Redundancy under Discussion[1]

Adèle HÉNOT-MORTIER — *Massachusetts Institute of Technology*

Abstract. We present novel data derived from a the structure ($p \vee p \vee q$) *via* the *or*-to-*if* tautology and core properties of disjunction (commutativity, associativity). The sentences at stake exhibit differing degrees of oddness, which represents a challenge for existing theories. Building on the QuD-informed model of oddness presented in the current volume (Hénot-Mortier, to appear), and much previous work, in particular Ippolito (2019) and Zhang (2022), we propose a solution covering the data at stake, and which can be extended to other classic instances of oddness, specifically Hurford phenomena Hurford (1974); Singh (2008); Mandelkern and Romoli (2018).

Keywords: oddness, redundancy, disjunction, conditionals, question under discussion

1. Introduction

The disjunctive sentences in (1), which are logically related to each other *via* applications of $\vee$-commutativity and $\vee$-associativity, appear sharply infelicitous.[2] Such sentences can be seen as odd due to them being contextually equivalent to their complex disjunct, $p \vee q$ or $q \vee p$ (Katzir and Singh, 2014).

(1) *Context: Jo got a paper accepted at SuB, but maybe he will not come in person.*
  a. # Either Jo is at SuB, or else he is at SuB or in Boston. $p \vee (p \vee q)$
  b. # Either Jo is at SuB, or else he is in Boston or at SuB. $p \vee (q \vee p)$
  c. # Either Jo is at SuB or in Boston, or else he is at SuB. $(p \vee q) \vee p$
  d. # Either Jo is in Boston or at SuB, or else he is at SuB. $(q \vee p) \vee p$

(2-5) show variants of (1a-1d) obtained *via* the *or*-to-*if* tautology. In each pair of sentences, the a. instances are derived by modifying the outer disjunction, while the the b. instances are derived by modifying the inner disjunction.[3]. Surprisingly, those variants exhibit different degrees of oddness: (2b) and (4b) escape infelicity, while the other variants do not.[4] This is unexpected given that all the sentences in (2-5) have same logical structure as the infelicitous sentences in (1a-1d), assuming implications are material. Particularly puzzling is the existence of a contrast *between* the different b. examples in (2-5), which are derived from (1a-1d) using the *same* transformation.

(2) Derived from (1a):
  a. # If Jo is not at SuB then he is at SuB or in Boston. $\neg p \rightarrow (p \vee q)$
  b. Either Jo is at SuB or if he is not at SuB then he is in Boston. $p \vee (\neg p \rightarrow q)$

[1]Many thanks to Amir Anvari, Athulya Aravind, Omri Doron, Danny Fox, Martin Hackl, Nina Haslinger, Viola Schmitt and Jad Wehbe for their input on that project. Thanks also to the audience and reviewers of SuB29 and of the 2024 BerlinBrnoVienna Workshop for relevant questions, datapoints and suggestions.

[2]More variants could be derived, for instance $q \vee (p \vee p)$. Here, we focus on the less obvious variants where two instances of p do not directly combine together.

[3]One could also apply the *or*-to-*if* tautology to *both* the inner and the outer disjunction in the sentences in (1). Nested conditionals however, are hard to judge. We will briefly cover them in Section 6.2.

[4](4b) sounds more degraded than (2b) however. We come back to this contrast in Section 6.1.

(3) Derived from (1b):

a. # If Jo is not at SuB then he is in Boston or at SuB. $\neg p \rightarrow (q \vee p)$

b. # Either Jo is at SuB or if he is not in Boston then he is at SuB. $p \vee (\neg q \rightarrow p)$

(4) Derived from (1c):

a. # If it's not true that Jo is at SuB or in Boston, then he is at SuB. $\neg(p \vee q) \rightarrow p$

b. ? Either Jo is in Boston if not at SuB, or he is at SuB. $(\neg p \rightarrow q) \vee p$

(5) Derived from (1d):

a. # If it's not true that Jo is in Boston or at SuB, then he is at SuB. $\neg(q \vee p) \rightarrow p$

b. # Either Jo is at SuB if not in Boston, or he is at SuB. $(\neg q \rightarrow p) \vee p$

The intuitive generalization seems to be the following: the sentences in (2-5) that retain an outer disjunction, and whose complex (conditional) disjunct has the negation of their simple disjunct as antecedent, are rescued. Building on the machinery laid out in Hénot-Mortier (to appear), we propose that this descriptive generalization follows from the idea that oddness arises when sentences cannot evoke any well-formed accommodated Questions under Discussion (henceforth QuD, Van Kuppevelt, 1995; Roberts, 1996); and that disjunctions and conditionals give rise to different QuDs. Specifically, we submit that disjunctions raise QuDs making both disjuncts at issue *in parallel* (Simons, 2001; Zhang, 2022), while conditionals "stack" the QuDs of their antecedent and consequent, and only treat the latter as topical. Assuming that the sole application $\vee$-commutativity does not affect oddness, we now focus on sentences (1a), (2a), (2b), (3b), and (4a), repeated in (6).

(6) a. # Either Jo is at SuB, or else he is at SuB or in Boston. $p \vee (p \vee q)$

b. # If Jo is not at SuB then he is at SuB or in Boston. $\neg p \rightarrow (p \vee q)$

c. Either Jo is at SuB or if he is not at SuB then he is in Boston. $p \vee (\neg p \rightarrow q)$

d. # Either Jo is at SuB or if he is not in Boston then he is at SuB. $p \vee (\neg q \rightarrow p)$

e. # If it's not true that Jo is at SuB or in Boston, then he is at SuB. $\neg(p \vee q) \rightarrow p$

The rest of this paper is structured as follows. The next Section reviews why some of the sentences in (6) are problematic for existing accounts of oddness. Section 3 introduces the model of accommodated QuDs laid out in Hénot-Mortier (to appear) and shows how it derives different QuDs for disjunctions and conditionals. Section 4 defines a new REDUNDANCY constraint targeting pairs formed by LFs and their accommodated QuD, and shows how this constraint captures the contrasts in (6). Section 5 compares the constraint to those posited by similar earlier accounts and further connects it to Grice's MAXIM OF MANNER. Section 6 discusses a few additional datapoints related to (6), and Section 7 concludes.

2. Previous accounts

In this section we briefly present three existing accounts of oddness: LOCAL REDUNDANCY CHECKING, SUPER-REDUNDANCY, and NON-TRIVIALITY. We show how the first two accounts fall short in explaining the contrast between the felicitous (6c) and the infelicitous (6b), (6d), and (6e). The last account *can* capture the pattern in (6), but at the cost of mispredicting the classic pattern of Hurford Disjunctions (Hurford, 1974).

2.1. Local Redundancy Checking

Katzir and Singh (2014) propose that the semantic computation evaluates, at certain nodes, whether the composition principle that applies there is non-vacuous. This gives rise to the principle in (7).

(7) LOCAL REDUNDANCY CHECKING. S is deviant if S contains γ s.t. $[\![\gamma]\!] = [\![O(\alpha,\beta)]\!] \equiv_c [\![\zeta]\!]$, $\zeta \in \{\alpha,\beta\}$.

This predicts the double disjunction (6a) to be deviant, because, at the level of the highest disjunction, it is contextually equivalent to its complex disjunct $(p \vee q)$. But, assuming conditionals denote material implications, this also predicts (6b-6d) to be deviant, because, in each of these cases, the highest node (which denotes the whole expression) is equivalent to its right daughter (consequent in (6b); right disjunct in (6c-6d)). Thus, the felicity of (6c) is not derived. Additionally, (6e) is incorrectly predicted to be fine. This is because, when evaluating (7) at the root level, the presence of a negation in the antecedent of the conditional prevents (7) to be triggered, and, at the level of the inner disjunction, (7) is not triggered either.

The issue in fact persists if we adopt a non-material analysis of conditionals. Under this assumption, a conditional is never contextually equivalent to its antecedent or consequent, regardless of what they denote. So, one can focus on disjunctive nodes when evaluating (7) against the sentences in (6). Under a (variably) strict analysis of conditionals, none of the disjunctive nodes in (6b-6e) are predicted to be equivalent to one of their daughters. Therefore, only (6a) is expected to be deviant. Although the felicity of (6c) is predicted under these assumptions, all the other conditional variants are predicted to be felicitous, as well. Again, this is not the expected pattern.

2.2. Super-Redundancy

Kalomoiros (2024), elaborating on Katzir and Singh (2014)'s view, introduces SUPER-REDUNDANCY to cover a wider variety of Hurford Phenomena (Hurford, 1974; Marty and Romoli, 2022; Mandelkern and Romoli, 2018). Roughly, a sentence S is super-redundant if it features a binary operation taking a constituent C as argument, and moreover there is no way of strengthening C to C^+ that would make the resulting sentence S^+ non-redundant (i.e., non-equivalent to its counterpart where C^+ got deleted).

(8) SUPER-REDUNDANCY. A sentence S is infelicitous if it contains $C * C'$ or $C' * C$, with $*$ a binary operation, s.t. $(S)^-_C$ is defined and for all D, $(S)^-_C \equiv S_{Str(C,D)}$. In this definition:
- $(S)^-_C$ refers to S where C got deleted;
- $Str(C,D)$ refers to a strengthening of C with D, defined inductively and whose key property is that it commutes with negation ($Str(\neg\alpha,D) = \neg(Str(\alpha,D))$), as well as with binary operators ($Str(O(\alpha,\beta),D) = O(Str(\alpha,D),Str(\beta,D))$);
- $S_{Str(C,D)}$ refers to S where C is replaced by $Str(C,D)$.

If we set C to be the first occurrence of p for $S \in \{$(6a)...(6e)$\}$, it can be shown that, assuming material implication, $S_{Str(C,D)}$ will always be $(p \wedge D) \vee (p \vee q)$, which turns out to be equivalent to $p \vee q \equiv \neg p \to q \equiv \neg q \to p \equiv (S)^-_C$, regardless of what D is. In other words, (6a-6e) are all super-redundant granted material implication. Assuming strict conditionals does not help. In that case, (6a) and (6b) are super-redundant (setting C as second occurrence of p in both cases), while (6c)-(6e) are not. Using a variably strict semantics may work better, but in any case would be problematic for the data SUPER-REDUNDANCY was originally designed to account for, namely Hurford Conditionals (Mandelkern and Romoli, 2018).

2.3. Non-triviality

Another line of work (Mayr and Romoli, 2016 i.a.), building on the notion of local contexts (Schlenker, 2009), associates oddness with triviality in the sense of (Stalnaker, 1999). This view is summarized in (9).

(9) NON-TRIVIALITY. A sentence S cannot be used in a context c if some part π of S is entailed or contradicted by the local context of π in c.

Assuming (i) that disjunctive local contexts are computed in a left-to-right fashion, i.e. that the local context of the first disjunct is the global context, and that the local context of the second disjunct corresponds to the negation of the first (intersected with the global context); and (ii) that the local context of the consequent of a conditional is the antecedent (intersected with the global context), the pattern in (6) is captured: in all cases but (6c), the second occurrence of p (which is sometimes negated), contradicts the local context set by the first occurrence of p. However, assuming that disjunctive local contexts are left-to-right poses issues for the kind of sentences NON-TRIVIALITY was originally designed to account for, namely Hurford Disjunctions (Hurford, 1974), of the form $r \vee r^+$ / $r^+ \vee r$, with $r^+ \vdash r$. Such sentences, exemplified in (10), are infelicitous regardless of the order of the disjuncts, and can only be captured by NON-TRIVIALITY assuming symmetric local contexts.

(10) a. # Jo studies in Noto or Sicily. b. # Jo studies in Sicily or Noto.

But this, in turn, makes wrong predictions in (6). Specifically, (6c) is predicted to be deviant, because the local context of its first disjunct (p), is taken to be $\neg(\neg p \rightarrow q) = (\neg p) \wedge (\neg q)$, contradicting p.[5]

We now introduce a QuD-based framework, in which the felicity of assertive sentences will be evaluated through the lens of their interaction with the possible QuDs they evoke, in the spirit of Katzir and Singh (2015).

3. Compositional QuDs as a source of oddness

3.1. Overview

Building on the model proposed for Hurford Sentences by Hénot-Mortier (to appear), previous work by Büring (2003); Katzir and Singh (2015); Onea (2016, 2019); Riester (2019); Ippolito (2019); Zhang (2022); Haslinger (2023), as well as current work represented in this volume (Zhang, to appear) we assume that Logical Forms evoke the implicit QuDs they could felicitously answer. Such QuDs are modeled as parse trees of the Context Set (henceforth CS, Stalnaker, 1974), following insights from Büring (2003); Ippolito (2019). We call such stuctures Qtrees, and propose that the Qtrees evoked by a complex LF are derived from the ones evoked by the LF's constitutive parts. As a result, disjunctions and conditionals will be predicted to

[5] Additionally, Hénot-Mortier (to appear) points out that NON-TRIVIALITY fails to account for Hurford Conditionals (Mandelkern and Romoli, 2018).

accommodate distinct Qtrees. Roughly, disjunctions evoke trees that make both disjuncts at issue at the same time, while conditionals evoke trees that makes the consequent at issue in the domain(s) of the CS where the antecedent holds.

This compositional machinery is supplemented by LF-Qtree well-formedness constraints, which rule-out specific Qtrees, derived from specific LFs. In this paper, we will focus on one such constraint, based on the concept of REDUNDANCY, which will rule out Qtrees derived from a sentence, if such Qtrees are also derived from a simplification of the sentence. The existence of LF-Qtree well-formedness constraints, implies that certain LFs may be unable to evoke any well-formed Qtrees. Such LFs will be deemed odd.

This model will predict that the Qtrees evoked by (6a) are all ruled out because they are also evoked by (6a)'s simplification $p \vee q$. Likewise, the Qtrees evoked by (6b) will be ruled out because of the q / $\neg p \rightarrow q$ simplifications, and the one evoked by (6d) will be as well, because of the p simplification. Lastly, the Qtree evoked by (6e) will be deemed ill-formed due to being answerless.

3.2. Key assumptions

We model QuDs as parse trees of the Context Set (Stalnaker, 1974), which can also be seen as nested partitions. (11) defines these structures, which are very similar to Ippolito's "structured sets of alternatives".

(11) *Structure of Question-trees (Qtrees).*
Qtrees are trees whose nodes are all subsets of the CS and s.t.:
- Their root denotes the CS;
- Any intermediate node is partitioned by the set of its children.

In such trees, the root can be seen as a tautology over the CS, and any other node, as a possible answer to the global question denoted by the tree. Intermediate nodes can generally be seen as non-maximal answers, while leaves can generally be seen as maximal ones. Any subtree rooted in a node N can be understood as conditional question taking for granted the proposition denoted by N. Finally, a path from the root to any node N can be seen as a strategy of inquiry (or a sequence of conditional questions) leading to the answer denoted by N.

We assume that out-of-the-blue LFs trigger a Qtree accommodation process that "retro-engineers" a Qtree from the sentence's structure.[6] When evoking a given Qtree, an LF "flags" specific nodes on the tree as maximal true answers. These nodes, that we dub *verifying nodes*, are typically the leaves of the Qtree which are subsets (i.e. entail) the proposition denoted by the LF. They will appear in $\boxed{\text{boxes}}$ in all subsequent figures. Just like Qtree structure, verifying nodes are compositionally derived. Moreover, an accommodated Qtree should allow the sentence evoking it to properly answer it; that is why we assume that any well-formed Qtree derived from a sentence should come with a non-empty set of verifying nodes; cf. (12). More generally, we assume that oddness results from the fact that a given sentence, through its LF, cannot give rise to any well-formed Qtree; cf. (13) and (14).

[6]Here, we do not cover the case of assertive sentences that are direct answers to an overt QuD. There is in fact an interesting line of work showing that overt QuDs can influence pragmatic oddness, especially when it comes to matters of REDUNDANCY (Haslinger, 2023).

(12) EMPTY FLAGGING. A Qtree T evoked by a sentence S exhibits an empty flagging if it does not have any verifying nodes.

(13) *Oddness of a Qtree, given a sentence.* If a sentence S evokes a Qtree T and the pair (S,T) violates a well-formedness constraint, e.g. EMPTY FLAGGING (12), or Q-REDUNDANCY (tbd), then T is deemed odd given S.

(14) *Oddness of a sentence.* A sentence S is odd if any Qtree it evokes is odd given S.

We now proceed to defining Qtrees evoked by "simplex" LFs, understood as LFs which do not contain a node of type t besides their root.

3.3. QuDs evoked by simplex LFs

We assume that a simplex LF denoting a proposition p can give rise to two types of Qtree:[7] a "polar-question" depth-1 Qtree whose leaves are the p and $\neg p$ worlds respectively; and a "*wh*-question" depth-1 Qtree whose leaves are p and relevant, mutually exclusive alternatives to p.[8] Moreover, verifying nodes are defined on such trees as simply the p-leaf (if present).

Looking back at (6a-6e), where S_p = *Jo is at SuB* denotes p and S_q = *Jo is in Boston* denotes q, it is reasonable to think S_p and S_q are exclusive mutual alternatives. Other alternatives may be S_r = *Jo is in Chicago* etc. As a result, the Qtrees compatible with S_p and S_q are given in Figures 1 and 2. The two Figures are equivalent *modulo* a permutation of p and q. Additionally, Figures 1B and 2B show that the "*wh*" Qtrees raised by S_p and S_q have similar structures, in the sense that their partitioning of the CS is the same, ignoring verifying nodes. The corresponding "polar" Qtrees in Figures 1A and 2A on the other hand, introduce different structures. We now define Qtrees raised by complex LFs, in particular, negated, disjunctive, and conditional LFs.

(A) "Polar" (B) "*Wh*"

Figure 1: Qtrees for S_p = *Jo is at SuB*. $\boxed{\text{Boxed}}$ nodes are verifying.

(A) "Polar" (B) "*Wh*"

Figure 2: Qtrees for S_q = *Jo is in Boston*. $\boxed{\text{Boxed}}$ nodes are verifying.

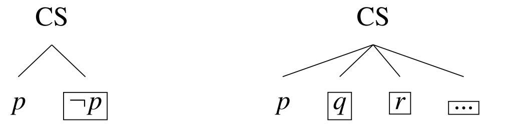
3.4. QuDs evoked by negated LFs

We take that a negated LF evokes the same kind of question as its positive counterpart, but identifies a disjoint set of true answers (verifying nodes). Given an LF X, evoking a Qtree T, a Qtree T' for $\neg X$ is obtained by retaining T's structure (nodes and edges), and "swapping" T's verifying nodes, by replacing any set of same-level verifying nodes in T by the set of non-verifying nodes at the same level in T. If the verifying nodes are all leaves, this operation simply corresponds to set complementation in the domain of leaves. This is done for $\neg S_p$ and

[7] This is a simplification; Hénot-Mortier (to appear) assumes that even simplex LFs can give rise to layered Qtrees, whose layers are ordered by some notion of granularity. But this assumption is not relevant here, because we assume p and q are same-granularity alternatives.

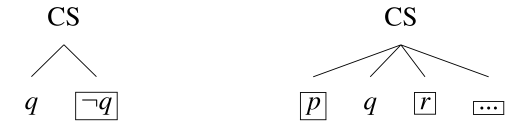
[8] This can be generalized to non-exclusive alternatives *via* Hamblin-style partitions Hamblin (1973), but leads to more complexity. See extended definition in Hénot-Mortier (to appear).

$\neg S_q$ in Figures 3 and 4.



(A) "Polar" (B) "*Wh*"

Figure 3: Qtrees for $\neg S_p =$ *Jo is not at SuB*. $\boxed{\text{Verifying nodes}}$ are all sisters of the p-node.



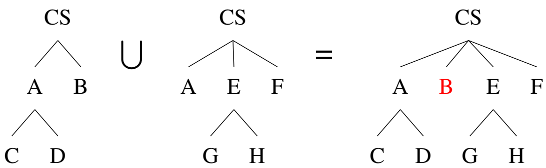
(A) "Polar" (B) "*Wh*"

Figure 4: Qtrees for $\neg S_q =$ *Jo is not in Boston*. $\boxed{\text{Verifying nodes}}$ are all sisters of the q-node.

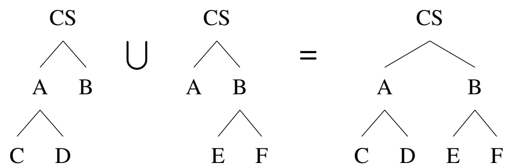
3.5. QuDs evoked by disjunctive LFs

A disjunctive Qtree should address the questions evoked by each disjunct *in parallel*, making them both at issue (Simons, 2001; Zhang, 2022). Structurally, disjunction thus returns all the well-formed unions of Qtrees evoked by its individual disjuncts. "Union" refers to that of nodes and edges; it is thus symmetric. The union of two Qtrees T and T' will be well-formed if there is no node N present in both T and T' that introduces different partitionings in T and T'. Ill- and well-formed unions of Qtrees are given in (15) and (16).

(15) An ill-formed union of Qtrees. Nodes with different labels denote different sets. By construction, E and F partition B. Therefore, A, B, E, and F are not disjoint and so do not properly partition the CS.



(16) A well-formed union of Qtrees. A and B each introduce independent partitionings, which can be fused while retaining Qtree structure.



The sets of verifying nodes attached to the two disjoined Qtrees, are also unioned. The only possible Qtree for $S_p \vee S_q$ / $S_q \vee S_p$ is given in Figure 5. It is obtained from Qtrees 1B and 2B, which have similar structures and as such can be properly unioned. Other possible unions of Qtrees are shown in Figure 6 but are ill-formed, because their leaves do not partition the CS.

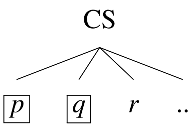
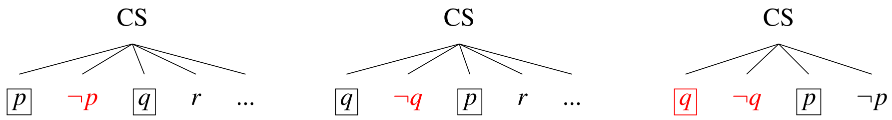


Figure 5: Only well-formed Qtree evoked by $S_p \vee S_q =$ *Jo is at SuB or in Boston*, obtained from Qtrees 1B and 2B. **This Qtree is also the only Qtree compatible with (6a)**.



(A) 1A $\vee$ 2B (B) 1B $\vee$ 2A (C) 1A $\vee$ 2A

Figure 6: Ill-formed Qtrees resulting from the union of the Qtrees in Figures 1 and 2. Red nodes are nodes that should be removed for the leaves to form a proper partition of the CS.

Likewise, the Qtrees in Figures 1B (for S_p) and 5 (for $S_p \vee S_q$), can be unioned to derive the only possible Qtree for (6a) = $S_p \vee (S_p \vee S_q)$. Because Qtree 5 has same structure as Qtree 1B, and its set of verifying nodes contains those of Qtree 1B, the result of their union is simply Qtree 5. In sum, Qtree 5 turns out to be compatible with *both* the simple disjunction $S_p \vee S_q$ and the more complex disjunction (6a). We will show in the next Section that this makes Qtree 5 "redundant" given (6a)–an in turn, predicts (6a) to be odd.

3.6. QuDs evoked by conditional LFs

We assume that conditionals typically evoke conditional questions, i.e. questions pertaining to their consequent, set in the domain(s) of the CS where the antecedent holds. Additionally, we assume some form of "neglect-zero" effect (Aloni, 2022; Flachs, 2023) in conditional Qtrees, by proposing that only the consequent of a conditional contributes verifying nodes in the resulting conditional Qtree. In particular, nodes falsifying the antecedent are not considered verifying in the resulting conditional Qtree.[9] This will be crucial to derive the absence of oddness in the case of (6c): disjoining $\neg p \rightarrow q$ with p creates a Qtree where p is at-issue (verifying), whereas in the Qtrees evoked by the simpler conditional statement $\neg p \rightarrow q$, p is "neglected" (non-verifying). In other words, disjunction has a non-redundant effect in (6c).

These intuitions are modeled by assuming that conditional Qtrees are derived by "plugging" a consequent Qtree T_C into the verifying nodes of antecedent Qtrees T_A. More concretely, for each verifying node N of T_A, N gets replaced by $N \cap T_C$. We call this entire process "intersection-and-plugging-in", noted $\cap\uparrow$. $\cap$ refers to node-wise intersection, defined in (17).[10]

(17) *Node-wise intersection.* Let T be a Qtree and N be a proposition. The node-wise intersection of N and T, noted $N \cap T$, is a tree T'

- whose nodes are the nodes of T intersected with N (excluding resulting empty nodes);
- whose verifying nodes are inherited from T (if N' is verifying in T and $N \cap N'$ is in T', then $N \cap N'$ is verifying in T');
- whose edges are inherited from T (if $N' - N''$ in T and both $N \cap N'$ and $N \cap N''$ are in T', then $N \cap N' - N \cap N''$ in T').

From a more algorithmic standpoint, node-wise intersection can be achieved by (i) intersecting all nodes of T with N; (ii) removing resulting empty nodes; (iii) removing

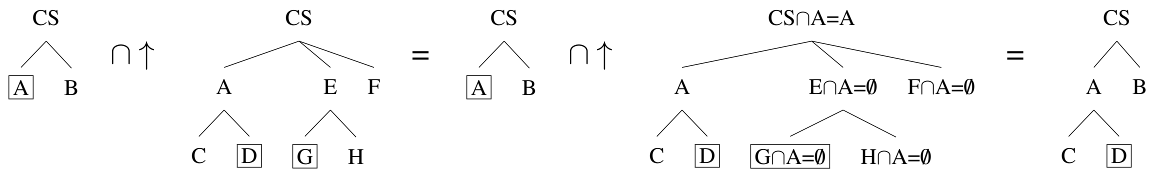
[9] This predicts that a sentence whose antecedent is falsified in the CS, evokes a Qtree without any verifying node, and thus checks the EMPTY FLAGGING condition (12)–causing oddness.

[10] This operation is structurally idle if N entails a leaf of T_C, because in this case, $N \cap T_C$ reduces to a root N, and replacing N by N in T_A is idle. However, it might still affect verifying nodes.

resulting dangling and unary edges.

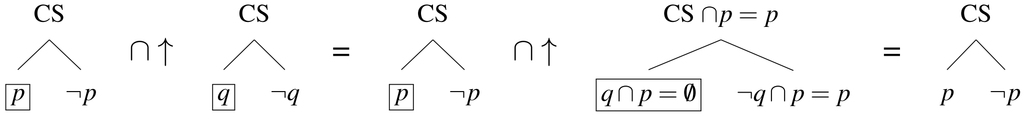
To model "neglect zero", we assume that the verifying nodes of a conditional Qtree are inherited from its input consequent Qtree only. Given that we defined conditional Qtree formation as "intersection-and-plugging-in", the verifying nodes that get retained in a conditional Qtree are those contributed by the consequent *and* compatible with the antecedent. The whole process is exemplified in (18).

(18) An example of conditional Qtree formation. First, the consequent Qtree (right operand of $\cap\uparrow$) gets node-wise intersected with node A, which is the only verifying node in the antecedent Qtree (left operand of $\cap\uparrow$). The resulting "restricted" consequent Qtree then replaces node A within the antecedent Qtree to form the conditional Qtree. In this Qtree, A is not longer verifying, but D, which was verifying in the consequent Qtree and is compatible with A, is.

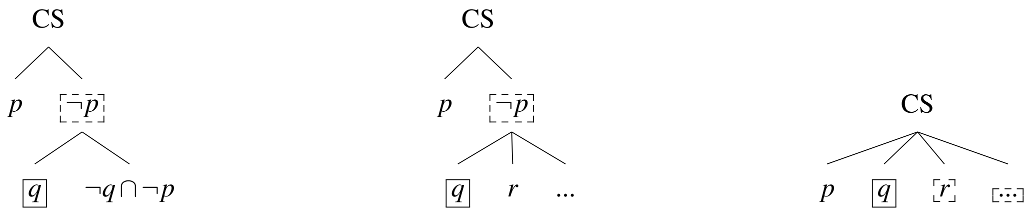


The core idea behind this operation is that conditionals introduce a hierarchy between antecedent (backgrounded) and consequent (at-issue): the consequent Qtree gets *restricted* by the antecedent Qtree. This predicts a sentence like *If Jo is at SuB then he is in Boston* = $S_p \rightarrow S_q$, in which the domain of the antecedent and that of the consequent are disjoint, to give rise to Qtrees without verifying nodes–leading to EMPTY FLAGGING (cf. (12)), and in turn, oddness. This is shown in (19).

(19) $S_p \rightarrow S_q$ (with p and q disjoint) triggers EMPTY FLAGGING and is thus odd. First, the consequent Qtree gets node-wise intersected with p which leads to a reduced Qtree with only a p-root, that is non-verifying. This Qtree replaces the p-leaf in the antecedent, leading to no change in structure, except the p-leaf is no longer verifying. Using any other Qtrees for S_p / S_q as input leads to the same kind of result.

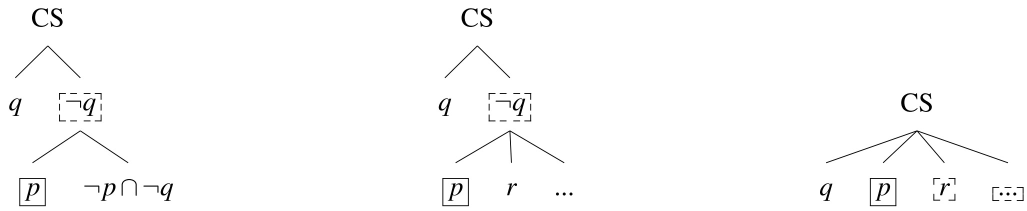


The same recipe is applied to $\neg S_p \rightarrow S_q$ in Figure 7, using Qtrees for $\neg S_p$ from Figure 3 and Qtrees for S_q from Figure 2. Figure 8 does the same for $\neg S_q \rightarrow S_p$. In these figures, nodes in dashed boxes refer to those that were verifying in the antecedent Qtree, but are no longer verifying in the conditional Qtree. They can be seen as "restrictor" nodes, which define the domain(s) of the CS in which the consequent Qtree is introduced. Nodes in solid boxes refer to the nodes that are verifying in the consequent Qtree, and are thus still verifying in the conditional Qtree.



(A) 3A → 2A. (B) 3A → 2B. (C) 3B → 2A / 2B.[11]

Figure 7: Qtrees for $\neg S_p \rightarrow S_q$ = *If Jo is not at SuB then he is in Boston*.



(A) 4A → 1A (B) 4A → 1B (C) 4B → 1A / 1B

Figure 8: Qtrees for $\neg S_q \rightarrow S_p$ = *If Jo is not in Boston then he is at SuB*; obtained *mutatis mutandis* from Figure 7

4. Capturing the target cases

Now that the compositional QuD machinery is set up, it can be used to derive implicit QuD for all the sentences in (6). Before doing this, we introduce a new version of REDUNDANCY which applies to LF-Qtree pairs.

4.1. Redundancy as a constraint on LF-Qtree pairs

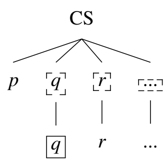
In the previous section, we already noted that the Qtree evoked by (6a) = $S_p \vee S_p \vee S_q$ (in Figure 5), was the same as the Qtree evoked by the simpler sentence $S_p \vee S_q$. We now argue that such configurations constitute violations of a specific implementation of NON-REDUNDANCY, inspired by Katzir and Singh (2014), but operating on LF-Qtree pairs, instead of just LFs. Basically, if a Qtree Q is evoked by a sentence S and also by one of the sentence's formal simplifications S' (in the sense of Katzir, 2007); then Q is deemed Q-REDUNDANT given S. This is formalized in (20a-20c).

(20) a. Q-REDUNDANCY (simplified from Hénot-Mortier, to appear). Let X be a LF and let $Qtrees(X)$ be the set of Qtrees evoked by X. For any $T \in Qtrees(X)$, T is deemed Q-REDUNDANT given X iff there exists a formal simplification of X, X', and $T' \in Qtrees(X')$, such that $T = T'$.

b. *Formal simplification.* X is a formal simplification of X if X' can be derived from X *via* a series of constituent-to-subconstituent substitutions (Katzir, 2007).

[11] This Qtree is derived *via* the "intersection-and-plugging-in" operation ($\cap\uparrow$). The Qtree obtained *before* the removal of empty nodes and dangling/unary-branching edges, is given in (i). The removal of empty nodes and dangling/unary-branching edges collapses the two q-nodes and makes the resulting node verifying; collapses the two r-nodes and makes the resulting node non-verifying; and so on for all other nodes different from the p-node.

(i)



c. *Qtree equality*. $T = T'$ iff T and T' have same structure and same verifying nodes.[12]

A *sentence S* will be deemed Q-REDUNDANT if *all* the Qtrees it evokes, are Q-REDUNDANT given *S*. This constitutes a special case of sentence oddness (as defined in (14)). (6a) is thus odd, because the only possible Qtree it evokes (in Figure 5), is evoked by $S_p \vee S_q$ which can be obtained from (6a) by substitution (regardless of bracketing). We now derive the Qtrees evoked by the infelicitous sentences (6b), (6d), and 6e), and show that they all appear problematic.

4.2. Ruling out the infelicitous (6b), (6d), and 6e)

For conciseness, we now use p and q as shorthands for the sentences denoting p and q, previously noted S_p = *Jo is at SuB* and S_q = *Jo is in Boston*. We start with (6b) $= \neg p \rightarrow (p \vee q)$, whose Qtrees are given in Figure 9, and are derived using Figures 3 (for $\neg p$), 5 (for $p \vee q$), and the combination rule for conditional Qtrees. In both cases, the output Qtree is also evoked by a simpler expressions, $\neg p \rightarrow q$ and q respectively. This stems from two features of conditional Qtree formation: (i) p in the disjunctive consequent of (6b) gets "ignored" at the Qtree level, due to the antecedent restricting the consequent Qtree to the $\neg p$-domain; and (ii) nodes falsifying the antecedent (here, p) are *not* treated as verifying. As a result, both Qtrees in Figure 9 are Q-REDUNDANT given (6b), and (6b) turns out to be a Q-REDUNDANT sentence.

(A) 3A $\rightarrow$ 5.
Same as Qtree 7B evoked by $\neg p \rightarrow q$.

(B) 3A $\rightarrow$ 5.[13]
Same as Qtree 2B evoked by q.

Figure 9: Qtrees for (6b) $= \neg p \rightarrow (p \vee q)$

Turning to (6d) $= p \vee (\neg q \rightarrow p)$, its only possible Qtree, shown in Figure 10, is obtained using Figures 1 (for p) and 8 (for $\neg q \rightarrow p$) along with the union rule for disjunctive Qtrees. Other Qtree combinations for p and $\neg q \rightarrow p$ cannot be properly disjoined (unioned), because the partitionings introduced by p and that introduced by $(\neg q \rightarrow p)$ at depth 1 differ from each other, leading to the kind of ill-formedness issue described in Figure 6. Because both input Qtrees in Figures 1 and 8 are the same, the (well-formed) output Qtree in 10 is also similar. It is therefore Q-REDUNDANT given (6d), and (6d) is deemed odd.

Finally, the only possible Qtree associated with (6e), given in Figure 11, is such that no verifying node remains after the conditional rule applies. It is thus considered ill-formed due to EMPTY FLAGGING (cf. (12)) and (6e) is in turn predicted to be odd as per (14).

[12]This is sufficient for our purposes here, but needs to be generalized to cover other cases of REDUNDANCY in this QuD-driven framework. The generalized concept of Qtree equality ("equivalence"), is based on structural equality, and equality between sets of minimal verifying paths; cf. Hénot-Mortier (to appear).

[13]This Qtree, just like the ones in Figures 7C and 8C, is derived *via* "plugging in". Before reduction, the Qtree looked like Qtree (i).

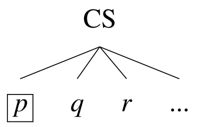


Figure 10: Only possible Qtree for (6d) $= p \vee (\neg q \rightarrow p)$, obtained from 1B $\vee$ 8C. **Same as Qtree 1B evoked by** p.

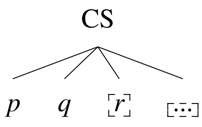
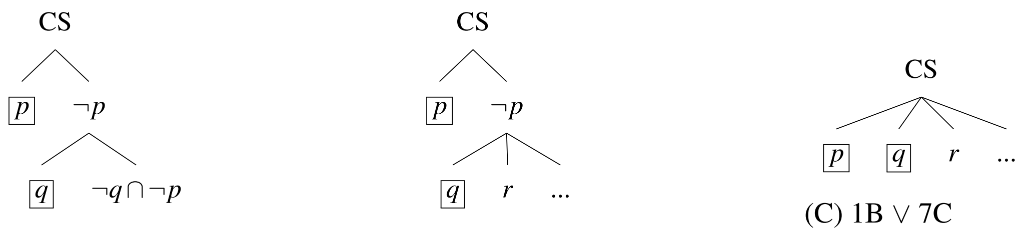


Figure 11: Only possible Qtree for (6e) $= \neg(p \vee q) \rightarrow p$, obtained from $\neg$ 5 $\rightarrow$ 1A / 1B. **EMPTY FLAGGING.**

4.3. Ruling in (6c)

We now show that Q-REDUNDANCY spares the felicitous (6c) $= p \vee (\neg p \rightarrow q)$. The relevant Qtrees, shown in Figure 12, are obtained using Figures 1 (for p) and 7 (for $\neg p \rightarrow q$), combined with the union rule for disjunctive Qtrees. Because the Qtrees evoked by p that are properly disjoinable with those evoked by $\neg p \rightarrow q$, are structurally contained in them, the Qtrees in Figure 12 are structurally similar to those from Figure 7 (corresponding to $\neg p \rightarrow q$), but, crucially, display an extra verifying p-leaf, contributed by the first disjunct of (6c).



(A) 1A $\vee$ 7A

(B) 1A $\vee$ 7B

(C) 1B $\vee$ 7C
Same as Qtree 5 evoked by $p \vee q$

Figure 12: Qtrees for (6c) $= p \vee (\neg p \rightarrow q)$

This extra leaf guarantees that Qtrees 12A and 12B *cannot* be evoked by a simplification of sentence (6c). To show this, let us review the nine possible simplifications of (6c), that we divide into three groups: p, q, $\neg p$, $p \vee q$, $p \vee p$, $p \vee \neg p$ (group 1); $\neg p \rightarrow q$ (group 2); $p \rightarrow q$, $p \vee (p \rightarrow q)$ (group 3). Starting with the simplifications in group 1, we notice that their corresponding Qtrees are always of depth 1, because they correspond to Figures 1, 2, or 5, or to slight variants thereof, where only the verifying nodes change (due to negation, or union). Such Qtrees are thus obviously distinct from Qtrees 12A and 12B, which have depth 2. So the simplifications in group 1 cannot make Qtrees 12A and 12B Q-REDUNDANT given (6c). Regarding the $\neg p \rightarrow q$ simplification (group 2), it was shown to gives rise to the Qtrees in Figure 7. Crucially, these Qtrees do not count the p-leaf as verifying–unlike those in Figures 12A and 12B. So they cannot be used to justify Q-REDUNDANCY. As for the simplifications in group 3, we showed in (19) that the Qtree for $p \rightarrow q$ had only one layer, and moreover displayed EMPTY FLAGGING. Again, this is clearly distinct from Qtrees 12A and 12B. Disjunction with a Qtree for p, to form a Qtree for $p \vee (p \rightarrow q)$, does not help either: the resulting Qtree gains one verifying p-leaf, but retains one single layer, making it distinct from Qtrees 12A and 12B. Therefore, no simplification of (6c) gives rise to Qtrees like 12A and 12B, and, as a result, such Qtrees are *not* Q-REDUNDANT given (6c). This means that (6b) should not be deemed odd on the basis of Q-REDUNDANCY, in line with intuitions.

To sum up, we accounted for the pattern in (6a-6e) by appealing to a model of compositional QuDs assigning disjunctions and conditionals different "inquisitive" contributions, and by redefining REDUNDANCY on pairs formed by sentences and their possible implicit QuD. In the next Section, we discuss how this new model relates to earlier similar approaches and to the MAXIM OF MANNER.

5. Taking stock

5.1. Comparison with similar approaches

Two recent approaches to Hurford phenomena (Hurford, 1974; Marty and Romoli, 2022; Mandelkern and Romoli, 2018 i.a.) exploit ideas similar to those presented here. Ippolito (2019) and Zhang (2022) in particular, both use structures very close to Qtrees, and propose that oddness can arise from specific configurations in these structures. In Zhang's framework, this takes the form of a distinctness constraint between answers to the same question; in Ippolito's, this is cashed out in terms of matching specificity between disjoined alternatives. In both cases, the constraints posited are mostly structural and only target *the* implicit alternative set/QuD evoked by any given sentence. Even if the constraints posited are very sensible, they are not directly motivated by familiar, competition-based, pragmatic principles. As we will further show in the next Section, our implementation of Q-REDUNDANCY fills this gap and provides a perhaps more explanatory account of QuD-driven cases of oddness. Zhang (to appear) proposes a different account that also goes in this direction. Additionally, previous accounts were mostly focused on disjunctions, or more generally, configurations where two subconstituents could be taken to answer the same QuD.[14] But it remained unclear how the overarching question was derived in each case, and whether it should be in the first place. Our system also fills this gap, in providing a set of recipes to compositionally derive implicit QuDs, instead of taking them for granted. Together with Q-REDUNDANCY, this machinery captured the target contrasts.

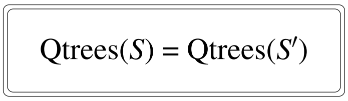
Our approach also differs from Inquisitive Semantics (Mascarenhas, 2008; Ciardelli, 2009; Groenendijk and Roelofsen, 2009; Ciardelli and Roelofsen, 2017; Ciardelli et al., 2018; Zhang, to appear), whereby assertive sentences and questions are fundamentally the same kinds of objects. Under the current view, sentences retain a "semantic", truth-conditional component, and evoke Qtrees at a distinct "inquisitive" level. While the semantic module is sensitive to truth conditions, the pragmatic module is assumed to be sensitive to the interaction between form, meaning, and inquisitive content. So our approach may be seen as an "inquisitive pragmatics". The difference between the two frameworks is particularly visible when it comes to negation: in Inquisitive Semantics, negation removes structure by collecting and collapsing all information states incompatible with those of the prejacent. In our framework, negation retains structure, and simply flips verifying nodes.

5.2. An "inquisitive" Maxim of Manner?

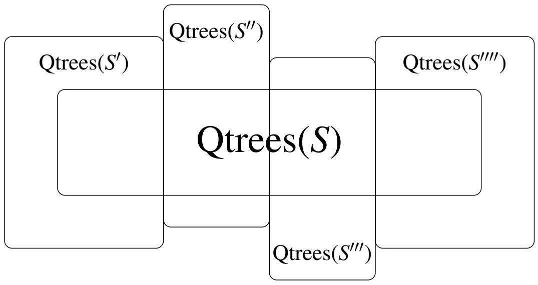
Earlier definitions of REDUNDANCY were linking this notion to Grice's MAXIM OF MANNER (MANNER for short), which roughly states that if two sentences have the same logical contribution, then the more concise one should be preferred. Is Q-REDUNDANCY a proper extension of MANNER at the inquisitive level? At first blush, not exactly. In particular, Q-REDUNDANCY does not state that, for a sentence S to be Q-REDUNDANT, *all* Qtrees compatible with S should be identified (*via* a bijective operation) to *all* Qtrees compatible with some simplification of S. This perhaps would have been the most intuitive extension of MANNER as the QuD level,

[14] Ippolito (2019) in fact argues that this comprises Sobel Sequences and sequences of superlatives.

and is depicted in Figure 13A. Instead, Q-REDUNDANCY states that for a sentence S to be Q-REDUNDANT, *each* Qtree compatible with S should be identified with *some* Qtree generated by *some* simplification of S. This configuration, depicted in Figure 13B, is significantly less strong, i.e. predicts *more* sentences to be redundant. For instance, we concluded that (6b) was Q-REDUNDANT because each of its Qtrees could be identified with Qtrees evoked by *distinct* simplifications of (6b)–namely q and $\neg p \rightarrow q$. Moreover, q and $\neg p \rightarrow q$ themselves led to Qtrees that were *not* compatible with (6b).



(A) What a more "intuitive" version of Q-REDUNDANCY could have been (S' refers to some simplification of S).



(B) What it takes for S to be Q-REDUNDANT (S', S'', S''', and S'''' refer to simplifications of S).

Figure 13: Comparing Q-REDUNDANCY to a more "intuitive" extension of MANNER to the QuD domain.

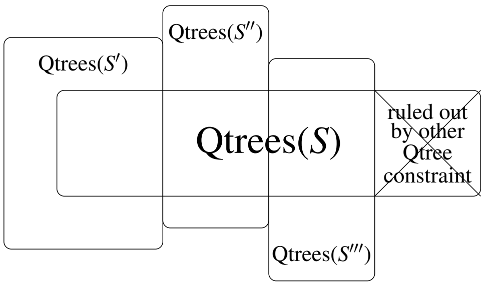


Figure 14: What it means for S to be odd partly due to Q-REDUNDANCY, partly due to other Qtree well-formedness constraints, e.g. EMPTY FLAGGING or Q-RELEVANCE.

Our definition of Q-REDUNDANCY also leaves space for other Qtree well-formedness constraints to contribute to a sentence's oddness. EMPTY FLAGGING (cf. (12)) is one such constraint. Following Hénot-Mortier (to appear), we can also assume that RELEVANCE (in the form of Q-RELEVANCE) filters out Qtrees.[15] A sentence S may then be deemed odd because *some* Qtrees compatible with S are Q-REDUNDANT given S, and the other Qtrees compatible with S are ruled-out by Q-RELEVANCE or the EMPTY FLAGGING condition. This mixed-oddness profile is schematized in Figure 14.

If Q-REDUNDANCY at the sentential level is not an intuitive extension of MANNER, Q-REDUNDANCY defined on LF-Qtree pairs (cf. 20a), seems to be. To see this, one must define

[15]Roughly, this constraint bans Qtrees whose derivation involves "shrinking" verifying nodes. It is not relevant to our current data, because the propositions we consider here are mutually exclusive, s.t. each time a verifying node gets intersected, it is not "shrunk", but instead reduced to the empty set.

the simplification of a LF-Qtree pair (S,T), where T is a Qtree evoked by S, as a pair (S', T') where S' is a formal simplification of S in the sense of (20b), and T' is a Qtree evoked by S'. Additionally, one must define equivalence between LF-Qtree pairs as equivalence between their Qtree-component. This yields a definition of Q-RELEVANCE-as-MANNER, given in (21) that is set as a two-dimensional optimization problem on both LFs (which calibrate conciseness, and, indirectly, informativeness) and Qtrees (which calibrate informativeness).

(21)
a. *LF-Qtree pair.* (S,T) is a well-formed LF-Qtree pair iff S evokes T.
b. Q-REDUNDANCY *as* MANNER. If (S,T) and (S', T') are two LF-Qtree pairs that are equivalent to each other, then the most concise of the two should be preferred.
c. *Conciseness of a LF-Qtree pair.* If (S,T) and (S', T') are two LF-Qtree pairs, (S', T') is more concise than (S,T) iff S' is a formal simplification of S as per (20b).
d. *Equivalence between LF-Qtree pairs.* If (S,T) and (S', T') are two LF-Qtree pairs, (S', T') is equivalent to (S,T) iff $T = T'$.[16]

6. Exploring elaborations of the target sentences

Now that we have better situated Q-REDUNDANCY as part of the existing work on pragmatic oddness, we explore its predictions beyond the data in (6).

6.1. Effect of disjunct ordering in the felicitous case

First, let us briefly come back to the pair (4b)-(2b), repeated in (22). The two sentences only differ in the ordering of their disjuncts. We predict both to escape Q-REDUNDANCY, and more generally oddness. This, again, is because the introduction of p as a disjunct makes it at-issue, while it would be "neglected" if the sentence were simplified to its conditional disjunct. (22b) however, sounds worse than (22a), and even more so if the conditional disjunct did not feature inversion.

(22)
a. Either Jo is at SuB or if he is not at SuB then he is in Boston. $p \vee (\neg p \rightarrow q)$
b. ? Either Jo is in Boston if not at SuB, or he is at SuB. $(\neg p \rightarrow q) \vee p$

We suggest this contrast is caused by an independent, incremental constraint targeting Qtree derivations. As observed in Section 4.3, the Qtrees evoked by p that are properly disjoinable with those evoked by $\neg p \rightarrow q$, are structurally contained in them, i.e. are less "specific". Therefore, the disjunction in (22a) takes two Qtrees of increasing specificity as input (from left to right), while the disjunction in (22b) takes two Qtrees of decreasing specificity. And it is reasonable to think that the latter order should be preferred. This is supported by the sequences of questions in (23): it appears more natural to ask a less specific question (e.g., about countries), before a more specific one (e.g., about cities), than the other way around. The latter ordering in fact seems to suggest that the more specific question would not allow to infer the exact answer to the less specific one.

(23)
a. In which country does Jo live? And in which city?
b. ? In which city does Jo live? And in which country?

[16] This is simplified for the purposes of this paper: equality could be replaced by any more elaborate relation between Qtrees; cf. footnote 12.

Assuming that the country-level question is structurally contained in the city-level question (Hénot-Mortier, to appear),[17] we derive the following generalization, which applies to both (22) and (23).[18]

(24) *Incremental Qtree Containment.* Let X and Y be LFs, and $\circ$ be a binary operator. If $X \circ Y$ and $Y \circ X$ have same meaning and same evoked Qtrees, and if $\forall T \in Qtrees(X \circ Y)$, T is obtained from $T' \in Qtrees(X)$ and $T'' \in Qtrees(Y)$ with $T' \subset T''$, then $X \circ Y$ should be preferred over $Y \circ X$.

6.2. Double or-to-if

We now briefly discuss more complex variants of (1), derived *via* two applications of the *or-to-if* tautology. (25a) sounds clearly redundant, while (25b) and (25b) sounds contradictory. (25d) and (25d) appear very tough to make sense of.

(25)
a. # If Jo isn't at SuB then, if he isn't at SuB then he is in Boston. $\neg p \to (\neg p \to q)$
b. # If Jo isn't at SuB then, if he isn't in Boston then he is at SuB. $\neg p \to (\neg q \to p)$
c. ?? If it's not that Jo is in Boston if not at SuB, then he isn't at SuB. $\neg(\neg p \to q) \to p$
d. # If it's not that Jo is at SuB if not in Boston, then Jo is at SuB. $\neg(\neg q \to p) \to p$

The model laid out in this paper predicts all the sentences in (25) to be odd, for different reasons. (25a) turns out Q-REDUNDANT, because all the Qtrees it gives rise to are the same as the ones generated by its consequent $\neg p \to q$ (cf. Figure 7). Both (25b) and (25d) generate Qtrees that invariably display EMPTY FLAGGING. In both cases this is because the "restrictor" nodes in which the consequent Qtree is plugged, are sets of $\neg p$-worlds, and p is precisely what the consequent would have contributed as verifying node. Lastly, (25c) represents a mixed case of oddness: most of the Qtrees it evokes display EMPTY FLAGGING, and one tree turns out Q-REDUNDANT due to the p-simplification of the sentence.

7. Conclusion and outlook

We presented novel data based on logical variants of $p \vee p \vee q$, that appeared challenging to account for while retaining classic results on other families of odd sentences, in particular Hurford Disjunctions and Conditionals. We proposed an account of this paradigm in the QuD-framework, based on the intuitive idea that sentences have to be good answer to good questions (Katzir and Singh, 2015), and that disjunctions and conditionals have distinct "inquisitive" contributions. In that framework, sentences compositionally evoke QuD-trees, and the interaction between sentences and evoked QuDs is evaluated to determine pragmatic oddness. This was shown to differ from Inquisitive Semantics in the sense that a sentence's "inquisitive" contribution is not meant to replace its truth-conditional meaning. Hénot-Mortier (to appear) successfully extends this framework to capture Hurford Disjunctions and Conditionals (Hurford, 1974; Mandelkern and Romoli, 2018). Beyond the data discussed, this framework suggests that oddness come in different "flavors" and that sentences may be odd due to a conspiration of these various factors. Future work will determine if these predictions indeed hold.

[17] More specifically, a country-question will take the form of a partition of the CS according to countries, while a city-question may take the form of a two-level Qtree, whose first layer is a by-country partition, and whose second layer is a by-city partition, properly connected to the first layer.

[18] (23) may seem reminiscent of Hurford Disjunctions (Hurford, 1974). It is worth noting that Hénot-Mortier (to appear) predicts Hurford Disjunctions to be bad in both orders, independently of the constraint in (24).

References

Aloni, M. (2022). Logic and conversation: The case of free choice. *Semantics and Pragmatics 15*(5), 1–60.

Büring, D. (2003). On D-Trees, Beans, and B-Accents. *Linguistics & Philosophy 26*(5), 511–545.

Ciardelli, I. (2009). Inquisitive semantics and intermediate logics. Master's thesis, University of Amsterdam.

Ciardelli, I., J. Groenendijk, and F. Roelofsen (2018). *Inquisitive Semantics*. Oxford University Press.

Ciardelli, I. and F. Roelofsen (2017). Hurford's constraint, the semantics of disjunction, and the nature of alternatives. *Natural Language Semantics 25*(3), 199–222.

Flachs, B. L. (2023). Neglect-Zero Effects on Indicative Conditionals: Extending BSML and BiUS with an implication. Master's thesis, University of Amsterdam. Available at `https://eprints.illc.uva.nl/id/eprint/2235/`.

Grice, H. P. (1975). Logic and Conversation. In D. Davidson (Ed.), *The logic of grammar*, pp. 64–75. Dickenson Pub. Co.

Groenendijk, J. and F. Roelofsen (2009). Inquisitive Semantics and Pragmatics. In *Workshop on Language, Communication and Rational Agency*.

Hamblin, C. L. (1973). Questions in Montague English. *Foundations of Language 10*(1), 41–53.

Haslinger, N. (2023). *Pragmatic constraints on imprecision and homogeneity*. Ph. D. thesis, Georg-August-Universität Göttingen.

Hurford, J. R. (1974). Exclusive or inclusive disjunction. *Foundations of language 11*(3), 409–411.

Hénot-Mortier, A. (to appear). "One tool to rule them all"? An integrated model of the QuD for Hurford Sentences. In *Proceedings of the 29th Sinn und Bedeutung*.

Ippolito, M. (2019). Varieties of Sobel sequences. *Linguistics and Philosophy 43*(6), 633–671.

Kalomoiros, A. (2024). An approach to Hurford Conditionals. *Semantics and Linguistic Theory*, 724–743.

Katzir, R. (2007). Structurally-defined alternatives. *Linguistics and Philosophy 30*(6), 669–690.

Katzir, R. and R. Singh (2014). Hurford disjunctions: embedded exhaustification and structural economy. In *Proceedings of Sinn und Bedeutung*, Volume 18, pp. 201–216.

Katzir, R. and R. Singh (2015). Economy of structure and information: Oddness, questions, and answers. In E. Csipak and H. Zeijlstra (Eds.), *Proceedings of Sinn und Bedeutung 19*, pp. 322–339. Georg August Universität Göttingen.

Mandelkern, M. and J. Romoli (2018). Hurford Conditionals. *Journal of Semantics 35*(2), 357–367.

Marty, P. and J. Romoli (2022). Varieties of Hurford disjunctions. *Semantics and Pragmatics 15*(3), 1–25.

Mascarenhas, S. (2008). Inquisitive Semantics and Logic. Master's thesis, University of Amsterdam.

Mayr, C. and J. Romoli (2016). A puzzle for theories of redundancy: Exhaustification, incrementality, and the notion of local context. *Semantics and Pragmatics 9*(7).

Onea, E. (2016). *Potential questions at the semantics-pragmatics interface*. Brill.

Onea, E. (2019). *Underneath Rhetorical Relations: the Case of Result*, pp. 194–250. BRILL.

Riester, A. (2019). *Constructing QUD Trees*, pp. 164–193. BRILL.

Roberts, C. (1996). Information structure in discourse: Towards an integrated formal theory of pragmatics. *Semantics and Pragmatics 5*.

Schlenker, P. (2009). Local Contexts. *Semantics and Pragmatics 2*.

Simons, M. (2001). Disjunction and Alternativeness. *Linguistics and Philosophy 24*(5), 597–619.

Singh, R. (2008). On the interpretation of disjunction: asymmetric, incremental, and eager for inconsistency. *Linguistics and Philosophy 31*(2), 245–260.

Stalnaker, R. (1974). Pragmatic Presuppositions. In *Context and Content*, pp. 47–62. Oxford University Press.

Stalnaker, R. (1999). Pragmatic Presuppositions. In R. Stalnaker (Ed.), *Context and Content: Essays on Intentionality in Speech and Thought*, pp. 47–62. Oxford University Press UK.

Van Kuppevelt, J. (1995). Main structure and side structure in discourse. *Linguistics 33*(4), 809–833.

Zhang, Y. (2022). New perspectives on Inquisitive Semantics.

Zhang, Y. (to appear). QUD-mediated redundancy. In *Proceedings of the 29th Sinn und Bedeutung*.