Interpreting movement

Semantics II

January 22 & 25, 2018

Semantics without assignments

$$\frac{[\![A]\!] = f : \sigma \to \tau \qquad [\![B]\!] = x : \sigma}{[\![A\ B]\!] = f\,x : \tau}\ \text{FA} \qquad \frac{[\![A]\!] = x : \sigma \qquad [\![B]\!] = f : \sigma \to \tau}{[\![A\ B]\!] = f\,x : \tau}\ \text{BA}$$

$$\frac{[\![A]\!] = f : \sigma \to \mathtt{t} \qquad [\![B]\!] = g : \sigma \to \mathtt{t}}{[\![A\ B]\!] = \lambda x. f\,x \wedge g\,x : \sigma \to \mathtt{t}}\ \text{PM}$$

Conventions I

The semantic value of a tree $\mathcal{T}$ is written $[\![\mathcal{T}]\!]$.

Our basic types are e and t. Complex types are defined inductively:

If σ and τ are both types, then $\sigma \to \tau$ is a type.

'$\sigma \to \tau$' names the space of functions f s.t. $\mathrm{Dom}\, f \subseteq \sigma$ and $\mathrm{Ran}\, f \subseteq \tau$.

Conventions II

We drop parentheses whenever possible:

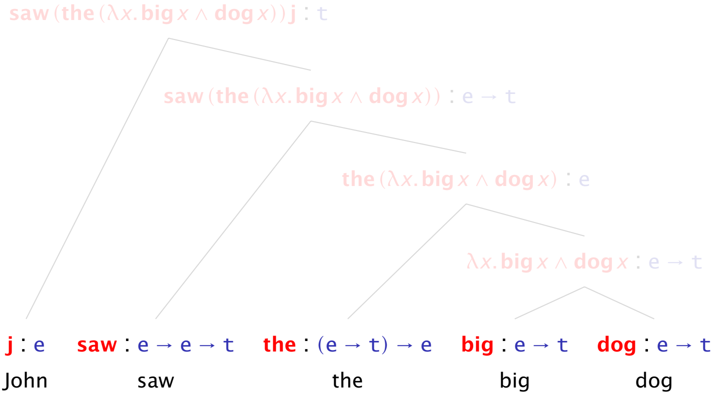
- Types associate to the **right**: $\rho \to \sigma \to \tau \equiv \rho \to (\sigma \to \tau)$.
- Function application associates to the **left**: $f\,x\,y \equiv (f\,x)\,y$.
- The scope of a λ extends as far to the right as possible: $f\,(\lambda x.\, f\,x\,y\,z)\,x$.

A sample derivation: *John saw the big dog*

$$
\dfrac{\mathbf{j} : \mathtt{e} \qquad \dfrac{\mathbf{saw} : \mathtt{e} \to \mathtt{e} \to \mathtt{t} \qquad \dfrac{\mathbf{the} : (\mathtt{e} \to \mathtt{t}) \to \mathtt{e} \qquad \dfrac{\mathbf{big} : \mathtt{e} \to \mathtt{t} \qquad \mathbf{dog} : \mathtt{e} \to \mathtt{t}}{\lambda x.\, \mathbf{big}\, x \wedge \mathbf{dog}\, x : \mathtt{e} \to \mathtt{t}}}{\mathbf{the}\,(\lambda x.\, \mathbf{big}\, x \wedge \mathbf{dog}\, x) : \mathtt{e}}}{\mathbf{saw}\,(\mathbf{the}\,(\lambda x.\, \mathbf{big}\, x \wedge \mathbf{dog}\, x)) : \mathtt{e} \to t}}{\mathbf{saw}\,(\mathbf{the}\,(\lambda x.\, \mathbf{big}\, x \wedge \mathbf{dog}\, x))\,\mathbf{j} : \mathtt{t}}
$$

(Leaving the object-language stuff implicit, as well as the inference rules.)

Equivalence to tree representations



Equivalence to tree representations

$\mathbf{saw}\,(\mathbf{the}\,(\lambda x.\,\mathbf{big}\,x \wedge \mathbf{dog}\,x))\,\mathbf{j} : \mathtt{t}$

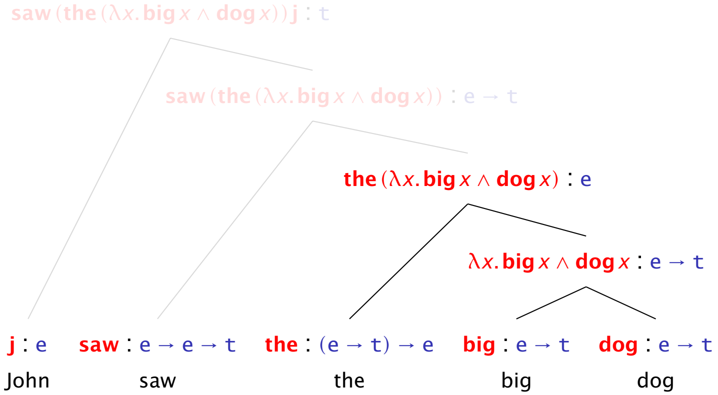
$\mathbf{saw}\,(\mathbf{the}\,(\lambda x.\,\mathbf{big}\,x \wedge \mathbf{dog}\,x)) : \mathtt{e} \to \mathtt{t}$

$\mathbf{the}\,(\lambda x.\,\mathbf{big}\,x \wedge \mathbf{dog}\,x) : \mathtt{e}$

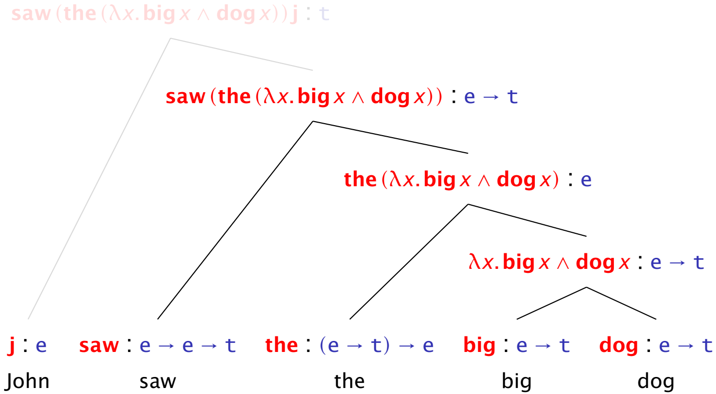
$\lambda x.\,\mathbf{big}\,x \wedge \mathbf{dog}\,x : \mathtt{e} \to \mathtt{t}$

$\mathbf{j} : \mathtt{e}$	$\mathbf{saw} : \mathtt{e} \to \mathtt{e} \to \mathtt{t}$	$\mathbf{the} : (\mathtt{e} \to \mathtt{t}) \to \mathtt{e}$	$\mathbf{big} : \mathtt{e} \to \mathtt{t}$	$\mathbf{dog} : \mathtt{e} \to \mathtt{t}$
John	saw	the	big	dog

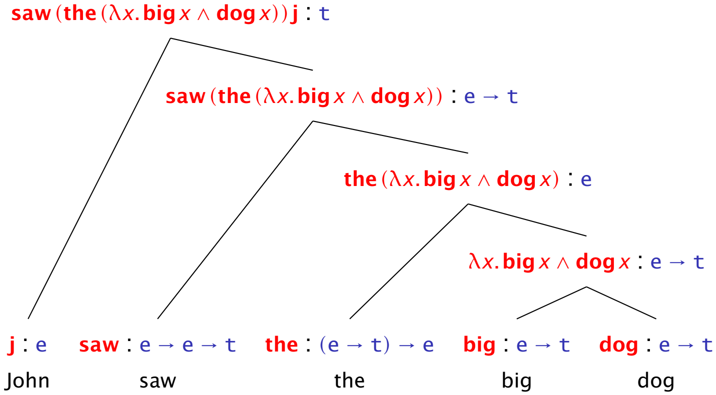
Equivalence to tree representations



Equivalence to tree representations



Equivalence to tree representations



Pronouns and traces

Pronouns

Pronouns are chameleons: their values shift with the context of utterance — and sometimes multiple times within a single sentence!

1. John saw **her**.
2. Every philosopher$_i$ thinks **they**$_i$'re a genius.

Assignments

We need some way to interpret pronouns. The standard device is an *assignment function*, essentially something whose only job is to value pronouns.

Simple assignments can be typed as functions from natural numbers (indices) to individuals: $\mathbb{N} \to \mathrm{e}$. To save parentheses, we can write g_n for $g\,n$.

Semantics with assignments

$$\frac{[\![A]\!]^g = f : \sigma \to \tau \qquad [\![B]\!]^g = x : \sigma}{[\![A\ B]\!]^g = f\,x : \tau}\ \text{FA} \qquad \frac{[\![A]\!]^g = x : \sigma \qquad [\![B]\!]^g = f : \sigma \to \tau}{[\![A\ B]\!]^g = f\,x : \tau}\ \text{BA}$$

$$\frac{[\![A]\!]^g = f : \sigma \to \mathtt{t} \qquad [\![B]\!]^g = g : \sigma \to \mathtt{t}}{[\![A\ B]\!]^g = \lambda x.\,f\,x \wedge g\,x : \sigma \to \mathtt{t}}\ \text{PM}$$

Example derivation: *the dog near her₁*

$$
\dfrac{\textbf{the} : (\mathtt{e} \to \mathtt{t}) \to \mathtt{e} \qquad \dfrac{\textbf{dog} : \mathtt{e} \to \mathtt{t} \qquad \dfrac{\textbf{near} : \mathtt{e} \to \mathtt{e} \to \mathtt{t} \qquad g_1 : \mathtt{e}}{\textbf{near}\, g_1 : \mathtt{e} \to \mathtt{t}}}{\lambda x.\, \textbf{dog}\, x \wedge \textbf{near}\, g_1\, x : \mathtt{e} \to \mathtt{t}}}{\textbf{the}\, (\lambda x.\, \textbf{dog}\, x \wedge \textbf{near}\, g_1\, x) : \mathtt{e}}
$$

What, precisely, this means depends on how g tells us to understand 1.

Predicate Abstraction

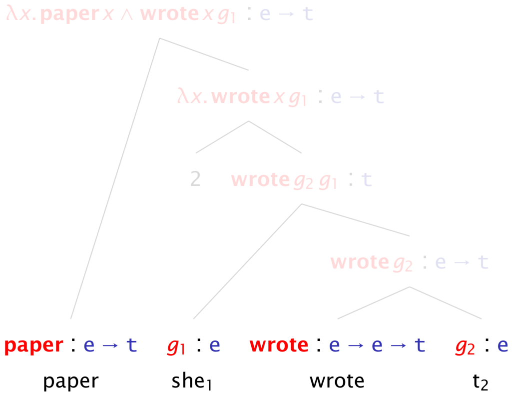
$$\frac{[\![A]\!]^g = \phi : \sigma}{[\![n\ A]\!]^g = \lambda x.\, [\![A]\!]^{g[n \to x]} : e \to \sigma}\ \text{PA}$$

Formulated in this way, the rule looks a little strange: the denotation of *A* at *g* doesn't matter at all for determining the denotation of *n A* at *g*!

This is why people sometimes say that rules like PA are non-compositional. This is not a deep feature of the system, and we will talk a bit more about this later on.

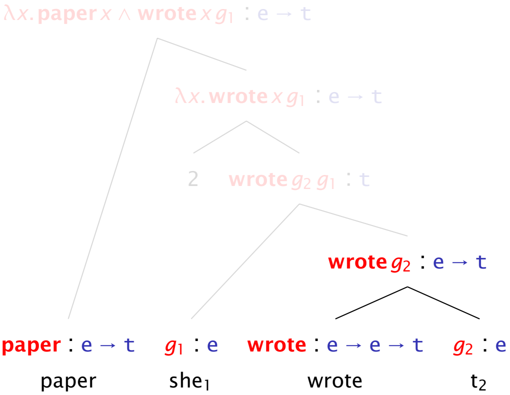
Practice with PA

Basic case



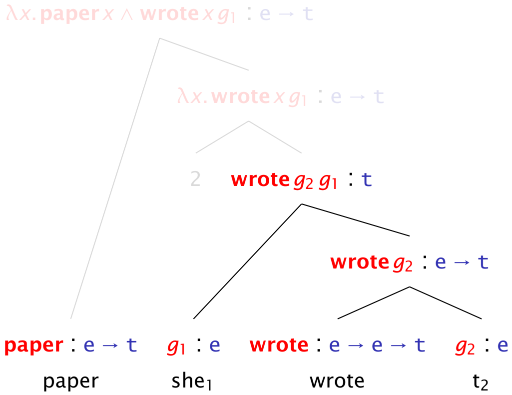
λx. paper x ∧ wrote x g1 : e → t
λx. wrote x g1 : e → t
2
wrote g2 g1 : t
wrote g2 : e → t
paper : e → t
g1 : e
wrote : e → e → t
g2 : e
paper
she1
wrote
t2

Basic case

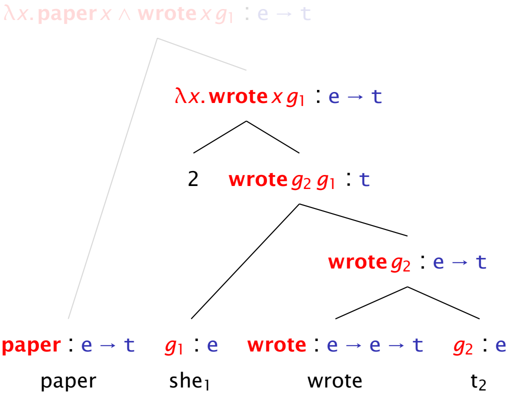


λx. paper x ∧ wrote x g₁ : e → t
λx. wrote x g₁ : e → t
2
wrote g₂ g₁ : t
wrote g₂ : e → t
paper : e → t
g₁ : e
wrote : e → e → t
g₂ : e
paper
she₁
wrote
t₂

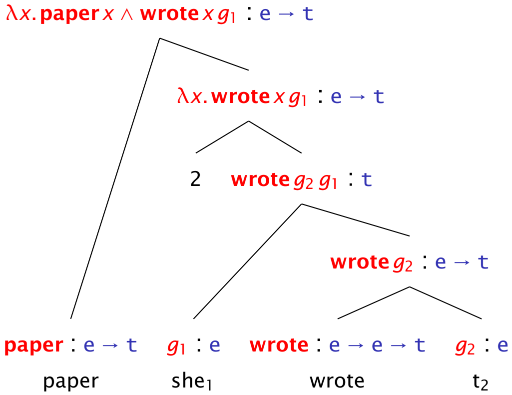
Basic case


$\lambda x.\, \mathbf{paper}\, x \wedge \mathbf{wrote}\, x\, g_1 : e \to t$
$\lambda x.\, \mathbf{wrote}\, x\, g_1 : e \to t$
2
$\mathbf{wrote}\, g_2\, g_1 : t$
$\mathbf{wrote}\, g_2 : e \to t$
$\mathbf{paper} : e \to t$
$g_1 : e$
$\mathbf{wrote} : e \to e \to t$
$g_2 : e$
paper
she_1
wrote
t_2

Basic case


$\lambda x.\,\mathbf{paper}\,x \wedge \mathbf{wrote}\,x\,g_1 : e \to t$
$\lambda x.\,\mathbf{wrote}\,x\,g_1 : e \to t$
2
$\mathbf{wrote}\,g_2\,g_1 : t$
$\mathbf{wrote}\,g_2 : e \to t$
$\mathbf{paper} : e \to t$
$g_1 : e$
$\mathbf{wrote} : e \to e \to t$
$g_2 : e$
paper
she_1
wrote
t_2

Basic case


$\lambda x.\,\mathbf{paper}\,x \wedge \mathbf{wrote}\,x\,g_1 : e \to t$
$\lambda x.\,\mathbf{wrote}\,x\,g_1 : e \to t$
2
$\mathbf{wrote}\,g_2\,g_1 : t$
$\mathbf{wrote}\,g_2 : e \to t$
$\mathbf{paper} : e \to t$
$g_1 : e$
$\mathbf{wrote} : e \to e \to t$
$g_2 : e$
paper
she$_1$
wrote
t$_2$

Parasitic gaps?

Using {FA,BA,PM,PA} can we analyze *parasitic gap* constructions like the following?

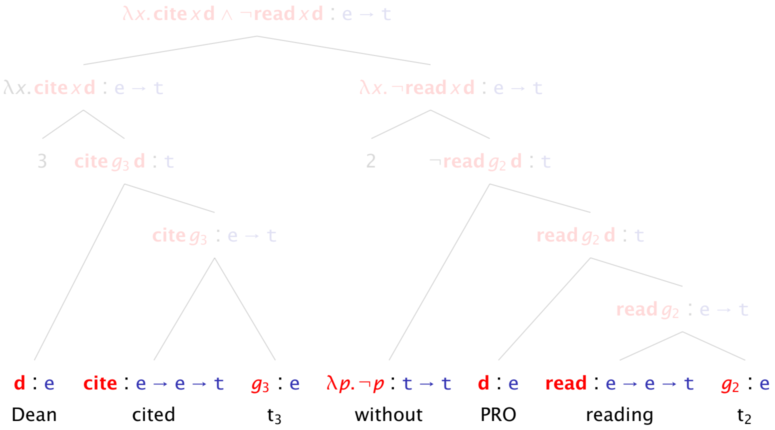
3. . . . the paper Dean cited _ without reading _ . . .

(Thanks to Akane for bringing up these kinds of cases.)

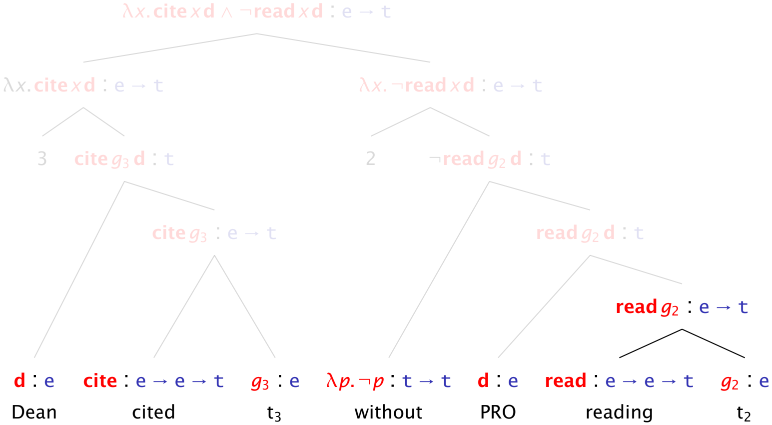
For the sake of discussion, we can make the following assumptions:

- *Reading* has a silent subject *DEAN*, i.e., $[\![\text{PRO}]\!]^g = \mathbf{d}$.
- $[\![\text{without}]\!]^g = [\![\text{not}]\!]^g = \lambda p.\neg p$ (type: $\mathtt{t} \to \mathtt{t}$)

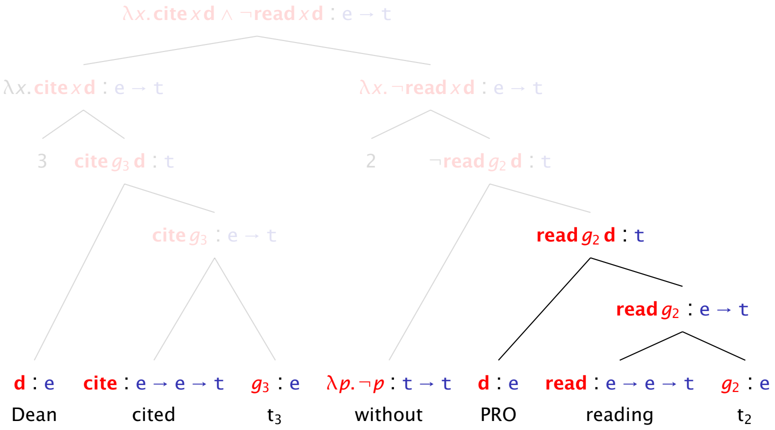
Parasitic gap analysis



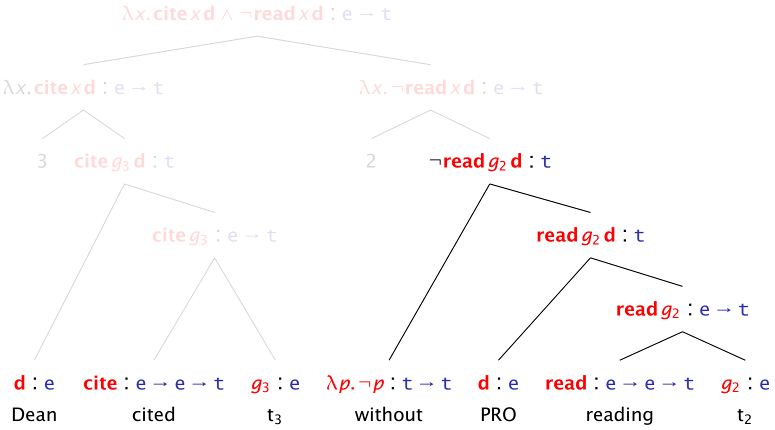
Parasitic gap analysis



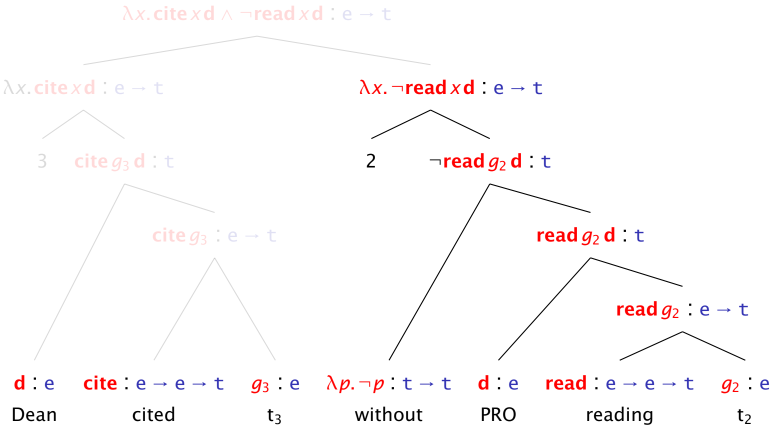
Parasitic gap analysis



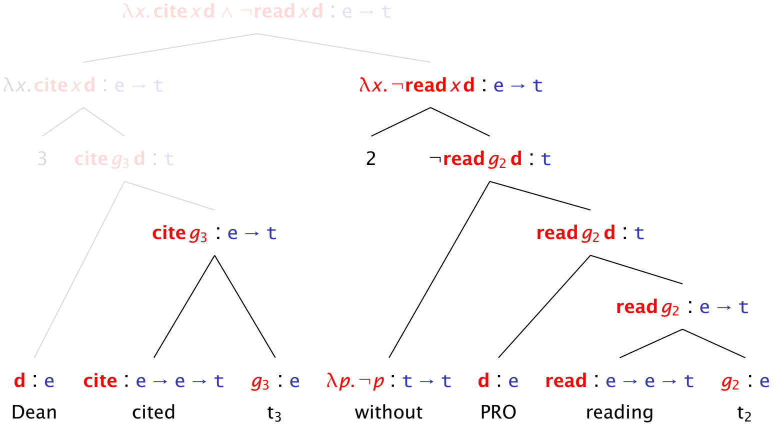
Parasitic gap analysis



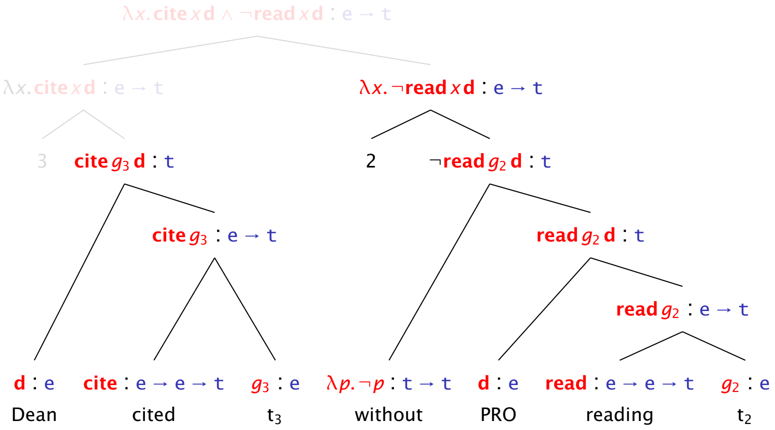
Parasitic gap analysis



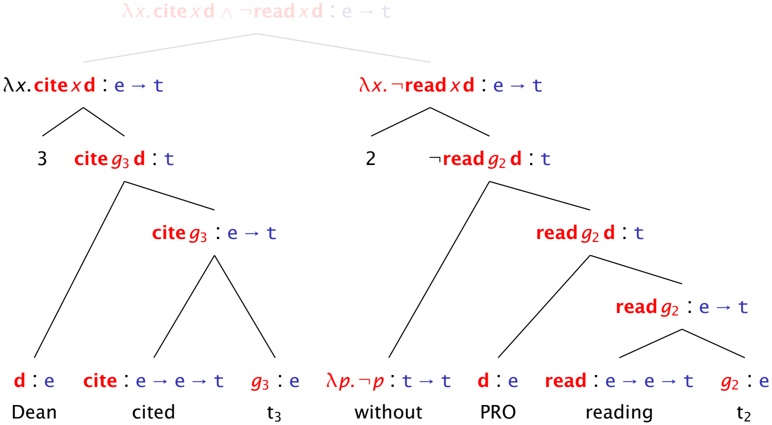
Parasitic gap analysis



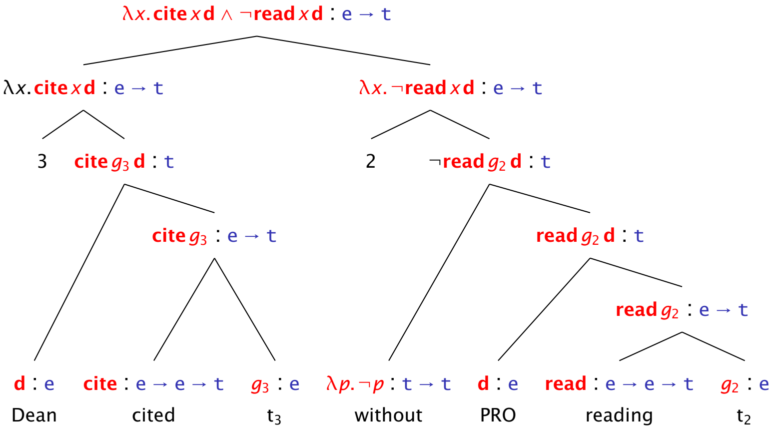
Parasitic gap analysis



Parasitic gap analysis



Parasitic gap analysis



Note about binding operators

It is frequently assumed that the nodes triggering PA are *introduced by movement*.

An alternative is to suppose that PA-triggering indices are freely inserted in the syntax (they're silent, after all).

Still, it is worth thinking about our last tree in terms of the first kind of assumption. What movements would introduce our two abstraction nodes?

Ellipsis

It's frequently noted that ambiguity doesn't multiply in ellipsis:

4. Mary [$_{VP}$ went to the bank], and John did Δ too.
5. Mary [$_{VP}$ saw the elk with the binoculars], and John did Δ too.

This suggests that ellipsis requires Δ to *mean the same thing* as its "antecedent" VP.

Nevertheless, we find certain cases in which the meaning *does* seem to shift:

6. Mary [$_{VP}$ cited her paper], and John did Δ too.

Quantifiers

English quantifiers occur in object positions, and participate in scope ambiguities:

7. John read every book.
8. A doctor examined every patient.
9. Everyone passenger wasn't screened.

Given a richer theory of the syntax and the architecture of the grammar, cases like (7) and (8) are automatically explained by our interpretive system. Cases like (9) turn out to be a bit trickier to account for (but they are within reach as well).

Scope inversion

In the **Y-model** of syntax, a post-surface level of representation is assumed. Crucially, at this level, quantifiers can silently participate in movement processes similar to the processes that produce relativization structures.

$$\begin{tikzcd}[column sep=small] & \text{D-structure} \arrow[d, "\text{overt movement}"] & \\ & \text{S-structure} \arrow[dl] \arrow[dr, "\text{covert movement}"] & \\ \text{PF} & & \text{LF} \end{tikzcd}$$

7. [every book] [1 [John read t_1]]
8. [a doctor] [1 [[every patient] [2 [t_1 examined t_2]]]]
   [every patient] [2 [[a doctor] [1 [t_1 examined t_2]]]]

Crossover and reconstruction

10. *Their$_i$ advisor cited no grad student$_i$.
11. *She$_i$ cited every linguist$_i$.
12. Their advisor$_i$ seems to every Ph.D. student$_i$ to be a genius.

What distinguishes the grammatical cases from the ungrammatical ones? What does this suggest about possible constraints on quantifier movement?

Roofing and inverse linking

Despite the fact that both of the following sentences contain two quantifiers, they turn out to be unambiguous. What is going on?

13. A candidate submitted every paper she had written.
14. A member of every committee voted to abolish it.