Context and well-formedness: the dynamics of ellipsis

Ronnie Cann (E'bgh) Ruth Kempson (KCL) Matthew Purver (CSLI)

Acknowledgements:
Wilfried Meyer-Viol, Masayuki Otsuka

18th August 2005

Outline

Context and well-formedness

1 Indexicality

(a) I bumped into Mary yesterday.
(b) #I will bump into Mary yesterday.
(c) #I bumped into Mary tomorrow.

Contradictory information leads to anomaly.

2 Anaphora

(a) Bill hit his head on the doorframe and he cried.
(b) #Mary hit her head on the doorframe and he cried.
(c) Sue detests her boss and thinks the foolish man is sleeping with his secretary.
(d) #Sue detests her desk and thinks the foolish man is sleeping with his secretary.

What is the status of (2.a,d) without appropriate antecedents for *he* and *the foolish man*?

Context and well-formedness

3 Intrasentential Ellipsis

(a) Mary washed her hair and so did Bill.
(b) Bill dislikes something but it's not clear what.
(c) Sue sang a ballad for John and some lieder too.
(d) Sue gave John a book and Bill a CD.

4 (a) *Mary was tall and so did Bill.
(b) #Bill dislikes coffee but it's not clear what.
(c) #Sue is sick, and some lieder too.
(d) *Sue sings well and Bill a CD.

Immediate linguistic context essential for licensing ellipsis.

Context and well-formedness

5 Intersentential Ellipsis

(a) A: Mary washed her hair. B: So did Bill.
(b) Bill dislikes coffee. I don't know why.
(c) Sue sang a ballad for John. Some lieder too.
(d) A: Sue gave John a book. B: And Bill a CD.

6 (a) A: Who washed the dishes? B: John (did).
(b) A: Who does Mary dislike? B: Herself.
(c) A: Who does everyone love? B: Their mother.
(d) A: How was the cat killed? B: I believe with a knife.

7 (a) A: Mary washed her hair. B: *So is Bill.
(b) Bill dislikes coffee. #I don't know what.
(c) Sue is sick. #Some lieder, too.
(d) Sue sings well. *And Bill a CD.

Discourse context essential for licensing ellipsis.

Context and well-formedness

8 Dialogue continuations:

(a) Ruth: What did Alex ....
Hugh: give Eliot? A rabbit.

(b) Ruth: Where have you got to...
Hugh: with your book? Not past the first page.

What is the grammatical status of the fragments in a discourse:

– Dialogue ellipsis independent of intrasentential ellipsis?

– Any fragment of dialogue is well-formed in its own right?

Syntax is context-dependent

The Flow of Language Understanding

Trees as representations of **semantic content** (LF) NOT representations of distributional properties of words or strings.

Syntax is the **process** by which such trees are constructed through the time-linear (top-down) parse of a string of words uttered in context. (Parsing as a grammatical formalism)

Parsing and generation use the same grammatical architecture.

Inferential processes interact with syntax to define well-formed output trees.

Context is necessary for the successful completion of the parsing process.

The Flow of Language Understanding -the framework

- Semantic structure is represented in terms of binary (argument/functor) trees.
- The process of tree-building is driven by concepts of underspecification encoded as REQUIREMENTS to specify certain types of information.

The starting point Content of*John upset Mary*

$$?Ty(t) \quad \mapsto \quad Ty(t), Fo(\text{Upset}'(\text{Mary}')(\text{John}')), \diamondsuit$$

$$Ty(e), Fo(\text{John}') \qquad Ty(e \rightarrow t), Fo(\text{Upset}'(\text{Mary}'))$$

$$Ty(e), Fo(\text{Mary}') \qquad Ty(e \rightarrow (e \rightarrow t)), Fo(\text{Upset}')$$

- **Grammaticality**: For every wellformed string at least one complete logical form can be constructed from the words in sequence, with no requirements outstanding.

The Flow of Language Understanding – Parsing

Parsing ‘ John upset Mary’

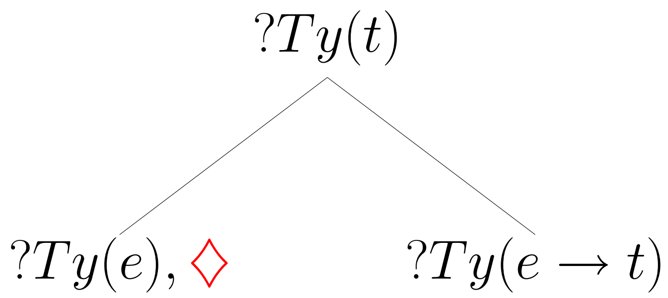
$$?Ty(t), \diamondsuit$$

The Starting Point: The Goal $?Ty(t)$: To establish some tree with a rootnode with a propositional formula as interpretation.

$\diamondsuit$ the ‘pointer’ indicating which node is under development.

The Flow of Language Understanding – Parsing

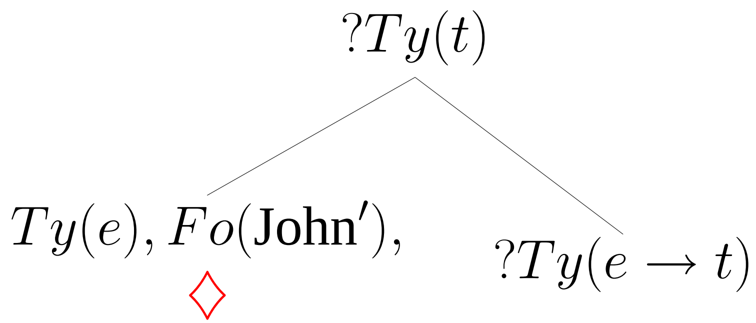
Parsing ‘John upset Mary’



Computational Actions (optional): provide general means of updating partial trees.

The Flow of Language Understanding – Parsing

Parsing ‘John upset Mary’

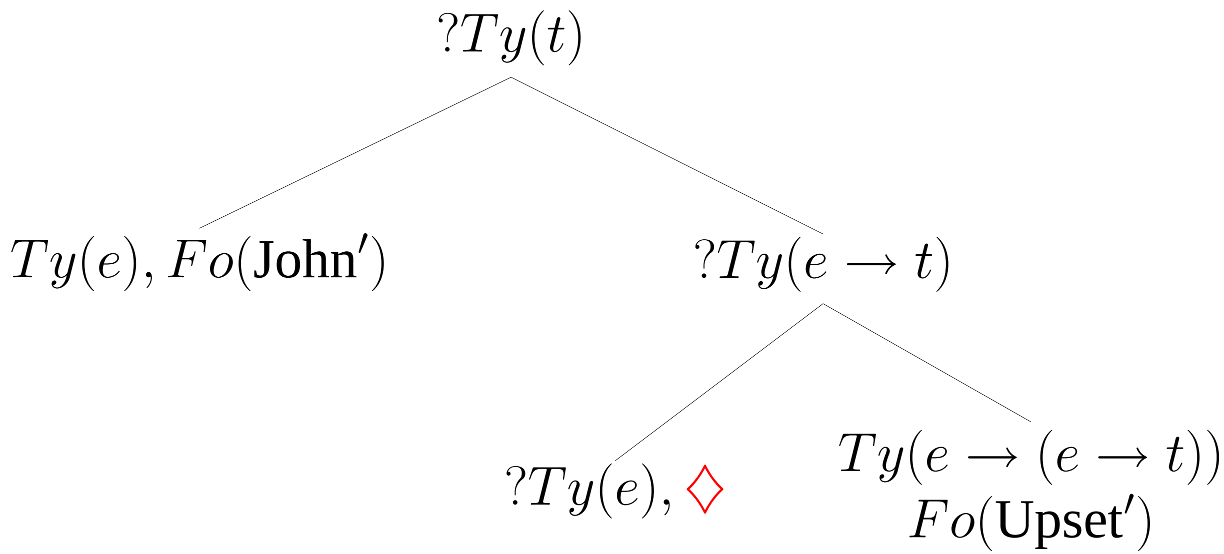


Lexical Actions (obligatory): words provide procedures for updating partial trees, adding nodes, requirements or formulae:

John	IF	$?Ty(e)$	*Trigger*
	THEN	$put(Ty(e), Fo(\text{John}'))$	*Actions*
	ELSE	ABORT	*Failure*

The Flow of Language Understanding – Parsing

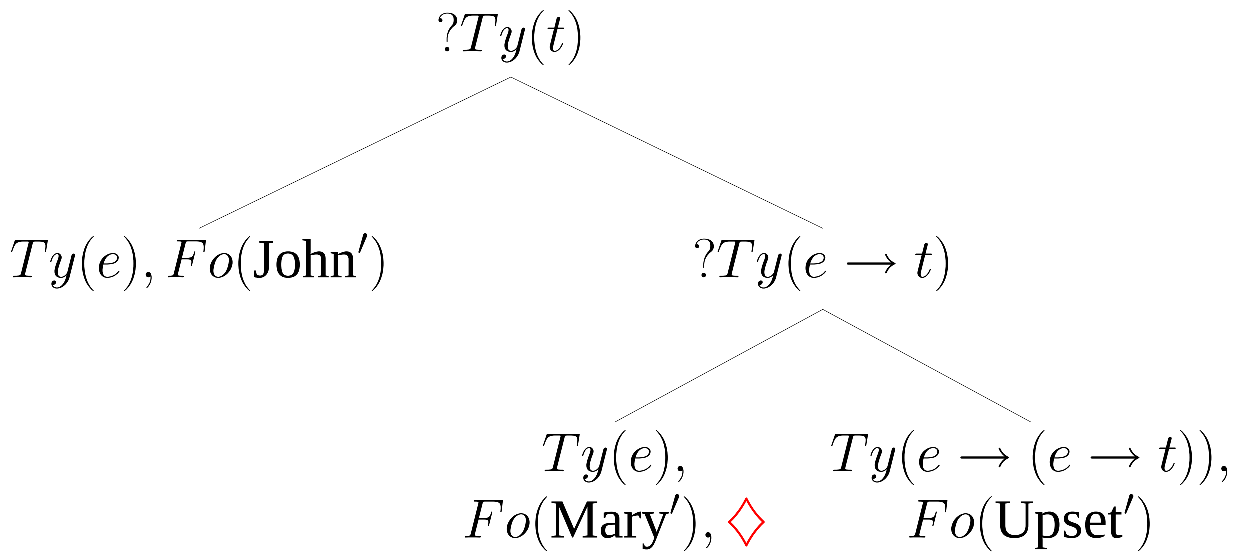
Parsing ‘John upset Mary’



Lexical actions may build nodes and add requirements, as well as merely annotating nodes.

The Flow of Language Understanding – Parsing

Parsing ‘John upset Mary’



The Flow of Language Understanding – Parsing

Parsing ‘John upset Mary’

$$Ty(t), Fo(\text{Upset}'(\text{Mary}')(\text{John}')), \Diamond$$

$$Ty(e), Fo(\text{John}') \qquad Ty(e \rightarrow t), Fo(\text{Upset}'(\text{Mary}'))$$

$$Ty(e), Fo(\text{Mary}') \qquad Ty(e \rightarrow (e \rightarrow t)), Fo(\text{Upset}')$$

Parses are completed by applying Functional Application over types

The Flow of Language Understanding – Parsing

LOFT (Logic of Finite Trees) (Blackburn and Meyer-Viol 1994)
$\langle\downarrow_0\rangle$: argument daughter of X.
$\langle\downarrow_1\rangle$: functor daughter of X.
$\langle\uparrow\rangle$: mother.
$\langle\uparrow_*\rangle$: dominated by.
$\langle\downarrow_*\rangle$: dominates.

Requirements: $?X$ for any X including modal statements – a requirement may be stated at one point in a parse that is to be satisfied at some later stage

(e.g. object case $?\langle\uparrow_0\rangle Ty(e \rightarrow t)$ - at some point current node must be dominated by a predicate node).

The Flow of Language Understanding – Parsing

Left Dislocation:

A subtree may be associated with an underspecified dominance relation with respect to some node with address $Tn(n)$

$$\langle\uparrow_*\rangle Tn(n)$$

(Tn = treenode) with a requirement to find a fixed position within the tree

$$?\exists \mathbf{x}.Tn(\mathbf{x})$$

$$\langle\uparrow_*\rangle\alpha \rightarrow \langle\uparrow\rangle\alpha \vee \langle\uparrow\rangle\langle\uparrow_*\rangle\alpha$$

The Flow of Language Understanding – Parsing

Parsing ‘Mary, John upset’

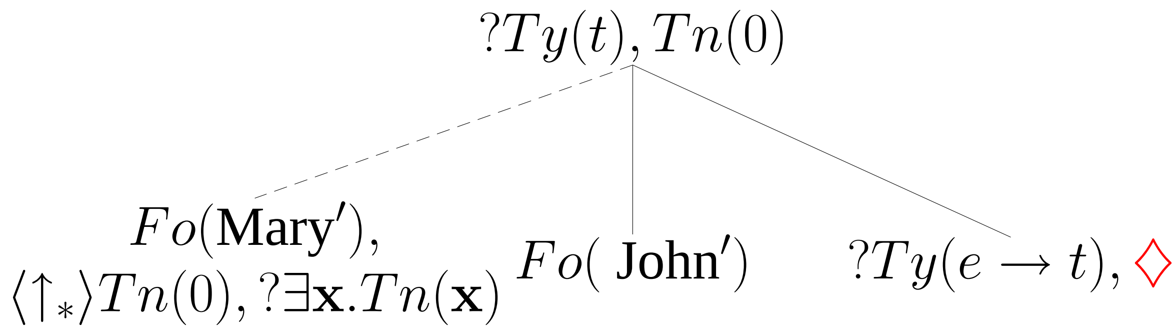
$$Tn(0), ?Ty(t), \diamondsuit$$

$$Fo(\text{ Mary}'), \langle \uparrow_* \rangle Tn(0), ?\exists \mathbf{x}.Tn(\mathbf{x})$$

The semantic function of $Fo(\text{Mary}')$ is underspecified.

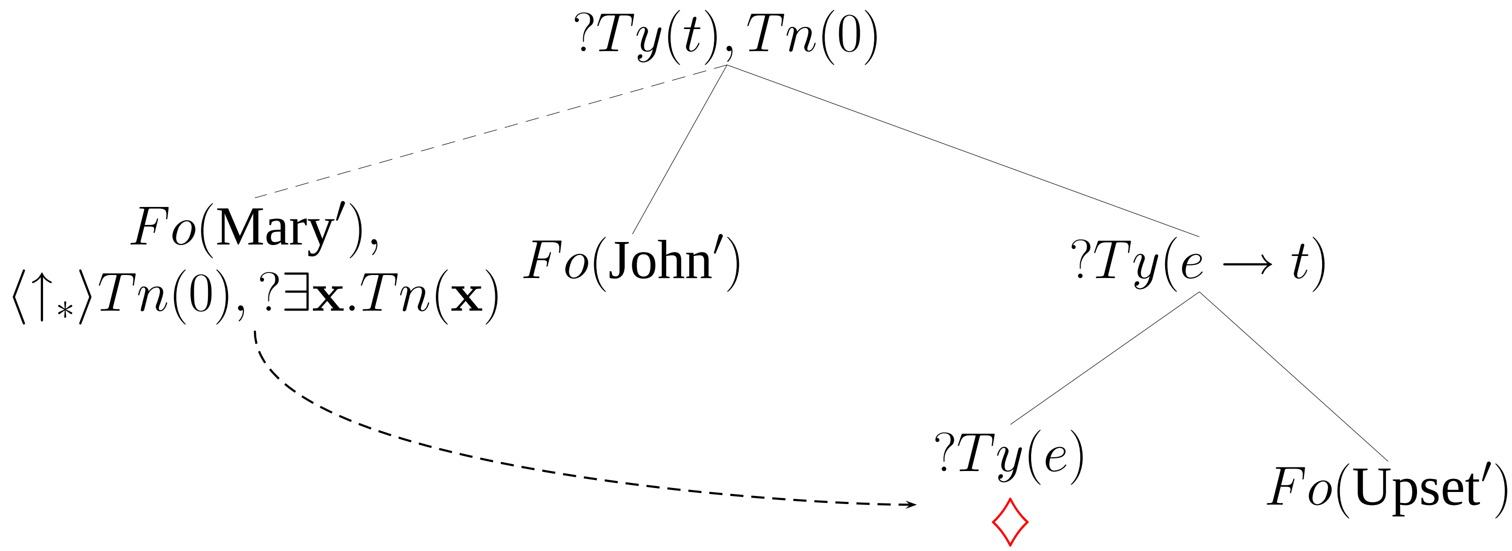
The Flow of Language Understanding – Parsing

Parsing ‘Mary, John upset’



The Flow of Language Understanding – Parsing

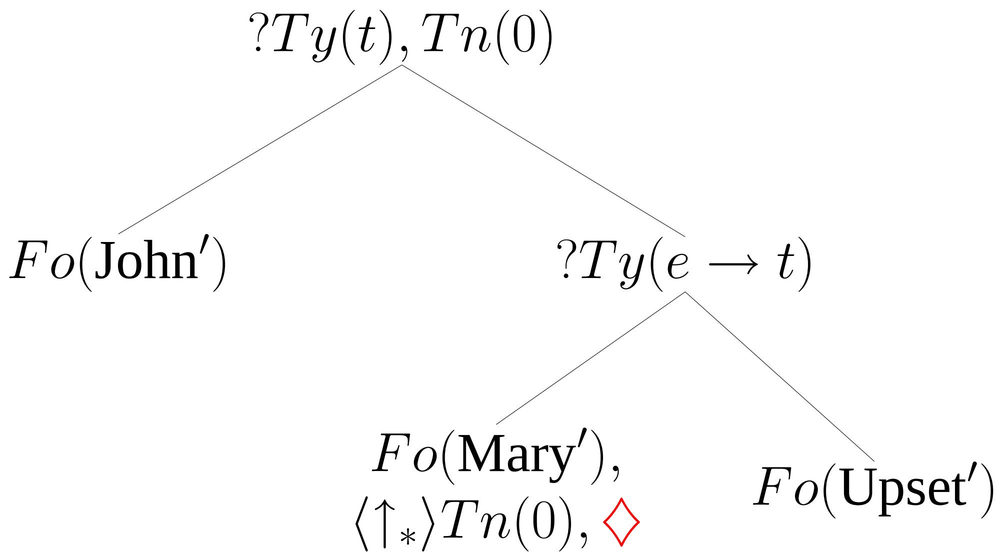
Parsing ‘Mary, John upset’



The position of the unfixed node is fixed through a process of unification.

The Flow of Language Understanding – Parsing

Parsing ‘Mary, John upset’



The output **tree** is identical to that produced by a parse of ‘John upset Mary’ and contains no trace of dislocated object.

But the set of **actions** (i.e. the syntax) used to construct the tree *does* carry this information.

The Flow of Language Understanding – Parsing

A PARSER STATE consists of a triple

$$\langle T, W, A \rangle$$

T a (possibly partial) propositional tree,

W a string of words so far parsed

A the set of actions (computational and lexical) used to construct T from W.

Initial parser state: $\langle \{?Ty(t), \Diamond\}, \emptyset, \emptyset \rangle$.

Final (acceptable) parser state: $\langle T_\phi, \phi, A_\phi \rangle$

where T_ϕ is a complete propositional tree derived from ϕ by A_ϕ.

The Flow of Language Understanding – Generation

A GENERATOR STATE G is a pair

$$(T_G, X)$$

of a GOAL TREE, T_G, representing the content of the utterance to be produced,

and a set X of pairs (S, P), where S is a candidate partial string and P is the associated PARSER STATE

(a set of $\langle T, W, A \rangle$ triples).

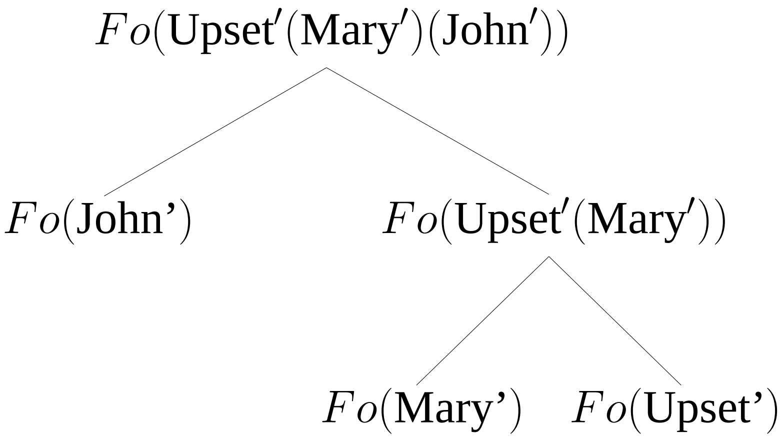
Generation is thus characterised in **exactly** the same terms as parsing except that the the current parse state is constrained by the requirement that the current partial tree subsumes the goal tree.

Initial generator state G_0 will (usually) be the pair $(T_G, \{(\emptyset, P_0)\})$.

The Flow of Language Understanding – Generation

Generating ‘John upset Mary’:

GOAL TREE:



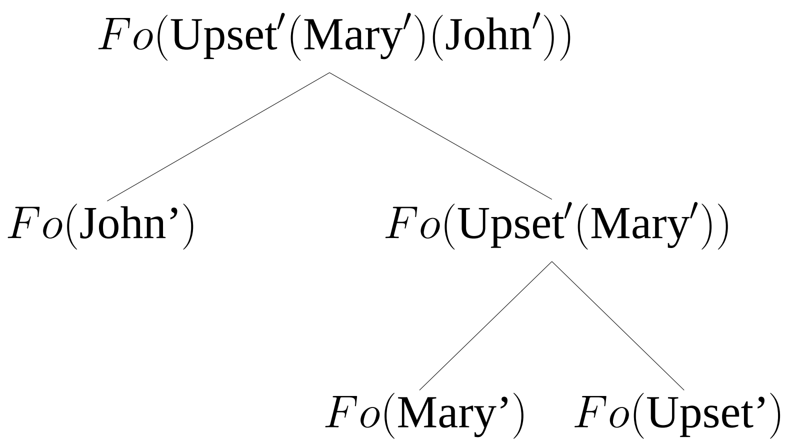
PARSER STATE

$\langle \{?Ty(t), \Diamond\}, \emptyset, \emptyset \rangle$

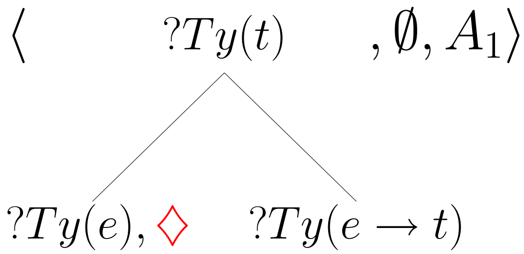
The Flow of Language Understanding – Generation

Generating 'John upset Mary':

GOAL TREE:



PARSER STATE



The Flow of Language Understanding – Generation

Generating ‘John upset Mary’:

GOAL TREE:

- $Fo(\text{Upset}'(\text{Mary}')(\text{John}'))$
  - $Fo(\text{John'})$
  - $Fo(\text{Upset}'(\text{Mary}'))$
    - $Fo(\text{Mary'})$
    - $Fo(\text{Upset'})$

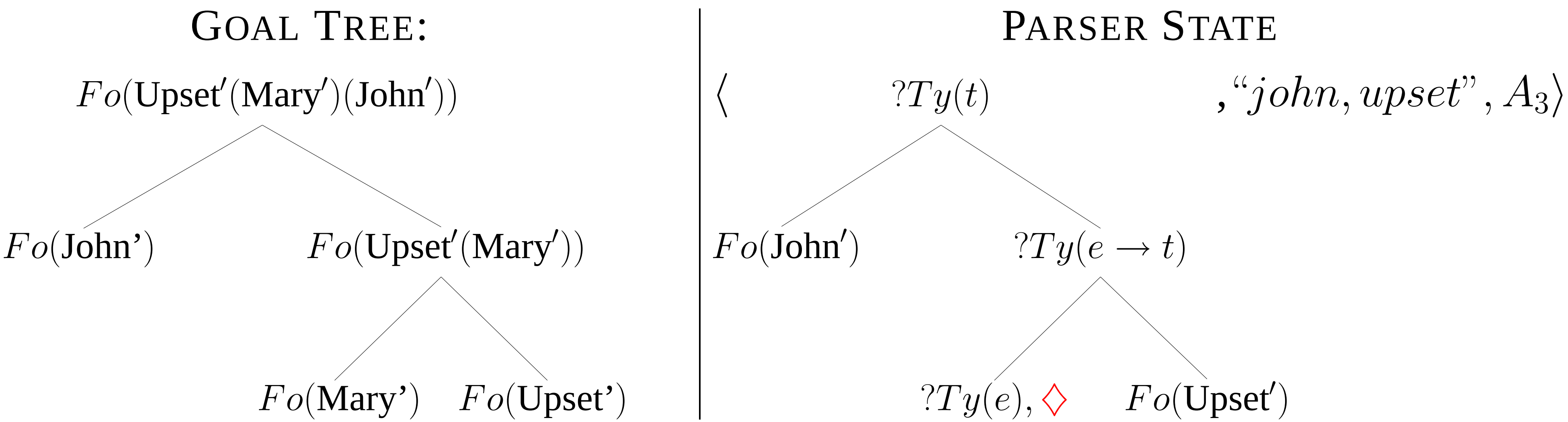
PARSER STATE

$\langle$ $?Ty(t)$, “$john$”, $A_2\rangle$

- $?Ty(t)$
  - $Fo(\text{John}')$
  - $?Ty(e \rightarrow t), \diamondsuit$

The Flow of Language Understanding – Generation

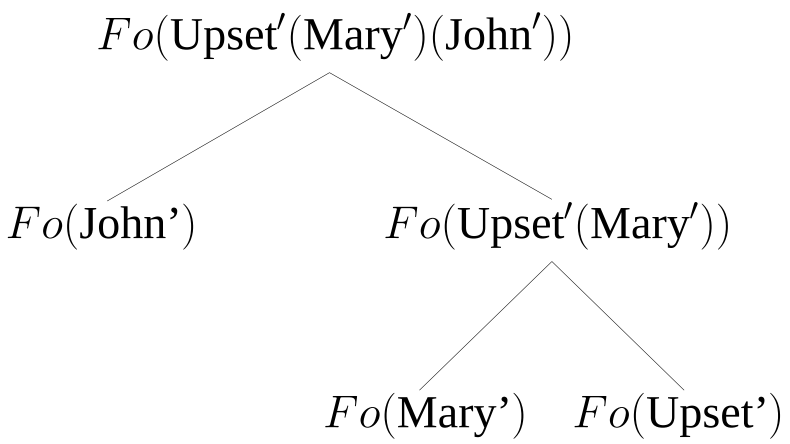
Generating ‘John upset Mary’:



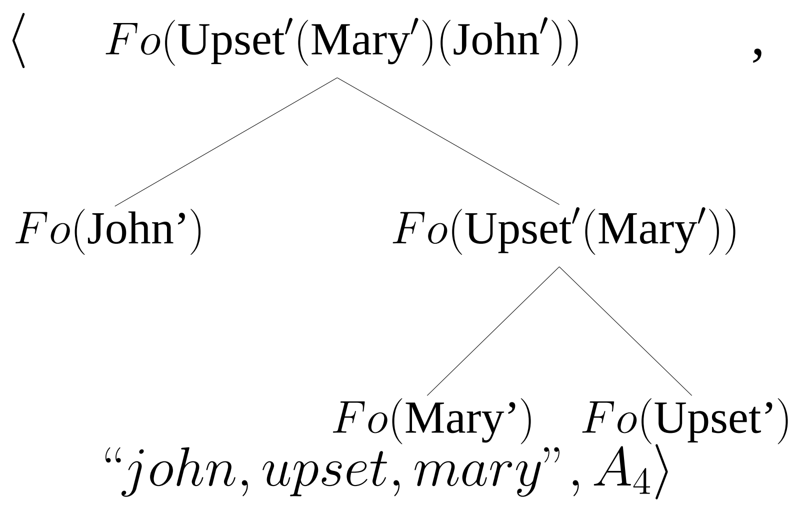
The Flow of Language Understanding – Generation

Generating ‘John upset Mary’:

GOAL TREE:



PARSER STATE



Context-dependence: anaphora

Pronouns project METAVARIABLES to be replaced by some selected term from context
through a pragmatic process of SUBSTITUTION,
constrained by conditions on ‘binding’, Relevance Theoretic principles AND any associated ‘presupposition’.

	IF	$?Ty(e)$	
	THEN	$put(Fo(\mathbf{U}_{FEMALE}),$	Metavariable plus ‘presupposition’
her		$Ty(e),$	Type
		$?\exists \mathbf{x}.Fo(\mathbf{x}))$	Formula requirement
	ELSE	ABORT	

9 **A**. Who upset Mary?
B. John upset her.

$\mathfrak{C} : \{Fo(\text{John}'), Fo(\text{Mary}')\}$ CONTEXT?
$Fo(\mathbf{U}_{FEMALE}) \Leftarrow Fo(\text{Mary}')$ SUBSTITUTION

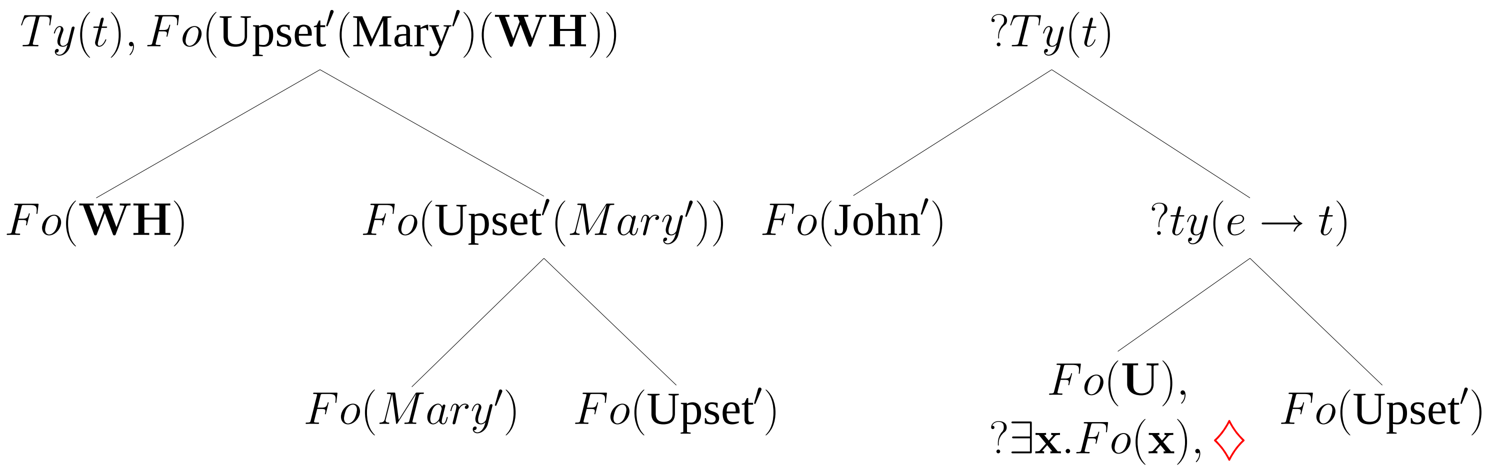
Context-dependence: relative clauses and anaphora

What is $\mathfrak{C}$?

Minimally the context contains the **tree** that provides the interpretation of the preceding utterance:

CONTEXT
Who upset Mary?

CURRENT TREE
John upset her.



WH a specialised metavariable

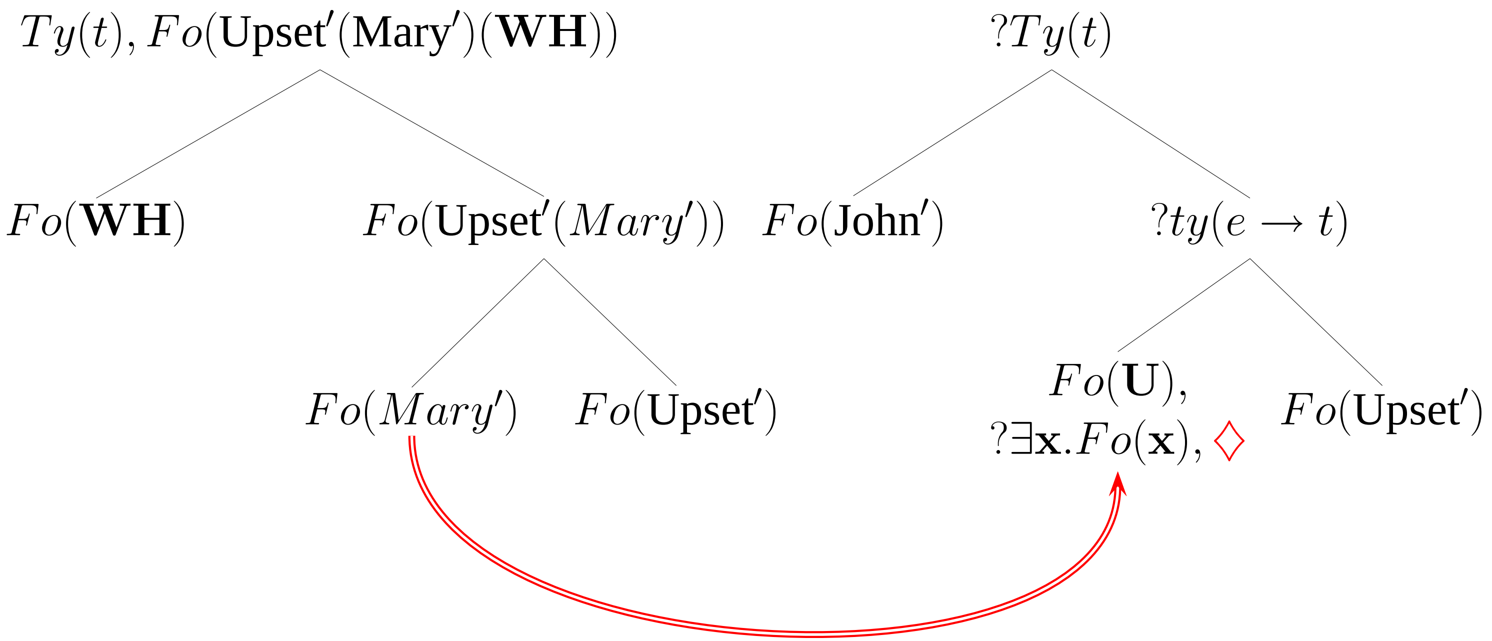
Context-dependence: relative clauses and anaphora

What is $\mathfrak{C}$?

Minimally the context contains the **tree** that provides the interpretation of the preceding utterance:

CONTEXT
Who upset Mary?

CURRENT TREE
John upset her.



SUBSTITUTION of term from context

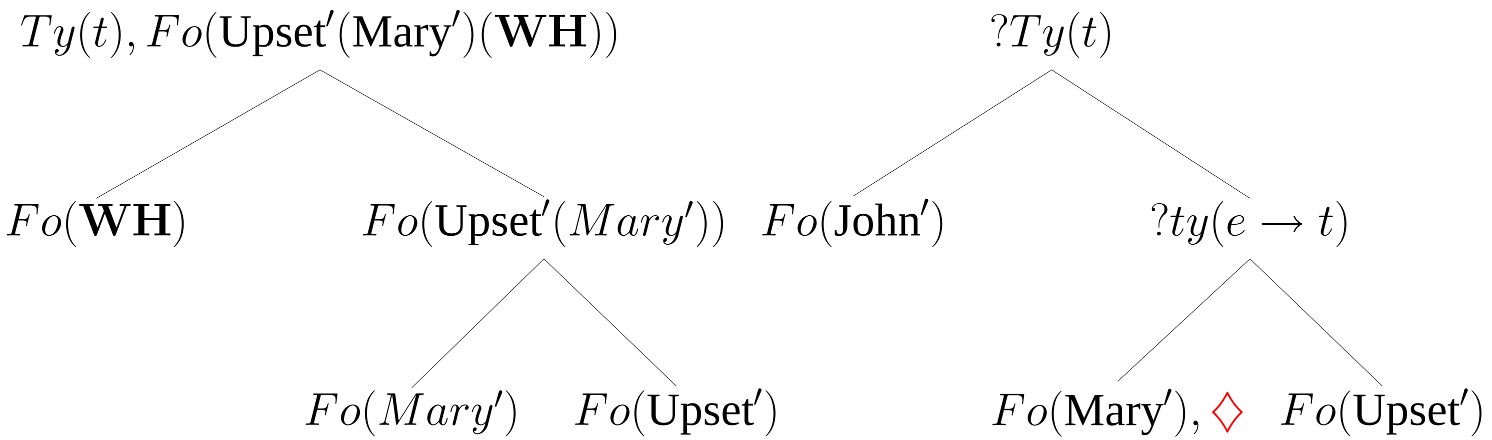
Context-dependence: relative clauses and anaphora

What is $\mathfrak{C}$?

Minimally the context contains the **tree** that provides the interpretation of the preceding utterance:

CONTEXT
Who upset Mary?

CURRENT TREE
John upset her.



SUBSTITUTION of term from context

Context-dependence: relative clauses and anaphora

Note that SUBSTITUTION MUST occur otherwise there remains an outstanding requirement ($?\exists \mathbf{x}.Fo(\mathbf{x})$) rendering the tree incomplete and the utterance ill-formed:

John upset her is **not** well-formed if there is no accessible antecedent for the pronoun.

Context for a particular (partial) tree T consists of:

(a) the triple $\langle T, W, A \rangle$ containing T in the current state;

(b) the ordered set of final parser states P_n (a set of triples) from the previous utterances.

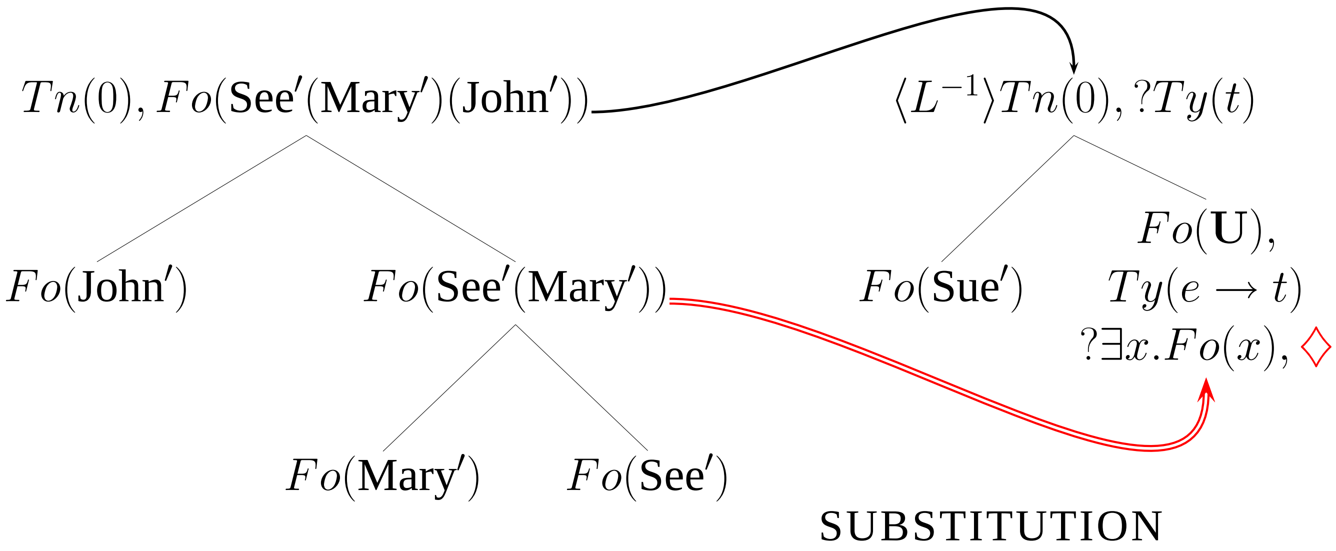
This definition applies BOTH to parsing and generation.

Context-dependence: ellipsis

Using **terms** taken from context (strict readings)

– *do* as a pro-(event-)verb projecting a predicate metavariable:

John saw Mary and Sue did, too.



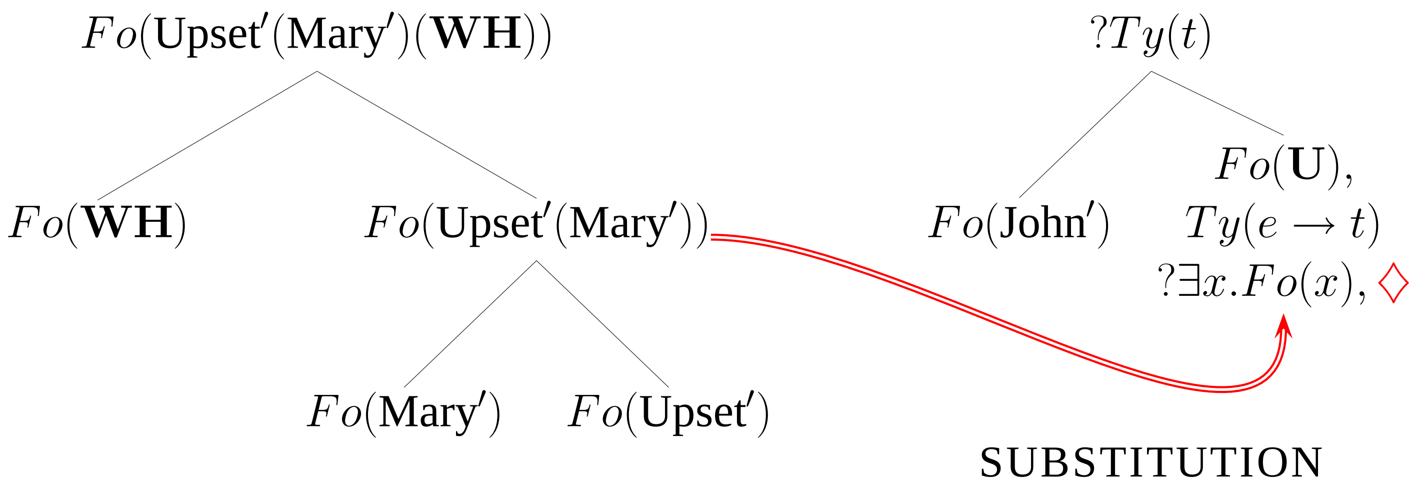
Output: $Fo(\text{See}'(\text{Mary}')(\text{John}') \wedge \text{See}'(\text{Mary}')(\text{Sue}'))$.

Context-dependence: ellipsis

The licensing tree need not be part of the interpretation of the current utterance:

CONTEXT — TREE UNDER CONSTRUCTION:

Q: Who upset Mary? — Ans: John did.



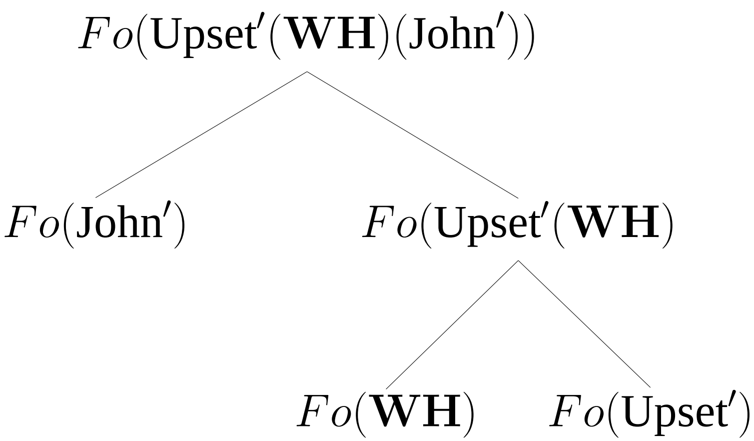
Output: $Fo(\text{Upset}'(\text{Mary}')(\text{John}'))$

Context-dependence: ellipsis

Using **structure** from context:

an interlocutor may use the structure provided by parsing another utterance to generate an answer.:

TREE AS CONTEXT:	**becomes**	TREE UNDER CONSTRUCTION:
Q: Who did John upset?		Ans: Himself.



himself

IF	$?Ty(e)$		
THEN	IF	$\langle\uparrow_0\rangle ?Ty(t)$	
	THEN	`Abort`	
	ELSE	IF	$\langle\uparrow_0\rangle\langle\uparrow_1^*\rangle\langle\downarrow_0\rangle Fo(\alpha)$
		THEN	$\texttt{put}(Ty(e), Fo(\alpha),)$
		ELSE	`Abort`
	ELSE	`Abort`	
ELSE	`Abort`		

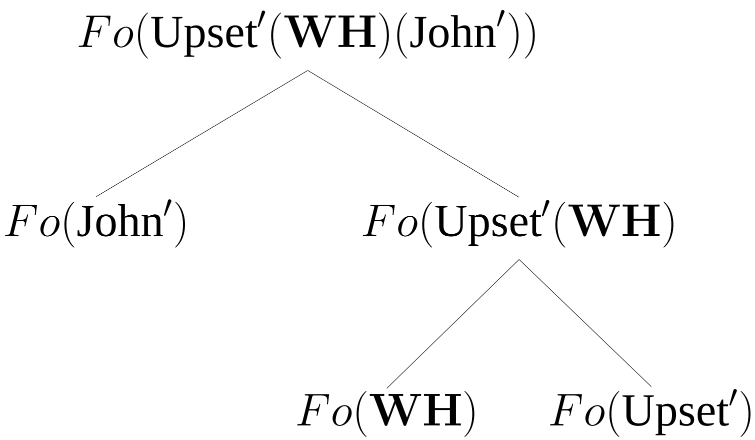
Context-dependence: ellipsis

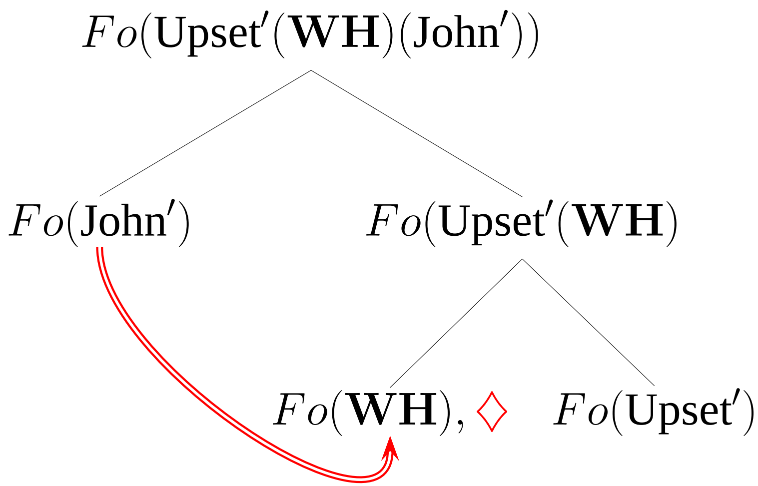
Using **structure** from context:

an interlocutor may use the structure provided by parsing another utterance to generate an answer.:

TREE AS CONTEXT: **becomes** TREE UNDER CONSTRUCTION:

Q: Who did John upset? Ans: Himself.





Output: $Fo(\text{Upset}'(\text{John}')(\text{John}'))$

Context-dependence: ellipsis

Such development of given tree provides straightforward analyses of scope relations and ambiguities:

10 (a) Q: Who did every student upset?
Ans: Their supervisor.

(b) Q: Who did every student upset?
Ans: A lecturer.

Context-dependence: ellipsis – using actions

Context includes not only **trees** but **actions**.

Re-running actions from context (as licensed by formula underspecification) provides a way of analysing sloppy readings.

11 Q: Who upset his mother?

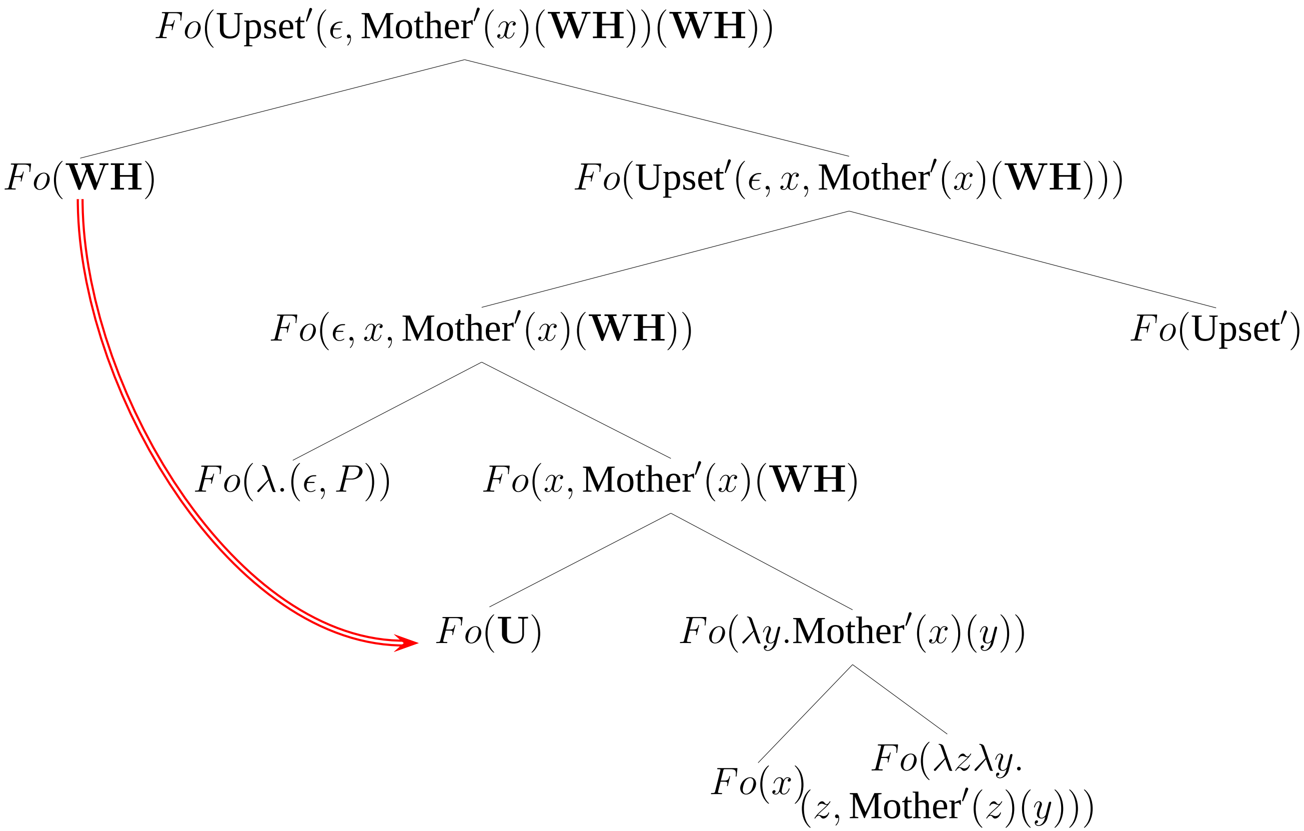
Ans: John did.

(John upset John's mother)

Context-dependence: ellipsis

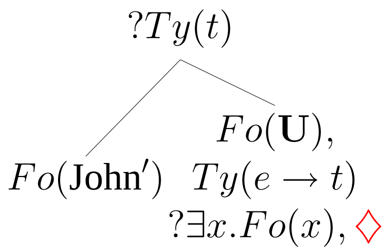
CONTEXT

Who upset his mother?



TREE UNDER CONSTRUCTION

John did.



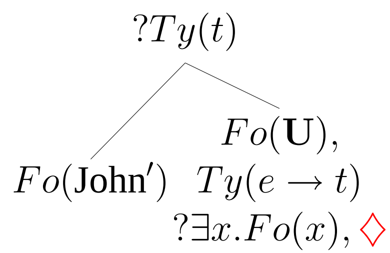
Context-dependence: ellipsis – using actions

CONTEXT	TREE UNDER CONSTRUCTION
Who upset his mother?	John did.

Actions: 'upset his mother'

IF $?Ty(e \rightarrow t)$

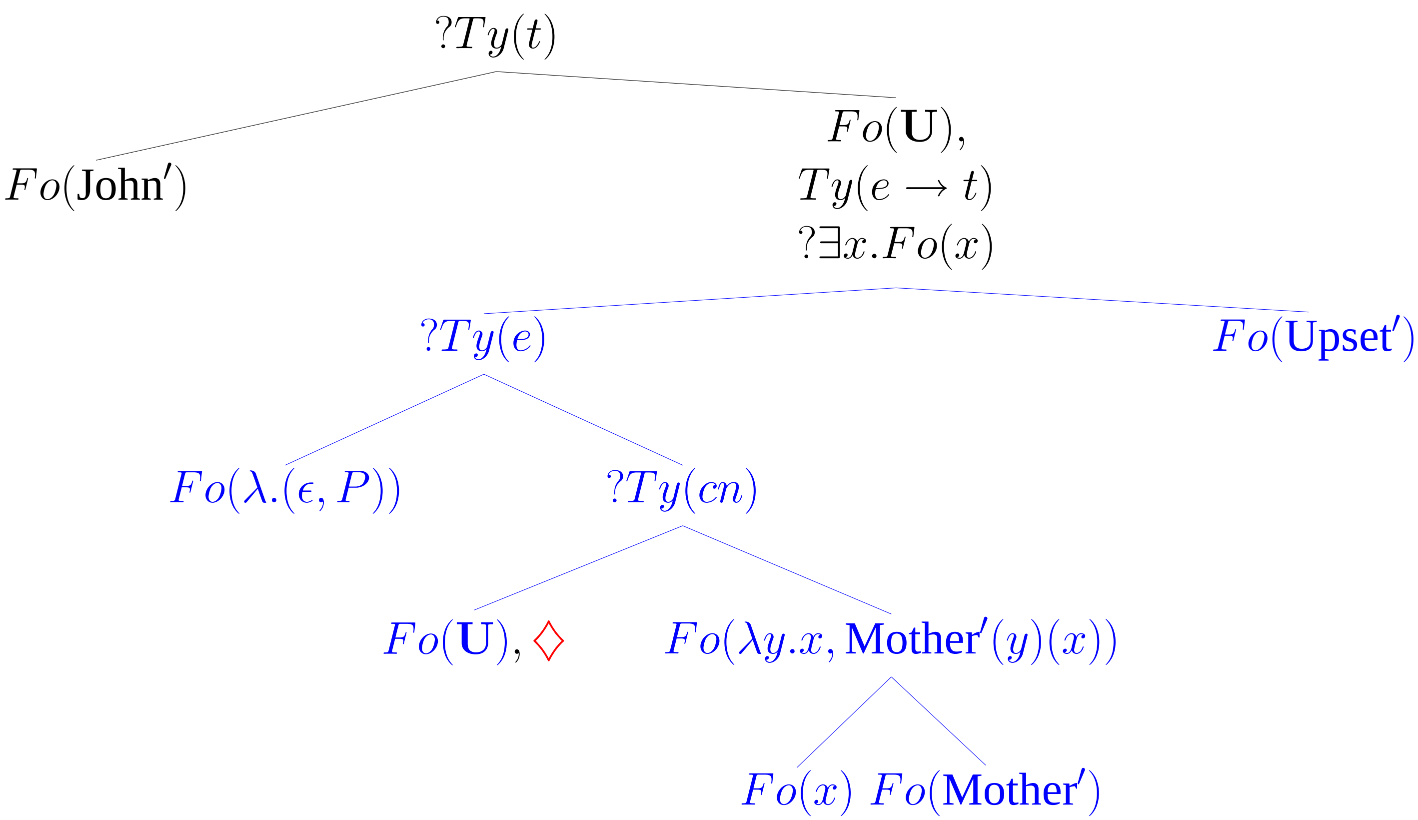
THEN make-go($\downarrow_1$); put(Fo(Upset$'$));
go($\uparrow_1$); make-go($\downarrow_0$);
put($?Ty(e)$); make-go($\downarrow_1$); put($\lambda P.\epsilon, P$);
go($\uparrow_1$); make-go($\downarrow_0\downarrow_0$);
put($Fo(\mathbf{U}, Ty(e)$); go($\uparrow_0$);
make-go($\downarrow_1\downarrow_0$); fresh-put($\mathbf{x}$);
go($\uparrow_0$); make-go($\downarrow_1$);
put($Fo(\text{Mother}'), Ty(e \rightarrow (e \rightarrow cn)))$)
go($\langle\uparrow_1\rangle$); put($Fo(\text{Mother}'(x))$);
go($\langle\uparrow_1\rangle$); go($\langle\downarrow_0\rangle$);
SUBSTITUTE($\mathbf{U}/Fo(\alpha)$, $\langle\uparrow_0\rangle\langle\uparrow_1\rangle\langle\uparrow_0\rangle\langle\uparrow_1\rangle\langle\downarrow_0\rangle Fo(\alpha)$)



Context-dependence: ellipsis – using actions

TREE UNDER CONSTRUCTION – AFTER ACTIONS

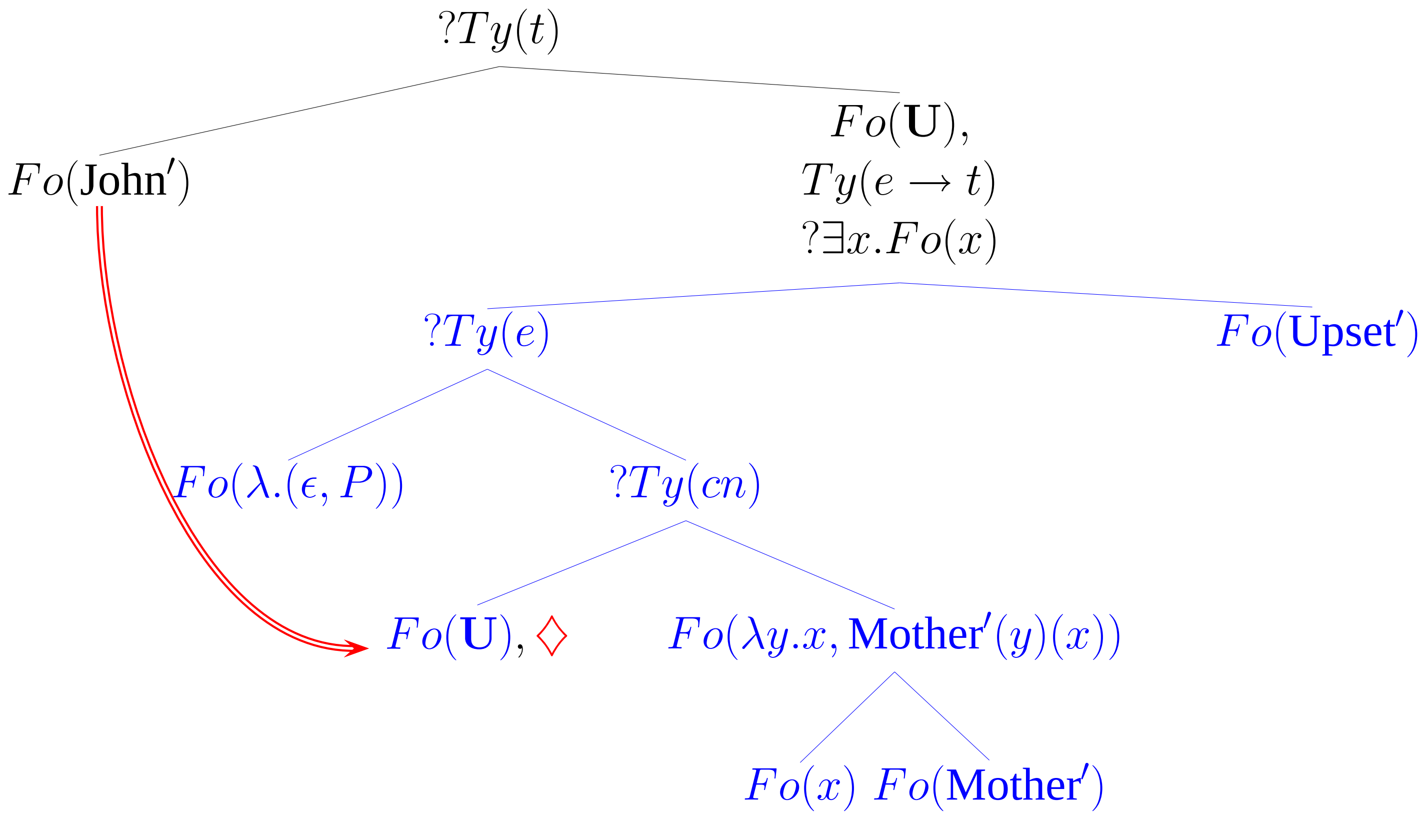
John did.



Context-dependence: ellipsis – using actions

TREE UNDER CONSTRUCTION – AFTER ACTIONS

John did.

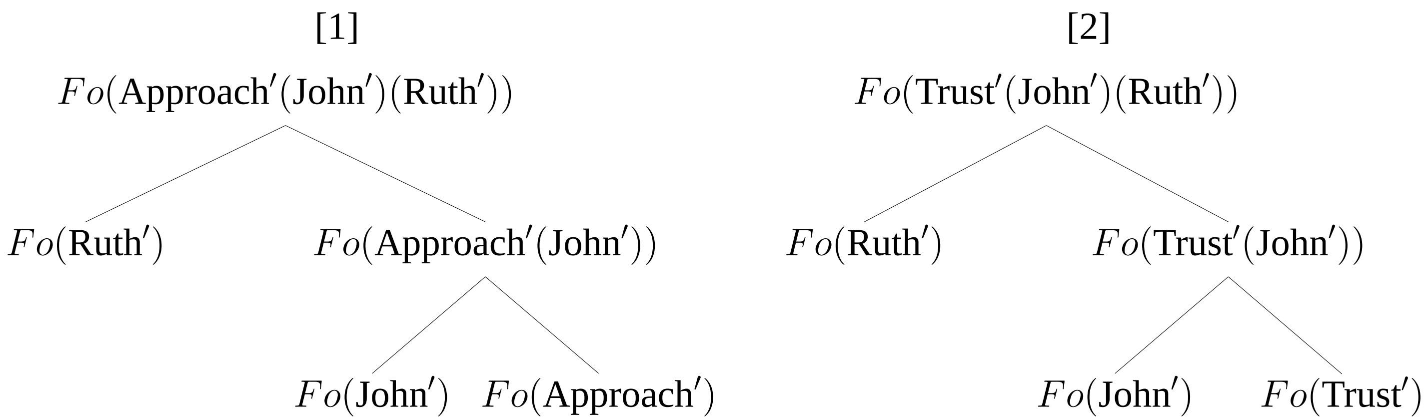


SUBSTITUTE$(\mathbf{U}/Fo(\alpha), \langle\uparrow_0\rangle\langle\uparrow_1\rangle\langle\uparrow_0\rangle\langle\uparrow_1\rangle\langle\downarrow_0\rangle Fo(\alpha))$

Context-dependence: ellipsis – using actions

12 [1] I'll approach John. [2] Him, I trust. [3] Tom, too.

CONTEXT:



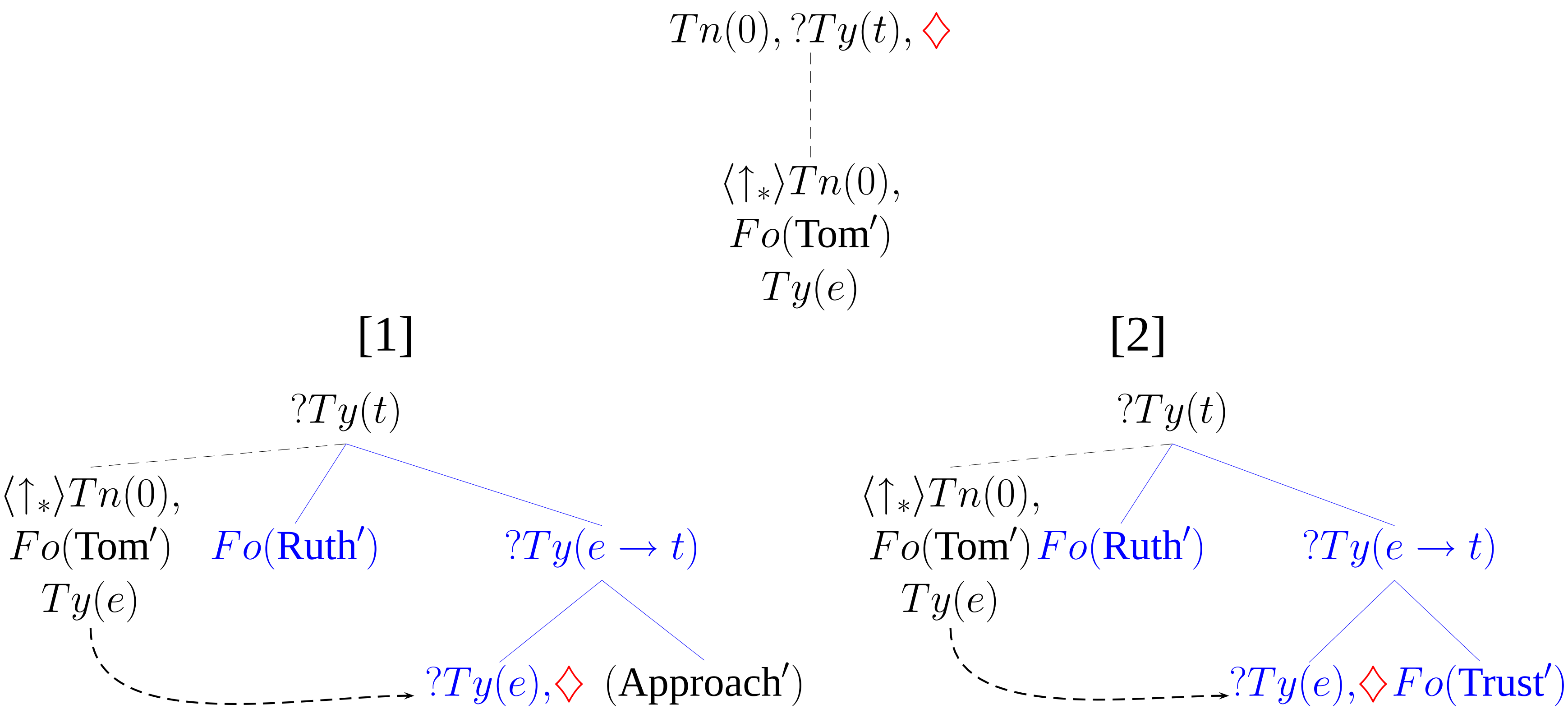
IF $?Ty(t)$
THEN $\texttt{make}(\downarrow_0); \texttt{go}(\downarrow_0);$
$\texttt{put}(Fo(\mathbf{U}'_{spkr}), Ty(e)); \texttt{go}(\uparrow_0);$
$\texttt{make}(\downarrow_1); \texttt{go}(\downarrow_1); \texttt{put}(?Ty(e \rightarrow t));$
$\texttt{make}(\downarrow_1); \texttt{go}(\downarrow_1);$
$\texttt{put}(fo(\text{Approach}'), Ty(e \rightarrow (e \rightarrow t)));$
$\texttt{go}(\uparrow_1); \texttt{make}(\downarrow_0);$
$\texttt{go}(\downarrow_0); \texttt{put}(?Ty(e))$

IF $?Ty(t)$
$\texttt{make}(\downarrow_0); \texttt{go}(\downarrow_0);$
$\texttt{put}(Fo(\mathbf{U}'_{spkr}), Ty(e)); \texttt{go}(\uparrow_0);$
$\texttt{make}(\downarrow_1); \texttt{go}(\downarrow_1);$
$\texttt{put}(?Ty(e \rightarrow t)); \texttt{make}(\downarrow_1); \texttt{go}(\downarrow_1);$
$\texttt{put}(Fo(\text{Trust}'), Ty(e \rightarrow (e \rightarrow t)));$
$\texttt{go}(\uparrow_1); \texttt{make}(\downarrow_0); \texttt{go}(\downarrow_0);$
$\texttt{put}(?Ty(e));$ MERGE

Context-dependence: ellipsis – using actions

12 [1] I'll approach John. [2] Him, I trust. [3] Tom, too.

TREE UNDER CONSTRUCTION[3]:



Context-dependence: ellipsis

13 The SUBSTITUTION process does not respect islands:

The man who arrested John failed to read him his rights.
So did the man who arrested Tom.
(= the man who arrested Tom$_i$ failed to read Tom$_i$ Tom$_i$'s rights)

14 Use of **actions** as opposed to **trees** allows parallelism of separate binding:

Mary submitted a proposal. So did Bill.

different proposals

$$\text{Actions}\left|\begin{array}{ll} \text{IF} & ?Ty(e \rightarrow t) \\ \text{THEN} & make-go(\langle\downarrow_1\rangle); \mathtt{put}(Ty(e \rightarrow (e \rightarrow t)), Fo(\text{Submit}'),); \\ & \mathtt{go}(\langle\uparrow_1\rangle); make-go(\langle\downarrow_0\rangle); \mathtt{put}(?Ty(e)); \\ & make-go(\langle\downarrow_1\rangle); \mathtt{put}(Ty(cn \rightarrow e), Fo(\lambda P.\epsilon, P),); \\ & \mathtt{go}(\langle\uparrow_1\rangle); make-go(\langle\downarrow_0\rangle); \mathtt{put}(?Ty(cn)); \\ & make-go(\langle\downarrow_1\rangle); \mathtt{put}(Ty(e \rightarrow cn), Fo(\text{Proposal}'),); \\ & \mathtt{go}(\langle\uparrow_1\rangle); \mathtt{put}(Ty(e)); \mathtt{freshput}(x) \end{array}\right.$$

Defining well-formedness

An utterance of a string ϕ in language L with respect to a context $\mathcal{C}$ is well-formed iff:

$$\mathcal{C} \cup P_0 \quad \overrightarrow{\phi_{\mathcal{L},\mathcal{K},\mathcal{P}}} \quad \{\ldots \langle T_\phi, \phi, A_\phi \rangle \ldots\}$$

where $\mathcal{C}$ is the prior context (a sequence of parser states);

$P_0 = \{\langle T_0, \emptyset, \emptyset \rangle\}$ is the standard initial state;

$\overrightarrow{\phi_{\mathcal{L},\mathcal{K},\mathcal{P}}}$ is the state transition licensed by the lexical ($\mathcal{L}$), computational ($\mathcal{K}$) and pragmatic ($\mathcal{P}$) actions (A_ϕ) used in parsing ϕ;

and T_ϕ is complete.

Felicitous utterance – proper DRS.

Defining well-formedness

A string ϕ is **fully grammatical** iff an utterance of ϕ is well-formed in all contexts:

$$\forall \mathcal{C}[\mathcal{C} \cup P_0 \quad \overrightarrow{\phi_{\mathcal{L},\mathcal{K},\mathcal{P}}} \quad \{\ldots, \langle T_\phi, \phi, A_\phi \rangle, \ldots\}]$$

(Equivalently) A string ϕ is **fully grammatical** iff an utterance of ϕ is well-formed in the null context:

$$\emptyset \cup P_0 \quad \overrightarrow{\phi_{\mathcal{L},\mathcal{K},\mathcal{P}}} \quad \{\ldots, \langle T_\phi, \phi, A_\phi \rangle, \ldots\}$$

(a) No man is mortal.

(b) A woman likes mustard though it makes her hot.

(c) If John is a teacher, he will have a degree.

(d) As for John, he is sick.

Defining well-formedness

A string ϕ is **fully ungrammatical** iff there is no context in which an utterance of ϕ is well-formed:

$$\neg\exists\mathcal{C}[\mathcal{C}\cup P_0 \quad \overrightarrow{\phi_{\mathcal{L},\mathcal{K},\mathcal{P}}} \quad \{\ldots,\langle T_\phi,\phi,A_\phi\rangle,\ldots\}]$$

(a) *The a in came.
(b) *Word every no salad sleeps snores.
(c) *Which man did you interview the man from London?
(d) *The man from London emerged that he is sick.
(e) *The man John saw whom is outside.
(f) *Who did you see the man who came in with?

Defining well-formedness

A string ϕ is well-formed iff an utterance of ϕ is well-formed in *some tree-complete* context:

$$\exists \mathcal{C}[\text{complete}(\mathcal{C}) \wedge \mathcal{C} \cup P_0 \quad \underset{\phi_{\mathcal{L},\mathcal{K},\mathcal{P}}}{\longrightarrow} \quad \{\ldots, \langle T_\phi, \phi, A_\phi \rangle, \ldots\}]$$

(a) He upset her.

(b) John did, too.

(c) John.

While liberal with respect to some data, the definition remains strict with respect to strings that cannot lead to well-formed complete proposition outputs:

(a) Have you read?

(b) Where are?

The *grammar* excludes only categorically unacceptable strings.

Gradient responses are context dependent.

So, the grammar defines satisfiability with respect to context.

Summary

- Point of departure:

  Fragments require a structural concept of context for interpretation.

- Background:

  With DS commitment to articulating concepts of
  structural underspecification and update (parsing and generation)
  defining a concept of context is essential to defining wellformedness

- Result 1:

  Context as representations of content and actions of building them provides a unitary basis for explaining ellipsis.

- Result 2:

  More fine-grained concepts of wellformedness.

- Conclusion:

  Characterising context dependence and the dynamics of its update is central to NL syntax