LING4400: Lecture Notes 15
Quantifier Scope

Contents

15.1 Quantifiers can raise scope

We've seen how to translate in-situ interpretations of nested quantifiers using schematization:

- $(\text{Two Person } (\text{All}_{\langle e,t\rangle} \text{ Booth In})) : t$
  - $(\text{Two Person}) : \langle\langle e,t\rangle,t\rangle$
    - $\text{Two} : \langle\langle e,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle$
      - *Two*
    - $\text{Person} : \langle e,t\rangle$
      - *people*
  - $(\text{All}_{\langle e,t\rangle} \text{ Booth In}) : \langle e,t\rangle$
    - $\text{In} : \langle e,\langle e,t\rangle\rangle$
      - *occupy*
    - $(\text{All}_{\langle e,t\rangle} \text{ Booth}) : \langle\langle\langle e,t\rangle,t\rangle,\langle e,t\rangle\rangle$
      - $\text{All}_{\langle e,t\rangle} : \langle\langle e,t\rangle,\langle\langle e,\langle e,t\rangle\rangle,\langle e,t\rangle\rangle\rangle$
        - *every*
      - $\text{Booth} : \langle e,t\rangle$
        - *booth*

But sometimes there's a preferred reading with the quantifiers scoped in the opposite order:

$$\text{All Booth } (\lambda_{y:e} \text{ Two Person } (\text{In } y))$$

Unfortunately, we can't get this preferred reading via schematization, so we need something else. . .

15.2 Scope via storage [Cooper, 1983, Keller, 1988]

Instead, we'll now augment each expression with a **store** or **context** Γ, Δ, Θ, delimited by '$\vdash$':

$$\underbrace{\underbrace{\Delta}_{\text{store}} \vdash \underbrace{\varphi : \langle \gamma_n, \gamma_{n-1} \rangle}_{\text{expression}}}_{\text{sequent}}$$

These are called **sequents**. The store Γ, Δ, Θ of each sequent is a list of other sequents and variables.

Quantifier functions φ can now be stored, leaving variables x of type δ_n in their place:

$$\underbrace{\Delta \vdash \varphi : \langle \gamma_n, \gamma_{n-1} \rangle}_{\text{sequent}} \;\Rightarrow\; (\underbrace{\Delta \vdash \varphi : \langle \gamma_n, \gamma_{n-1} \rangle}_{\text{stored sequent}}, \underbrace{x : \delta_n}_{\text{variable}}) \vdash x : \delta_n \qquad \text{(Quantifier Storage)}$$

Stored functions φ are then retrieved and applied to bind the new variable x at a wider scope ψ:

$$\Gamma, (\underbrace{\Delta \vdash \varphi : \langle \gamma_n, \gamma_{n-1} \rangle}_{\text{stored sequent}}, \underbrace{x : \delta_n}_{\text{variable}}), \Theta \vdash \psi : \gamma_{n-1} \quad \Rightarrow \quad \Gamma, \Delta, \Theta \vdash (\varphi\ (\lambda_{x:\delta_n}\ \psi)) : \gamma_{n-1} \quad \text{(Quantifier Retrieval)}$$

Note the retrieved sequents and functions need not be retrieved in the same order they are stored!

Other rules are then augmented with stores that are just concatenated without being changed.

$$\Gamma \vdash \varphi : \langle \alpha, \beta \rangle \quad \Delta \vdash \chi : \alpha \quad \Rightarrow \quad \Gamma, \Delta \vdash (\varphi\ \chi) : \beta \qquad \text{(Forward Function Application)}$$

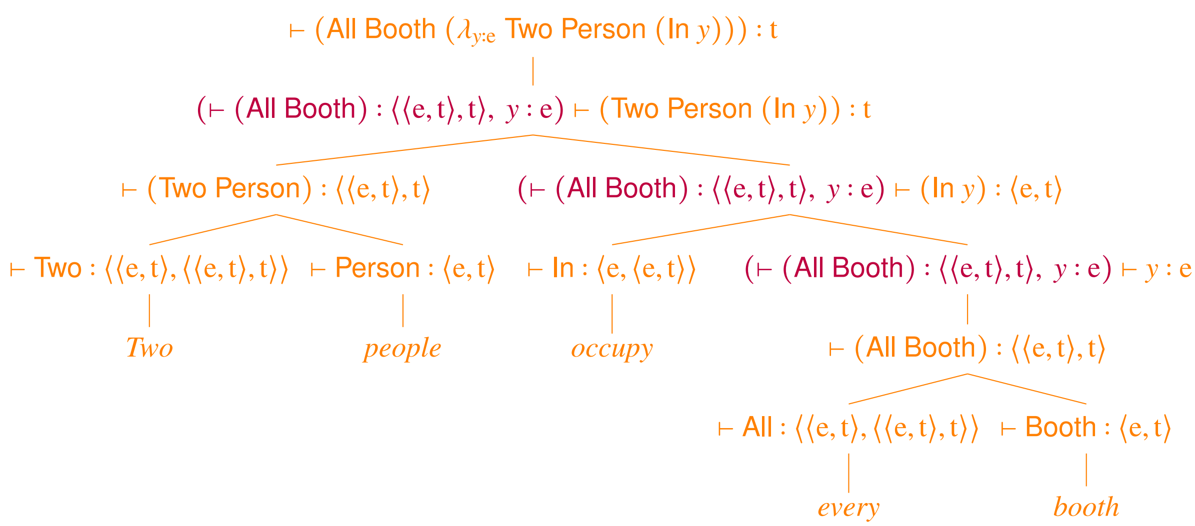
$$\Gamma \vdash \chi : \alpha \quad \Delta \vdash \varphi : \langle \alpha, \beta \rangle \quad \Rightarrow \quad \Gamma, \Delta \vdash (\varphi\ \chi) : \beta \qquad \text{(Backward Function Application)}$$

and similarly for Modification and Closure rules.

Now stored sequents can propagate up the translation and be retrieved in any order!

This approach to quantifier scope raising is called **(nested) Cooper storage**.

Now we can translate our preferred reading of *Two people occupy every booth*:



We can also translate our in-situ reading without using schematized quantifiers:

$$(\textsf{Two Person}\ (\lambda_{x:\mathrm{e}}\ \textsf{All Booth}\ (\lambda_{y:\mathrm{e}}\ \textsf{In}\ y\ x))) : \mathrm{t}$$

$$(\vdash (\textsf{Two Person}) : \langle\langle \mathrm{e},\mathrm{t}\rangle,\mathrm{t}\rangle,\ x : \mathrm{e}) \vdash (\textsf{All Booth}\ (\lambda_{y:\mathrm{e}}\ \textsf{In}\ y\ x)) : \mathrm{t}$$

$$(\vdash (\textsf{Two Person}) : \langle\langle \mathrm{e},\mathrm{t}\rangle,\mathrm{t}\rangle,\ x : \mathrm{e}), (\vdash (\textsf{All Booth}) : \langle\langle \mathrm{e},\mathrm{t}\rangle,\mathrm{t}\rangle,\ y : \mathrm{e}) \vdash (\textsf{In}\ y\ x) : \mathrm{t}$$

$$(\vdash (\textsf{Two Person}) : \langle\langle \mathrm{e},\mathrm{t}\rangle,\mathrm{t}\rangle,\ x : \mathrm{e}) \vdash x : \mathrm{e} \qquad (\vdash (\textsf{All Booth}) : \langle\langle \mathrm{e},\mathrm{t}\rangle,\mathrm{t}\rangle,\ y : \mathrm{e}) \vdash (\textsf{In}\ y) : \langle \mathrm{e},\mathrm{t}\rangle$$

$$\vdash (\textsf{Two Person}) : \langle\langle \mathrm{e},\mathrm{t}\rangle,\mathrm{t}\rangle \qquad \vdash \textsf{In} : \langle \mathrm{e},\langle \mathrm{e},\mathrm{t}\rangle\rangle \qquad (\vdash (\textsf{All Booth}) : \langle\langle \mathrm{e},\mathrm{t}\rangle,\mathrm{t}\rangle,\ y : \mathrm{e}) \vdash y : \mathrm{e}$$

$$\vdash \textsf{Two} : \langle\langle \mathrm{e},\mathrm{t}\rangle,\langle\langle \mathrm{e},\mathrm{t}\rangle,\mathrm{t}\rangle\rangle \qquad \vdash \textsf{Person} : \langle \mathrm{e},\mathrm{t}\rangle \qquad \textit{occupy} \qquad \vdash (\textsf{All Booth}) : \langle\langle \mathrm{e},\mathrm{t}\rangle,\mathrm{t}\rangle$$

$$\textit{Two} \qquad \textit{people} \qquad \vdash \textsf{All} : \langle\langle \mathrm{e},\mathrm{t}\rangle,\langle\langle \mathrm{e},\mathrm{t}\rangle,\mathrm{t}\rangle\rangle \qquad \vdash \textsf{Booth} : \langle \mathrm{e},\mathrm{t}\rangle$$

$$\textit{every} \qquad \textit{booth}$$

We can likewise model negation without schemas, since (recall) it can be modeled as quantification.

Practice 15.1: trees with sequents

Draw **a translation tree with logical sequents at each branch** for the phrase:

each country

in which *each country* undergoes storage.

Practice 15.2: trees with sequents

Draw **a translation tree with logical sequents at each branch** for the following sentence:

A city in each country is coastal.

in which *each country* is scoped high.

15.3 Scoping over eventualities

Quantifier storage gives us a better analysis of eventualities, too.

Here's the old schematized analysis, which makes the two volcanoes share the same event:

- $(\mathsf{Some}\ (\mathsf{Two}_{\langle e,t\rangle}\ \mathsf{Volcano}\ \mathsf{Erupt})\ (\lambda_{e:e}\ \mathsf{True})) : t$
  - $(\mathsf{Some}\ (\mathsf{Two}_{\langle e,t\rangle}\ \mathsf{Volcano}\ \mathsf{Erupt})) : \langle\langle e,t\rangle,t\rangle$
    - $(\mathsf{Two}_{\langle e,t\rangle}\ \mathsf{Volcano}\ \mathsf{Erupt}) : \langle e,t\rangle$
      - $(\mathsf{Two}_{\langle e,t\rangle}\ \mathsf{Volcano}) : \langle\langle e,\langle e,t\rangle\rangle,\langle e,t\rangle\rangle$
        - $\mathsf{Two}_{\langle e,t\rangle} : \langle\langle e,t\rangle,\langle\langle e,\langle e,t\rangle\rangle,\langle e,t\rangle\rangle\rangle$
          - *Two*
        - $\mathsf{Volcano} : \langle e,\langle e,t\rangle\rangle$
          - *volcanoes*
      - $\mathsf{Erupt} : \langle e,\langle e,t\rangle\rangle$
        - *erupted*

These eruptions are not necessarily a contiguous region of time, so this e is not a single event.

But now we can scope the eventualities low, with one for each volcano, using quantifier storage:

- $\vdash (\mathsf{Two}\ \mathsf{Volcano}\ (\lambda_{x:e}\ \mathsf{Some}\ (\mathsf{Erupt}\ x)\ (\lambda_{e:e}\ \mathsf{True}))) : t$
  - $(\vdash (\mathsf{Two}\ \mathsf{Volcano}) : \langle\langle e,t\rangle,t\rangle,\ x : e) \vdash (\mathsf{Some}\ (\mathsf{Erupt}\ x)\ (\lambda_{e:e}\ \mathsf{True})) : t$
    - $(\vdash (\mathsf{Two}\ \mathsf{Volcano}) : \langle\langle e,t\rangle,t\rangle,\ x : e) \vdash (\mathsf{Some}\ (\mathsf{Erupt}\ x)) : \langle\langle e,t\rangle,t\rangle$
      - $(\vdash (\mathsf{Two}\ \mathsf{Volcano}) : \langle\langle e,t\rangle,t\rangle,\ x : e) \vdash (\mathsf{Erupt}\ x) : \langle e,t\rangle$
        - $(\vdash (\mathsf{Two}\ \mathsf{Volcano}) : \langle\langle e,t\rangle,t\rangle,\ x : e) \vdash x : e$
          - $\vdash (\mathsf{Two}\ \mathsf{Volcano}) : \langle\langle e,t\rangle,t\rangle$
            - $\vdash \mathsf{Two} : \langle\langle e,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle$
              - *Two*
            - $\vdash \mathsf{Volcano} : \langle e,t\rangle$
              - *volcanoes*
        - $\vdash \mathsf{Erupt} : \langle e,\langle e,t\rangle\rangle$
          - *erupted*

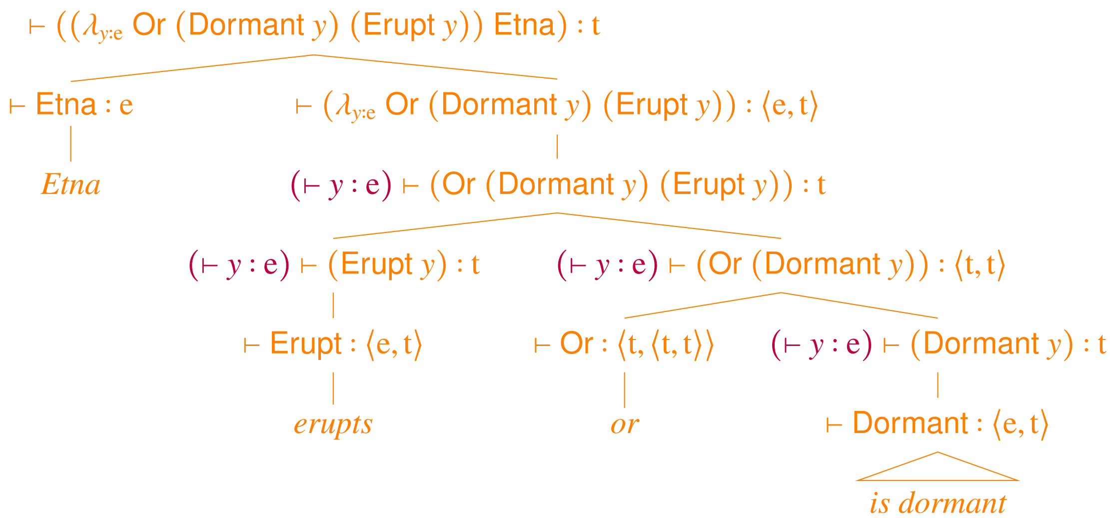
This would match eventualities in a world model!

Practice 15.3: rule labeling

Label the **rules** in the above tree for *Two volcanoes erupted*.

15.4 Schematization via storage

This approach also eliminates the need for schematization of conjunction and disjunction:



But we need different storage and retrieval rules, and a conjunction function that matches stores.

References

[Cooper, 1983] Cooper, R. (1983). *Quantification and syntactic theory*. Dordrecht, Holland: D. Reidel.

[Keller, 1988] Keller, W. R. (1988). Nested cooper storage: The proper treatment of quantifiers in ordinary noun phrases. In E. U. Reyle & E. C. Rohrer (Eds.), *Natural Language Parsing and Linguistic Theories* (pp. 432–447). D. Reidel.