# Grammar Formalisms and Explanations of Dialogue

Ronnie Cann, Ruth Kempson, and Matthew Purver

Edinburgh, King’s College London, CSLI

Acknowledgements: Eleni Gregoromichelaki, Jieun Kiaer, Wilfried Meyer-Viol, Masayuki Otsuka,

http://semantics.phil.kcl.ac.uk/ldsnl

Cann, R., Kempson,R., and Marten, L., 2005 *The Dynamics of Language* Elsevier.

January 30, 2006

# The Dialogue Challenge

- The Pickering and Garrod challenge of evaluating (grammar) formalisms by how well they correspond to dialogue phenomena
- Meeting the challenge of extending the remit of NL grammars without writing discourse grammars ?
- Dynamic Syntax as a basis for dialogue modelling:
  - Syntax as the architecture for real-time parsing in context Underspecification-plus-enrichment in syntax
  - Context-dependent generation using the parsing architecture
- Defining context-dependent concepts of wellformedness, hence defining what it means to be a wellformed fragment, BUT
- retaining a concept of a natural-language grammar.

# The Dialogue Challenge

- context-dependence:

  — ellipsis/pronoun/parallelism effects:

  (i) Mary: What did you give Eliot?
  Ruth: Some lego.
  Mary: I gave him some playdoh.
  Bill: So did I.

  —– speaker/hearer exchange of roles:

  (ii) Ruth: Who does everyone admire?
  Hugh: Their mother.

  (iii) Ruth: What did Alex design for
  Hugh: Eliot? A kaleidoscope.

  (iv) Ruth: What did she
  Hugh: design for herself? A self-loading washing-machine.

  (v) Ruth: Have you spoken to...
  Hugh: any of the doctors? Not yet.

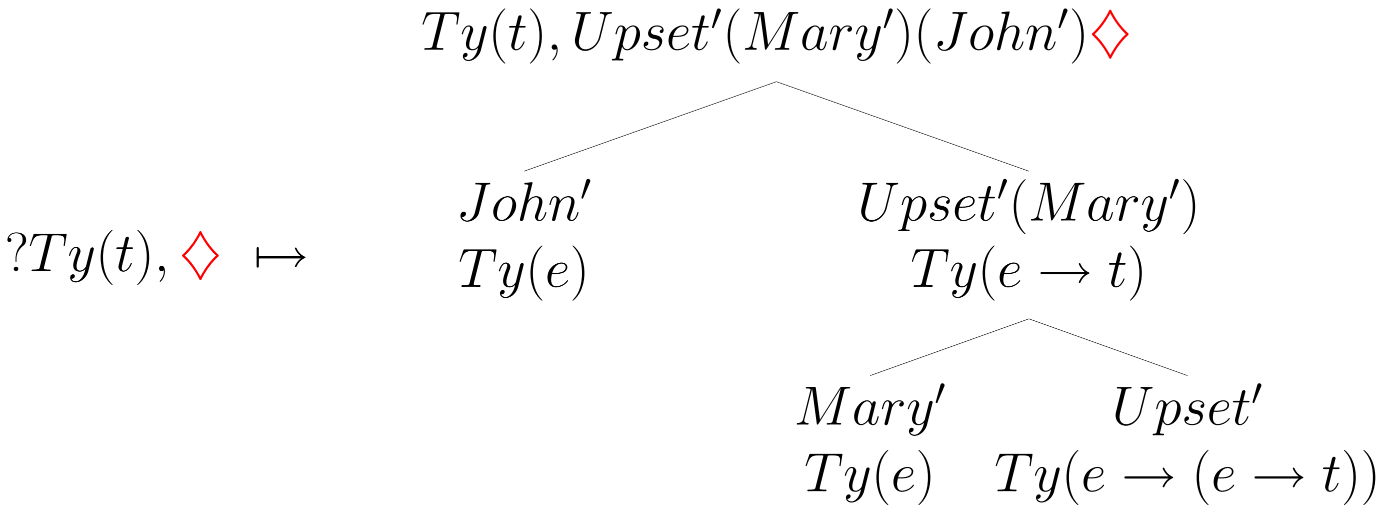
  (vi) Ruth: Old McDonald had a farm... And on that farm he had a
  Eliot: cow.

- Pragmatics, or syntax?

# The Flow of Language Understanding

- Building representations of content as monotonic tree growth process defined across the left-right sequence of words.

  Parsing *John upset Mary*:



$?Ty(t), \Diamond \quad \mapsto$

$Ty(t), Upset'(Mary')(John')\Diamond$

$John'$ $Ty(e)$

$Upset'(Mary')$ $Ty(e \rightarrow t)$

$Mary'$ $Ty(e)$

$Upset'$ $Ty(e \rightarrow (e \rightarrow t))$

The epsilon calculus (with lambda terms) as the basis for formula decorations:

$Upset'$ abbreviates $\lambda x \lambda y[Upset'(x)(y)]$
$John'$ abbreviates $(\iota, x, John'(x))$

Using structural and formula under-specification in combination

# The Flow of Language Understanding – Parsing

**LOFT (Logic of Finite Trees)** (Blackburn and Meyer-Viol 1994)

$\langle\downarrow_0\rangle X$ X holds at argument daughter of some tree-node n (Tn(n)).
$\langle\downarrow_1\rangle X$ X holds at functor daughter of Tn(n).
$\langle\uparrow\rangle X$ X holds at mother of Tn(n).
$\langle\downarrow_*\rangle X$ Tn(n) dominates X.
$\langle\uparrow_*\rangle X$ Tn(n) is dominated by X.
$\langle L\rangle X$ the LINK relation (between nodes in distinct trees)
$\langle L^{-1}\rangle X$ the inverse LINK relation.
$\langle D\rangle$ the reflexive transitive closure of $\langle\downarrow\rangle$, $\langle L\rangle$ relations.

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

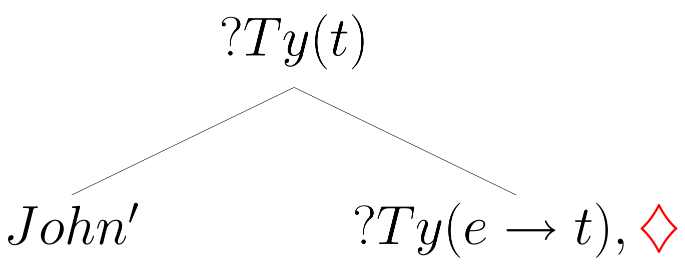
e.g. accusative case $?\langle\uparrow_0\rangle Ty(e \rightarrow t)$ - at output, current node must be dominated by a predicate node.

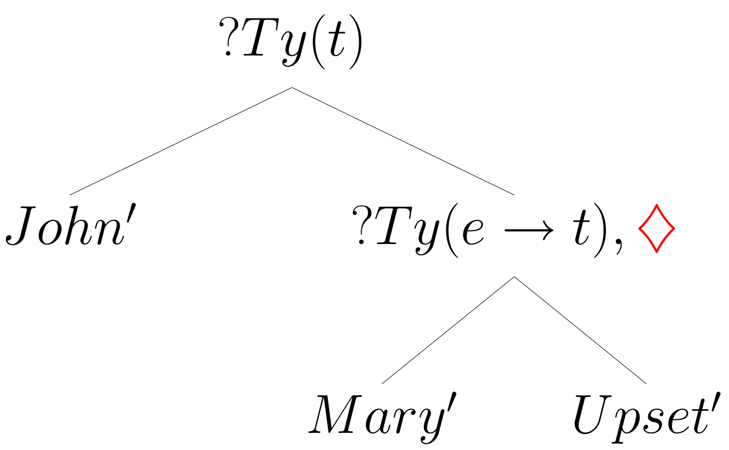
**Grammaticality**: For every wellformed string at least one complete logical form can be constructed from the words in sequence, with no requirements outstanding.

# Structural underspecification and dynamically resolving it (1)

- Computational and lexical actions progressively introduce structure to be inhabited by formulae.
- The notion of "requirements" to be filled later is central

(i) Parsing *John* in *John upset Mary* (ii) Parsing *John upset Mary*



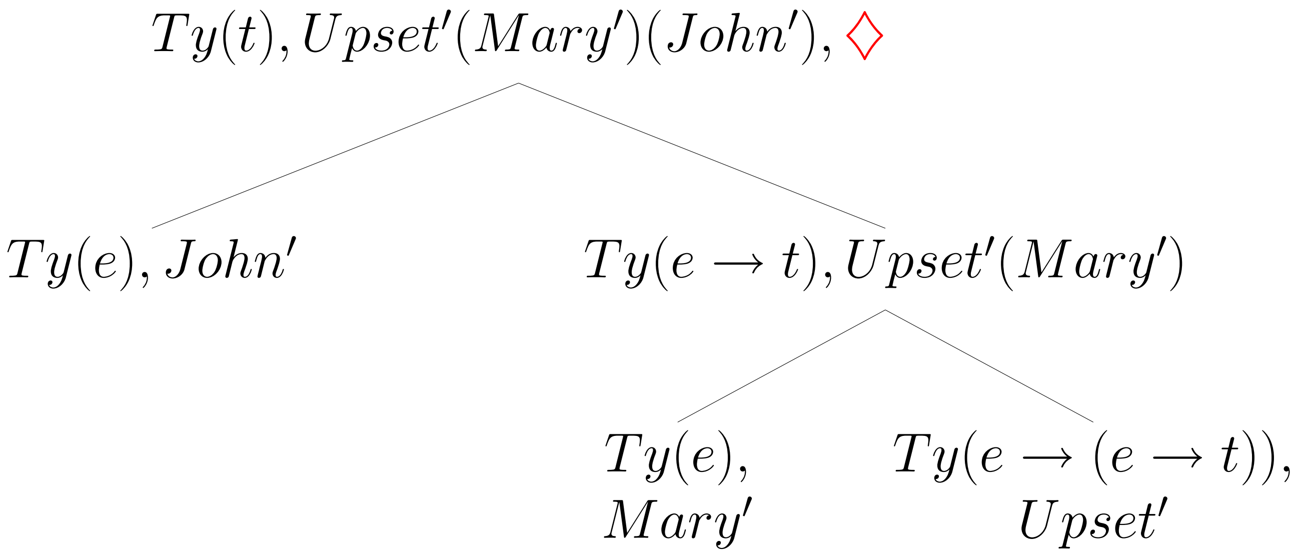


```
upset   IF     $?Ty(e \rightarrow t)$
        THEN   make($\langle\downarrow_1\rangle$);go($\langle\downarrow_1\rangle$);
               put($Fo(Upset')$,
               $Ty(e \rightarrow (e \rightarrow t))$
               go($\langle\uparrow_1\rangle$); make ($\langle\downarrow_0\rangle$); go($\langle\downarrow_0\rangle$);
               put($?Ty(e)$)
        ELSE   ABORT
```

Words may introduce nodes and decorate them.

# Compositionality reflected in the evaluation of non-terminal nodes (1)

Parsing ‘John upset Mary’



$Ty(t), Upset'(Mary')(John'), \diamondsuit$

$Ty(e), John'$

$Ty(e \rightarrow t), Upset'(Mary')$

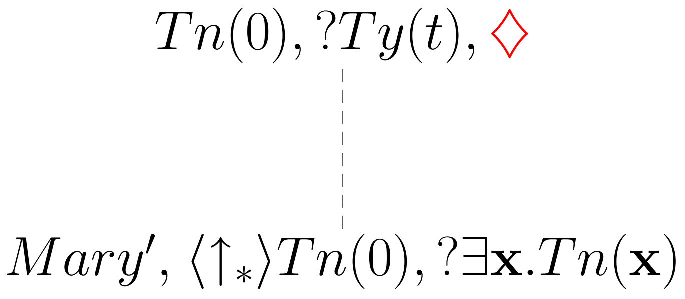
$Ty(e),$ $Mary'$

$Ty(e \rightarrow (e \rightarrow t)),$ $Upset'$

Parses are completed by applying Functional Application over types. Step-by-step compositionality defined on the resulting tree

# Structural underspecification and updating it (2)

Parsing ‘Mary, John upset’



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

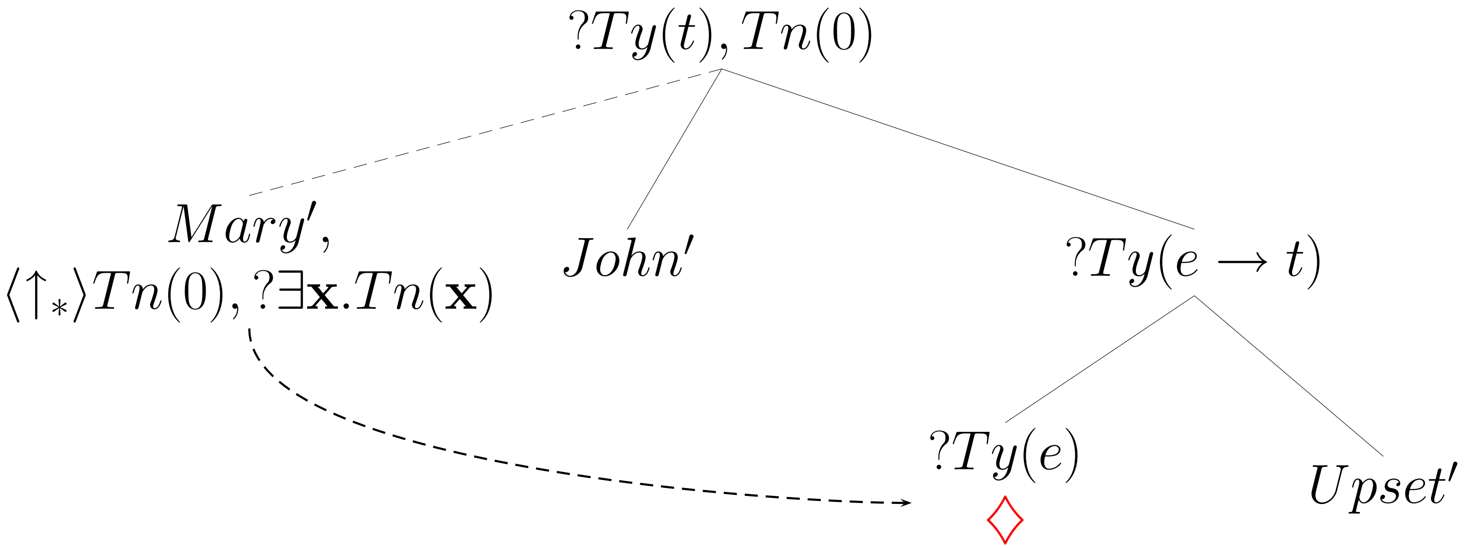
$Mary', \langle\uparrow_*\rangle Tn(0), ?\exists\mathbf{x}.Tn(\mathbf{x})$

The contribution of $Mary'$ to the resulting formula is underspecified.

(cp. *functional uncertainty* of LFG)

# Structural underspecification and updating it (2)

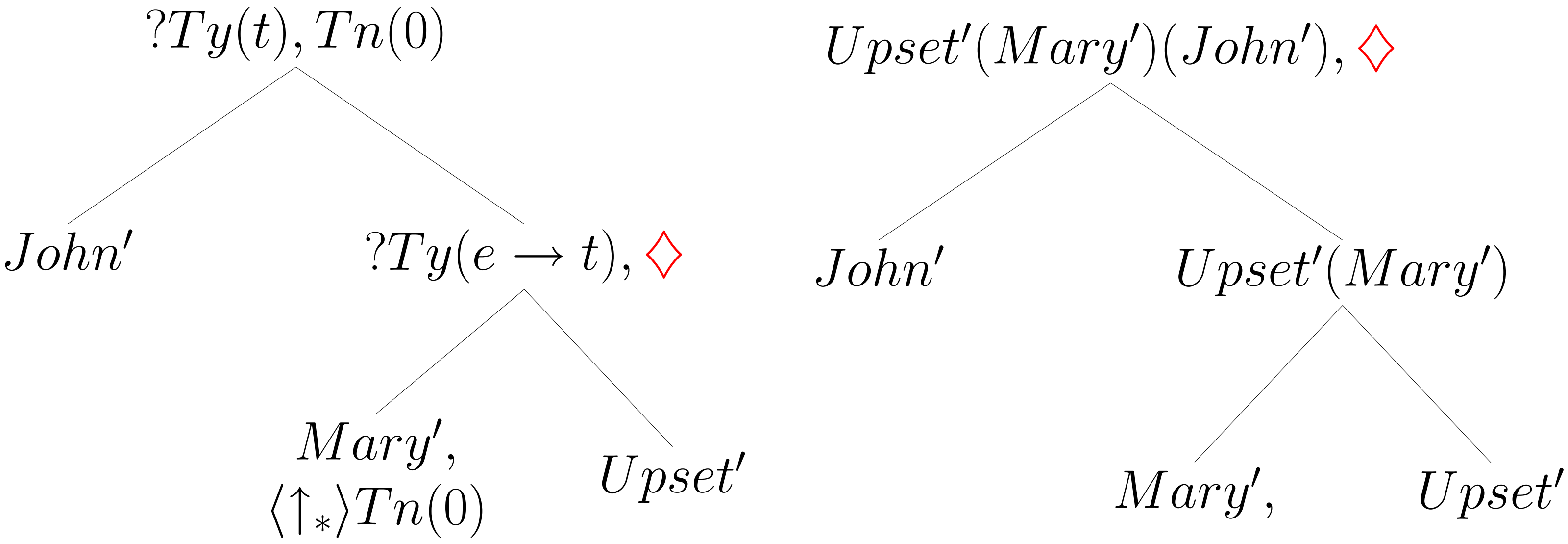
Parsing ‘Mary, John upset’



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

# Structural underspecification and updating it (2)

Parsing ‘Mary, John upset’



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

$John'$

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

$Mary',$
$\langle\uparrow_*\rangle Tn(0)$

$Upset'$

$Upset'(Mary')(John'), \diamondsuit$

$John'$

$Upset'(Mary')$

$Mary',$

$Upset'$

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

At any one transition in the construction process, there is a triple $\mathfrak{T}_W$:

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

$T$ a (possibly partial) propositional tree,

$W$ a string of words so far parsed

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

A PARSE STATE consists of a set of such triples

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

Final (acceptable) parse state: a non-empty set of triples of the form:

$\langle T_\phi, \phi, A_\phi\rangle$

$T_\phi$ a complete propositional tree derived from $\phi$ by $A_\phi$
(with no requirements outstanding).

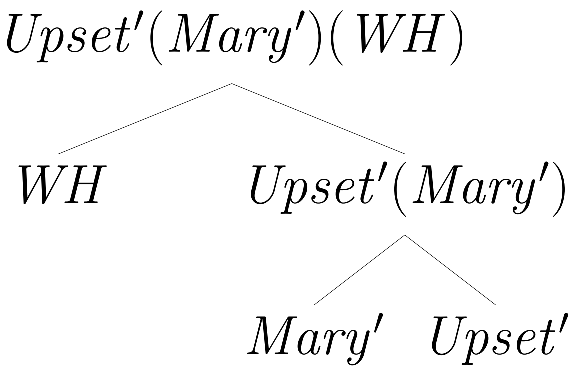
# Context-dependence: pronouns as place-holders

- pronouns: interpreted as META-VARIABLES (**U**), substituted by some term from context as part of construction process.

(1) Q: Who upset Mary? Ans: John upset her.

TREE AS CONTEXT: TREE UNDER CONSTRUCTION:

(i)

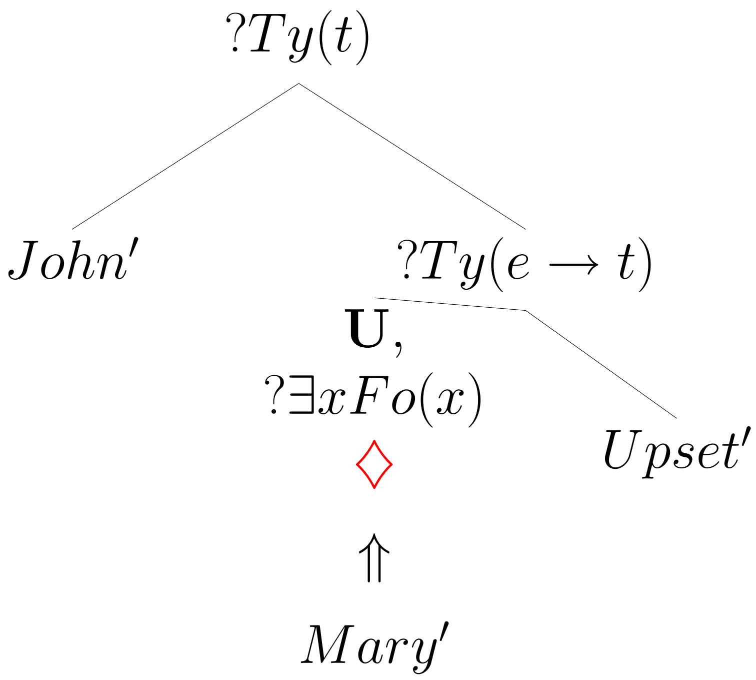


$Upset'(Mary')(WH)$

$WH$ $Upset'(Mary')$

$Mary'$ $Upset'$

(ii)



$?Ty(t)$

$John'$ $?Ty(e \rightarrow t)$

$\mathbf{U}$, $?\exists x Fo(x)$

$\diamondsuit$

$\Uparrow$

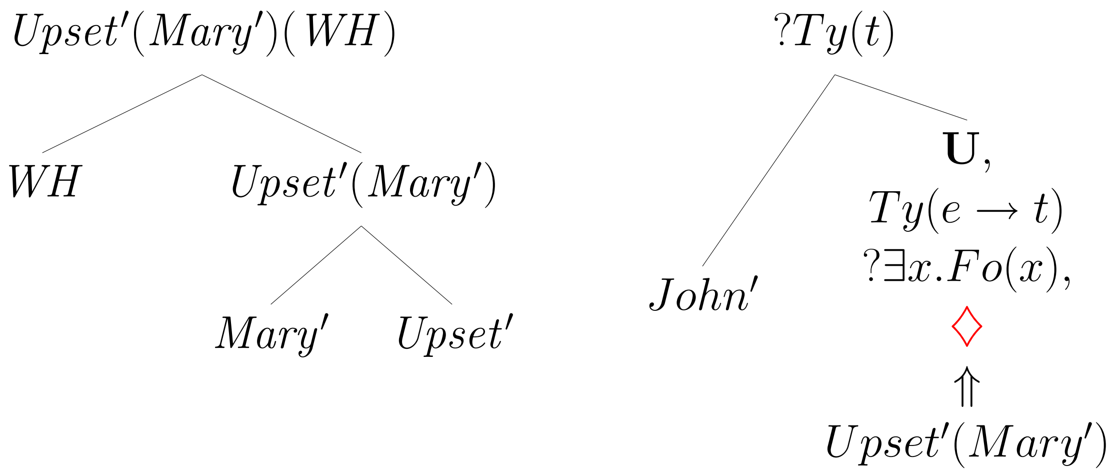
$Mary'$

$Upset'$

# Ellipsis as evidence of context: (a) - re-use of terms

- Using **terms** from context – strict reading:

  (2) Q: Who upset Mary? Ans: John did.



CONTEXT

$Upset'(Mary')(WH)$

$WH$ $Upset'(Mary')$

$Mary'$ $Upset'$

TREE UNDER CONSTRUCTION:

$?Ty(t)$

$John'$

$\mathbf{U}$, $Ty(e \rightarrow t)$ $?\exists x.Fo(x)$, $\diamondsuit$

$\Uparrow$

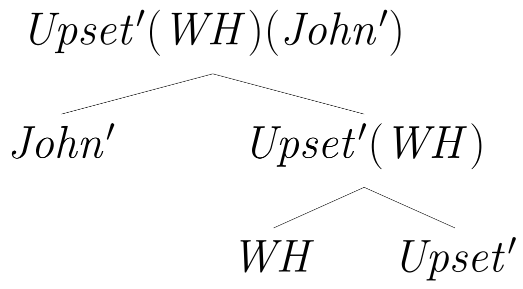
$Upset'(Mary')$

# Context-dependence: ellipsis (b)

- re-use of **structure** : parser starts from **partial** tree

(3) Q: Who did John upset? Ans: Himself.

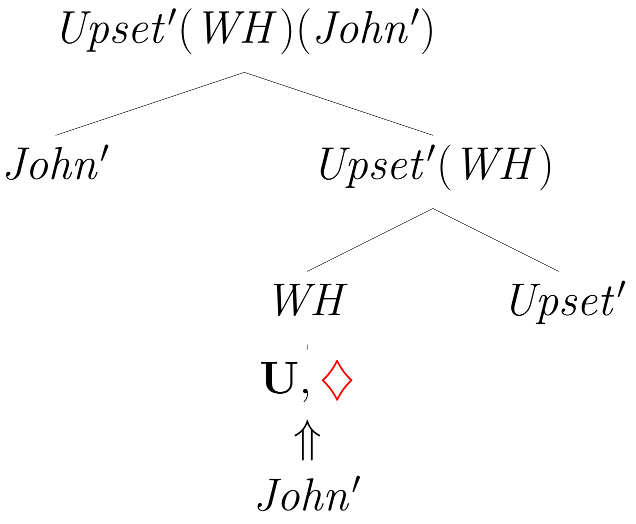
TREE AS CONTEXT: **becomes** TREE UNDER CONSTRUCTION:



$Upset'(WH)(John')$

$John'$ $Upset'(WH)$

$WH$ $Upset'$



$Upset'(WH)(John')$

$John'$ $Upset'(WH)$

$WH$ $Upset'$

**U**, $\diamondsuit$

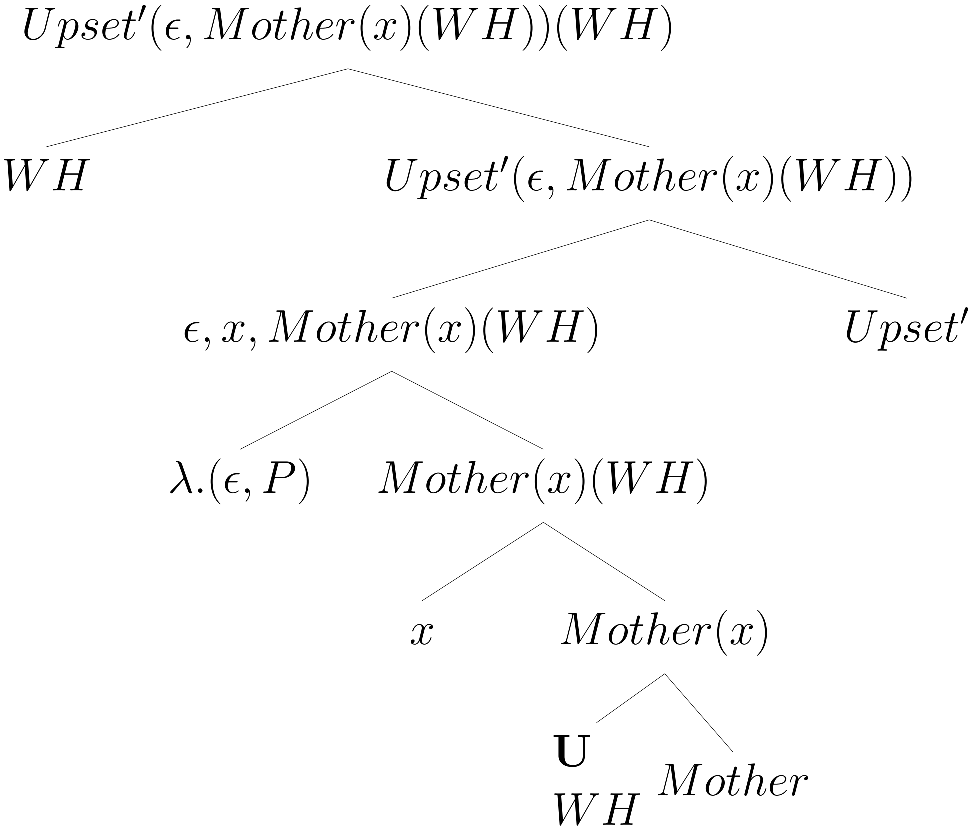
$\Uparrow$

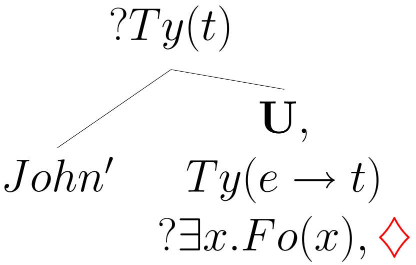
$John'$

(4) Who did everyone ignore? Their husband.

# Context-dependence -ellipsis(c): re-use of actions

- (5) Who upset his mother? John did.


$Upset'(\epsilon, Mother(x)(WH))(WH)$
$WH$
$Upset'(\epsilon, Mother(x)(WH))$
$\epsilon, x, Mother(x)(WH)$
$Upset'$
$\lambda.(\epsilon, P)$
$Mother(x)(WH)$
$x$
$Mother(x)$
$\mathbf{U}$
$WH$
$Mother$



```
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($Fo(\mathbf{U}, Ty(e)$);
        SUBSTITUTE($\alpha$, $\alpha \in T_{Context}$); go($\uparrow_0$);
        make-go($\downarrow_1\downarrow_0$); fresh-put($\mathbf{x}$); go($\uparrow_0$);
        make-go($\downarrow_1$);
        put($Fo(Mother'), Ty(e \rightarrow (e \rightarrow cn))$))
```

(6) The man who arrested John failed to read him his rights.
So did the man who arrested Tom.

(7) John interviewed everyone Bill did.

Note: Tree under construction also part of context

# Contexts for Parse States

A context $\mathfrak{C}$ for some partial tree $T_\phi$ established in uttering a string $\phi$ is a sequence:

$$\mathfrak{C} = \mathfrak{C}_D \oplus \mathfrak{T}_\phi$$

$\mathfrak{C}_D$ is a sequence of *inactive* triples (without a pointer)
– a ‘discourse context’,
$\mathfrak{T}_\phi$ is a current active triple of $T_\phi$, the string $\phi$, and actions $A_\phi$

A final parse state may contain more than one triple
(reflecting ambiguity):

(8) (a) A: Mary’s at the cricket ground.
(b) B: Right.
(c) A: I saw her duck.
(d) B: uh huh.
(e) A: It was waddling around on the boundary.
(e’) A: Lucky she did – the ball almost hit her right in the head.
(e”) A: She was very disappointed that she scored no runs.

Parsing thus involves constructing sets of contexts.

# Incremental Generation

# The Flow of Language Understanding – Generation

A GENERATOR STATE $G$ is a pair

$$(X, T_G)$$

of a set $X$ of pairs $(S, P)$, where

$S$ is a candidate partial string

$P$ is the associated PARSE STATE,

and

$T_G$ a GOAL TREE,

which represents the content of the utterance to be produced.

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 $(\{(\emptyset, P_0), T_G\})$:

i.e. the null string and the initial parse state, plus the goal tree.

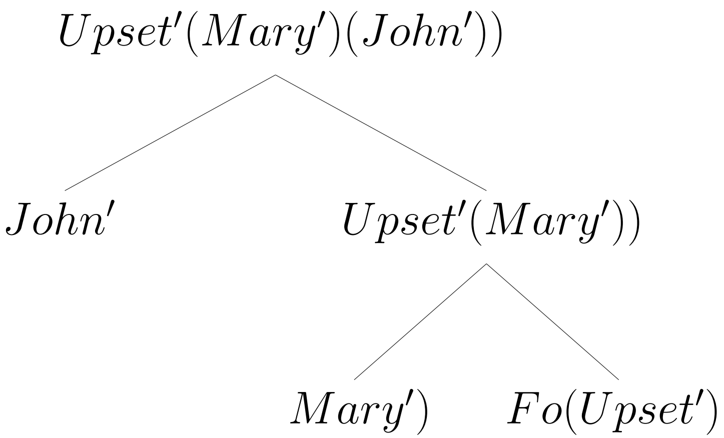
# The Flow of Language Understanding – Generation

Generating ‘John upset Mary’:

PARSER TRIPLE:

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

GOAL TREE



$Upset'(Mary')(John'))$

$John'$

$Upset'(Mary'))$

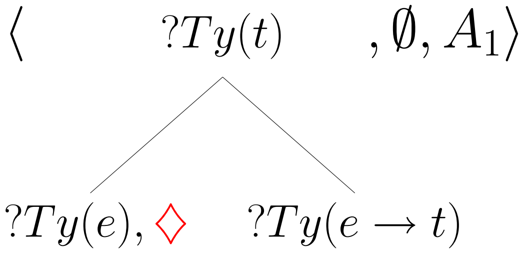
$Mary')$

$Fo(Upset')$

# The Flow of Language Understanding – Generation

Generating ‘John upset Mary’:

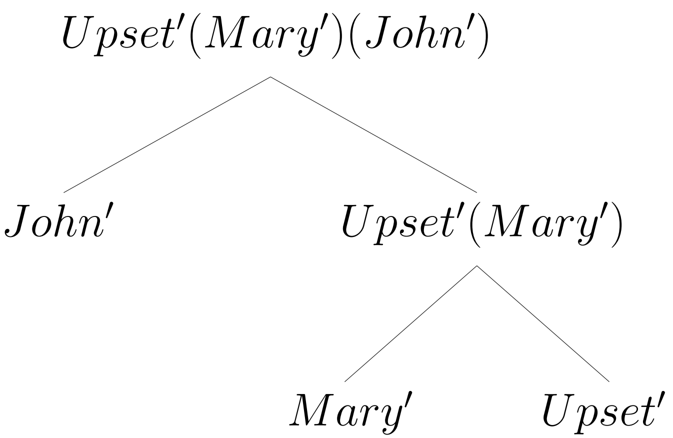
PARSE TRIPLE



$\langle \quad ?Ty(t) \quad , \emptyset, A_1 \rangle$

$?Ty(e), \diamondsuit$ $\quad ?Ty(e \rightarrow t)$

GOAL TREE



$Upset'(Mary')(John')$

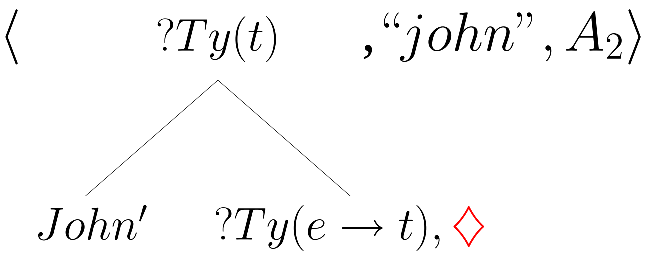
$John'$ $\quad Upset'(Mary')$

$Mary'$ $\quad Upset'$

# The Flow of Language Understanding – Generation

Generating ‘John upset Mary’:

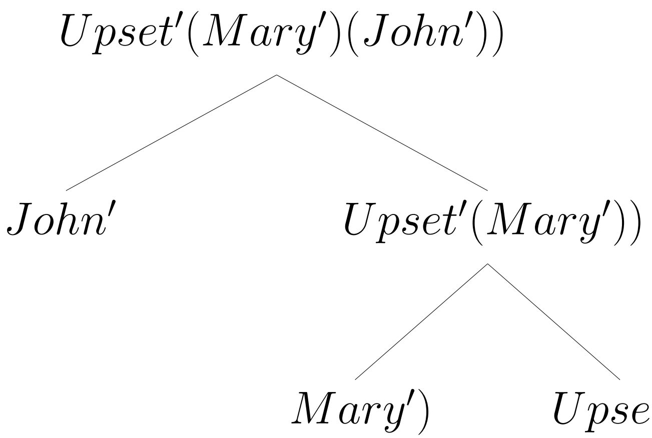
PARSE TRIPLE



$\langle \quad ?Ty(t) \quad , \text{“}john\text{”}, A_2\rangle$

$John'$ $\quad ?Ty(e \rightarrow t), \diamondsuit$

GOAL TREE



$Upset'(Mary')(John'))$

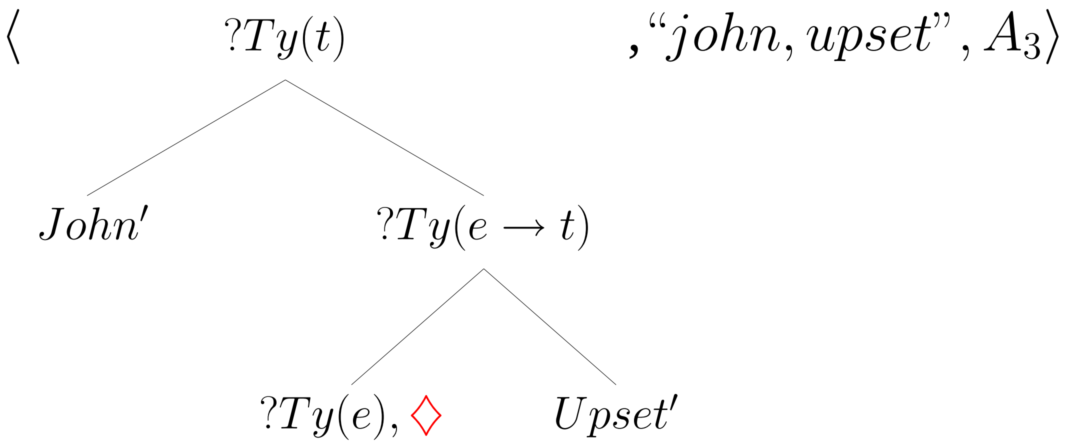
$John'$ $\quad Upset'(Mary'))$

$Mary')$ $\quad Upset'$

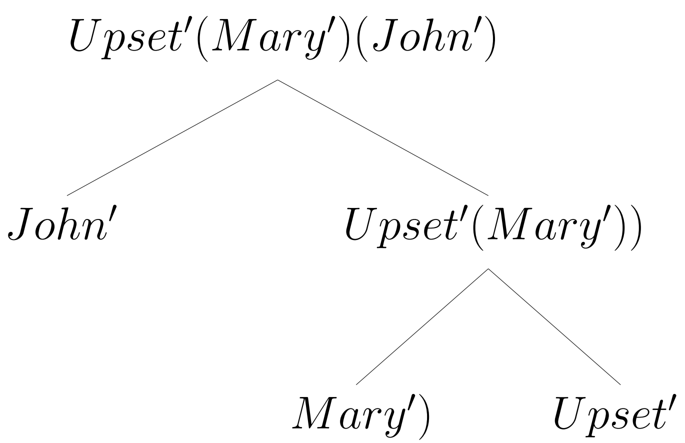
# The Flow of Language Understanding – Generation

Generating 'John upset Mary':

PARSE TRIPLE:



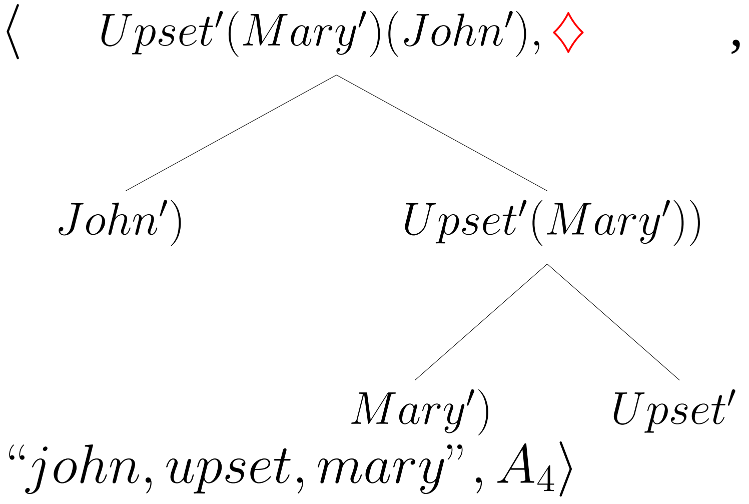
GOAL TREE



# The Flow of Language Understanding – Generation

Generating ‘John upset Mary’:

PARSE TRIPLE



$\langle$ $Upset'(Mary')(John'), \diamondsuit$ ,

$John')$

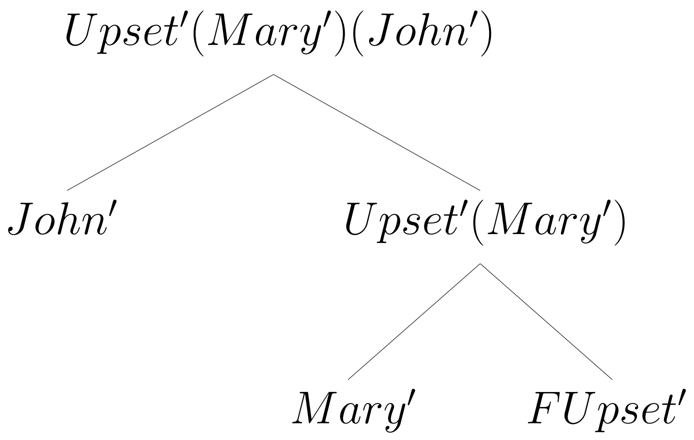
$Upset'(Mary'))$

$Mary')$

$Upset'$

“$john, upset, mary$”, $A_4\rangle$

GOAL TREE



$Upset'(Mary')(John')$

$John'$

$Upset'(Mary')$

$Mary'$

$FUpset'$

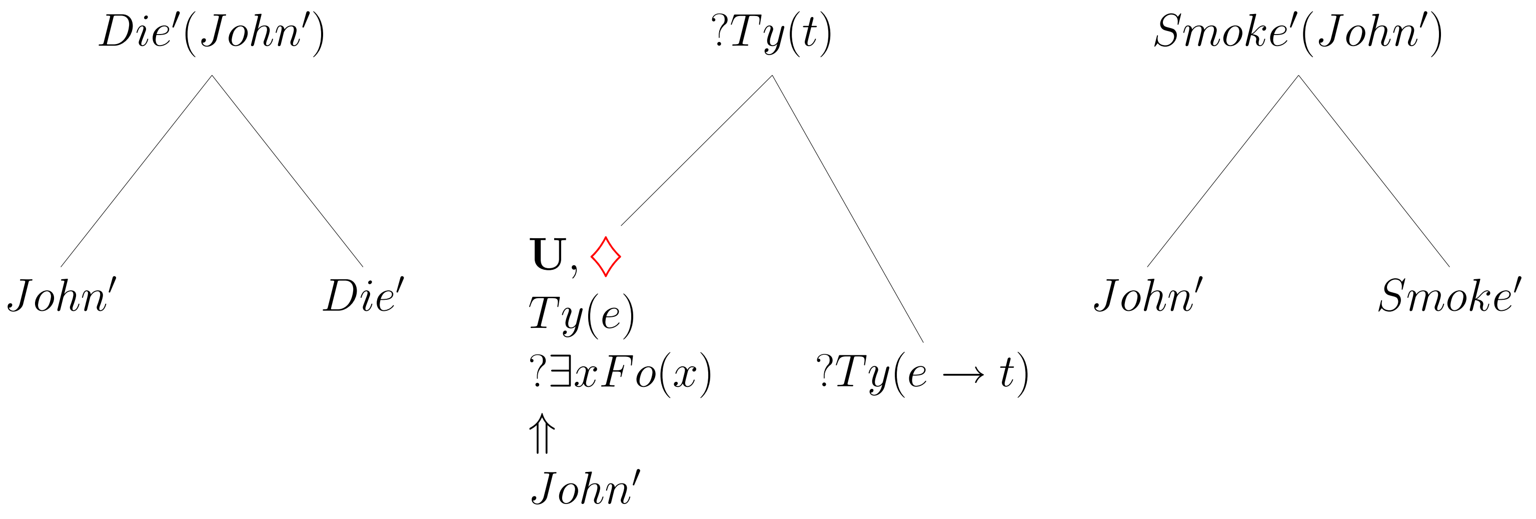
# Generation relative to Context? – Avoiding full lexicon search

- Pronouns require re-use of term from context
- As generation uses the same parsing actions, it uses context in the same way as parsing:

  (9) John died. He smoked.

  Parsing *he*:

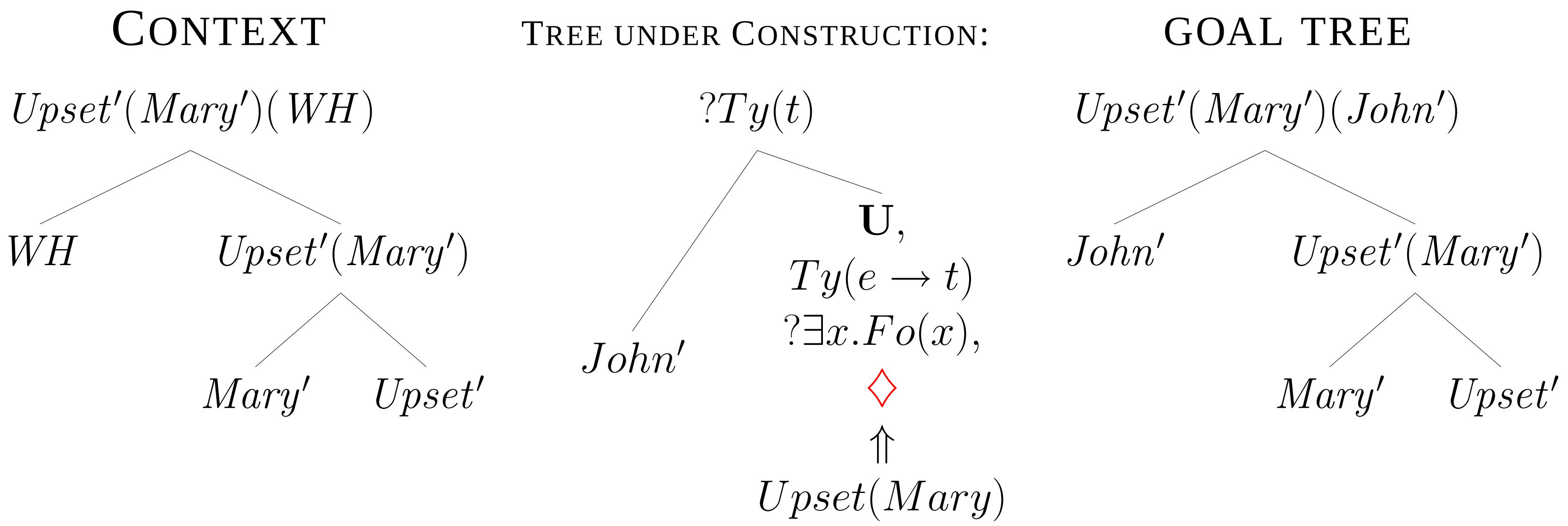
CONTEXT TREE UNDER CONSTRUCTION: GOAL TREE



# Context-dependent generation : ellipsis (a)

- Ellipsis also requires use of **terms** from context, bypassing lexicon search:

  (10) Q: Who upset Mary? Ans: John did.



CONTEXT

$Upset'(Mary')(WH)$

$WH$ $Upset'(Mary')$

$Mary'$ $Upset'$

TREE UNDER CONSTRUCTION:

$?Ty(t)$

$John'$

$\mathbf{U}$, $Ty(e \rightarrow t)$ $?\exists x.Fo(x)$, $\diamondsuit$

$\Uparrow$

$Upset(Mary)$

GOAL TREE

$Upset'(Mary')(John')$

$John'$ $Upset'(Mary')$

$Mary'$ $Upset'$

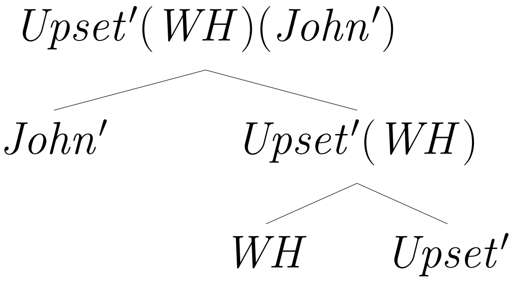
(11) A: The man from next door who owns a Cadillac crashed it into our wall.
B: Did he?

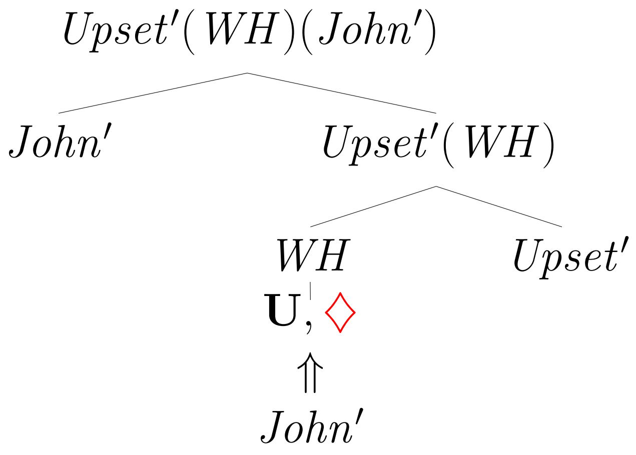
# Context-dependent generation : ellipsis (b)

- Using **structure** from context - generator starts from partial tree:

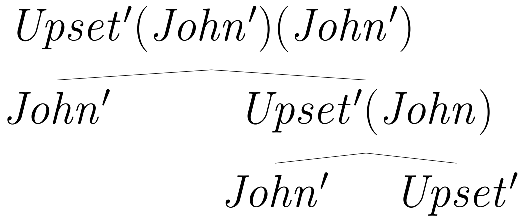
  (12) Q: Who did John upset? Ans: Himself.

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





GOAL TREE



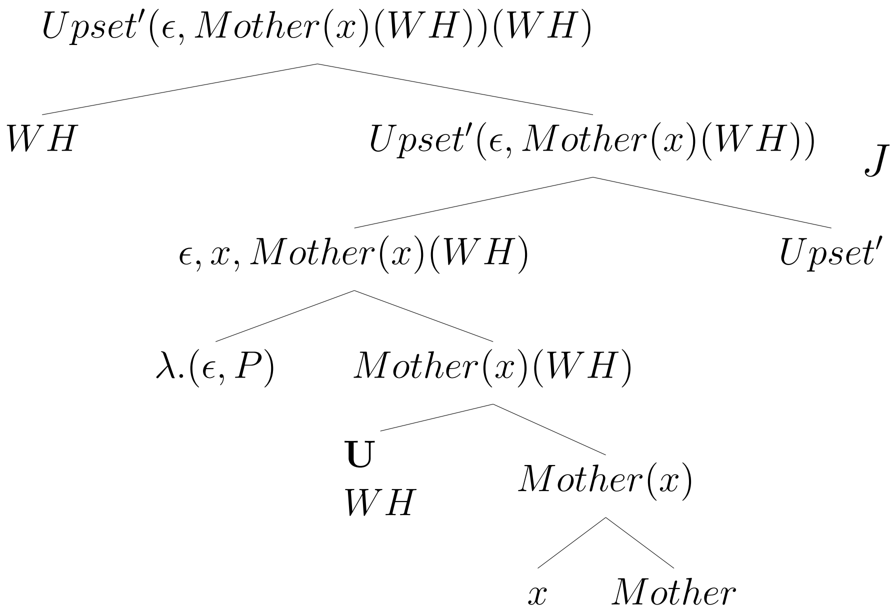
(13) Who did everyone ignore? Their husband.

# Context-dependent generation : ellipsis (c)

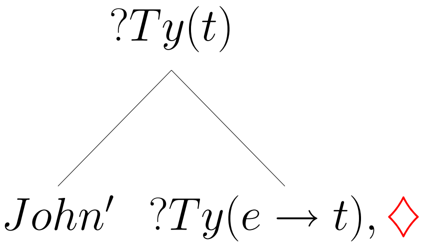
- Using (parsed) **actions** from context also avoids full lexicon search

  Who upset his mother? John did.

CONTEXT


$Upset'(\epsilon, Mother(x)(WH))(WH)$
$WH$
$Upset'(\epsilon, Mother(x)(WH))$
$\epsilon, x, Mother(x)(WH)$
$Upset'$
$\lambda.(\epsilon, P)$
$Mother(x)(WH)$
**U**
$WH$
$Mother(x)$
$x$
$Mother$

TREE UNDER CONSTRUCTION



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($Fo(\mathbf{U}, Ty(e)$);
SUBSTITUTE($\alpha$, $\alpha \in T_{Context}$); go($\uparrow_0$);
make-go($\downarrow_1\downarrow_0$); fresh-put(**x**); go($\uparrow_0$);
make-go($\downarrow_1$);
put($Fo(Mother'), Ty(e \rightarrow (e \rightarrow cn)))$)

GOALTREE
$Upset'(\epsilon x.Mother'(x)(John'))(John')$

Other eg's *John interviewed everyone Bill did* , ....

## Dialogue: Parsing and Generation in Context

(i) Shared utterances

(ii) Alignment

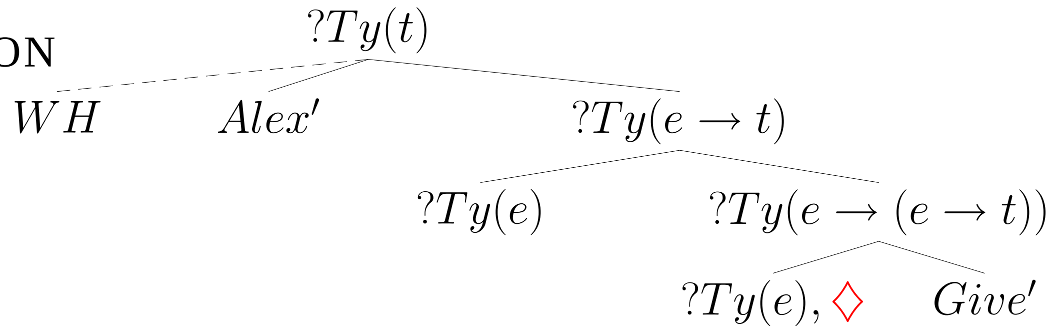
# Shared Utterances: Coordination of Parsing and Generation

- Parsing and Generation *necessarily* share same intermediate representations: both can start from **any** parse state

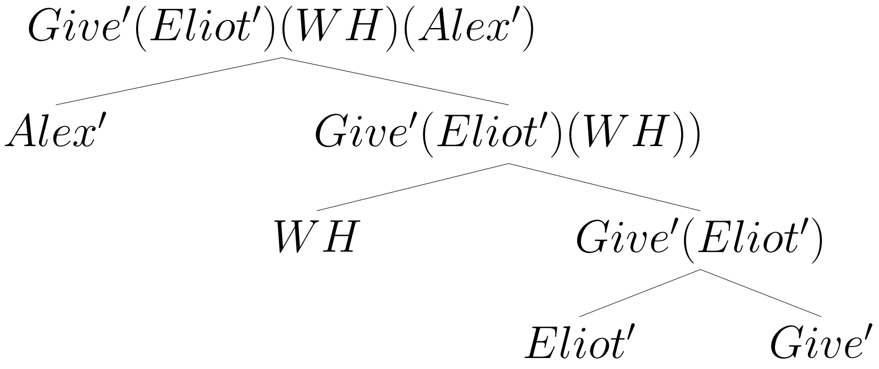
(14) *Ruth:* What did Alex give . . . *Hugh:* . . . Eliot? A teddy-bear.

PRIOR TO SPEAKER/HEARER TRANSITION:

TREE UNDER CONSTRUCTION



GOAL TREE



# Alignment: Using Actions from Context

- lexical/syntactic/semantic alignment all as Lexical Search Minimisation (economising processing costs):
  - instead of looking in lexicon, use lexical action from context
- repeated use of words and subcategorisation pattern established (Branigan et al., 2000):

  (15) A:John gave Mary a teddy-bear.
  B: And Tom gave Sue a music-box.
- repetition of computational actions:

  (16) A: How are your new neighbours?
  B: John, I like. Sue, I don't much care for.
- repetition of word-term pairing:

  (17) M: Have you set up things at the bank yet?
  R: I tried to, but when the bank insisted on having your signature, I left.

Unlike split utterances, fragments, etc, alignment does not involve grammar-induced trigger.

# Defining well-formedness

- A sentence string $\phi$ uttered with respect to a discourse context, $\mathfrak{C}_D$, is well-formed iff:

$$\mathfrak{C}_D \oplus \mathfrak{T}_0 \quad \longmapsto_{A_\phi} \quad \mathfrak{C}_D \oplus \mathfrak{T}_\phi$$

$\mathfrak{C}_D$ is the context given by the prior discourse
(a sequence of inactive triples);

$\mathfrak{T}_0$ is $\langle T_0, \emptyset, \emptyset \rangle$, the standard initial state;

$A_\phi$ is the set of lexical, computational and pragmatic actions used in parsing $\phi$ on a strictly time-linear basis;

and $\mathfrak{T}_\phi$ is complete (i.e. $\mathfrak{T}_\phi = \langle T_\phi, \phi, A_\phi \rangle$ where $T_\phi$ is a complete tree).

(18) John fainted. He was sick.

(19) John fainted. ♯She was sick.

This is the concept of felicitous use of DRT, Heim, etc.

# Retaining Traditional wellformedness

- A string $\phi$ is fully grammatical iff an utterance of $\phi$ is well-formed in the null context:

  $$\emptyset \oplus \mathfrak{T}_0 \quad \mapsto_{A_\phi} \quad \emptyset \oplus \mathfrak{T}_\phi$$

  $\mathfrak{T}_0$ is $\langle T_0, \emptyset, \emptyset \rangle$, the standard initial state;
  $A_\phi$ is the set of lexical, computational and pragmatic actions used in parsing $\phi$ on a strictly time-linear basis;
  $\mathfrak{T}_\phi$ is complete (i.e. $\mathfrak{T}_\phi = \langle T_\phi, \phi, A_\phi \rangle$ where $T_\phi$ is a complete tree).

- A string is fully ungrammatical iff there is no context in which an utterance of $\phi$ is wellformed.

# Defining grammaticality

A string $\phi$ is grammatical iff an utterance of $\phi$ is well-formed in SOME discourse context, i.e. as long as the actions in processing that fragment in conjunction with that context yield a complete tree:

$$\exists\mathfrak{C}[\mathfrak{C} \oplus \mathfrak{T}_\psi \quad \mapsto_{A_\phi} \quad \mathfrak{C} \oplus \mathfrak{T}_\phi]$$

$\mathfrak{C}$ is the context (a sequence of inactive triples plus one active triple);
$\mathfrak{T}_\psi$ is some active triple $\langle T_\psi, \psi, A_\psi \rangle$ taken from a parse state $P_\psi$ used to parse a previously uttered string $\psi$ ;
$A_\phi$ is the set of lexical, computational and pragmatic actions used in parsing $\phi$ on a strictly time-linear basis;
$\mathfrak{T}_\phi$ is complete.

(i) He did, too.

(ii) Himself.

(iii) A: When will you get home? Are you listening?
B: Sorry, I was miles away. 8.15.

The grammar excludes only strings that cannot lead to wellformed complete proposition outputs:

(iv) Have you read?

(v) Where are?

# Grammars and Dialogue as Core Data

- With syntax as progressive/incremental building of logical forms, we have:
  - a unitary basis for characterising utterance-internal and cross-utterance ellipsis
  - a context-dependent characterisation of wellformedness,
  - a system that is adapted to modelling dialogue as core data,
  - a broader remit of data including split utterances

  BUT we retain:
- the grammar as a set of constraints
- the grammar does not determine mechanisms for interpretation selection
- the grammar does not define a concept of “well-formed dialogue”, and is not a “dialogue grammar”.

# Publications

- Blackburn, P. and Meyer-Viol, W. 1994. Linguistics, Logic and Finite Trees. Bulletin of the IGPL 2, pp.3–31.

- Cann, R. Kempson, R, Marten, L. 2005. *The Dynamics of Language*. Oxford: Elsevier.

- Kempson, R. Meyer-Viol, W. Gabbay, D. 2002. *Dynamic Syntax: The Flow of Language Understanding.* Oxford: Blackwell

- Purver, M. Cann, R., Kempson, R. (in press). Grammars as parsers: meeting the dialogue challenge. In Piwek, P and Kibble, R. (ed) issue of *Research in Language and Computation*

For items of related interest, keep an eye on:

http://semantics.phil.kcl.ac.uk/ldsnl
http://dialoguematters.stanford.edu/twiki-dm/bin/view/Main/WebHome