The *can do* and *is* of exhaustification and "only"

Alastair Butler
University of Amsterdam

1 Introduction

This paper focuses on what exhaustification *can do* and *is*. It offers a proposal for what the assertional content is of a sentence with exhaustification, and what the role and effect is of the models/domains/contexts used to evaluate such sentences. It ends with a look at "only," which is argued to be a listing operation.

2 Core data: what exhaustification can do

A number of sentence types, closely related in terms of their syntax (including: questions, clefts, free relatives, comparatives, degree relatives), either invoke or rely on exhaustive interpretations. For example, this leads to a maximum value being returned by the pivot of the cleft in (1) and a minimum value being returned in (2).

(1) It is $[_{\text{pivot}}$four sheep] that John owns.
(4 *is the maximum number* n *for which John has* n *sheep.*)

(2) It is inside $[_{\text{pivot}}$four minutes] that Roger ran the mile.
(4 *is the minimum number* n *for which Roger ran the mile inside* n *minutes.*)

Exhaustification needs to say how and why different value types (maximum and minimum) are returned in (1) and (2).

3 Strict implication and the assertional content of exhaustification

With strict implication, a notion of strength can be formalised as in (3).

(3) ϕ is stronger than or equal to ψ if and only if $\Box(\phi \rightarrow \psi)$.

(3) can be used to say, for example, that a is stronger than or equal to b in predicate context P, by requiring (4) to hold.

(4) $\Box(P(a) \rightarrow P(b))$.

Using (4), exhaustification can take the form of an operation that forces quantifiers to pick strongest values — a technique introduced in Zeevat (1994). (5) formulates such an operation for giving an exhaustive value to variable x in predicate context P.

(5) $\exists x(P(x) \wedge \neg\exists y(x \neq y \wedge P(y) \wedge \Box(P(y) \rightarrow P(x))))$.

This allows us to translate (1) and (2) as (6) and (7), respectively.

(6) $\exists n(\textit{john-owns-sheep}(n) \wedge \neg\exists m(n \neq m \wedge \textit{john-owns-sheep}(m) \wedge \Box(\textit{john-owns-sheep}(m) \rightarrow \textit{john-owns-sheep}(n))) \wedge n = 4)$.

(7) $\exists n(\textit{roger-ran-inside}(n) \wedge \neg\exists m(n \neq m \wedge \textit{roger-ran-inside}(m) \wedge \Box(\textit{roger-ran-inside}(m) \rightarrow \textit{roger-ran-inside}(n))) \wedge n = 4)$.

Notably, (6) returns a maximum value because $\textit{john-owns-sheep}(x)$ allows inferences from larger values to smaller values, while (7) rerurns a minimum value because $\textit{roger-ran-inside}(x)$ has the reverse property of allowing inferences from smaller values to larger values.

4 On an apparent breakdown of exhaustification

However, it seems that not all clefts give rise to an exhaustive interpretation. For example, a world in which you can buy an Italian newspaper in a shop other than the one down the road remains compatible with (8). Interestingly, in (9), "only" has the effect of restoring exhaustification. How can this be?

(8) It is [$_{\text{pivot}}$in the shop down the road] that you can buy an Italian newspaper.

(9) It is [$_{\text{pivot}}$only in the shop down the road] that you can buy an Italian newspaper.

An important side effect of the set-up of the exhaustification instruction in (5) is that it does not require exhaustive values to be unique. If no entailment relation holds between two values then they are equally strong and so equally exhaustive. Now suppose we translate (8) as (10), using P for *You can buy an Italian newspaper in*:

(10) $\exists x(P(x) \wedge \neg\exists y(x \neq y \wedge P(y) \wedge \Box(P(y) \rightarrow P(x))) \wedge x = \textit{shop-down-road})$.

and evaluate (10) with respect to the models of (11) and (12). Notably, (11) and (12) represent the same facts about worlds w_1–w_8. For example, in w_2 of both models, you can buy an Italian newspaper only in shops "a" and "b", in w_7, only in shop "c", etc. However, they differ in their domains: (11) has a plural domain, while (12) does not.

(11) $M_1 = \langle W_1, D_1, R_1, V_1\rangle$, where:
$W_1 = \{w_i : 1 \leq i \leq 8\}$
$D_1 = \{\{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$
R_1 is a reflexive, symmetric and transitive relation on W_1
$V_1(\textit{shop-down-road}) = \{a\}$
$V_1(P, w_1) = \{\langle\{a\}\rangle, \langle\{b\}\rangle, \langle\{c\}\rangle, \langle\{a, b\}\rangle, \langle\{a, c\}\rangle, \langle\{b, c\}\rangle, \langle\{a, b, c\}\rangle\}$
$V_1(P, w_2) = \{\langle\{a\}\rangle, \langle\{b\}\rangle, \langle\{a, b\}\rangle\}$
$V_1(P, w_3) = \{\langle\{a\}\rangle, \langle\{c\}\rangle, \langle\{a, c\}\rangle\}$
$V_1(P, w_4) = \{\langle\{b\}\rangle, \langle\{c\}\rangle, \langle\{b, c\}\rangle\}$
$V_1(P, w_5) = \{\langle\{a\}\rangle\}$
$V_1(P, w_6) = \{\langle\{b\}\rangle\}$
$V_1(P, w_7) = \{\langle\{c\}\rangle\}$
$V_1(P, w_8) = \emptyset$

(12) $M_2 = \langle W_2, D_2, R_2, V_2\rangle$, where:
$W_2 = \{w_i : 1 \leq i \leq 8\}$
$D_2 = \{a, b, c\}$
R_2 is a reflexive, symmetric and transitive relation on W_2
$V_2(\textit{shop-down-road}) = a$
$V_2(P, w_1) = \{\langle a\rangle, \langle b\rangle, \langle c\rangle\}$
$V_2(P, w_2) = \{\langle a\rangle, \langle b\rangle\}$
$V_2(P, w_3) = \{\langle a\rangle, \langle c\rangle\}$
$V_2(P, w_4) = \{\langle b\rangle, \langle c\rangle\}$
$V_2(P, w_5) = \{\langle a\rangle\}$
$V_2(P, w_6) = \{\langle b\rangle\}$
$V_2(P, w_7) = \{\langle c\rangle\}$
$V_2(P, w_8) = \emptyset$

For model (11), (10) is true only in world w_5. This is because in worlds w_6–w_8, you cannot buy an Italian newspaper in "$\{a\}$", and in worlds w_1–w_4 there are stronger true values (e.g., in w_2, "$\{a, b\}$" is stronger than "$\{a\}$"). For model (12) the situation is very different, as (10) is true in worlds w_1–w_3 and w_5. This is because in each of these worlds you can buy an Italian newspaper in "a" and there are no values stronger than "a" (e.g., although "b" is an alternative value to "a" in w_2, it is not the case that "b" is true in all worlds in which "a" is true). Importantly, w_1–w_3 of (12) are worlds in which you can buy an Italian newspaper in a shop other than the one down the road, and yet they still support (10). It follows that such apparent breakdowns in the exhaustification instruction a cleft gives rise to are (at least in principle) compatible with the idea that clefts uniformally assert exhaustification via the technique of (5).

5 A role for "only"

The findings of the previous section do not account for the effect of "only" in (9), which is one of restoring a demand for an absolute exhaustivity. Suppose the addition of "only" in the pivot of a cleft has the effect of making the denotation of the pivot correspond to a list of all values the cleft construction licences as being exhaustive. For example, this would give (9) the approximate "setof" translation in (13) (again using P for *You can buy an Italian newspaper in*).

(13) $\exists x(setof(y, P(y) \wedge \neg\exists z(y \neq z \wedge P(z) \wedge \Box(P(z) \rightarrow P(y))), x) \wedge x = \textit{shop-down-road})$.

It (trivially) follows that for model (11), (13) is true only in world w_5, since this is the only world in which *shop-down-road* is an exhaustive value. But for model (12), it is also the case that (13) is true only in world w_5. This holds despite *shop-down-road* being an exhaustive value in w_1–w_3; the problem being that the full list of exhaustive values for these worlds is longer than just *shop-down-road*.

6 Conclusion

In this paper, exhaustification was argued to be an operation that licences a value from the domain if it holds true and has no other alternative true value necessarily entailing it. This sometimes gives maximum, sometimes minimum, sometimes unique, and sometimes non-unique values. Licenced values depend on the inferential properties of their contexts. This left an important role for "only" to fill, as forming the demand for an absolute exhaustivity, by licencing only those denotations that correspond to the list of all values licenced by exhaustification.

References

Henk Zeevat. Applying an exhaustivity operator in update semantics. In Hans Kamp, editor, *Ellipsis, Tense, and Questions*, volume R2.2B of *DYANA-deliverable*, pages 233–269, Amsterdam: ILLC, 1994.