Redundancy under Discussion[1]

Adèle HÉNOT-MORTIER — *Massachusetts Institute of Technology*

Abstract. This paper presents novel data taking the form of a family of sentences, all derived from a redundant structure ($p \vee p \vee q$) *via* the *or*-to-*if* tautology and core properties of disjunction (commutativity, associativity). In a way similar to Hurford Disjunctions vs. Conditionals, the sentences at stake exhibit differing degrees of pragmatic oddness, which represents a challenge for standard theories of redundancy. Building on a recent QuD-informed model of pragmatic oddness, we propose a solution covering almost all the cases at stake, and discuss how the remaining problematic cases might be solved assuming an incremental view on Redundancy.

Keywords: redundancy, relevance, question under discussion

1. Introduction

Disjunctive sentences featuring a repeated disjunct, such as (1a), (1b) and (1c), appear sharply infelicitous. From a logical point of view, these three sentences are related to each other *via* applications of $\vee$-commutativity and $\vee$-associativity.[2] From a pragmatic point of view, these sentences can be argued to be odd because they are all contextually equivalent to their complex disjunct, $p \vee q$ or $q \vee p$.

(1) a. # Either Ido is at SuB, or else he is at SuB or in Cambridge. $p \vee (p \vee q)$
b. # Either Ido is at SuB, or else he is in Cambridge or at SuB. $p \vee (q \vee p)$
c. # Either Ido is at SuB or in Cambridge, or else he is at SuB. $(p \vee q) \vee p$
d. # Either Ido is in Cambridge or at SuB, or else he is at SuB. $(q \vee p) \vee p$

(2-5) below show variants of (1a-1d) obtained *via* the *or*-to-*if* tautology. In each pair of sentences, the a. instances are derived by applying the tautology to the outer disjunction, while the the b. instances are derived by applying the tautology to the inner disjunction.[3]. Surprisingly, those variants exhibit different degrees of oddness: (2b) and (4b) are the only two variants which seem to escape infelicity. 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.

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

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

[1]Many thanks to Amir Anvari, Athulya Aravind, Danny Fox, Nina Haslinger, and Viola Schmitt for their precious input on that project. All mistakes are my own.

[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. That is why we omit them in this introduction.

(4) Derived from (1c):

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

b. ? Either Ido is in Cambridge 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 Ido is in Cambridge or at SuB, then he is at SuB. $\neg(q \vee p) \rightarrow p$

b. # Either Ido is at SuB if not in Cambridge, 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 (2024), we propose that this descriptive generalization follows from the idea that oddness arises when sentences cannot evoke any well-formed accommodated Questions under Discussion (Van Kuppevelt, 1995; Roberts, 1996); and that disjunctions and conditionals have different inquisitive contributions, in the sense that disjunctions force their disjuncts to raise *parallel* QuDs, while conditionals "stack" the QuDs of their antecedent and consequent. More generally, this predicts "connectivity effects" in disjunctions-of-conditionals, in that the antecedents and consequents respectively have to address similar QuDs; and no such effect in conditionals-of-disjunctions, in that disjuncts coming from the antecedent and consequent may be inquisitively unrelated. Assuming that $\vee$-commutativity does not affect oddness (in line with the data presented here), we now focus on sentences (1a), (2a), (2b), (3b), and (4a), repeated in (6) below.

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

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

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

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

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

The rest of this paper is structured as follows. In the next Section, we briefly review why some of the sentences in (6) are problematic for existing accounts of redundancy. In Section 3 we show how the model of accommodated QuDs laid out in Hénot-Mortier (2024) captures the target asymmetries. In Section 7 we explore further predictions of the model in elaborations of the sentences in (6). Section 8 concludes.

2. Previous accounts

In this section we briefly present three existing accounts of redundant structures: Local Redundancy Checking, Non-triviality, and Super-Redundancy. We show how the they straightforwardly account for the double disjunction case (6a), but fall short in explaining the contrast between the felicitous (6c) vs. (6b), (6d), and (6e).

2.1. Local Redundancy Checking

Katzir and Singh (2014) propose that the semantic computation evaluates, at certain nodes, whether the semantic 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 it is contextually equivalent to its complex disjunct $(p \vee q)$. But, assuming conditionals denote material implications, this also predicts (6b-6e) to be deviant, meaning, the felicity of (6c) is not derived.

The issue persists if we adopt a non-material analysis of conditionals. Under this assumption, the whole conditional will never be contextually equivalent to its antecedent or consequent regardless of what they denote. Candidate simplifications for (6b-6c) are then $\neg p \rightarrow q$, and for (6d-6e), $\neg q \rightarrow p$. Under a strict analysis of conditionals, (6b-6e) can be shown to be equivalent to their respective simplifications and thus devious, while (6c-6d) are not. But this is not the expected contrast. Under a variably strict analysis, all the cases but (6b) are predicted into felicity. Again, it is not the expected contrast.

2.2. Non-triviality

Another line of work, building on the notion of local contexts (Schlenker, 2009), associates redundancy with triviality (Stalnaker, 1999): a sentence should not contain a part that is trivially true or false when evaluated against its local context (Mayr and Romoli, 2016).

(8) *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 the local context of the second disjunct is the negation of the first, and the local context of a consequent is the antecedent, (6a-6e) are all predicted to be deviant under a material implication analysis: in all cases, the second occurrence of p gets interpreted in a local context entailing $\neg p$.

nonmaterial conditionals TODO

2.3. Super-Redundancy

Kalomoiros (2024) proposes an adaptation of the REDUNDANCY view, based on the novel notion of SUPER-REDUNDANCY. Roughly, a sentence is super-redundant if there is no way of strengthening one of its subconstituents that that would make the resulting sentence non-redundant.

(9) *Super-redundancy.* A sentence S is infelicitous if it contains a subconstituent C s.t. $(S)^-_C$ is defined and for all D, $(S)^-_C \equiv S_{Str(C,D)}$.

Roughly, $(S)^-_C$ in the above definition designates S where C got deleted, while $Str(C,D)$ 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 $Str(O(\alpha,\beta),D) = O(Str(\alpha,D), Str(\beta,D))$. $S_{Str(C,D)}$ designates S where C is replaced by $Str(C,D)$

This account elegantly solved the puzzle posed by Hurford Disjunctions (Hurford, 1974) and Conditionals (Mandelkern and Romoli, 2018), by predicting that, when deriving a logical expression from another, double negation introduction influences redundancy. However, this account fails to predict any contrast for (6a-6e), precisely because those sentences are isomorphic *without* any appeal to double negation introduction. More specifically, if we set $C = p$ 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 $p \vee q \equiv \neg p \rightarrow q \equiv \neg q \rightarrow p \equiv (S)^{-}_{C}$, regardless of what D is. In other words, (6a-6e) are all super-redundant.

3. A QuD-based account

Building on the model proposed for Hurford Sentences by Hénot-Mortier (2024) and previous work by Zhang (2024); Haslinger (2023), we assume that Logical Forms evoke accommodated QuDs in the form of parse trees of the Context Set (Qtrees); and that the Qtrees evoked by a complex LF are derived from the Qtrees evoked by the LF's constitutive parts in a compositional way. This compositional machinery is supplemented by Qtree-LF well-formedness constraints (RELEVANCE, REDUNDANCY), which rule-out certain derived Qtrees. As a result, specific LFs can end up with no accommodated Qtrees, and are therefore deemed odd.

For the present case study, we will need two key ingredients borrowed from Hénot-Mortier (2024): that disjunctions and conditionals give rise to distinct kinds of Qtrees, and that derived Qtrees-LFs pairs are subject to a REDUNDANCY constraint, stating that a Qtree evoked by a LF is suboptimal if it is also evoked by a simplification of this LF. The interaction between these two ingredients predicts that the Qtrees evoked by (6a) are redundant given $p \vee q$, those evoked by (6b) are redundant given $p \vee q / \neg p \rightarrow q$, the one evoked by (6d) is redundant given p, and the one evoked by (6e) is answerless. In the following two sections, we sketch the account presented in more detail in Hénot-Mortier (2024).

3.1. General picture

We model QuDs as parse trees of the Context Set (Stalnaker, 1974), which can also be seen as nested partitions. The definition of such trees ("Qtrees") is given in (10).

(10) *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.

The nodes of such trees can be assigned the following interpretation. The root denotes a tautology over the CS, and any other node, 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 answers. 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.[4] When evoking a Qtree, a given LFs is assumed to "flag" specific nodes on the tree as maximal true answers. These nodes, that we dub *verifying nodes*,

[4]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).

are typically the leaves of the Qtree which are subsets (i.e. entail) the proposition denoted by the LF. Those verifying nodes, just like Qtree structure, 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. This is repeated in (11). 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. This is repeated in (12) and (13).

(11) *Vacuous labeling of verifying nodes.* If a sentence S evokes a Qtree T but does not flag any node as verifying on T, then T is deemed odd given S.

(12) *Oddness of a Qtree, given a sentence.* If a sentence S evokes a Qtree T and the pair (S, T) is REDUNDANT (tbd) or induces a vacuous labeling of verifying nodes, then T is deemed odd given S.

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

Before defining Qtrees for simplex LFs, let us clarify that all Qtrees will be defined and derived *modulo* a reduction function, which, given a tree T removes any empty nodes and trivial edges from T.

(14) *Qtree reduction.*
If T is a tree whose nodes are sets, and endowed with a set of distinguished (e.g. verifying) nodes, a reduction of T is obtained by:
- Removing all empty nodes (and resulting dangling edges) from T;
- Removing all trivial links from T, in the following way:
  - if N has N' as only child, and neither N nor N' are verifying, replace the edge $N - N'$ by N;
  - if N has N' as only child, and either N or N' is verifying, replace the edge $N - N'$ by N, where N is labeled as verifying.

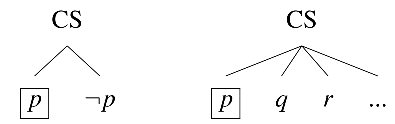
3.2. QuDs evoked by simplex LFs

Now reasoning *modulo* reduction, we assume that a simplex LF denoting a proposition p can give rise to two types of Qtree:[5] 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. Moreover, verifying nodes are defined on such trees as simply the p-leaf (if present). This predicts a simplex LF denoting a tautology to only be compatible with one Qtree, namely the Qtree whose root and unique (verifying) leaf is the whole CS. This also predicts a simplex LF denoting a contradiction to be compatible with Qtrees whose leaves are non-contradictory alternatives to the prejacent proposition, and whose set of verifying nodes is empty. In other words, contradictions cannot answer any suitable question and as such should be odd (cf. condition (11)).

Looking back at (6a-6e), where S_p = *Ido is at SuB* denotes p and S_q = *Ido is in Cambridge* denotes q, it is reasonable to think S_p and S_q are exclusive mutual alternatives. Other similar

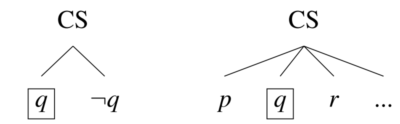
[5] This is a simplification; Hénot-Mortier (2024) assumes that even simplex LFs can give rise to layered Qtrees, whose layers are ordered by some notion of granularity. But this assumption is not relevant here, because we assume p and q are same-granularity alternatives.

alternatives may be S_r = *Ido is in Paris*, S_s = *Ido is in Chicago* etc. As a result, the Qtrees compatible with S_p and S_q are given in Figures 1 and 2. From those Figures it can be noticed that the "*wh*" Qtrees raised by S_p and S_q have similar structures (ignoring verifying nodes) – while the corresponding "polar" Qtrees do not.



(a) "Polar" (b) "*Wh*"

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

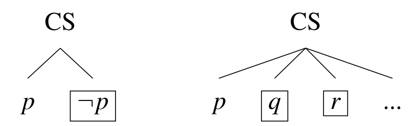


(a) "Polar" (b) "*Wh*"

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

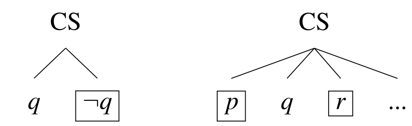
3.3. QuDs evoked by complex LFs

We now proceed to define Qtrees raised by complex LFs, in particular, negated, disjunctive, and conditional LFs. The negative case is straightforward: for each verifying node, find its same level, non-verifying siblings/cousins and swap the "verifying" status between the two groups. This is done for S_p and S_q in Figures 3 and 4



(a) "Polar" (b) "*Wh*"

Figure 3: Qtrees for $\neg S_p$ = *Ido is not at SuB*. $\boxed{\text{Boxed}}$ nodes are verifying.



(a) "Polar" (b) "*Wh*"

Figure 4: Qtrees for $\neg S_q$ = *Ido is not in Cambridge*. $\boxed{\text{Boxed}}$ nodes are verifying.

Disjunctive and conditional Qtree involve a heavier machinery, whose complete definitions and predictions can be found in Hénot-Mortier (2024). Here it is enough to say that disjunction returns all the well-formed unions of Qtrees evoked by its individual disjuncts. A 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'. The set of verifying nodes attached to the two disjoined Qtrees, are also unioned. A disjunctive Qtree is thus a Qtree adressing the questions evoked by each disjunct *in parallel*, making both disjuncts at issue. It is a symmetric operation: the order of the disjuncts does not influence the output. 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, as previously observed, have similar structures. Other possible unions of Qtrees are shown in Figure 6 but appear ill-formed, because the leaves of such Qtrees do not properly partition the CS. Following a similar line of reasoning, one can use the Qtrees in Figures 1b (for p) and 5 (for $p \vee q$), to derive the only possible Qtree for (6a) = $p \vee (p \vee q)$. Because Qtrees 1b and 5 have same structure, and are s.t. the former Qtree's set of verifying nodes is a subset of the latter Qtree's set of verifying nodes, Qtree union simply returns Qtree 5 as output. So Qtree 5 turns out to be compatible with both $p \vee q$ and (6a).

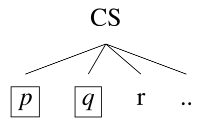
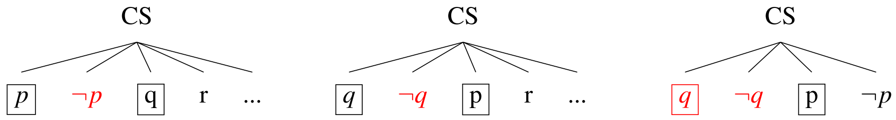


Figure 5: Only well-formed Qtree evoked by $S_p \vee S_q =$ *Ido is at SuB or in Cambridge*, obtained from 1b $\vee$ 2b. **This Qtree is also the only Qtree compatible with (6a)**.



(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.

Let us now turn to the conditional case. Following Hénot-Mortier (2024), we assume that the “inquisitive” contribution of $\rightarrow$ is *not* material, meaning, a conditional Qtree is not derived by disjoining the negation of its antecedent Qtrees, with its consequent Qtrees. Rather, we propose that conditional Qtrees are derived by “plugging” consequent Qtree into the verifying nodes of antecedent Qtrees – where “plugging” technically refers to Qtree-node intersection (supplemented by reduction as defined in (14)). Note that in certain cases, this operation is vacuous; for instance, when each verifying leaves in the antecedent Qtree entails a specific leaf of the consequent Qtree. The verifying nodes of an output conditional Qtrees are inherited from its input *consequent* Qtree; meaning, verifying nodes from the antecedent Qtree are *disregarded*. This whole operation is schematized in Figure 7.

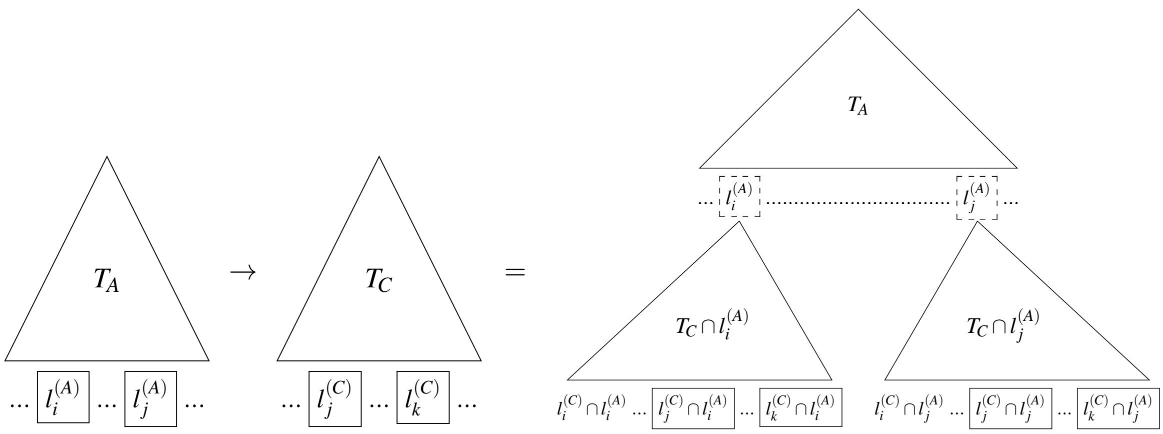
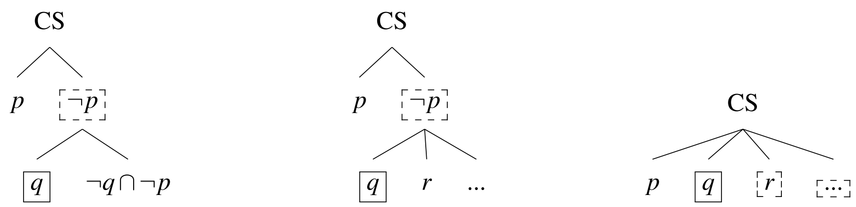


Figure 7: Schema of the derivation of a conditional Qtree. Nodes in dashed boxes refer to the nodes that were verifying in the input antecedent Qtree, but are no longer verifying in the output conditional Qtree. Nodes in solid boxes refer to the nodes that were verifying in the input consequent Qtree, and are thus still verifying in the output conditional Qtree.

The core idea behind this operation is that conditionals do not make antecedent and consequent QuDs at issue at the same time; rather, they introduce a hierarchy between these two objects, by raising the consequent QuD only in the cells of the CS (as defined by the antecedent QuD), where the antecedent holds. Yet another way to phrase this is by saying that, through the process of Qtree-conditionalization, the consequent Qtree gets *restricted* by the antecedent Qtree. This is done for $\neg S_p \rightarrow S_q$ in Figure 8, using Qtrees for $\neg S_p$ from Figure 3 and Qtrees for S_q from Figure 2. Figure 9, does the same for $\neg S_q \rightarrow S_p$, just swapping the roles of p and q.

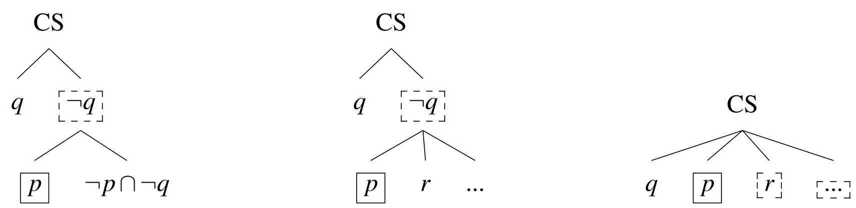


(a) "Polar" antecedent (3a) and "polar" consequent (2a).

(b) "Polar" antecedent (3a) and "*wh*" consequent (2b).

(c) "*Wh*" antecedent (3b) and "polar"/"*wh*" consequent (2a)/(2b).[6]

Figure 8: Qtrees for $\neg S_p \rightarrow S_q =$ *If Ido is not at SuB then he is in Cambridge*. Nodes in dashed boxes refer to the nodes that were verifying in the input antecedent Qtree, but are no longer verifying in the output conditional Qtree. Nodes in $\boxed{\text{solid boxes}}$ refer to the nodes that were verifying in the input consequent Qtree, and are thus still verifying in the output conditional Qtree.



(a) "Polar" antecedent (4a) and "polar" consequent (1a).

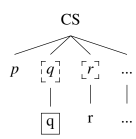
(b) "Polar" antecedent (4a) and "*wh*" consequent (1b).

(c) "*Wh*" antecedent (4b) and "polar"/"*wh*" consequent (1a)/(1b).

Figure 9: Qtrees for $\neg S_q \rightarrow S_p =$ *If Ido is not in Cambridge then he is at SuB*; obtained *mutatis mutandis* from Figure 8

[6]Note that this Qtree is derived *via* the "plugging in" operation, plus Qtree reduction. The Qtree obtained from (3b) and (2a)/(2b) *before* reduction is given in (i). Reduction 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)



4. Capturing the target cases

4.1. Accommodated Qtrees for (6b-6e)

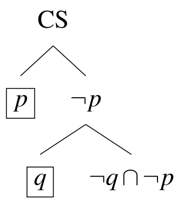
It now becomes possible to derive the accommodated QuD of sentences (6b-6e), using the “helper” Qtrees already derived, and our semantics for negated, disjunctive and conditional Qtrees. We start with (6b), whose Qtrees are given in Figure 10. Those two Qtrees are derived using Figures 3 (for $\neg p$), 5 (for $p \vee q$), and the combination rule for conditional Qtrees. Note that in both cases, the output Qtrees are the same as Qtrees associated with other “simpler” expressions, $\neg p \rightarrow q$ and q respectively.

(a) “Polar” antecedent 3a and unique possible disjunctive consequent 5. **Note that this Qtree is the same as 8b**

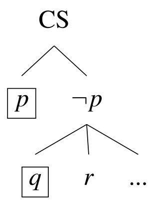
(b) “*Wh*” antecedent 3a and unique possible disjunctive consequent 5.[7]**Note that this Qtree is the same as 2b**

Figure 10: Qtrees for (6b) $= \neg p \rightarrow (p \vee q)$

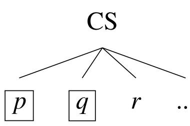
For (6c), the relevant Qtrees, shown in Figure 11, are obtained using Figures 1 (for p) and 8 (for $\neg p \rightarrow q$). Note that, because the Qtrees evoked by p that are properly disjoinable with those evoked by $\neg p \rightarrow q$, are structurally contained in them, the Qtrees in Figure 11 are simply those from Figure 8, with an extra verifying p-leaf.



(a) Obtained by disjoining a “polar” Qtree for p 1a, and the conditional Qtree 8a



(b) Obtained by disjoining a “polar” Qtree for p 1a, and the conditional Qtree 8b



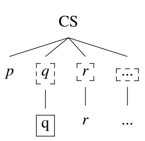
(c) Obtained by disjoining a “*wh*” Qtree for p 1b, and the conditional Qtree 8c. **Note that this Qtree is the same as 5.**

Figure 11: Qtrees for (6c) $= p \vee (\neg p \rightarrow q)$

For (6d), the only possible Qtree, shown in Figure 12, is obtained using Figures 1 (for p) and 9

[7]This Qtree, just like the one in Figure 8c, is derived *via* “plugging in” and reduction. Before reduction, the Qtree looked like (i). Reduction of the q—q trivial edge yields a verifying q-leaf; reduction of all other edges yield non-verifying leaves.

(i)



(for $\neg q \rightarrow p$). Because both input Qtrees are the same, the output Qtree is also similar. Finally, the only possible Qtree associated with (4a), given in Figure 13, ends up being structurally similar to a disjunctive Qtree for $p \vee q$ (Figure 5), except that no verifying nodes remains after conditionalization. This Qtree is thus considered ill-formed.

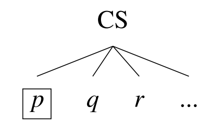


Figure 12: Qtrees for (6d) $= p \vee (\neg q \rightarrow p)$, obtained by disjoining a "*wh*" Qtree for p 1b and the conditional Qtree 9c. **Note that this Qtree is the same as 1b.**

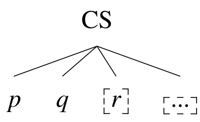


Figure 13: Qtree for (6e) $= \neg(p \vee q) \rightarrow p$, obtained by using the negation of Qtree 5 as antecedent and any Qtree for p as consequent.

4.2. Rephrasing Redundancy

In the previous section, we noted cases where the Qtrees derived from our target sentences turned out to be identical to Qtrees evoked by other, "simpler" sentences. Here, we argue that those cases are in fact problematic and constitute violations of a specific implementation of REDUNDANCY, inspired from Katzir and Singh (2014); Meyer (2013); Mayr and Romoli (2016). More specifically, we use a simplified notion of the REDUNDANCY constraint on Qtree-LF pairs introduced in Hénot-Mortier (2024). This constraint is sensitive to how accommodated QuDs package information, allowing us to introduce a contrast between $\vee$ and $\rightarrow$. More specifically, if a question is evoked by a sentence S and also by one the sentence's formal simplifications S'; then the question is redundant w.r.t. S, and as such should be ruled out from the set of possible Qtrees of S. This is spelled out in (15a-15c).

(15) a. *QuD-driven* REDUNDANCY (simplified from Hénot-Mortier (2024)). Let X be a LF and let $Qtrees(X)$ be the set of the Qtrees compatible with X. For any $T \in Qtrees(X)$, T is deemed Q-REDUNDANT with respect to 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.

c. *Qtree equality.* $T = T'$ iff T and T' have same structure and same set of verifying nodes.

Regarding (6a)$= p \vee (p \vee q)$, we noted that its only possible Qtree, shown in Figure 5, was also compatible with $p \vee q$, which is a simplification of $p \vee (p \vee q)$. So, after checking Q-REDUNDANCY, (6a) is no longer compatible with any Qtree and correctly deemed odd.

Regarding (6b)$= \neg p \rightarrow (p \vee q)$, we noted that its two possible Qtrees, shown in Figure 10, were also compatible with $\neg p \rightarrow q$, and q, respectively. And both $\neg p \rightarrow q$ and q are simplifications of $\neg p \rightarrow (p \vee q)$. So, after checking Q-REDUNDANCY, (6b) is no longer compatible with any Qtree and correctly deemed odd.

Regarding (6c)$= p \vee (\neg p \rightarrow q)$, we noted that one of its three possible Qtrees, shown in Figure 11, was also compatible with $p \vee q$, which is a simplification of $\neg p \rightarrow (p \vee q)$. What about the two other trees, in Figures 11a and 11b? The six possible simplifications of (6c) are p, $\neg p \rightarrow q$, $p \rightarrow q$, $p \vee \neg p$, $p \vee q$, and $p \vee p$. As for p, $p \vee p$, and $p \vee \neg p$, they give rise to Qtrees whose structures are given by Figure (1); which are obviously different from those in Figures 11a and 11b. $\neg p \rightarrow q$ and $p \rightarrow q$ on the other hand, give rise to Qtrees whose structures are equal to, or analog to, those in Figure 8c, but crucially, do not treat the p leaf as verifying – unlike the Qtrees in Figures 11a and 11b. Lastly, $p \vee q$ gives rise to the Qtree in Figure 5, which is obviously different from those in Figures 11a and 11b. Therefore, there is no simplification of (6c) giving rise to Qtrees like 11a and 11b, and, as a result, those Qtrees remain compatible with the sentence after checking Q-REDUNDANCY. This means that (6b) should not be deemed odd, in line with our expectations.

Regarding (6d) $= p \vee (\neg q \rightarrow p)$, we noted that its only possible Qtree, shown in Figure 12, was also compatible with p, which is a simplification of $p \vee (p \vee q)$. So, after checking Q-REDUNDANCY, (6d) is no longer compatible with any Qtree and correctly deemed odd.

Finally, we already settled the case of (6e) in the previous section: because this sentence only gives rise to a tree without any verifying leaf, it should be deemed odd as per (11) and (13).

5. Taking stock

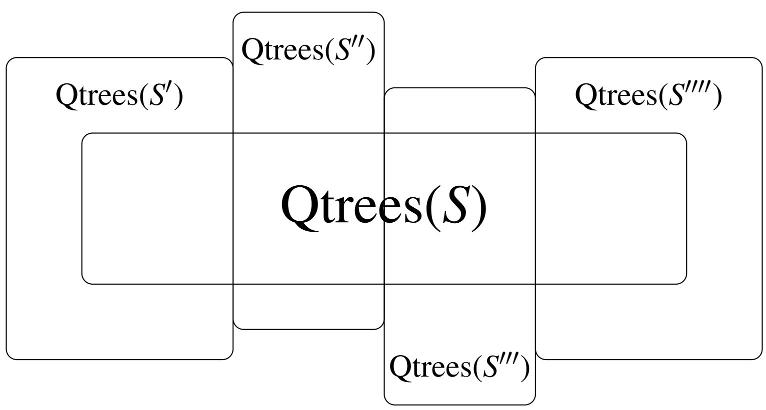
6. Parallel with Hurford Sentences

$p \vee q$ is like r and p like r^+. The only diff is that $p \vee q$, unlike r has internal structure. Given that $p \vee q$ is logically weaker than p, one may want to define $s^+ := p$ and $s := p \vee q$, s.t. (6a) and (6b) can be seen as

6.1. Comparison with Inquisitive Semantics

6.2. Brevity at the inquisitive level?

Earlier definitions of REDUNDANCY were linking this notion to Grice's Maxim of Brevity: if two sentences have the same logical contribution, prefer the more concise one. Is Q-REDUNDANCY a proper extension of Brevity at the inquisitive level? Not exactly. In particular, Q-REDUNDANCY does not state that, for a sentence S to be redundant, *all* Qtrees compatible with S should be identified *via* a bijective operation to *all* Qtrees compatible with a simplification of S – which would have been the most intuitive extension of brevity. For instance, we stated that (6b) was Q-REDUNDANT because each of its Qtrees could be identified with Qtrees coming from *distinct* simplifications of (6b). Moreover, the relevant simplifications of (6b) themselves led to Qtrees that (6b) was not compatible with.



extends brevity if we consider the optimization of 2 variables: LF and QuD.

6.3.

7. Exploring extensions and elaborations of the target sentences

8. Conclusion and outlook

(16) Double *or*-to-*if*

a. # If Ido is not at SuB then, if he is not at SuB then he is in Cambridge. $\neg p \rightarrow (\neg p \rightarrow q)$

b. # If Ido is not at SuB then, if he is not in Cambridge then he is at SuB. $\neg p \rightarrow (\neg q \rightarrow p)$

c. If it's not that Ido is in Cambridge if not at SuB, then Ido is not at SuB. $\neg(\neg p \rightarrow q) \rightarrow p$

d. # If it's not that Ido is at SuB if not in Cambridge, then Ido is at SuB. $\neg(\neg q \rightarrow p) \rightarrow p$

ido had dessert or cheese and dessert if ido did not have dessert he had cheese and dessert if ido did not have cheese and dessert he had dessert

9. Conclusion

Stress the fact the Hurford Senteces covered by other paper did not allow to deriectly tease apart my account from Kalomoiros' but the sentences at stake in this paper can.

References

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. (2024). "one tool to rule them all"? an integrated model of the qud for hurford sentences. In *Proceedings of the 29th Sinn und Bedeutung*. To appear.

Kalomoiros, A. (2024, January). An approach to hurford conditionals. *Semantics and Linguistic Theory*, 724–743.

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

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

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

Meyer, M.-C. (2013). *Ignorance and Grammar*. Ph. D. thesis, Massachusetts Institute of Technology.

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

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

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. (2024). A questions-under-discussion-based account of redundancy. *Unpublished manuscript*.