FOCUS ON "ONLY" AND "NOT"

Allan Ramsay
Dept. of Computer Science, University College Dublin, Belfield, Dublin 4, Ireland
allan@monkey.ucd.ie

Krifka [1993] has suggested that focus should be seen as a means of providing material for a range of semantic and pragmatic functions to work on, rather than as a specific semantic or pragmatic function itself. The current paper describes an implementation of this general idea, and applies it to the interpretation of *only* and *not*.

1 Background

Consider the following sentences:

(1) *I only borrowed your car*

(2) *I only borrowed your car*

(3) *I only borrowed your car*

All of them entail the same basic message, namely that I borrowed your car. In addition to the basic message, however, they also carry information about what I didn't do. (1) says that I didn't borrow any of your other possessions, (2) says that I didn't borrow anyone else's car, and (3) says that I didn't do anything else to your car. It seems as though the word *only* and the focus marker (indicated here by underlining the stressed word) combine to add an extra message about what I didn't do.

A similar phenomenon appears to be taking place in the next set of sentences:

(4) *I didn't steal your car*

(5) *I didn't steal your car*

(6) *I didn't steal your car*

Each of these says that I didn't steal your car, but again they each carry some extra message. (4) says that I did steal something which belongs to you, (5) says that I stole somebody's car, but not yours, and (6) says that I did do something to your car (I probably borrowed it, though that is not entailed by (6)).

Krifka [1993] argues that in (1–3) and (4–6), and in a number of other situations as well, the focus marker[1] is used to extract part of the interpretation. Operators like *only* and so-called "focussed negation" then combine the extracted element of the interpretation with what was left behind to carry complex messages of the kind discussed above.

The current paper shows how to implement this general notion, without following Krifka's analysis in detail. The crucial point is the provision of some way of storing the extracted part of the interpretation and making it available when required. The interpretation of *only* and focussed negation is fairly straightforward, so long as the treatment of the focussed item itself is coherent.

2 Abstraction and Focus

The general aim of this paper is to show how to use focus to decompose the interpretation of a phrase into two parts, where one part is the interpretation of the focussed item and the other is some object with which this can combine. Suppose, for example, we thought that the VP *ate a peach* should be interpreted as:

$$\lambda A \exists Y (event(Y) \wedge type(Y, eat) \wedge past(Y) \wedge agent(Y, A) \wedge \exists X (peach(X) \wedge object(Y, X)))$$

In other words, this VP is an abstraction over events where somebody ate a peach. Then we would want the two objects corresponding to the interpretation of *ate a peach* to be something like:

$$\lambda Z peach(Z)$$

[1] The term FOCUS has been used in a wide variety of ways. In the present paper I simply use it to denote the part(s) of an utterance to which attention is drawn by stress markers.

and

$$\lambda P \lambda A \exists Y(event(Y) \wedge type(Y, eat) \wedge past(Y) \wedge agent(Y, A) \wedge \exists X P.X \wedge object(Y, X)))$$

Here we have extracted the denotation of *peach* as the property of being a peach, and converted the interpretation of the VP to an abstraction which will combine appropriately with this property to reproduce the original interpretation [2].

Where else do we see a phenomenon of this kind? Consider the following phrases:

(7) *the man who stole your bike*

(8) *the man who I wanted you to meet*

In (7) the property of being a man combines with the property of being someone who stole your bike to construct a contextually minimal unique characterisation of the relevant individual, and similarly in (8). To achieve this, we need to interpret the relative pronouns in the two relative clauses as leaving a hole in the interpretation of clause and then abstracting with respect to that hole. This is clear for (8), but it also holds for (7) if we want to interpret a sentence like *a man stole a bike* as

$$\exists Y(event(Y) \wedge type(Y, steal) \wedge past(Y) \wedge \exists Z(man(Z) \wedge agent(Y, Z)) \wedge \exists X(bike(X) \wedge object(Y, X)))$$

where the quantifier introduced by the subject does not in fact have maximal scope (an analysis I have argued for elsewhere [Ramsay 1992a]).

The treatment of (8) clearly requires much the same mechanism as we will require if we want to deal with focus as outlined above, and this may or may not also hold for (7). Any serious NLP system will include some way of dealing with the interpretation of cases like (8), and almost any such mechanism should be open to adaptation to deal with focus along the suggested lines. One such approach is outlined below.

[2] You cannot freely mix λ-calculus and the truth functional connectives of predicate calculus as we have here without running into the paradoxes of self-reference — Russell's paradox, the Liar, and so on. The notation used in this paper looks, for the sake of familiarity, like a combination of λ-calculus and predicate calculus, but is in fact grounded in the revision-based semantics of Turner's [1987] PROPERTY THEORY.

3 Quantification, Presupposition, Abstraction and Focus

We expect to interpret relative clauses (uncontroversially) and phrases with focussed constituents (more controversially) as abstractions over the interpretations of simple sentences. In order to construct interpretations of the kinds of objects we are interested in, then, we have to start by looking at simple sentences. The analyses presented in this paper start from the following observations, most of which are fairly orthodox:

- Indefinite NPs should be viewed as a way of introducing items (or preferably sets of items) into the discourse. Universally quantified NPs say that all items of the specified type satisfy some property.
- VPs should be viewed as a way of introducing events or possibly sets of events into the discourse.
- If you construct interpretations by paraphrasing NL sentences into a formal language which extends predicate calculus, you have to realise that the scope of quantifiers in your paraphrases may not be determined by simple structural properties of the source text.
- Definite NPs and other presuppositional constructions place constraints on the discourse, so that a sentence containing the phrase *the man* will be uninterpretable in contexts not containing a unique man (a version of this point has been made by, among others, Barwise & Perry [1983], Kamp [1984], Groenendijk & Stokhof [1987]).
- There are interactions of scope between definite NPs and other types of expression: in *Each man kills the thing he loves*, the presuppositional construct *the thing he loves* requires the existence of a single target of affection *per man*.

The standard way to deal with the potential discrepancy between where a phrase appears and the width of its scope is by storing quantifiers on a quantifier stack until the entire sentence has been interpreted, and then using explicit information about the priority of various quantifiers to sort things out [Cooper 1983, Vestre 1991]. The work

reported here follows this treatment, but extends it by introducing quantifier-like entities for dealing with presuppositional items such as definite NPs (see Ramsay [1992b, 1994] for a formal account of such CONSTRAINTS on whether a sentence is meaningful with respect to a situation). As an example, the sentence *the woman stole a bike* is interpreted as

$$\begin{array}{l}\exists A\ A < now\\ \quad \wedge\ \iota B : (\forall C\ member(C,B) \rightarrow\ woman(C)\\ \qquad\quad \wedge\ |B| = 1)\\ \qquad \exists D\ \forall E\ member(E,D) \rightarrow\ bike(E)\\ \qquad\quad \wedge\ |D| = 1\\ \qquad\quad \wedge\ simple(A, \lambda F\ event(F)\\ \qquad\qquad\qquad\qquad \wedge\ type(F, steal)\\ \qquad\qquad\qquad\qquad \wedge\ agent(F,B)\\ \qquad\qquad\qquad\qquad \wedge\ object(F,D))\end{array}$$

This says that the relationship *simple* holds between some past instant A and the property of being a certain sort of event. What sort of event? One where a bike is stolen by someone (or rather, where a singleton set of bikes is stolen). Writing something like $\iota B\ :\ (\forall C\ member(C,B) \rightarrow\ woman(C) \wedge\ |B| = 1)W$, where W may contain occurrences of B, says that W holds for the contextually unique individual B which satisfies the restriction that B is a woman (is a singleton set of women). If this restriction fails to pick out a unique individual the whole expression is meaningless in the context.

Most of this analysis is fairly orthodox. The two main points that might require some defence are the analysis of aspect in terms of a relationship between temporal objects and event types, which is discussed in [Ramsay 1993], and the treatment of definite reference in terms of constraints on meaningfulness. Neither of these is crucial to the remainder of the paper, but if you don't like them you will have to replace them with something better, and you are unlikely to find something which is both better and simpler.

The analysis above was obtained in a framework where quantifier scope is determined on the basis of information explicitly associated with a form of COOPER STORAGE ([Cooper 1983]), using abstraction operators of the form $\lambda W \exists X W$, $\lambda W \forall X W$ or $\lambda W \iota X : RW$ which can be applied to a formula to bind its free variables. Within this framework, it is perfectly easy to deal with cases like (8) by allowing the relative pronoun to add the expression $\lambda W \lambda X W$ to the quantifier store, annotated to specify that this expression has maximal scope. If this expression is applied to a formula containing a free occurrence of X it will return an abstraction with respect to X — exactly what we want. The requirement that this should have maximal scope will ensure that X is the last free variable in W.

But if we can use this mechanism to construct an abstraction as the interpretation of a relative clause, we can also use it to construct an abstraction as the interpretation of a phrase containing a focussed item. The only extra work we have to perform is that we have to find somewhere to put the interpretation of the focussed item itself. To do this, all that is needed is an extra feature `focus` in the descriptions of linguistic items. The value of `focus` is the focussed item itself. `focus` behaves like a GPSG FOOT FEATURE, in that at most one daughter of an item can have a non-vacuous value for `focus`, and that if an item does have exactly one daughter with a non-vacuous value for this feature then the item will share this value with that daughter. `focus` is thus very like the standard feature `slash` which is used for dealing with left extraposition — it is a foot feature whose value is some item which is somehow "out of position".

4 Applications of Focus

Once we have this mechanism, we can use it to construct interpretations of sentences like (1)–(6). Consider, for instance, the example:

(9) *I only borrowed a car*

$$\begin{array}{l}only(\lambda A car(A),\\ \quad \lambda B\ \exists C\ C < now\\ \qquad \wedge\ \exists D\ \forall E\ member(E,D) \rightarrow\ B.E\\ \qquad\quad \wedge\ |D| = 1\\ \qquad\quad \wedge\ \iota F : (\forall G member(G,F)\\ \qquad\qquad\qquad\qquad \rightarrow\ speaker(G)\\ \qquad\qquad\quad \wedge\ |F| = 1)\\ \qquad\qquad simple(C,K)))\end{array}$$

where K is $\lambda H event(H) \wedge\ type(H, borrow) \wedge agent(H,F) \wedge\ object(H,D)$ (this has been extracted from the displayed formula to get it inside the available space — it is in fact part of that formula).

This says that the relationship *only* holds between the property of being a car and some other object. This is fine as far as it goes, but it isn't worth

very much unless we spell out the conditions under which this relationship holds. The following meaning postulate does just that:

$$\forall P \forall Q(only(P,Q) \rightarrow\ Q.P \wedge (\forall P'(Q.P' \rightarrow P' = P)))$$

In other words, if $only(P,Q)$ holds then P satisfies Q and nothing else does. In the present case, the first of these consequences means that I did indeed borrow a car:

$$\begin{array}{l}\exists C\ C < now \\ \quad \wedge \exists D\ \forall E\ member(E,D) \rightarrow\ car(E) \wedge\ |D| = 1 \\ \qquad \wedge\ \iota F : (\forall G\ member(G,F) \rightarrow\ speaker(G) \\ \qquad\qquad \wedge\ |F| = 1) \\ \qquad\quad simple(C,K)))\end{array}$$

where $K = \lambda H event(H) \wedge type(H, borrow) \wedge agent(H,I) \wedge object(H,D)))$ has again been extracted to save space. This was obtained from the meaning postulate by substituting $\lambda A car(A))$ for B and using $(\lambda A car(A)).E \equiv car(E)$.
The second consequence of the MP for *only* says that there is no other category of item which satisfies the abstraction – that the only thing I borrowed was a car.

If we put the focus somewhere else, we get another interpretation:

(10) *I only borrowed a car*

$$\begin{array}{l}only(\lambda A\ \lambda B\ \lambda C\ event(C) \wedge\ type(C, borrow) \\ \qquad\qquad \wedge\ B.\lambda D(agent(C,D)) \\ \qquad\qquad \wedge\ A.\lambda E(object(C,E)), \\ \quad \lambda F\ \exists G\ G < now \\ \qquad \wedge\ \exists H\ \forall I\ member(I,H) \rightarrow\ car(I) \\ \qquad\qquad \wedge\ |H| = 1 \\ \qquad\qquad \wedge\ \iota J : (\forall K member(K,J) \\ \qquad\qquad\qquad\qquad \rightarrow\ speaker(K) \\ \qquad\qquad\qquad \wedge\ |J| = 1) \\ \qquad\qquad\quad simple(G,K))\end{array}$$

where $K = (F.\lambda L(L.H)).\lambda M(M.J)))$

This says that *only* holds between a description of the type of event C where somebody B borrows something A, and an abstraction over situations in which I did something to some car. Then the first consequence of *only* says that what I did to this car was I borrowed it: substituting the description of the event type for the abstracted variable F produces $((\lambda A \lambda B \lambda C event(C) \wedge type(C, borrow) \wedge B.\lambda D(agent(C,D)) \wedge A.\lambda E(object(C, E))).\lambda L(L.H)).\lambda M(M.J)$ as the second argument of *simple*, and this reduces to $\lambda C event(C) \wedge type(C, borrow) \wedge agent(C, J)) \wedge object(C, H)$, which is what we want. The second says that I didn't do anything else to it.

Much the same will happen with

(11) *I didn't steal it*

$$\begin{array}{l}not(\lambda A\ \lambda B\ \lambda C\ event(C) \wedge\ type(C, steal) \\ \qquad\qquad \wedge\ B.\lambda D(agent(C,D)) \\ \qquad\qquad \wedge\ A.\lambda E(object(C,E)), \\ \quad \lambda F\ \exists G\ G < now \\ \qquad \wedge\ \iota H : (\forall I\ member(I,H) \rightarrow\ neuter(I) \\ \qquad\qquad \wedge\ |H| = 1) \\ \qquad\quad \iota J : (\forall K\ member(K,J) \\ \qquad\qquad\qquad\qquad \rightarrow\ speaker(K) \\ \qquad\qquad\quad \wedge\ |J| = 1) \\ \qquad\qquad simple(G,K))\end{array}$$

where $K = F.\lambda L(L.H).\lambda M(M.J)))$

Here we have a 2-place relation *not*, which is backed up by the following MP:

$$\forall P \forall Q(not(P,Q) \rightarrow (\neg Q.P \wedge \exists P'(Q.P')))$$

This says that this form of negation holds between P and Q if Q does not hold of P, but does hold for some other entity P'. In the present case, this means that I did do something to it (whatever "it" is), but what I did was not stealing.

This contrasts with simple negation, with no focussed item, as in:

(12) *I didn't steal it*

$$\begin{array}{l}\neg(\exists A\ A < now \\ \qquad \wedge\ \iota B : (\forall C\ member(C,B) \rightarrow\ neuter(C) \\ \qquad\qquad \wedge\ |B| = 1) \\ \qquad\quad \iota D : (\forall E\ member(E,D) \\ \qquad\qquad\qquad\qquad \rightarrow\ speaker(E) \\ \qquad\qquad\quad \wedge\ |D| = 1) \\ \qquad\qquad simple(A,K)))\end{array}$$

where $K = \lambda F event(F) \wedge type(F, steal) \wedge agent(F,D) \wedge object(F,B)))$

This simply says that it is not the case that there is a past stealing event involving me and it. The choice between the two is forced by the presence or absence of a focussed item.

As a final example, consider a sentence which contains a focussed item but no operator for using it up:

(13) *A man ate it*

The analysis of this is an abstraction over kinds of individuals who ate it

$$\lambda A\ \exists B\ B < now \wedge \exists C\ \forall D\ member(D, C) \rightarrow A.D \wedge |C| = 1 \wedge \iota E : (\forall F\ member(F, E) \rightarrow neuter(F) \wedge |E| = 1) simple(B, K)$$

with $K = \lambda Gevent(G) \wedge type(G, eat) \wedge agent(G, C) \wedge object(G, E))$, and with the focus set as the description (including the semantic analysis) of the focussed phrase *man*. This is just the kind of object required for a discourse operator such as contrast or elaboration — exactly which such operator is appropriate depends on factors not visible in (13) itself, but whatever it is it will require a pair of arguments of this kind.

5 Conclusions

The discussion above shows what can be achieved by treating focus as a syntactic marker which makes information available to a variety of operators. The mechanism for doing this involves introducing a foot feature to carry the focussed item around, and constructing appropriate abstractions by using the standard quantifier scoping mechanism which is required for other phenomena anyway. Different NLP systems will deal with the syntax and semantics of phenomena such as left- and right-extraposition in different ways. What I have argued is that almost any approach to these phenomena can be adapted to deal with focus as well. The examples in Section 4 showed how you can combine these analyses of focus with a variety of operators to convey a range of interpretations of the same sequence of words. It is important to recall at this point that the interpretation language being used here is a highly intensional logic which permits quantification over arbitrary kinds of individual, including quantification over properties and propositions. I have argued elsewhere that such a language is required for a wide variety of phenomena. The interpretation of focus is just another example.

6 Implementation

All the analyses in this paper were produced, and λ-reduced (and turned into LaTeX!), by a version of the system described in [Ramsay 1992a]. This consists of a highly lexical grammar with a compositional semantics, parsed via a bi-directional head-driven chart parser. I believe it is virtually impossible to do this kind of work without embodying it in a working system. You simply cannot explore the consequences of doing something one way rather than another, or of combining an analysis of this with an analysis of that, unless activities such as compositional construction and subsequent λ-reduction of interpretations is done for you by machine.

References

Barwise J. & Perry J. (1983): *Situations and Attitudes*: Bradford Books, Cambridge, MA.

Cooper R. (1983): *Quantification and Syntactic Theory*: Reidel, Dordrecht.

Groenendijk J. & Stokhof M. (1987): *Dynamic Predicate Logic: Towards a Compositional and Non-representational Discourse Theory*, Technical Report, University of Amsterdam.

Kamp H. (1984): A Theory of Truth and Semantic Representation, in *Formal Methods in the Study of Language* (eds. J. Groenendijk, J. Janssen & M. Stokhof): Foris Publications, Dordrecht: 277–322.

Krifka M. (1993): Focus, Presupposition and Dynamic Interpretation, *Journal of Semantics* 10.

Ramsay A.M. (1992a): Bare Plural NPs and Habitual VPs, *COLING-92*, Nantes: 226-231.

Ramsay A.M. (1992b): Presuppositions and WH-clauses, *Journal of Semantics* 9: 251–186.

Ramsay A.M. (1993): *Aspect and Aktionsart Without Coercion*, submitted to ACL-94.

Ramsay A.M. (1994): Meanings as Constraints on Information States, in *Constraints, Language and Computation* (eds. C.J. Rupp, M.A. Rosner & R.L. Johnson): Academic Press, London: 249–276.

Turner R. (1987): A Theory of Properties, *Journal of Symbolic Logic* 52(2): 455–472.

Vestre E. (1991): An Algorithm for Generating Non-redundant Quantifier Scopings, *Fifth Conference of the European Chapter of the Assoc. for Computational Linguistics*, Berlin: 251–256.