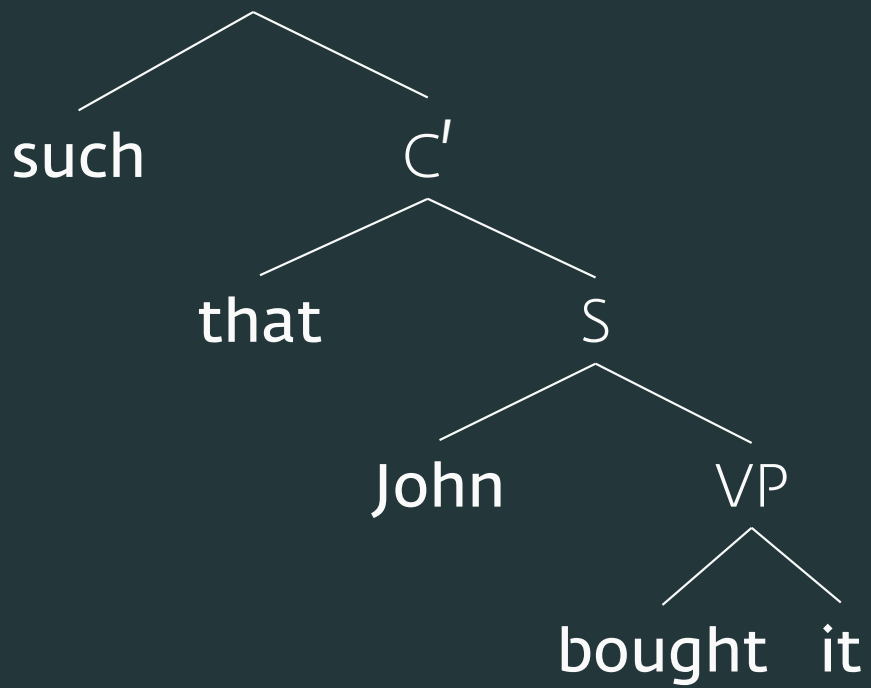
- We've come across two new rules, which we can state in a more explicit format.

(17) **Pronoun Rule**
If α is a pronoun, then for **any assignment** $a \in D$, $[\![\alpha]\!]^a = a$

(18) **Predicate Abstraction**
If α is a branching node, and $\{\beta, \gamma\}$ its daughters, **where β is a relative pronoun or β ='such'**, then $[\![\alpha]\!] = \lambda x \in D\,.\,[\![\gamma]\!]^x$.

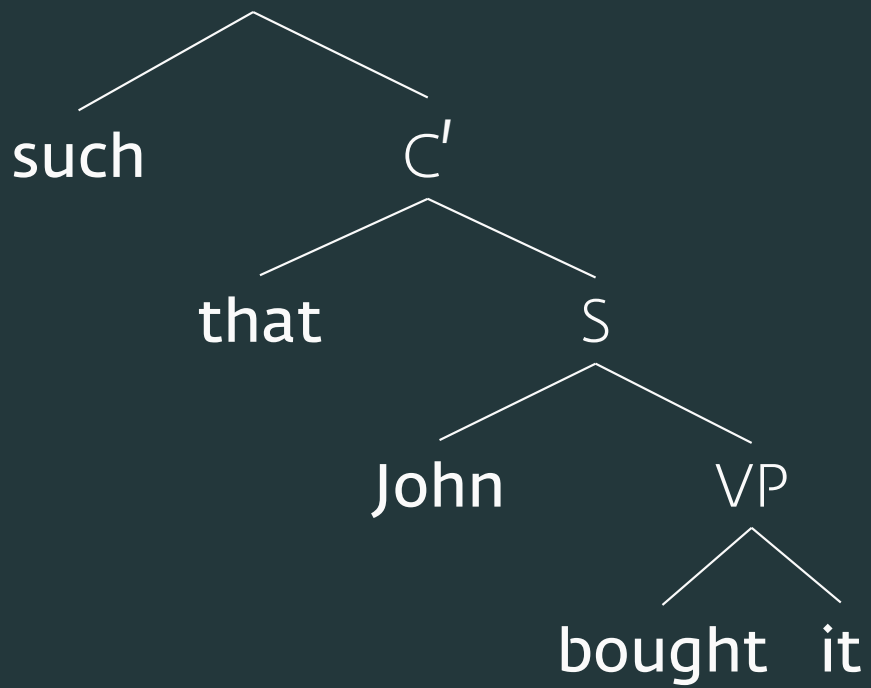
- Let's, then, try calculating a semantic value for the tree below:

(19)



- Let's, then, try calculating a semantic value for the tree below:

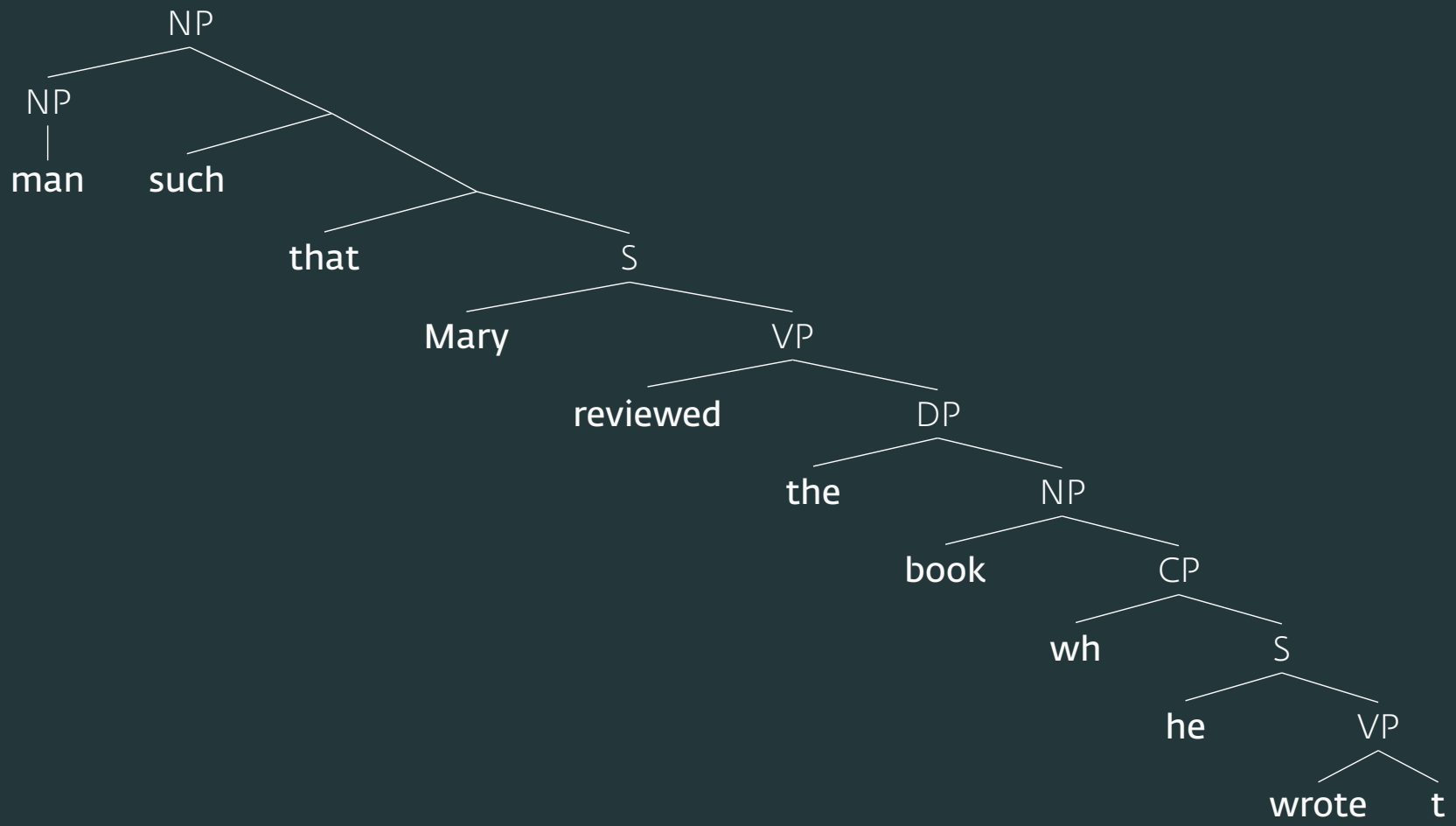
(19)



- =⟦which John bought⟧

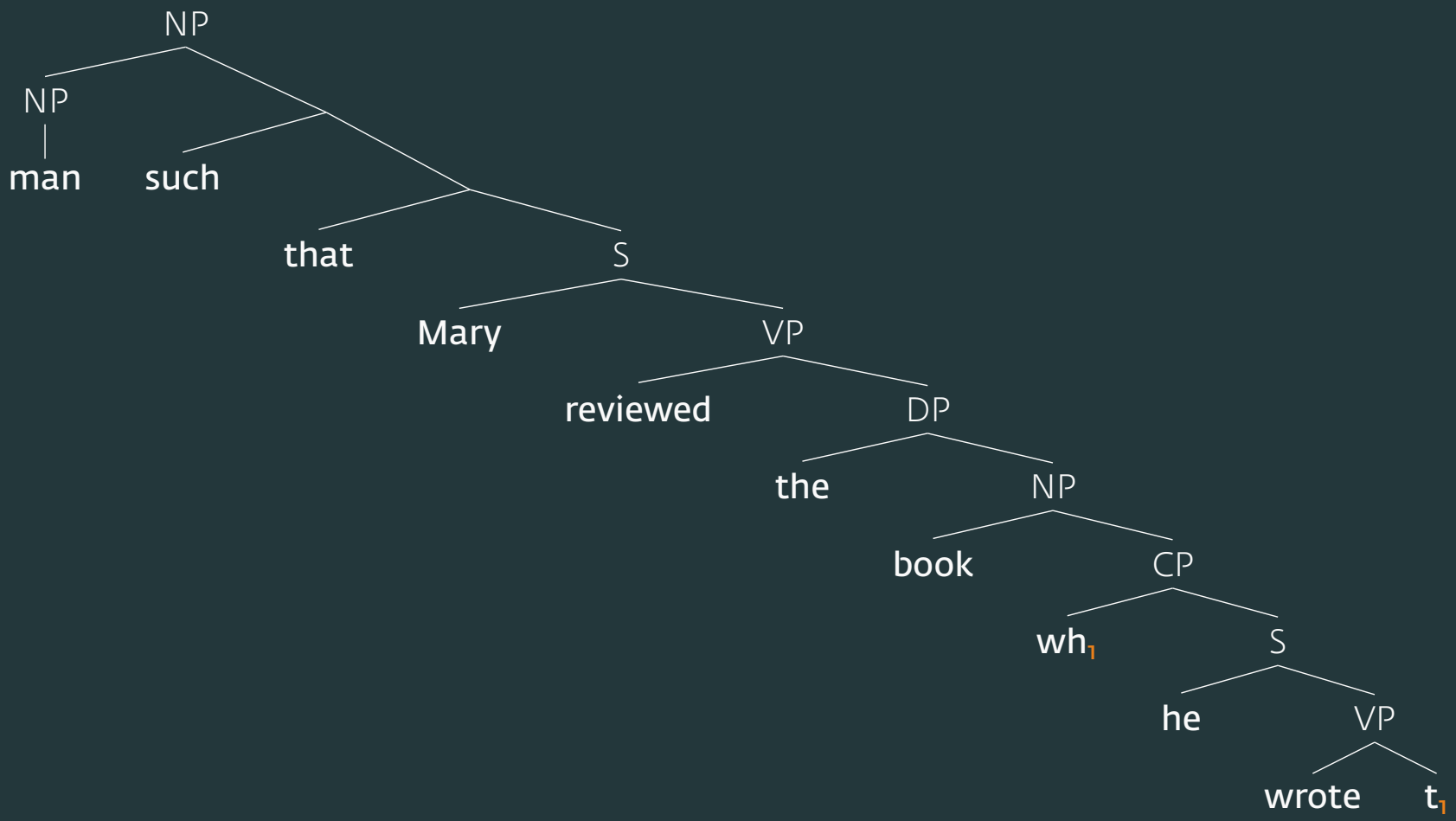
MORE THAN ONE VARIABLE? A RELATIVE INSIDE A RELATIVE.

(20) man such that Mary reviewed the book he wrote.



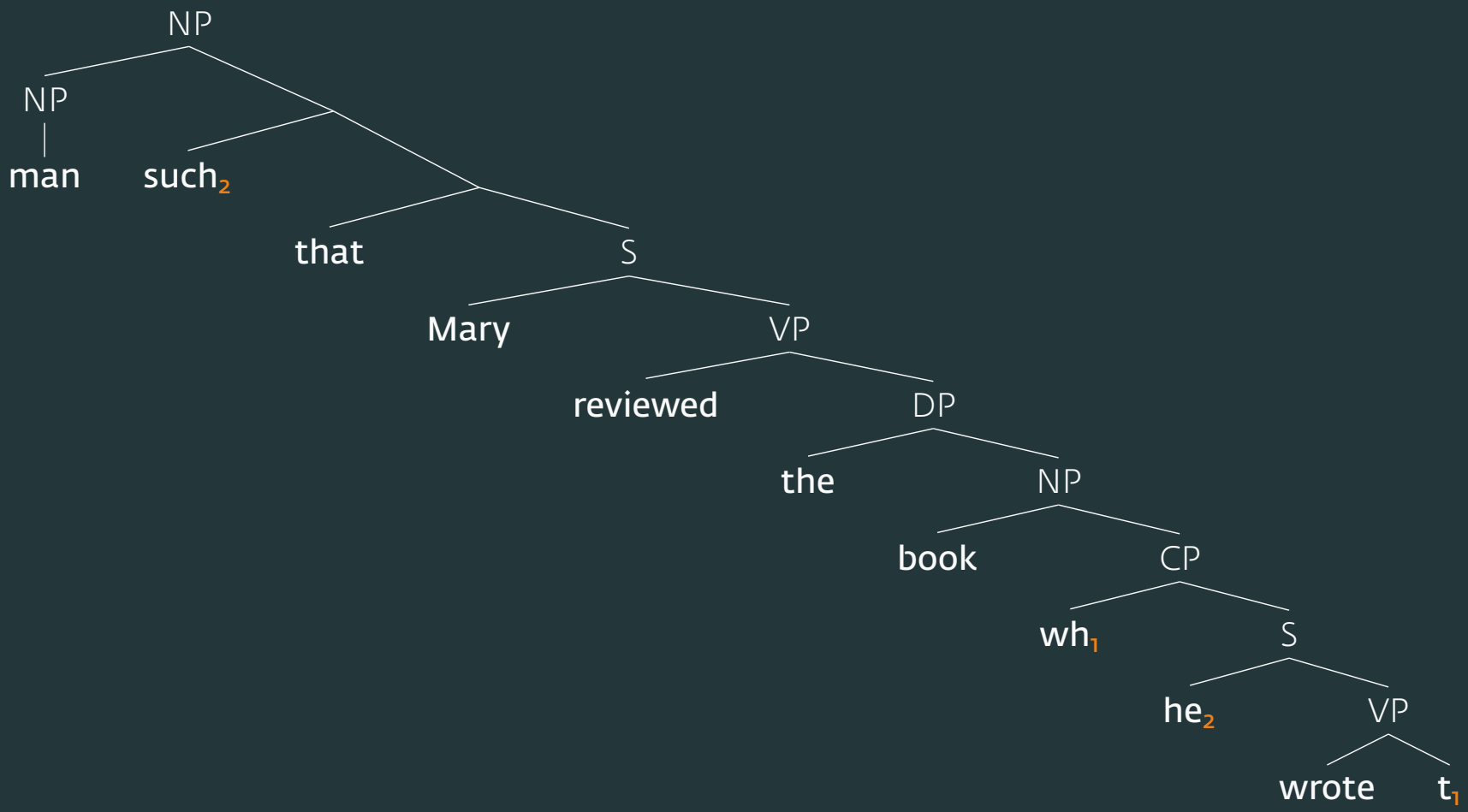
MORE THAN ONE VARIABLE? A RELATIVE INSIDE A RELATIVE.

(20) man such that Mary reviewed the book he wrote.



MORE THAN ONE VARIABLE? A RELATIVE INSIDE A RELATIVE.

(20) man such that Mary reviewed the book he wrote.



- Co-indexed words make sense: we need semantics to see those indexes.

CO-INDEXING & ASSIGNMENTS

AMENDING OUR SEMANTICS WITH CO-INDEXING

- We need to add to our system so that co-indexed items *mean* the same thing.

Amending our semantics with co-indexing

- We need to add to our system so that co-indexed items *mean* the same thing.
- How do you think we can accomplish this?

AMENDING OUR SEMANTICS WITH CO-INDEXING

- We need to add to our system so that co-indexed items *mean* the same thing.
- How do you think we can accomplish this?
- We'll change what assignment denote.
  - So far, assignments were individuals ($a \in D$). This worked for single variables (e.g., "which is empty").
  - This does not work for relatives with multiple variables. Hence, we need to amend the notion of assignment.

Assignment (new def.)

A (variable) **assignment** is a partial function from $\mathbb{N}$ into $D_{\mathbf{e}}$

Assignment (new def.)

A (variable) **assignment** is a partial function from $\mathbb{N}$ into $D_{\mathbf{e}}$

- Hence, it is a function from indices to individuals.

Assignment (new def.)

A (variable) **assignment** is a partial function from $\mathbb{N}$ into D_e

- Hence, it is a function from indices to individuals.

Example

$$\begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{bmatrix}, \quad \begin{bmatrix} 2 \mapsto \text{John} \\ 5 \mapsto \text{Mary} \\ 7 \mapsto \text{Ann} \end{bmatrix}, \quad \begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{John} \end{bmatrix}$$

- Assignments have a domain and range. What are they?

NB assignment domain is DOM since D_a would be confusing.

(21) **Traces and Pronouns Rule**

If α is a pronoun or a trace, a is a variable assignment, and $i \in \textsc{dom}(a)$, then $[\![\alpha_i]\!]^a = a(i)$.

- Let's apply (21) to the following:

(22) $[\![\textbf{he}_2]\!]^{\begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Joe} \end{bmatrix}} =$

- Let's apply (21) to the following:

$$(22) \quad [\![\textbf{he}_2]\!]^{\begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Joe} \end{bmatrix}} = \begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Joe} \end{bmatrix}$$

- Let's apply (21) to the following:

$$(22) \quad [\![\textbf{he}_2]\!]^{\begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Joe} \end{bmatrix}} = \begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Joe} \end{bmatrix}(2) =$$

- Let's apply (21) to the following:

$$(22) \quad [\![\textbf{he}_2]\!]^{\begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Joe} \end{bmatrix}} = \begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Joe} \end{bmatrix}(2) = \text{Joe}$$

- Let's apply (21) to the following:

(22) $[\![\textbf{he}_2]\!]^{\begin{bmatrix}1 \mapsto \text{Sue}\\ 2 \mapsto \text{Joe}\end{bmatrix}} = \begin{bmatrix}1 \mapsto \text{Sue}\\ 2 \mapsto \text{Joe}\end{bmatrix}(2) = \text{Joe}$

(23) $[\![\textbf{t}_1]\!]^{\begin{bmatrix}1 \mapsto \text{Sue}\\ 2 \mapsto \text{Joe}\end{bmatrix}} =$

- Let's apply (21) to the following:

(22) $[\![\textbf{he}_2]\!]^{\begin{bmatrix}1 \mapsto \text{Sue}\\ 2 \mapsto \text{Joe}\end{bmatrix}} = \begin{bmatrix}1 \mapsto \text{Sue}\\ 2 \mapsto \text{Joe}\end{bmatrix}(2) = \text{Joe}$

(23) $[\![\textbf{t}_1]\!]^{\begin{bmatrix}1 \mapsto \text{Sue}\\ 2 \mapsto \text{Joe}\end{bmatrix}} = \begin{bmatrix}1 \mapsto \text{Sue}\\ 2 \mapsto \text{Joe}\end{bmatrix}$

- Let's apply (21) to the following:

$$(22) \quad [\![\mathbf{he}_2]\!]^{\begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Joe} \end{bmatrix}} = \begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Joe} \end{bmatrix}(2) = \text{Joe}$$

$$(23) \quad [\![\mathbf{t}_1]\!]^{\begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Joe} \end{bmatrix}} = \begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Joe} \end{bmatrix}(1) =$$

- Let's apply (21) to the following:

$$(22) \quad [\![\mathbf{he}_2]\!]^{\begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Joe} \end{bmatrix}} = \begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Joe} \end{bmatrix}(2) = \text{Joe}$$

$$(23) \quad [\![\mathbf{t}_1]\!]^{\begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Joe} \end{bmatrix}} = \begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Joe} \end{bmatrix}(1) = \text{Sue}$$

- Let's apply (21) to the following:

(22) $[\![\textbf{he}_2]\!]^{\begin{bmatrix}1 \mapsto \text{Sue}\\ 2 \mapsto \text{Joe}\end{bmatrix}} = \begin{bmatrix}1 \mapsto \text{Sue}\\ 2 \mapsto \text{Joe}\end{bmatrix}(2) = \text{Joe}$

(23) $[\![\textbf{t}_1]\!]^{\begin{bmatrix}1 \mapsto \text{Sue}\\ 2 \mapsto \text{Joe}\end{bmatrix}} = \begin{bmatrix}1 \mapsto \text{Sue}\\ 2 \mapsto \text{Joe}\end{bmatrix}(1) = \text{Sue}$

(24) Is $[\![\textbf{he}_2]\!]$ in the domain of $[\![\ \ ]\!]^{[1 \mapsto \text{Joe}]}$, $[\![\ \ ]\!]^{[3 \mapsto \text{Joe}]}$, or $[\![\ \ ]\!]^{\varnothing}$?

- Let's apply (21) to the following:

(22) $[\![\textbf{he}_2]\!]^{\begin{bmatrix}1 \mapsto \text{Sue}\\ 2 \mapsto \text{Joe}\end{bmatrix}} = \begin{bmatrix}1 \mapsto \text{Sue}\\ 2 \mapsto \text{Joe}\end{bmatrix}(2) = \text{Joe}$

(23) $[\![\textbf{t}_1]\!]^{\begin{bmatrix}1 \mapsto \text{Sue}\\ 2 \mapsto \text{Joe}\end{bmatrix}} = \begin{bmatrix}1 \mapsto \text{Sue}\\ 2 \mapsto \text{Joe}\end{bmatrix}(1) = \text{Sue}$

(24) Is $[\![\textbf{he}_2]\!]$ in the domain of $[\![\ \ ]\!]^{[1 \mapsto \text{Joe}]}$, $[\![\ \ ]\!]^{[3 \mapsto \text{Joe}]}$, or $[\![\ \ ]\!]^{\varnothing}$? No.

WHAT ABOUT THE EMPTY ASSIGNMENT ∅?

- As regards $[\![\ \ ]\!]^{\varnothing}$, **no pronoun or trace is in its domain**.

WHAT ABOUT THE EMPTY ASSIGNMENT ∅?

- As regards $[\![\ ]\!]^{\varnothing}$, **no pronoun or trace is in its domain**.
- What about other expressions?

WHAT ABOUT THE EMPTY ASSIGNMENT ∅?

- As regards $[\![\ ]\!]^{\varnothing}$, **no pronoun or trace is in its domain**.
- What about other expressions?
- Yes, but only those expressions which are in the domain of $[\![\ ]\!]^{a}$ for **every** a.

WHAT ABOUT THE EMPTY ASSIGNMENT ∅?

- As regards $[\![\ ]\!]^{\varnothing}$, **no pronoun or trace is in its domain**.
- What about other expressions?
- Yes, but only those expressions which are in the domain of $[\![\ ]\!]^{a}$ for **every** a.
- This is the case of other 'full' lexical items. (e.g., **John**, **smoke**, etc.)

WHAT ABOUT THE EMPTY ASSIGNMENT ∅?

- As regards $[\![\ ]\!]^{\varnothing}$, **no pronoun or trace is in its domain**.
- What about other expressions?
- Yes, but only those expressions which are in the domain of $[\![\ ]\!]^{a}$ for **every** a.
- This is the case of other 'full' lexical items. (e.g., **John**, **smoke**, etc.)
- From now on, we will think of all such 'full' meanings (≠variables)as having an empty assignment (since they have a denotation independently of the value of assignment).

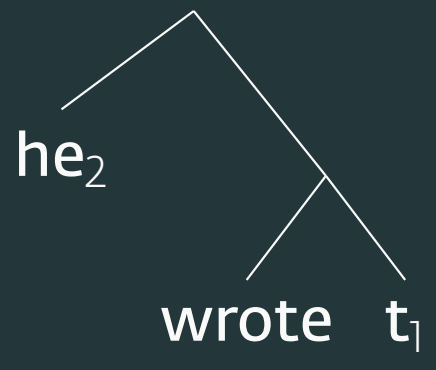
WHAT ABOUT THE EMPTY ASSIGNMENT ∅?

- As regards $[\![\ \ ]\!]^{\varnothing}$, **no pronoun or trace is in its domain**.
- What about other expressions?
- Yes, but only those expressions which are in the domain of $[\![\ \ ]\!]^{a}$ for **every** a.
- This is the case of other 'full' lexical items. (e.g., **John**, **smoke**, etc.)
- From now on, we will think of all such 'full' meanings (≠variables)as having an empty assignment (since they have a denotation independently of the value of assignment).

(25) $[\![\alpha]\!] := [\![\alpha]\!]^{\varnothing}$

- We can now calculate semantic values under given assignments for many larger phrases containing lexical items and variables.

- We can now calculate semantic values under given assignments for many larger phrases containing lexical items and variables.
- Let's prove that (**??**)= 1 iff Chomsky wrote "Barriers".

(26) $[\![$ 

$]\!]^{\begin{bmatrix} 1 \mapsto \text{"Barriers"} \\ 1 \mapsto \text{Chomsky} \end{bmatrix}}$

MODIFIED ASSIGNMENTS

MODIFIED ASSIGNMENTS

- We now need a way to modify an existing assignment.

(27) Let a be an assignment, $i \in \mathbb{N}$, and $x \in D$. Then $a^{\frac{x}{i}}$ (read: "a **modified so as to assign** x **to** i) is the unique assignment which fulfils the following conditions:

a. $\textsc{dom}(a^{\frac{x}{i}}) = \textsc{dom}(a) \cup \{i\}$,

b. $a^{\frac{x}{i}}(i) = x$, and

c. For every $j \in \textsc{dom}(a^{\frac{x}{i}})$ such that $j \neq i$: $a^{\frac{x}{i}}(j) = a(j)$.

- We can now have a new assignment $a^{\frac{x}{i}}$.

EXAMPLE OF THIS NEW ASSIGNMENT

$$(28) \quad \begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{bmatrix}^{\frac{\text{Mary}}{2}} = \begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{bmatrix}$$

EXAMPLE OF THIS NEW ASSIGNMENT

(28) $$\begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{bmatrix}^{\frac{\text{Mary}}{2}} = \begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{bmatrix}$$

(29) $$\begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{John} \end{bmatrix}^{\frac{\text{Mary}}{2}} = \begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{bmatrix}$$

EXAMPLE OF THIS NEW ASSIGNMENT

$$(28)\quad \begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{bmatrix}^{\frac{\text{Mary}}{2}} = \begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{bmatrix}$$

$$(29)\quad \begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{John} \end{bmatrix}^{\frac{\text{Mary}}{2}} = \begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{bmatrix}$$

$$(30)\quad \begin{bmatrix} 1 \mapsto \text{John} \end{bmatrix}^{\frac{\text{Mary}}{2}} = \begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{bmatrix}$$

ANOTHER EXAMPLE OF THIS NEW ASSIGNMENT

$$(31) \quad \left[\left[1 \mapsto \text{John}\right]^{\frac{\text{Mary}}{2}} = \begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{bmatrix}\right]^{\frac{\text{Sue}}{1}} =$$

ANOTHER EXAMPLE OF THIS NEW ASSIGNMENT

$$(31) \quad \left[\left[1 \mapsto \text{John}\right]^{\frac{\text{Mary}}{2}} = \begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{bmatrix}\right]^{\frac{\text{Sue}}{1}} = \begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{bmatrix}^{\frac{\text{Sue}}{1}} =$$

ANOTHER EXAMPLE OF THIS NEW ASSIGNMENT

$$(31)\quad \left[\left[1 \mapsto \text{John}\right]^{\frac{\text{Mary}}{2}} = \left[\begin{array}{c} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{array}\right]\right]^{\frac{\text{Sue}}{1}} = \left[\begin{array}{c} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{array}\right]^{\frac{\text{Sue}}{1}} = \left[\begin{array}{c} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Mary} \end{array}\right]$$

ANOTHER EXAMPLE OF THIS NEW ASSIGNMENT

$$(31)\quad \left[\left[1 \mapsto \text{John}\right]^{\frac{\text{Mary}}{2}} = \begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{bmatrix}\right]^{\frac{\text{Sue}}{1}} = \begin{bmatrix} 1 \mapsto \text{John} \\ 2 \mapsto \text{Mary} \end{bmatrix}^{\frac{\text{Sue}}{1}} = \begin{bmatrix} 1 \mapsto \text{Sue} \\ 2 \mapsto \text{Mary} \end{bmatrix}$$

- The extra bracket was added for clarity.
- We will write (31) as:

$$(32)\quad \left[1 \mapsto \text{John}\right]^{\frac{\text{Mary}}{2}, \frac{\text{Sue}}{1}}$$