Call-by-name Linguistic Side Effects

http://okmij.org/ftp/Computation/gengo/

Symmetric calculi and Ludics for the semantic interpretation
ESSLLI 2008 Workshop
Hamburg, Germany   August 6, 2008

Outline

Phenomena

Mary sees everyone's mother.

John, Mary saw.

Mary saw who?
Who did Mary see?
Whose mother did Mary see?

John's mother saw him.

Everyone$_i$'s father loves their$_i$ mother.
*Their$_i$ father loves everyone$_i$'s mother.

Who _ saw who?
*Who did who see _?

Who$_i$ _ saw his$_i$ mother?
*Who$_i$ did his$_i$ mother see _?

Mary's present for him$_i$, every boy$_i$ saw.
*Every boy his mother likes.

We handle quantification; fronted phrases, in-situ and raised wh-questions, binding. And more complex things: quantification and binding, superiority, binding in raised wh-questions, and even more complex cases.
This talk will only describe superiority; anaphora and its various interactions are described in the paper.
All these phenomena have been explained in separate ways. What we strive is uniformity and parsimony: explain all these and more phenomena using simple means. We want to avoid type raising. In particular, John here and John there, Mary here and Mary there are uniformly denoted by the domain constant of the same simple type e. The last two lines are controversial. The cataphora sounds better if ”Mary’s” is replaced with ”whose”. In general, cataphora is very rare. Carl Pollard gave two examples, both literary: ”When he woke up, Roger found that he turned into a rabbit” ”The first dollar he earned is the businessman longest memory.”

Phenomena

Mary sees everyone's mother.

John, Mary saw.

Mary saw who?
Who did Mary see?
Whose mother did Mary see?

John's mother saw him.

Everyone$_i$'s father loves their$_i$ mother.
*Their$_i$ father loves everyone$_i$'s mother.

Who _ saw who?
*Who did who see _?

Who$_i$ _ saw his$_i$ mother?
*Who$_i$ did his$_i$ mother see _?

Mary's present for him$_i$, every boy$_i$ saw.
*Every boy his mother likes.

I draw attention to these incorrect phrases, which our theory should be able to rule out (and it does). It is relatively easy to build a theory that accepts everything; it’s hard to make the theory more discriminating. Taking type categorial approach as an example, it is relatively easy to show something is derivable: one just exhibits the derivation. It is quite difficult to show something is not derivable. It is rather hard to prove negative.

Summary

Developing the approach pioneered by Shan and Barker

Uniformity

- Evaluation order: not fixed left-to-right, demand-driven
- No need for thunks and type raising
- The type of an individual: e, *always*

Typing

- Assigning types to terms and contexts
- The type of a term describes possible effects of the term
- The type of a term describes what the term *produces* and what context the term *requires*
- Ruling on incomplete phrases

Main result

Both typing and CBN are needed for correct prediction of superiority and binding in wh-questions with topicalization

Here is a brief summary of the rest of the talk, before I get to the extended introduction.
Again, we strive for uniformity and parsimony, following the approach established by Shan and Barker. We further develop the approach in terms of uniformity and typing. We introduce a CBN calculus, evaluation order is not fixed left-to-right, determined by the demand on values. Therefore, we need no thunks or type-raising. The calculus has context as first-class entities.

The calculus is typed; both terms and contexts have types. An individual is *always* represented by a term of the type e. Given a term, we can determine its type – or determine that the term cannot be given any type. We pronounce such phrases ungrammatical. We can make this determination *even* for terms that do not represent complete sentences. Thus we predict that any sentence including such a phrase (e.g., a gapped clause with a question word before the gap) is unacceptable.

Denotations

Mary saw John.

- ▷ **see(john,mary)**

For unacceptable sentences, we want to describe what is wrong with them. For acceptable sentences, we wish to give their denotations, in the form that lets us determine their consequences. We chose to specify denotations as logic formulas, like this formula in FOL. Here **see** is a 2-place predicate and **john** and **mary** are constants, or zero-arity function symbols.

Denotations

Mary saw John.

- ▷ see john mary

john	$: i$
mary	$: i$
see	$: i \rightarrow (i \rightarrow o)$

Instead of FOL, we chose a form of a higher-order classical logic: simple type theory (STT). I don't think that I make the use of higher-order features, but I sure like the notation, like this. Here, see and john and mary are constants, *typed* constants. Church, who originated STT, would give the constants these types.

Denotations

Mary saw John.

- ▷ see john mary

$$\begin{aligned}\text{john} &: \text{e}\\ \text{mary} &: \text{e}\\ \text{see} &: \text{e} \rightharpoondown \text{e} \rightharpoondown \text{t}\end{aligned}$$

I would use a more familiar notation for types. The white space in this formula (juxtaposition) is an application. We already deal with two languages here: the *subject* language in which the phrase ‘Mary saw John’ is written, and the *object, or target* language of STT in which the denotation is written. We shall establish the mapping between the two. Note the different arrow, to emphasize we are talking about object language and types in that object language.

Natural language denotations

Mary saw John.

- ▷ see john mary

Natural language denotations

Mary saw John.

- $\triangleright$ see john mary

Mary sees everyone's mother.

- $\triangleright$ $\forall_c$ see(mother c) mary

Mary saw who?

- $\triangleright$ ∂_c see c mary

John's mother saw him.

- $\triangleright$ see john(mother john)

Who _ saw who?

- $\triangleright$ $\partial_{c_1}\partial_{c_2}$ see $c_2 c_1$

The first mapping is straightforward, mere local reshuffling. Here, the quantifier shows up ‘outside’. This is no longer localized reshuffling. Ditto for questions. The denotation of a question is a characteristic function of a set of true answers (I guess I do use HOF in STT). In this example, there is no trace of the pronoun in the object language phrase. Finally, here our mapping should explain the order of these bindings. Thus, mapping the subject language to the object language is not that straightforward. The subject and object languages aren’t the same: we have pronouns in the subject language but not in the object language.

We also have the third language: the meta-language, in which we describe the mapping. So far, the subject language and the meta-language were both English. They don’t have to be the same: I were to speak Russian (given this is a Russian session); the meta-language would have been Russian. Using a natural language to describe how form relates to meaning is what, I believe, Karttunen calls ‘paper-and-pencil linguistics’. He submitted that it may not be enough. A more precise and mechanical ways are needed. For that, we turn to the dark side, CS. We’ll use the language we all hate to love.

Computer language denotations

Mary saw J\'on.

- ▷ Mary saw Jón.

TeX: the subject language is what we write in the TeX file; the object language is the typeset text, usually DVI or PDF. The denotation is what we see on the paper or screen. The metalanguage: TeX or Pascal. Subject and object languages are certainly different.

Other possible examples: from ML to assembly, or from Prolog to WAM. Here is our first example: simple mapping. It is akin to this natural language example. In both cases, the output is essentially like input, with some localized processing such as dis-inflection.

Computer language denotations

Mary saw J\'on.

▷ Mary saw Jón.

Mary saw John. ▷ see john mary

Computer language denotations

Mary saw J\'on.

$\triangleright$ Mary saw Jón.

Mary sees mothers\footnote{of everyone}.

$\triangleright$ Mary sees mothers^1. $\cdots$ $\cdots$ ^{1}of everyone

This example is like quantification. The occurrence of ‘footnote’ (the quantifier) is replaced with some ‘variable’, which is bound somewhere *far on the outside*.

Computer language denotations

Mary saw J\'on.

▷ Mary saw Jón.

Mary sees mothers\footnote{of everyone}.

▷ Mary sees mothers[1]. · · · · · · [1]of everyone

Mary sees everyone's mother. ▷ $\forall_c\, \mathrm{see}(\mathrm{mother}\, c)\, \mathrm{mary}$

Computer language denotations

Mary saw J\'on.

- ▷ Mary saw Jón.

Mary sees mothers\footnote{of everyone}.

- ▷ Mary sees mothers[1]. ⋯ ⋯ [1]of everyone

`\begin{block}`{Sentence}
The type of \insertblocktitle: t`\end{block}`

- ▷ **Sentence** The type of Sentence: t

Pronouns are pervasive in programming languages and in TeX. For example, current-date, citation reference, or, this example. Again we see the correspondence with the natural language. In both cases, the denotation has no traces of pronouns. Pronouns exists only in subject language, but not in object language.

Computer language denotations

Mary saw J\'on.

▷ Mary saw Jón.

Mary sees mothers\footnote{of everyone}.

▷ Mary sees mothers[1]. · · · · · · [1]of everyone

`\begin{block}`{Sentence}
The type of \insertblocktitle: t`\end{block}`

▷ **Sentence** The type of Sentence: t

John's mother saw him. ▷ see john (mother john)

Computer language denotations

Mary saw J\'on.

▷ Mary saw Jón.

Mary sees mothers\footnote{of everyone}.

▷ Mary sees mothers^1. ⋯ ⋯ ^{1}of everyone

\begin{block}{Sentence}
The type of \insertblocktitle: t\end{block}

▷ **Sentence** The type of Sentence: t

\def\gap{The type of \insertblocktitle: t}
\begin{block}{Sentence}A few types. \gap\end{block}

▷ **Sentence** A few types. The type of Sentence: t

A pronoun does not have to refer to something that textually occurs before. It may also refer to something that textually occurs after. In both cases, although textually the reference occurs before the definition, it is acted upon, or *evaluated* later.

Computer language denotations

Mary saw J\'on.

▷ Mary saw Jón.

Mary sees mothers\footnote{of everyone}.

▷ Mary sees mothers[1]. ⋯ ⋯ [1]of everyone

\begin{block}{Sentence}
The type of \insertblocktitle: t\end{block}

▷ **Sentence** The type of Sentence: t

\def\gap{The type of \insertblocktitle: t}
\begin{block}{Sentence}A few types. \gap\end{block}

▷ **Sentence** A few types. The type of Sentence: t

Mary's present for him$_i$, every boy$_i$ saw.

Compilation

Computer Language

Denotation: .obj file

Code generation

Type checking

Parsing and elaboration into Core

Lexing

In computer science, the transformation from a subject to an object language is typically called compilation. It is a complex process, with many phases. The bottom two phases are syntactical. That it is not to say they are easy. Many programming languages aren't context-free. Many languages include pre-processors (e.g., Lisp, C, C++, OCaml – and TeX), which are Turing-complete. Many languages (e.g., SML, Haskell) are defined by elaboration of a surface syntax into a *core*. The core is a simplified, intermediate language, often represented only as an AST. The compiler type-checks the core language and then transforms it to an object language (assembly or DVI) – our denotation.

Other phases are more semantical. The end result is the code in the object language (e.g., PDF or assembly) – which is the desired denotation.

Errors may occur on many levels. An error means that the subject language text is not acceptable, not 'grammatical'.

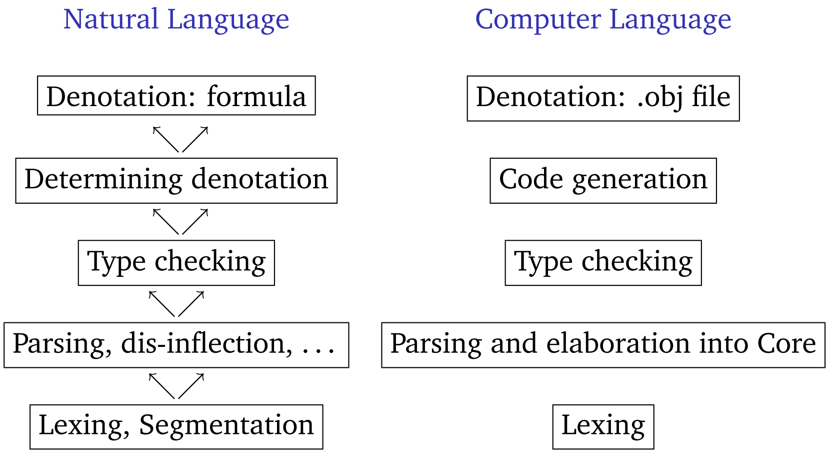
The diagram is coarse-grained: parsing is often done in several phases; code generation may include many optimization phases, etc.

Compilation

Natural Language	Computer Language
Denotation: formula	Denotation: .obj file
Determining denotation	Code generation
Type checking	Type checking
Parsing, dis-inflection, . . .	Parsing and elaboration into Core
Lexing, Segmentation	Lexing

We note that ‘understanding’ natural language, deriving meaning from an utterance, is also a process with many phases. Here too errors may occur on many levels. An error means that there is something wrong with the sentence: spelling error, incorrect conjugation, wrong word order – or more subtle problems demonstrated in the phrases at the beginning. These subtle problems show up at these higher stages, when determining denotation.

Compilation



But is this analogy good enough for us, to base our decisions on? Natural languages are very ambiguous.

Compilation

$$\begin{tikzcd}
\text{Natural Language} & \text{Computer Language} \\
\boxed{\text{Denotation: formula}} & \boxed{\text{Denotation: .obj file}} \\
\boxed{\text{Determining denotation}} \arrow[u, shift left] \arrow[u, shift right] & \boxed{\text{Code generation}} \arrow[u, shift left] \arrow[u, shift right] \\
\boxed{\text{Type checking}} \arrow[u, shift left] \arrow[u, shift right] & \boxed{\text{Type checking}} \arrow[u] \\
\boxed{\text{Parsing, dis-inflection, \ldots}} \arrow[u, shift left] \arrow[u, shift right] & \boxed{\text{Parsing and elaboration into Core}} \arrow[u, shift left] \arrow[u, shift right] \\
\boxed{\text{Lexing, Segmentation}} \arrow[u, shift left] \arrow[u, shift right] & \boxed{\text{Lexing}} \arrow[u]
\end{tikzcd}$$

But so are computer languages. Ambiguity first comes in parsing: OCaml grammar has numerous shift/reduce conflicts. The conflicts are resolved by precedence rules and various heuristics like the max munch rule. There are also ambiguity in the evaluation order, in OCaml. The same OCaml program, compiled with two different compilers may give different results. Scheme, OCaml and C and many languages are deliberately underspecified.

Compilation

Natural Language | Computer Language

$$\begin{tikzcd}
\boxed{\text{Denotation: formula}} & \boxed{\text{Denotation: .obj file}} \\
\boxed{\text{Determining denotation}} \arrow[u, phantom, "\nwarrow\nearrow"] & \boxed{\text{Code generation}} \arrow[u] \\
\boxed{\text{Type checking}} \arrow[u, phantom, "\nwarrow\nearrow"] & \boxed{\text{Type checking}} \arrow[u] \\
\boxed{\text{Parsing, dis-inflection, \ldots}} \arrow[u, phantom, "\nwarrow\nearrow"] & \boxed{\text{Parsing and elaboration into Core}} \arrow[u] \\
\boxed{\text{Lexing, Segmentation}} \arrow[u, phantom, "\nwarrow\nearrow"] & \boxed{\text{Lexing}} \arrow[u]
\end{tikzcd}$$

Normally though, a particular compiler given a particular set of compilation flags and a particular moon phase – yes, it has been known to matter – arbitrarily resolves the conflict, so we have this.

Compilation

Natural Language

$$\begin{tikzcd}
\boxed{\text{Denotation: formula}} \\
\boxed{\text{Determining denotation}} \arrow[u] \\
\boxed{\text{Type checking}} \arrow[u] \\
\boxed{\text{Parsing, dis-inflection, \ldots}} \arrow[u, shift left] \arrow[u, shift right] \\
\boxed{\text{Lexing, Segmentation}} \arrow[u, shift left] \arrow[u, shift right]
\end{tikzcd}$$

Computer Language

$$\begin{tikzcd}
\boxed{\text{Denotation: .obj file}} \\
\boxed{\text{Code generation}} \arrow[u] \\
\boxed{\text{Type checking}} \arrow[u] \\
\boxed{\text{Parsing and elaboration into Core}} \arrow[u] \\
\boxed{\text{Lexing}} \arrow[u]
\end{tikzcd}$$

For natural languages, we assume this. Ambiguity is somehow resolved at the syntactic level, so mapping from the core to denotation is deterministic. That is our working assumption.

Compilation

$$\begin{tikzcd}
\text{Natural Language} & \text{Computer Language} \\
\boxed{\text{Denotation: formula}} & \boxed{\text{Denotation: .obj file}} \\
\boxed{\text{Determining denotation}} \arrow[u] & \boxed{\text{Code generation}} \arrow[u] \\
\boxed{\text{Type checking}} \arrow[u] & \boxed{\text{Type checking}} \arrow[u] \\
\boxed{\text{Parsing, dis-inflection, \ldots}} \arrow[u, shift left] \arrow[u, shift right] & \boxed{\text{Parsing and elaboration into Core}} \arrow[u] \\
\boxed{\text{Lexing, Segmentation}} \arrow[u, shift left] \arrow[u, shift right] & \boxed{\text{Lexing}} \arrow[u]
\end{tikzcd}$$

In the following, we will be dealing only with the following portion: mapping from the Core language, intermediate language, to the denotation. The role of the type system: to delineate a set of terms whose reduction never fails to produce denotation. That is, we'd like errors to be reported at the type-checking stage. We'd like code/denotation generation to be error free: deterministic and total. Being well-typed is thus an alternative criterion of grammaticality.

Outline

Specifying rewriting rules

Mary sees John.	Subject language
$\therefore$ mary \\ (see / john)	*elaborated to* Core language
$\triangleright$ see john mary	*compiled to* Object language: denotation

Let us state our goals more precisely. We have this subject language phrase. Parsing and elaboration gives us a phrase in the core language. And from it we wish to produce the denotation, a phrase in the object language, STT. The first part, elaboration, production of the core language phrase, is outside the scope of this research. We assume that we are somehow given this, and have to produce this. What is this core language? One may think it is similar to TeX: it includes the object language (ordinary words) as literal constants, and adds markup. Our core language here likewise includes the object language, STT, literally (as literal constants). It also adds *phonetically silent* ‘markup’, so to speak.

Specifying rewriting rules

Mary sees John.

$\therefore$ mary \\(see / john)

$\triangleright$ see john mary

Mary sees John's mother.

$\therefore$ mary \\(see /(john \\ mother))

$\triangleright$ see(mother john) mary

The rewriting rules should be compositional. If we re-write as in the first example, we expect the result of the second example to be like this. So, our goal is to specify this highlighted relation, the re-writing rules. One can do this in many ways. We note that this kind of compositional re-writing is similar to evaluation in lambda-calculus. The operational semantics of lambda-calculus is re-writing!

Specifying rewriting rules

Mary sees John.

$\therefore$ mary \ (see / john)

▷ see john mary

Mary sees John's mother.

$\therefore$ mary \ (see / (john \ mother))

▷ see(mother john) mary

$$/ \overset{\text{def}}{=} \lambda^! u.\, \lambda^! v.\, uv$$
$$\backslash \overset{\text{def}}{=} \lambda^! u.\, \lambda^! v.\, vu$$

So, our intermediate, core language is lambda-calculus! Typed lambda-calculus, as we shall see soon. This is indeed the case for many programming languages, like ML and Haskell. So, the re-writing rules, the *compilation* of the core to denotation – is the *evaluation* of lambda-expressions. Denotations are computed. Indeed, if we assume these definitions, core phrases evaluate to the denotations. Disregard exclamation marks on these lambdas for now.

Non-compositionality: Questions

Mary saw who?
$\therefore$ mary \ (see / who)

So far, so good: semantics is compositional, and so are evaluation rules. Alas, semantics can also appear non-compositional, for example, in questions. Can we deal with that?

‘Who’ as the ‘breakpoint’ in natural languages.

Non-compositionality: Questions

Mary saw who?

$\therefore\ \#\ \$\ \mathrm{mary} \backslash (\mathrm{see} \,/\, \mathrm{who})$ $\qquad$ $\mathrm{who} \overset{\mathrm{def}}{=} \text{⧢} f.\ \partial_c(f\ \$\ c)$

Yes, with the help of control operators, e.g., shift. The subject phrase is elaborated into the following term in our calculus. We see operations shift, plug and the top context (top co-term). We see how they work on the example of the detailed derivation.

Non-compositionality: Questions

Mary saw who?

$$\begin{aligned} &\therefore\ \#\ \$\ \mathrm{mary} \backslash (\mathrm{see} \,/\, \mathrm{who}) \qquad\qquad \mathrm{who} \overset{\mathrm{def}}{=} \text{ш} f.\, \partial_c (f \ \$\ c) \\ &=\ \#\ \$\ \mathrm{mary} \backslash (\mathrm{see} \,/\, \text{ш} f.\, \partial_c (f \ \$\ c)) \end{aligned}$$

Non-compositionality: Questions

Mary saw who?

$$\begin{aligned} \therefore\ & \#\ \$\ \text{mary} \setminus (\text{see} \mathbin{/} \text{who}) \qquad\qquad \text{who} \overset{\text{def}}{=} \text{Ш} f.\, \partial_c (f\ \$\ c) \\ =\ & \#\ \$\ \text{mary} \setminus (\text{see} \mathbin{/} \text{Ш} f.\, \partial_c (f\ \$\ c)) \end{aligned}$$

We substitute the definition of ‘shift’. We determine the delimited context of shift. A context is what remains of a term if we take out shift and leave out the hole. A delimited context: we consider not the whole term but rather a subterm, starting with the closest plug. In this case, it is the whole term... We then do shift-transition: we replace the whole subterm with the body of shift plugged into the empty context. We also bind the removed context, the yellow part, to a co-variable here.

Non-compositionality: Questions

Mary saw who?

$$\begin{aligned}
&\therefore\ \#\ \$\ \text{mary} \backslash (\text{see} \,/\, \text{who}) && \text{who} \overset{\text{def}}{=} \text{Ш} f.\, \partial_c(f\ \$\ c) \\
&=\ \#\ \$\ \text{mary} \backslash (\text{see} \,/\, \text{Ш} f.\, \partial_c(f\ \$\ c)) && \\
&\rightsquigarrow\ \#\ \$\ \partial_c(f\ \$\ c) && f \overset{\text{def}}{=} \#\ \$\ \text{mary} \backslash (\text{see} \,/ [\ ])
\end{aligned}$$

Non-compositionality: Questions

Mary saw who?

$$\begin{aligned}
&\therefore\ \#\ \$\ \text{mary} \backslash (\text{see} \,/\, \text{who}) && \text{who} \overset{\text{def}}{=} \text{Ш} f.\ \partial_c (f\ \$\ c) \\
&=\ \#\ \$\ \text{mary} \backslash (\text{see} \,/ \text{Ш} f.\ \partial_c (f\ \$\ c)) \\
&\leadsto\ \#\ \$\ \partial_c (f\ \$\ c) && f \overset{\text{def}}{=} \#\ \$\ \text{mary} \backslash (\text{see} \,/ [\ ]) \\
&=\ \#\ \$\ \partial_c (\#\ \$\ \text{mary} \backslash (\text{see} \,/ [\ ])\ \$\ c)
\end{aligned}$$

Non-compositionality: Questions

Mary saw who?

$$\begin{aligned}
&\therefore\ \#\,\$\,\text{mary}\,\backslash(\text{see}\,/\,\text{who}) && \text{who} \overset{\text{def}}{=} \text{Ш}f.\,\partial_c(f\ \$\ c) \\
&=\ \#\,\$\,\text{mary}\,\backslash(\text{see}\,/\,\text{Ш}f.\,\partial_c(f\ \$\ c)) && \\
&\leadsto\ \#\,\$\,\partial_c(f\ \$\ c) && f \overset{\text{def}}{=} \#\,\$\,\text{mary}\,\backslash(\text{see}\,/[\,]) \\
&=\ \#\,\$\,\partial_c(\#\,\$\,\text{mary}\,\backslash(\text{see}\,/[\,])\ \$\ c) && \\
&\leadsto\ \#\,\$\,\partial_c(\#\,\$\,\text{mary}\,\backslash(\text{see}\,/c)) &&
\end{aligned}$$

We have plugged the hole. We now do regular beta-reductions.

Non-compositionality: Questions

Mary saw who?

$$\begin{aligned}
&\therefore\ \#\ \$\ \text{mary} \backslash (\text{see} / \text{who}) && \text{who} \overset{\text{def}}{=} \text{出} f.\, \partial_c(f\ \$\ c) \\
&=\ \#\ \$\ \text{mary} \backslash (\text{see} / \text{出} f.\, \partial_c(f\ \$\ c)) \\
&\rightsquigarrow\ \#\ \$\ \partial_c(f\ \$\ c) && f \overset{\text{def}}{=} \#\ \$\ \text{mary} \backslash (\text{see} / [\,]) \\
&=\ \#\ \$\ \partial_c(\#\ \$\ \text{mary} \backslash (\text{see} / [\,])\ \$\ c) \\
&\rightsquigarrow\ \#\ \$\ \partial_c(\#\ \$\ \text{mary} \backslash (\text{see} / c)) \\
&\rightsquigarrow\ \#\ \$\ \partial_c(\#\ \$\ \text{see}\ c\ \text{mary})
\end{aligned}$$

see *c* mary is a value: an object language constant. A value plugged into the top context reduces to itself. We thus discharge the top context, again and again, until we obtain the final result.

Non-compositionality: Questions

Mary saw who?

$$
\begin{array}{ll}
\therefore & \#\ \$\ \text{mary} \backslash (\text{see} \,/\, \text{who}) \qquad\qquad \text{who} \overset{\text{def}}{=} ⧢ f.\, \partial_c(f\ \$\ c) \\
= & \#\ \$\ \text{mary} \backslash (\text{see} \,/ ⧢ f.\, \partial_c(f\ \$\ c)) \\
\rightsquigarrow & \#\ \$\ \partial_c(f\ \$\ c) \qquad\qquad f \overset{\text{def}}{=} \#\ \$\ \text{mary} \backslash (\text{see} \,/ [\,]) \\
= & \#\ \$\ \partial_c(\#\ \$\ \text{mary} \backslash (\text{see} \,/ [\,])\ \$\ c) \\
\rightsquigarrow & \#\ \$\ \partial_c(\#\ \$\ \text{mary} \backslash (\text{see} \,/ c)) \\
\rightsquigarrow & \#\ \$\ \partial_c(\#\ \$\ \text{see}\ c\ \text{mary}) \\
\rightsquigarrow & \#\ \$\ \partial_c(\text{see}\ c\ \text{mary})
\end{array}
$$

Non-compositionality: Questions

Mary saw who?

$$
\begin{aligned}
&\therefore\ \#\,\$\,\mathrm{mary}\backslash(\mathrm{see}\,/\,\mathrm{who}) && \mathrm{who} \overset{\mathrm{def}}{=} \text{⧢}f.\,\partial_c(f\ \$\ c)\\
&=\ \#\,\$\,\mathrm{mary}\backslash(\mathrm{see}\,/\,\text{⧢}f.\,\partial_c(f\ \$\ c))\\
&\rightsquigarrow\ \#\,\$\,\partial_c(f\ \$\ c) && f \overset{\mathrm{def}}{=} \#\,\$\,\mathrm{mary}\backslash(\mathrm{see}\,/[\,])\\
&=\ \#\,\$\,\partial_c(\#\,\$\,\mathrm{mary}\backslash(\mathrm{see}\,/[\,])\ \$\ c)\\
&\rightsquigarrow\ \#\,\$\,\partial_c(\#\,\$\,\mathrm{mary}\backslash(\mathrm{see}\,/c))\\
&\rightsquigarrow\ \#\,\$\,\partial_c(\#\,\$\,\mathrm{see}\,c\,\mathrm{mary})\\
&\rightsquigarrow\ \#\,\$\,\partial_c(\mathrm{see}\,c\,\mathrm{mary})\\
&\rightsquigarrow\ \partial_c(\mathrm{see}\,c\,\mathrm{mary})
\end{aligned}
$$

Non-compositionality: Gaps

John, Mary saw.

$$\therefore\ \#\ \$\ \mathrm{john} \mathbin{\backslash\!\backslash} (\#\ \$\ \mathrm{mary} \mathbin{\backslash} (\mathrm{see} \mathbin{/} \_))$$

$$\_ \overset{\text{def}}{=} \text{ш}f.\ \lambda x.(f\ \$\ x)$$

$$\mathbin{\backslash\!\backslash} \overset{\text{def}}{=} \lambda x.\ \lambda y.\ y \mathbin{/\!/} x$$

Non-compositionality: Gaps

John, Mary saw.

$$\therefore\ \#\ \$\ \text{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \_))$$

$$\_ \overset{\text{def}}{=} \text{Ш} f.\ \lambda x.(f\ \$\ x)$$

$$\mathbin{\backslash\!\!\backslash} \overset{\text{def}}{=} \lambda x.\ \lambda y.\ y \mathbin{/\!\!/} x$$

$$\rightsquigarrow\ \#\ \$\ \text{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \text{Ш} f.\ \lambda x.(f\ \$\ x)))$$

Note the occurrences of plug inside. The context now is truly delimited.

Non-compositionality: Gaps

John, Mary saw.

$$\therefore\ \#\ \$\ \text{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \_)) \qquad \_ \overset{\text{def}}{=} ⧢f.\ \lambda x.(f\ \$\ x)$$

$$\mathbin{\backslash\!\!\backslash} \overset{\text{def}}{=} \lambda x.\ \lambda y.\ y \mathbin{/\!\!/} x$$

$$\leadsto\ \#\ \$\ \text{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} ⧢f.\ \lambda x.(f\ \$\ x)))$$

$$\leadsto\ \#\ \$\ \text{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \lambda x.(f\ \$\ x)) \qquad f \overset{\text{def}}{=} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} [\,]))$$

Non-compositionality: Gaps

John, Mary saw.

$$\therefore\ \#\ \$\ \mathrm{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \mathrm{mary} \mathbin{\backslash} (\mathrm{see} \mathbin{/} \_)) \qquad \_ \overset{\text{def}}{=} Ш f.\ \lambda x.(f\ \$\ x)$$

$$\mathbin{\backslash\!\!\backslash} \overset{\text{def}}{=} \lambda x.\ \lambda y.\ y \mathbin{/\!\!/} x$$

$$\rightsquigarrow\ \#\ \$\ \mathrm{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \mathrm{mary} \mathbin{\backslash} (\mathrm{see} \mathbin{/} Ш f.\ \lambda x.(f\ \$\ x)))$$

$$\rightsquigarrow\ \#\ \$\ \mathrm{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \lambda x.(f\ \$\ x)) \qquad f \overset{\text{def}}{=} (\#\ \$\ \mathrm{mary} \mathbin{\backslash} (\mathrm{see} \mathbin{/} [\ ]))$$

$$\rightsquigarrow\ \#\ \$\ \mathrm{john} \mathbin{\backslash\!\!\backslash} (\lambda x.(f\ \$\ x))$$

Non-compositionality: Gaps

John, Mary saw.

$$\therefore\ \#\ \$\ \text{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \_))$$

$$\_ \overset{\text{def}}{=} \text{ш}f.\ \lambda x.(f\ \$\ x)$$

$$\mathbin{\backslash\!\!\backslash} \overset{\text{def}}{=} \lambda x.\ \lambda y.\ y \mathbin{/\!\!/} x$$

$$\rightsquigarrow\ \#\ \$\ \text{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \text{ш}f.\ \lambda x.(f\ \$\ x)))$$

$$\rightsquigarrow\ \#\ \$\ \text{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \lambda x.(f\ \$\ x)) \qquad f \overset{\text{def}}{=} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} [\,]))$$

$$\rightsquigarrow\ \#\ \$\ \text{john} \mathbin{\backslash\!\!\backslash} (\lambda x.(f\ \$\ x))$$

$$\rightsquigarrow\ \#\ \$\ (\lambda x.(f\ \$\ x)) \mathbin{/\!\!/} \text{john}$$

Non-compositionality: Gaps

John, Mary saw.

$$\therefore\ \#\ \$\ \text{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \_)) \qquad \_ \overset{\text{def}}{=} \text{丗} f.\ \lambda x.(f\ \$\ x)$$

$$\mathbin{\backslash\!\!\backslash} \overset{\text{def}}{=} \lambda x.\ \lambda y.\ y \mathbin{/\!\!/} x$$

$$\leadsto\ \#\ \$\ \text{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \text{丗} f.\ \lambda x.(f\ \$\ x)))$$

$$\leadsto\ \#\ \$\ \text{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \lambda x.(f\ \$\ x)) \qquad f \overset{\text{def}}{=} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} [\,]))$$

$$\leadsto\ \#\ \$\ \text{john} \mathbin{\backslash\!\!\backslash} (\lambda x.(f\ \$\ x))$$

$$\leadsto\ \#\ \$\ (\lambda x.(f\ \$\ x)) \mathbin{/\!\!/} \text{john}$$

$$\leadsto_{\beta}\ \#\ \$\ f\ \$\ \text{john}$$

Non-compositionality: Gaps

John, Mary saw.

$\therefore\ \#\ \$\ \text{john} ⑊ (\#\ \$\ \text{mary} \backslash (\text{see} / \_))$ $\qquad \_ \overset{\text{def}}{=} \text{⧢} f.\ \lambda x.(f\ \$\ x)$

$\qquad ⑊ \overset{\text{def}}{=} \lambda x.\ \lambda y.\ y ⫽ x$

$\rightsquigarrow\ \#\ \$\ \text{john} ⑊ (\#\ \$\ \text{mary} \backslash (\text{see} / \text{⧢} f.\ \lambda x.(f\ \$\ x)))$

$\rightsquigarrow\ \#\ \$\ \text{john} ⑊ (\#\ \$\ \lambda x.(f\ \$\ x))$ $\qquad f \overset{\text{def}}{=} (\#\ \$\ \text{mary} \backslash (\text{see} / [\ ]))$

$\rightsquigarrow\ \#\ \$\ \text{john} ⑊ (\lambda x.(f\ \$\ x))$

$\rightsquigarrow\ \#\ \$\ (\lambda x.(f\ \$\ x)) ⫽ \text{john}$

$\rightsquigarrow\ \#\ \$\ (\#\ \$\ \text{mary} \backslash (\text{see} / [\ ]))\ \$\ \text{john}$

Non-compositionality: Gaps

John, Mary saw.

$\therefore\ \#\ \$\ \mathrm{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \mathrm{mary} \mathbin{\backslash} (\mathrm{see} \mathbin{/} \_))$ $\qquad$ $\_ \overset{\text{def}}{=} \text{Ш} f.\, \lambda x.(f\ \$\ x)$

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\mathbin{\backslash\!\!\backslash} \overset{\text{def}}{=} \lambda x.\, \lambda y.\, y \mathbin{/\!\!/} x$

$\leadsto\ \#\ \$\ \mathrm{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \mathrm{mary} \mathbin{\backslash} (\mathrm{see} \mathbin{/} \text{Ш} f.\, \lambda x.(f\ \$\ x)))$

$\leadsto\ \#\ \$\ \mathrm{john} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \lambda x.(f\ \$\ x))$ $\qquad$ $f \overset{\text{def}}{=} (\#\ \$\ \mathrm{mary} \mathbin{\backslash} (\mathrm{see} \mathbin{/} [\ ]))$

$\leadsto\ \#\ \$\ \mathrm{john} \mathbin{\backslash\!\!\backslash} (\lambda x.(f\ \$\ x))$

$\leadsto\ \#\ \$\ (\lambda x.(f\ \$\ x)) \mathbin{/\!\!/} \mathrm{john}$

$\leadsto\ \#\ \$\ (\#\ \$\ \mathrm{mary} \mathbin{\backslash} (\mathrm{see} \mathbin{/} [\ ]))\ \$\ \mathrm{john}$

$\leadsto\ \#\ \$\ (\#\ \$\ \mathrm{mary} \mathbin{\backslash} (\mathrm{see} \mathbin{/} \mathrm{john}))$

Non-compositionality: Gaps

John, Mary saw.

$$\therefore\ \#\,\$\,\mathrm{john} \mathbin{\backslash\!\backslash} (\#\,\$\,\mathrm{mary} \mathbin{\backslash} (\mathrm{see} \mathbin{/} \_)) \qquad \_ \overset{\mathrm{def}}{=} Ш f.\ \lambda x.(f\ \$\ x)$$

$$\mathbin{\backslash\!\backslash} \overset{\mathrm{def}}{=} \lambda x.\ \lambda y.\ y \mathbin{/\!/} x$$

$$\rightsquigarrow\ \#\,\$\,\mathrm{john} \mathbin{\backslash\!\backslash} (\#\,\$\,\mathrm{mary} \mathbin{\backslash} (\mathrm{see} \mathbin{/} Ш f.\ \lambda x.(f\ \$\ x)))$$

$$\rightsquigarrow\ \#\,\$\,\mathrm{john} \mathbin{\backslash\!\backslash} (\#\,\$\,\lambda x.(f\ \$\ x)) \qquad f \overset{\mathrm{def}}{=} (\#\,\$\,\mathrm{mary} \mathbin{\backslash} (\mathrm{see} \mathbin{/} [\,]))$$

$$\rightsquigarrow\ \#\,\$\,\mathrm{john} \mathbin{\backslash\!\backslash} (\lambda x.(f\ \$\ x))$$

$$\rightsquigarrow\ \#\,\$\,(\lambda x.(f\ \$\ x)) \mathbin{/\!/} \mathrm{john}$$

$$\rightsquigarrow\ \#\,\$\,(\#\,\$\,\mathrm{mary} \mathbin{\backslash} (\mathrm{see} \mathbin{/} [\,]))\ \$\ \mathrm{john}$$

$$\rightsquigarrow\ \#\,\$\,(\#\,\$\,\mathrm{mary} \mathbin{\backslash} (\mathrm{see} \mathbin{/} \mathrm{john}))$$

$$\rightsquigarrow^{+}\ \mathrm{see\ john\ mary}$$

Two effects: Raised questions

Who did Mary see?

$\therefore\ \#\,\$\,\text{who} \mathbin{\backslash\!\!\backslash} (\#\,\$\,\text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \_))$

We were able to uniformly analyze questions and raised phrases. Can we handle more complex cases, like raised questions? Which of the two control operators to do first? Left-to-right? Can we substitute who as it is? In our calculus, we can. We get to the result we've seen, which is quite satisfying from the point of view of uniformity. The raised question is transformed to the in-situ question.

Two effects: Raised questions

Who did Mary see?

$\therefore\ \# \ \$ \ \text{who} \mathbin{\backslash\!\backslash} (\# \ \$ \ \text{mary} \backslash (\text{see} / \_))$

$\rightsquigarrow\ \# \ \$ \ (\text{山} f.\, \partial_c (f \ \$ \ c)) \mathbin{\backslash\!\backslash} (\# \ \$ \ \text{mary} \backslash (\text{see} / \_))$

If we evaluate left-to-right, we would get this…

Two effects: Raised questions

Who did Mary see?

$$\therefore\ \#\ \$\ \text{who} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} / \_))$$

$$\leadsto\ \#\ \$\ (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} / \_)) \mathbin{/\!\!/} \text{who}$$

$$\leadsto\ \#\ \$\ (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} / \text{出} f.\, \lambda x. (f\ \$\ x))) \mathbin{/\!\!/} \text{who}$$

$$\mathbin{\backslash\!\!\backslash} \overset{\text{def}}{=} \lambda x.\, \lambda y.\, y \mathbin{/\!\!/} x$$

But that is not what happens. The top operation here is $\mathbin{\backslash\!\backslash}$; Here we recall its definition. Note the regular lambda. It is a non-strict operation: it permutes the terms without evaluating them. Now we get this. The $/\!\!/$ operation is eliminated by a beta-redex. We need to see if the first operand is a lambda-expression; we need to evaluate it. That is what we do now.

Two effects: Raised questions

Who did Mary see?

$$\begin{aligned}
&\therefore\ \#\ \$\ \text{who} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \_))\\
&\rightsquigarrow\ \#\ \$\ (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \_)) \mathbin{/\!\!/} \text{who}\\
&\rightsquigarrow\ \#\ \$\ (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \text{ш} f.\ \lambda x.(f\ \$\ x))) \mathbin{/\!\!/} \text{who}\\
&\rightsquigarrow^{+}\ \#\ \$\ (\lambda x.\ \#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} [\,])\ \$\ x) \mathbin{/\!\!/} \text{who}
\end{aligned}$$

Two effects: Raised questions

Who did Mary see?

$\therefore\ \#\ \$\ \text{who} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} / \_))$

$\rightsquigarrow\ \#\ \$\ (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} / \_)) \mathbin{/\!\!/} \text{who}$

$\rightsquigarrow\ \#\ \$\ (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} / \text{出} f.\, \lambda x.(f\ \$\ x))) \mathbin{/\!\!/} \text{who}$

$\rightsquigarrow^{+}\ \#\ \$\ (\lambda x.\, \#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} / [\,])\ \$\ x) \mathbin{/\!\!/} \text{who}$

Should we evaluate who now? No, we see $/\!\!/$ operation and the regular, un-annotated λ. In CBN, the beta-reduction substitutes the argument of the application as it is.

Two effects: Raised questions

Who did Mary see?

$\therefore\ \#\ \$\ \text{who} \mathbin{\backslash\!\backslash} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \_))$

$\leadsto\ \#\ \$\ (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \_)) \mathbin{/\!/} \text{who}$

$\leadsto\ \#\ \$\ (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \text{ш} f.\ \lambda x.(f\ \$\ x))) \mathbin{/\!/} \text{who}$

$\leadsto^{+}\ \#\ \$\ (\lambda x.\ \#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} [\ ])\ \$\ x) \mathbin{/\!/} \text{who}$

$\leadsto\ \#\ \$\ \#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \text{who})$

And so we get this: which is equal to the result of elaboration of the in-situ question ‘Mary saw who?’ We have thus transformed a raised question to an in-situ one. We have already computed the denotation for the in-situ question, which is this.

Two effects: Raised questions

Who did Mary see?

$$\begin{aligned}
&\therefore\ \#\ \$\ \text{who} \mathbin{\backslash\!\backslash} (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \_\!\_))\\
&\rightsquigarrow\ \#\ \$\ (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \_\!\_)) \mathbin{/\!\!/} \text{who}\\
&\rightsquigarrow\ \#\ \$\ (\#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \text{Ш} f.\, \lambda x. (f\ \$\ x))) \mathbin{/\!\!/} \text{who}\\
&\rightsquigarrow^{+}\ \#\ \$\ (\lambda x.\ \#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} [\,])\ \$\ x) \mathbin{/\!\!/} \text{who}\\
&\rightsquigarrow\ \#\ \$\ \#\ \$\ \text{mary} \mathbin{\backslash} (\text{see} \mathbin{/} \text{who})\\
&\rightsquigarrow^{+}\ \partial_c(\text{see}\ c\ \text{mary})
\end{aligned}$$

Superiority

Who saw whom?

$\therefore\ \# \$ \text{who} \mathbin{\backslash\!\!\backslash} (\# \$ \_ \mathbin{\backslash} (\text{see} / \text{who}))$

Superiority

Who saw whom?

$\therefore$ # \$ who $\mathbin{\backslash\!\!\backslash}$(# \$ _ \ (see / who))

$\rightsquigarrow^{+}$ # \$ (# \$ _ \ (see / who)) $/\!\!/$ who

Now we have three control operators, three control effects. We first proceed as before, taking advantage of the fact $\backslash\!\backslash$ permutes the terms as they are. As before, $/\!/$ needs to evaluate the left argument. Now, we have two control operators in the left argument. Which to do first? We recall the definition of $\backslash$ and $/$. Note these marks on lambda: strictness marks. These lambdas demand values: applications involving these lambdas must evaluate the argument. Note, the left-most is evaluated first. So, trace is evaluated, as the left argument of the top-most operator, $\backslash$

Superiority

Who saw whom?

$\therefore\ \#\ \$\ \text{who} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \_ \mathbin{\backslash} (\text{see} / \text{who}))$

$\rightsquigarrow^{+}\ \#\ \$\ (\#\ \$\ \_ \mathbin{\backslash} (\text{see} / \text{who})) \mathbin{/\!\!/} \text{who}$

$$/ \overset{\text{def}}{=} \lambda^! u.\, \lambda^! v.\, uv$$
$$\backslash \overset{\text{def}}{=} \lambda^! u.\, \lambda^! v.\, vu$$

Superiority

Who saw whom?

$$\begin{aligned}
&\therefore\ \#\,\$\,\text{who} \mathbin{\backslash\!\!\backslash} (\#\,\$\,\_ \backslash (\text{see} / \text{who}))\\
&\rightsquigarrow^{+}\ \#\,\$\,(\#\,\$\,\_ \backslash (\text{see} / \text{who})) \mathbin{/\!\!/} \text{who}\\
&=\ \#\,\$\,(\#\,\$\,\_ \backslash (\text{see} / \text{who})) \mathbin{/\!\!/} \text{who}\\
&\rightsquigarrow^{+}\ \#\,\$\,(\#\,\$\ \text{who} \backslash (\text{see} / \text{who}))
\end{aligned}$$

Superiority

Who saw whom?

$$
\begin{aligned}
\therefore\ & \#\ \$\ \text{who} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \_ \backslash (\text{see} / \text{who})) \\
\leadsto^{+}\ & \#\ \$\ (\#\ \$\ \_ \backslash (\text{see} / \text{who})) \mathbin{/\!\!/} \text{who} \\
=\ & \#\ \$\ (\#\ \$\ \_ \backslash (\text{see} / \text{who})) \mathbin{/\!\!/} \text{who} \\
\leadsto^{+}\ & \#\ \$\ (\#\ \$\ \text{who} \backslash (\text{see} / \text{who}))
\end{aligned}
$$

We again have the same dilemma: which of two who to deal with first. Again, the strictness, or the demand for values, decides.

Superiority

Who saw whom?

$\therefore\ \#\,\$\,\text{who}\,\backslash\!\!\backslash(\#\,\$\,\_\,\backslash\,(\text{see}\,/\,\text{who}))$

$\rightsquigarrow^{+}\ \#\,\$\,(\#\,\$\,\_\,\backslash\,(\text{see}\,/\,\text{who}))\,/\!\!/\,\text{who}$

$=\ \#\,\$\,(\#\,\$\,\_\,\backslash\,(\text{see}\,/\,\text{who}))\,/\!\!/\,\text{who}$

$\rightsquigarrow^{+}\ \#\,\$\,(\#\,\$\ \text{who}\,\backslash(\text{see}\,/\,\text{who}))$

Superiority

Who saw whom?

$$
\begin{aligned}
&\therefore\ \#\ \$\ \text{who} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \_ \backslash (\text{see} / \text{who})) \\
&\rightsquigarrow^{+}\ \#\ \$\ (\#\ \$\ \_ \backslash (\text{see} / \text{who})) \mathbin{/\!\!/} \text{who} \\
&=\ \#\ \$\ (\#\ \$\ \_ \backslash (\text{see} / \text{who})) \mathbin{/\!\!/} \text{who} \\
&\rightsquigarrow^{+}\ \#\ \$\ (\#\ \$\ \text{who} \backslash (\text{see} / \text{who})) \\
&\rightsquigarrow^{+}\ \#\ \$\ (\#\ \$\ \partial_{c_1} (\#\ \$\ c_1 \backslash (\text{see} / \text{who})))
\end{aligned}
$$

Superiority

Who saw whom?

$$
\begin{aligned}
&\therefore\ \#\,\$\,\text{who} \mathbin{\backslash\!\!\backslash} (\#\,\$\,\_ \mathbin{\backslash} (\text{see} \mathbin{/} \text{who}))\\
&\rightsquigarrow^{+}\ \#\,\$\,(\#\,\$\,\_ \mathbin{\backslash} (\text{see} \mathbin{/} \text{who})) \mathbin{/\!\!/} \text{who}\\
&=\ \#\,\$\,(\#\,\$\,\_ \mathbin{\backslash} (\text{see} \mathbin{/} \text{who})) \mathbin{/\!\!/} \text{who}\\
&\rightsquigarrow^{+}\ \#\,\$\,(\#\,\$\ \text{who} \mathbin{\backslash} (\text{see} \mathbin{/} \text{who}))\\
&\rightsquigarrow^{+}\ \#\,\$\,(\#\,\$\,\partial_{c_1}(\#\,\$\,c_1 \mathbin{\backslash} (\text{see} \mathbin{/} \text{who})))
\end{aligned}
$$

Superiority

Who saw whom?

$$\begin{aligned}
&\therefore\ \#\,\$\,\text{who} \mathbin{\backslash\!\!\backslash} (\#\,\$\,\_ \backslash (\text{see} / \text{who}))\\
&\leadsto^{+}\ \#\,\$\,(\#\,\$\,\_ \backslash (\text{see} / \text{who})) \mathbin{/\!\!/} \text{who}\\
&=\ \#\,\$\,(\#\,\$\,\_ \backslash (\text{see} / \text{who})) \mathbin{/\!\!/} \text{who}\\
&\leadsto^{+}\ \#\,\$\,(\#\,\$\ \text{who} \backslash (\text{see} / \text{who}))\\
&\leadsto^{+}\ \#\,\$\,(\#\,\$\,\partial_{c_1}(\#\,\$\,c_1 \backslash (\text{see} / \text{who})))\\
&\leadsto^{+}\ \#\,\$\,\#\,\$\,\partial_{c_1}(\#\,\$\,\partial_{c_2}(\#\,\$\,c_1 \backslash (\text{see} / c_2)))
\end{aligned}$$

Superiority

Who saw whom?

$$
\begin{aligned}
&\therefore\ \# \,\$\, \text{who} \mathbin{\backslash\!\!\backslash} (\# \,\$\, \_ \backslash (\text{see} / \text{who})) \\
&\rightsquigarrow^{+}\ \# \,\$\, (\# \,\$\, \_ \backslash (\text{see} / \text{who})) \mathbin{/\!\!/} \text{who} \\
&=\ \# \,\$\, (\# \,\$\, \_ \backslash (\text{see} / \text{who})) \mathbin{/\!\!/} \text{who} \\
&\rightsquigarrow^{+}\ \# \,\$\, (\# \,\$\, \text{who} \backslash (\text{see} / \text{who})) \\
&\rightsquigarrow^{+}\ \# \,\$\, (\# \,\$\, \partial_{c_1} (\# \,\$\, c_1 \backslash (\text{see} / \text{who}))) \\
&\rightsquigarrow^{+}\ \# \,\$\, \# \,\$\, \partial_{c_1} (\# \,\$\, \partial_{c_2} (\# \,\$\, c_1 \backslash (\text{see} / c_2))) \\
&\rightsquigarrow^{+}\ \partial_{c_1} (\partial_{c_2} (\text{see}\, c_2 c_1))
\end{aligned}
$$

*Superiority

*Who did who see?

$\therefore\ \#\ \$\ \text{who} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \text{who} \backslash (\text{see} / \_))$

*Superiority

*Who did who see?

$\therefore\ \#\ \$\ \text{who} \mathbin{\backslash\!\backslash} (\#\ \$\ \text{who} \mathbin{\backslash} (\text{see} \mathbin{/} \_))$

$\leadsto^{+}\ \#\ \$\ (\#\ \$\ \text{who} \mathbin{\backslash} (\text{see} \mathbin{/} \_)) \mathbin{/\!/} \text{who}$

*Superiority

*Who did who see?

$\therefore\ \# \,\$\, \text{who} \mathbin{\backslash\!\!\backslash} (\# \,\$\, \text{who} \mathbin{\backslash} (\text{see} / \_))$

$\leadsto^{+}\ \# \,\$\, (\# \,\$\, \text{who} \mathbin{\backslash} (\text{see} / \_)) \mathbin{/\!\!/} \text{who}$

$\leadsto^{+}\ \# \,\$\, (\partial_{c_1} (\# \,\$\, c_1 \mathbin{\backslash} (\text{see} / \_))) \mathbin{/\!\!/} \text{who}$

A bit simplified. Beta-reduction cannot be performed. We are stuck. Computing denotation may fail. Which we take as an indication of the sentence to be ungrammatical.

CBN $\lambda\text{╫}$-calculus

$$\begin{array}{ll}
\text{Primitive Constants} & D ::= \text{john} \mid \text{mary} \mid \text{see} \mid \text{tall} \mid \text{mother} \\
\text{Constants} & C ::= D \mid CC \mid c \mid \forall_c \mid \partial_c \\
\text{Terms} & E, F ::= V \mid x \mid F \,/\!\!/\, E \mid E \,/\, F \mid Q \,\$\, E \mid \text{╫}k : S.\, E \\
\text{Values} & V ::= C \mid u \mid \lambda x{:}T.\, E \mid W \\
\text{Strict Values} & W ::= \lambda^! u{:}U.\, E \\
\text{Coterms} & Q ::= k \mid \# \mid E, Q \mid Q;_! W \mid E,_c Q \mid Q;_c V
\end{array}$$

Term equalities

$$\begin{array}{ll}
Q \,\$\, F \,/\!\!/\, E = E, Q \,\$\, F & Q \,\$\, W \,/\!\!/\, E = Q;_! W \,\$\, E \\
Q \,\$\, F \,/\, E = E,_c Q \,\$\, F & Q \,\$\, V \,/\, E = Q;_c V \,\$\, E \\
\# \,\$\, V = V &
\end{array}$$

Transitions

$$\begin{array}{lll}
Q_1 \,\$\, \cdots \,\$\, Q_n \,\$\, (\lambda x.\, E) \,/\!\!/\, F & \rightsquigarrow & Q_1 \,\$\, \cdots \,\$\, Q_n \,\$\, E\{x \mapsto F\} \\
Q_1 \,\$\, \cdots \,\$\, Q_n \,\$\, (\lambda^! x.\, E) \,/\!\!/\, V & \rightsquigarrow & Q_1 \,\$\, \cdots \,\$\, Q_n \,\$\, E\{x \mapsto V\} \\
Q_1 \,\$\, \cdots \,\$\, Q_n \,\$\, C_1 \,/\, C_2 & \rightsquigarrow & Q_1 \,\$\, \cdots \,\$\, Q_n \,\$\, C_1 C_2 \\
Q_1 \,\$\, \cdots \,\$\, Q_n \,\$\, Q \,\$\, \text{╫}k.\, E & \rightsquigarrow & Q_1 \,\$\, \cdots \,\$\, Q_n \,\$\, \# \,\$\, E\{k \mapsto Q\}
\end{array}$$

CBN $\lambda\text{╫}$-calculus

Primitive Constants	$D ::= \text{john} \mid \text{mary} \mid \text{see} \mid \text{tall} \mid \text{mother}$
Constants	$C ::= D \mid CC \mid c \mid \forall_c \mid \partial_c$
Terms	$E, F ::= V \mid x \mid F \mathbin{/\!/} E \mid E \mathbin{/} F \mid Q \mathbin{\$} E \mid \text{╫}k : S. E$
Values	$V ::= C \mid u \mid \lambda x{:}T.E \mid W$
Strict Values	$W ::= \lambda^! u{:}U.E$
Coterms	$Q ::= k \mid \# \mid E, Q \mid Q ;_! W \mid E ,_c Q \mid Q ;_c V$

Term equalities

$$\begin{array}{ll} Q \mathbin{\$} F \mathbin{/\!/} E = E, Q \mathbin{\$} F & Q \mathbin{\$} W \mathbin{/\!/} E = Q ;_! W \mathbin{\$} E \\ Q \mathbin{\$} F \mathbin{/} E = E ,_c Q \mathbin{\$} F & Q \mathbin{\$} V \mathbin{/} E = Q ;_c V \mathbin{\$} E \\ \# \mathbin{\$} V = V & \end{array}$$

Transitions

$$\begin{array}{lcl} Q_1 \mathbin{\$} \cdots \mathbin{\$} Q_n \mathbin{\$} (\lambda x. E) \mathbin{/\!/} F & \rightsquigarrow & Q_1 \mathbin{\$} \cdots \mathbin{\$} Q_n \mathbin{\$} E\{x \mapsto F\} \\ Q_1 \mathbin{\$} \cdots \mathbin{\$} Q_n \mathbin{\$} (\lambda^! x. E) \mathbin{/\!/} V & \rightsquigarrow & Q_1 \mathbin{\$} \cdots \mathbin{\$} Q_n \mathbin{\$} E\{x \mapsto V\} \\ Q_1 \mathbin{\$} \cdots \mathbin{\$} Q_n \mathbin{\$} C_1 \mathbin{/} C_2 & \rightsquigarrow & Q_1 \mathbin{\$} \cdots \mathbin{\$} Q_n \mathbin{\$} C_1 C_2 \\ Q_1 \mathbin{\$} \cdots \mathbin{\$} Q_n \mathbin{\$} Q \mathbin{\$} \text{╫}k. E & \rightsquigarrow & Q_1 \mathbin{\$} \cdots \mathbin{\$} Q_n \mathbin{\$} \# \mathbin{\$} E\{k \mapsto Q\} \end{array}$$

Symmetry of shift and lambda: both are binding forms; one binds a variable, the other binds a co-variable. When lambda-abstraction is applied, it takes a term on the right and substitutes into the body. When shift is applied, it takes a co-term on the left and substitutes into the body.

Outline

Types

Types	$T ::= U \mid S \downarrow U$
Pure types	$U ::= U \rightharpoonup T \mid T \rightarrow T \mid B$
Base types	$B ::= \mathrm{t} \mid \mathrm{e} \mid B \rightharpoondown B$
Cotypes	$S ::= U \uparrow U$

Typing of constants

$\mathrm{john} : \mathrm{e}$	$\mathrm{mary} : \mathrm{e}$
$\mathrm{tall} : \mathrm{e} \rightharpoondown \mathrm{t}$	$\mathrm{mother} : \mathrm{e} \rightharpoondown \mathrm{e}$
$\mathrm{see} : \mathrm{e} \rightharpoondown \mathrm{e} \rightharpoondown \mathrm{t}$	$c : \mathrm{e}$
$\forall_c : \mathrm{t} \rightharpoondown \mathrm{t}$	$\partial_{cB} : B \rightharpoondown (\mathrm{e} \rightharpoondown B)$

$$\frac{C_1 : (B_2 \rightharpoondown B) \quad C_2 : B_2}{C_1 C_2 : B}$$

We note that types can be pure U (or, imposing no constraint on the contexts they may appear in, except that the context should accept a value of the type U), and impure $S \downarrow U$. The latter require the context to handle the effect (in the simlest case: the context should handle the execption.)

Typing judgments: arrow introduction

$$\dfrac{\begin{matrix}[u : U]\\ \vdots\\ E : T\end{matrix}}{\lambda^! u{:}U.\,E : U \rightharpoonup T}\;\lambda^! \qquad \dfrac{\begin{matrix}[x : T_1]\\ \vdots\\ E : T_2\end{matrix}}{\lambda x{:}T_1.\,E : T_1 \to T_2}\;\lambda \qquad \dfrac{\begin{matrix}[k : S]\\ \vdots\\ \#\ \$\ E : U\end{matrix}}{\uplus k : S.\,E : S \downarrow U}\;\uplus$$

$$\dfrac{\begin{matrix}[u : U_1]\\ \vdots\\ Q\ \$\ u : U_2\end{matrix}}{Q : U_1 \uparrow U_2}\;{\uparrow}\mathrm{I}$$

Typing judgments: arrow elimination

$$\frac{E_1 : (B_2 \rightharpoondown B) \quad E_2 : B_2}{E_1 \mathbin{/} E_2 : B} \rightharpoondown\mathrm{E} \qquad \frac{F : U_1 \rightharpoonup T \quad E : U_2 \quad U_2 \leq U_1}{F \mathbin{/\!\!/} E : T} \rightharpoonup\mathrm{E}$$

$$\frac{F : U_1 \rightharpoonup T_1 \quad E : U_2 \uparrow U_I \downarrow U_R \qquad \begin{array}{c}[u : U_2 \quad k : U_I \uparrow U_R] \\ \vdots \\ k \mathbin{\text{¢}} Fu : T\end{array}}{F \mathbin{/\!\!/} E : T} \rightharpoonup\mathrm{E}_1$$

$$\frac{F : T_1 \rightarrow T \quad E : T_2 \quad T_2 \leq T_1}{F \mathbin{/\!\!/} E : T} \rightarrow\mathrm{E}$$

$$\frac{F : U_1 \uparrow U_I \downarrow U_R \quad E : T_2 \qquad \begin{array}{c}[u : U_1 \quad k : U_I \uparrow U_R] \\ \vdots \\ k \mathbin{\text{¢}} uE : T\end{array}}{F \mathbin{/\!\!/} E : T} \rightarrow\mathrm{E}_1$$

$$\frac{Q : U_1 \uparrow U \quad E : U_2 \quad U_2 \leq U_1}{Q \mathbin{\$} E : U} \uparrow\mathrm{E} \qquad \frac{Q : S_1 \quad E : S_2 \downarrow U \quad S_1 \leq S_2}{Q \mathbin{\$} E : U} \downarrow\mathrm{E}$$

Typing judgments: effect composition

$$\frac{Q : U_I \uparrow U_R \quad E : U}{Q \mathbin{¢} E : U \uparrow U_I \downarrow U_R}\, ¢_U \qquad \frac{Q : U_I \uparrow U_R \quad E : S \downarrow U_2 \quad U_2 \leq U_I}{Q \mathbin{¢} E : S \downarrow U_R}\, ¢_T$$

Subtyping

$$\frac{}{T \leq T} \qquad \frac{U \leq U_A \quad U_I \leq U_R}{U \leq U_A \uparrow U_I \downarrow U_R} \qquad \frac{U'_A \leq U_A \quad T_R \leq T'_R}{(U_A \rightharpoonup T_R) \leq (U'_A \rightharpoonup T'_R)}$$

$$\frac{T'_A \leq T_A \quad T_R \leq T'_R}{(T_A \rightarrow T_R) \leq (T'_A \rightarrow T'_R)} \qquad \frac{(U_A \rightharpoonup T_R) \leq (U'_A \rightharpoonup T'_R)}{(U_A \rightharpoonup T_R) \leq (U'_A \rightarrow T'_R)}$$

$$\frac{U_A \leq U'_A \quad U'_I \leq U_I \quad U_R \leq U'_R}{(U_A \uparrow U_I \downarrow U_R) \leq (U'_A \uparrow U'_I \downarrow U'_R)} \qquad \frac{U'_A \leq U_A \quad U_I \leq U'_I}{(U_A \uparrow U_I) \leq (U'_A \uparrow U'_I)}$$

Subtyping essentially shows that a pure term can always be used where an effectful term is expected, and if the function whose argument is of the pure type is expected, a corresponding strict function can be used instead.

*Superiority revisited

*Who did who see?

$\therefore\ \#\ \$\ \text{who} \mathbin{\backslash\!\!\backslash} (\#\ \$\ \text{who} \mathbin{\backslash} (\text{see} / \_))$

$\text{who} \mathbin{\backslash} (\text{see} / \_) \not: T$

We saw the evaluation of that phrase failed. If we apply the typing rules, we see that the term cannot be typed. Moreover, we can see that even this subterm cannot be typed. This subterm cannot be evaluated: there are two control operators, but there are no plugs. But we can attempt to type such terms nevertheless. This term does not type: details are in the paper. This term corresponds to an incomplete phrase, ‘who see trace’. We predict that any sentence with that phrase ungrammatical.

*Superiority revisited

*Who did who see?

$\therefore\ \# \mathbin{\$} \text{who} \mathbin{\backslash\!\!\backslash} (\# \mathbin{\$} \text{who} \backslash (\text{see} / \_))$

$\text{who} \backslash (\text{see} / \_) \not: T$

$*\cdots$ who see $\_\cdots$

Outline

Conclusions

- Typed CBN calculus with delimited control
- Improvement in uniformity and typing. No type raising
- Types reflect effects, show context-dependence
- Uniform analyses avoiding overgeneration require both CBN and typing
- Types permit ruling on incomplete sentences
- The calculus implemented
  All analyses mechanically verified

Linguistics offers the first interesting application of the typed CBN shift/reset

We introduce a type system to delineate a set of terms whose reduction never fails to produce denotation. Being well-typed is thus an alternative criterion of grammaticality. Typeability, unlike reducibility, has an advantage of being applicable to separate phrases rather than the whole sentences.

Both typing and CBN are needed for correct prediction of superiority and binding in wh-questions with topicalization while maintaining the uniformity.

The calculus (dynamic and static semantics) has been implemented, and all the analyses have been mechanically verified.Color-coding