
QUD
ONE TOOL TO RULE THEM ALL??
DROP FORGED

"One tool to rule them all"? An integrated model of the QuD for Hurford sentences[1]

Adèle Hénot-Mortier (MIT)

September 20, 2024

Workshop "Does semantics have a 'too many tools' problem?" of the 29th Sinn und Bedeutung Conference

[1]Many thanks to Amir Anvari, Athulya Aravind, Danny Fox, Nina Haslinger, and Viola Schmitt for their precious input on that project. Thanks also to the audience of the BerlinBrnoVienna Workshop for relevant questions, datapoints and suggestions. All mistakes are my own.

Introduction

Data at stake

- Hurford Disjunctions (**HD**, cf. Hurford, 1974) like (1-2), which feature entailing disjuncts, feel redundant.

(1) # Mary lives in **Noto** or she lives in **Italy**. $\mathbf{p}^+ \vee \mathbf{p}$

(2) # Mary lives in **Italy** or she lives in **Noto**. $\mathbf{p} \vee \mathbf{p}^+$

- Hurford *Conditionals* (**HC**, cf. Mandelkern and Romoli, 2018), like (3-4) are isomorphic variants of (1) assuming material implication and (for (4)) a variable change of the form $\neg p := q^+ / p^+ := \neg q$.

(3) # If Mary does **not** live in **Noto**, she lives in **Italy**. $\neg\mathbf{p}^+ \rightarrow \mathbf{p}$

(4) If Mary lives in **Italy**, she does **not** live in **Noto**. $\neg\neg\mathbf{p} \rightarrow \neg\mathbf{p}^+ \equiv \neg\mathbf{q}^+ \rightarrow \mathbf{q}$

- Yet, (3) is odd while (4) is felicitous. This is challenging for existing accounts of Hurford sentences relying on a classical interpretation of $\vee$, $\rightarrow$, and $\neg$.

Data at stake

- Hurford Disjunctions (**HD**, cf. Hurford, 1974) like (1-2), which feature entailing disjuncts, feel redundant.

(1) # Mary lives in **Noto** or she lives in **Italy**. $\mathbf{p}^+ \vee \mathbf{p}$

(2) # Mary lives in **Italy** or she lives in **Noto**. $\mathbf{p} \vee \mathbf{p}^+$

- Hurford *Conditionals* (**HC**, cf. Mandelkern and Romoli, 2018), like (3-4) are isomorphic variants of (1) assuming material implication and (for (4)) a variable change of the form $\neg p := q^+ / p^+ := \neg q$.

(3) # If Mary does **not** live in **Noto**, she lives in **Italy**. $\neg\mathbf{p}^+ \rightarrow \mathbf{p}$

(4) If Mary lives in **Italy**, she does **not** live in **Noto**. $\neg\neg\mathbf{p} \rightarrow \neg\mathbf{p}^+ \equiv \neg\mathbf{q}^+ \rightarrow \mathbf{q}$

- Yet, (3) is odd while (4) is felicitous. This is challenging for existing accounts of Hurford sentences relying on a classical interpretation of $\vee$, $\rightarrow$, and $\neg$.

Data at stake

- Hurford Disjunctions (**HD**, cf. Hurford, 1974) like (1-2), which feature entailing disjuncts, feel redundant.

(1) # Mary lives in **Noto** or she lives in **Italy**. $\mathbf{p}^+ \vee \mathbf{p}$

(2) # Mary lives in **Italy** or she lives in **Noto**. $\mathbf{p} \vee \mathbf{p}^+$

- Hurford *Conditionals* (**HC**, cf. Mandelkern and Romoli, 2018), like (3-4) are isomorphic variants of (1) assuming material implication and (for (4)) a variable change of the form $\neg p := q^+ / p^+ := \neg q$.

(3) # If Mary does **not** live in **Noto**, she lives in **Italy**. $\neg \mathbf{p}^+ \rightarrow \mathbf{p}$

(4) If Mary lives in **Italy**, she does **not** live in **Noto**. $\neg\neg \mathbf{p} \rightarrow \neg \mathbf{p}^+ \equiv \neg \mathbf{q}^+ \rightarrow \mathbf{q}$

- Yet, (3) is odd while (4) is felicitous. This is challenging for existing accounts of Hurford sentences relying on a classical interpretation of $\vee$, $\rightarrow$, and $\neg$.

Previous work on Hurford sentences

- **Kalomoiros (2024)** proposes an interesting solution to the puzzle of HCs and HDs based on the concept of *Super-Redundancy*, which gives a specific status to overt negation. However nothing is said about how Hurford sentences can be repaired.
- **Haslinger (2023)** accounts for HDs and other related cases, e.g. coordination, *via* some intuitions about the Question under Discussion (**QuD**, Van Kuppevelt, 1995; Roberts, 1996).
- **Zhang (2024)**, building on (Simons, 2001; Büring, 2003) proposes another view on HDs and how to fix them, based on intuitions about QuD *trees*; however, no compositional machinery is proposed to derive those trees.
- **We will be trying to build on Haslinger's and Zhang's insights to propose a way to retro-engineer and constrain questions raised by sentences**, allowing to derive the pattern of HCs and HDs.

Previous work on Hurford sentences

- **Kalomoiros (2024)** proposes an interesting solution to the puzzle of HCs and HDs based on the concept of *Super-Redundancy*, which gives a specific status to overt negation. However nothing is said about how Hurford sentences can be repaired.
- **Haslinger (2023)** accounts for HDs and other related cases, e.g. coordination, *via* some intuitions about the Question under Discussion (**QuD**, Van Kuppevelt, 1995; Roberts, 1996).
- **Zhang (2024)**, building on (Simons, 2001; Büring, 2003) proposes another view on HDs and how to fix them, based on intuitions about QuD *trees*; however, no compositional machinery is proposed to derive those trees.
- **We will be trying to build on Haslinger's and Zhang's insights to propose a way to retro-engineer and constrain questions raised by sentences**, allowing to derive the pattern of HCs and HDs.

Previous work on Hurford sentences

- **Kalomoiros (2024)** proposes an interesting solution to the puzzle of HCs and HDs based on the concept of *Super-Redundancy*, which gives a specific status to overt negation. However nothing is said about how Hurford sentences can be repaired.
- **Haslinger (2023)** accounts for HDs and other related cases, e.g. coordination, *via* some intuitions about the Question under Discussion (**QuD**, Van Kuppevelt, 1995; Roberts, 1996).
- **Zhang (2024)**, building on (Simons, 2001; Büring, 2003) proposes another view on HDs and how to fix them, based on intuitions about QuD *trees*; however, no compositional machinery is proposed to derive those trees.
- **We will be trying to build on Haslinger's and Zhang's insights to propose a way to retro-engineer and constrain questions raised by sentences**, allowing to derive the pattern of HCs and HDs.

Previous work on Hurford sentences

- **Kalomoiros (2024)** proposes an interesting solution to the puzzle of HCs and HDs based on the concept of *Super-Redundancy*, which gives a specific status to overt negation. However nothing is said about how Hurford sentences can be repaired.
- **Haslinger (2023)** accounts for HDs and other related cases, e.g. coordination, *via* some intuitions about the Question under Discussion (**QuD**, Van Kuppevelt, 1995; Roberts, 1996).
- **Zhang (2024)**, building on (Simons, 2001; Büring, 2003) proposes another view on HDs and how to fix them, based on intuitions about QuD *trees*; however, no compositional machinery is proposed to derive those trees.
- **We will be trying to build on Haslinger's and Zhang's insights to propose a way to retro-engineer and constrain questions raised by sentences**, allowing to derive the pattern of HCs and HDs.

At a glance

- We account for HCs and HDs assuming that:
  - questions have different levels of **granularity**;
  - sentences raise implicit questions (Katzir and Singh, 2015 i.a.) in the form of **trees**, in such a way that conditionals and disjunctions end up having different contributions;
  - RELEVANCE and REDUNDANCY constraints restrict the computation of implicit questions.
- The problem with the infelicitous HC (3) will boil down to the fact that the question raised by its consequent is "coarser-grained" than that of its antecedent, and therefore appears IRRELEVANT, granted the antecedent.
- The problem with the infelicitous HDs (1-2) will come from the fact their disjuncts raise 2 REDUNDANT strategies of inquiry.

At a glance

- We account for HCs and HDs assuming that:
  - questions have different levels of **granularity**;
  - sentences raise implicit questions (Katzir and Singh, 2015 i.a.) in the form of **trees**, in such a way that conditionals and disjunctions end up having different contributions;
  - RELEVANCE and REDUNDANCY constraints restrict the computation of implicit questions.
- The problem with the infelicitous HC (3) will boil down to the fact that the question raised by its consequent is "coarser-grained" than that of its antecedent, and therefore appears IRRELEVANT, granted the antecedent.
- The problem with the infelicitous HDs (1-2) will come from the fact their disjuncts raise 2 REDUNDANT strategies of inquiry.

At a glance

- We account for HCs and HDs assuming that:
  - questions have different levels of **granularity**;
  - sentences raise implicit questions (Katzir and Singh, 2015 i.a.) in the form of **trees**, in such a way that conditionals and disjunctions end up having different contributions;
  - RELEVANCE and REDUNDANCY constraints restrict the computation of implicit questions.
- The problem with the infelicitous HC (3) will boil down to the fact that the question raised by its consequent is “coarser-grained” than that of its antecedent, and therefore appears IRRELEVANT, granted the antecedent.
- The problem with the infelicitous HDs (1-2) will come from the fact their disjuncts raise 2 REDUNDANT strategies of inquiry.

At a glance

- We account for HCs and HDs assuming that:
  - questions have different levels of **granularity**;
  - sentences raise implicit questions (Katzir and Singh, 2015 i.a.) in the form of **trees**, in such a way that conditionals and disjunctions end up having different contributions;
  - RELEVANCE and REDUNDANCY constraints restrict the computation of implicit questions.
- The problem with the infelicitous HC (3) will boil down to the fact that the question raised by its consequent is "coarser-grained" than that of its antecedent, and therefore appears IRRELEVANT, granted the antecedent.
- The problem with the infelicitous HDs (1-2) will come from the fact their disjuncts raise 2 REDUNDANT strategies of inquiry.

At a glance

- We account for HCs and HDs assuming that:
  - questions have different levels of **granularity**;
  - sentences raise implicit questions (Katzir and Singh, 2015 i.a.) in the form of **trees**, in such a way that conditionals and disjunctions end up having different contributions;
  - Relevance and Redundancy constraints restrict the computation of implicit questions.
- The problem with the infelicitous HC (3) will boil down to the fact that the question raised by its consequent is "coarser-grained" than that of its antecedent, and therefore appears Irrelevant, granted the antecedent.
- The problem with the infelicitous HDs (1-2) will come from the fact their disjuncts raise 2 Redundant strategies of inquiry.

At a glance

- We account for HCs and HDs assuming that:
  - questions have different levels of **granularity**;
  - sentences raise implicit questions (Katzir and Singh, 2015 i.a.) in the form of **trees**, in such a way that conditionals and disjunctions end up having different contributions;
  - Relevance and Redundancy constraints restrict the computation of implicit questions.
- The problem with the infelicitous HC (3) will boil down to the fact that the question raised by its consequent is "coarser-grained" than that of its antecedent, and therefore appears Irrelevant, granted the antecedent.
- The problem with the infelicitous HDs (1-2) will come from the fact their disjuncts raise 2 Redundant strategies of inquiry.

Linking assertions to questions

Background on question semantics

- Questions are usually seen as **partitions** of the Context Set (**CS**, Stalnaker, 1974).
- For any set of worlds S, a partition of S can be generated from a set of propositions by simply grouping together the worlds of S that "agree" on all those propositions (Hamblin, 1973). Let's call that operation $\textsc{Partition}(S,\ p_1, ...p_k)$. Special cases:
  - You only consider one proposition p that is not settled by the CS; the partition obtained intuitively corresponds to the **polar question** of *whether p* ($\{p, \neg p\}$).
  - You consider a set of propositions corresponding to focus alternatives; the partition obtained intuitively corresponds to a **wh-question** inquiring about the focused material.
    - **Special subcase**: if the propositions are all possible and mutually exclusive in S, the corresponding question partition is just the set of those propositions: $\textsc{Partition}(S,\ p_1, ...p_k) = \{p_1, ...p_k\}$.

Background on question semantics

- Questions are usually seen as **partitions** of the Context Set (**CS**, Stalnaker, 1974).
- For any set of worlds S, a partition of S can be generated from a set of propositions by simply grouping together the worlds of S that "agree" on all those propositions (Hamblin, 1973). Let's call that operation $\textsc{Partition}(S,\ p_1, ...p_k)$. Special cases:
  - You only consider one proposition p that is not settled by the CS; the partition obtained intuitively corresponds to the **polar question** of *whether p* ($\{p, \neg p\}$).
  - You consider a set of propositions corresponding to focus alternatives; the partition obtained intuitively corresponds to a **wh-question** inquiring about the focused material.
    - **Special subcase**: if the propositions are all possible and mutually exclusive in S, the corresponding question partition is just the set of those propositions: $\textsc{Partition}(S,\ p_1, ...p_k) = \{p_1, ...p_k\}$.

Background on question semantics

- Questions are usually seen as **partitions** of the Context Set (**CS**, Stalnaker, 1974).
- For any set of worlds S, a partition of S can be generated from a set of propositions by simply grouping together the worlds of S that "agree" on all those propositions (Hamblin, 1973). Let's call that operation $\textsc{Partition}(S,\ p_1, ...p_k)$. Special cases:
  - You only consider one proposition p that is not settled by the CS; the partition obtained intuitively corresponds to the **polar question** of *whether p* ($\{p, \neg p\}$).
  - You consider a set of propositions corresponding to focus alternatives; the partition obtained intuitively corresponds to a **wh-question** inquiring about the focused material.
    - **Special subcase**: if the propositions are all possible and mutually exclusive in S, the corresponding question partition is just the set of those propositions: $\textsc{Partition}(S,\ p_1, ...p_k) = \{p_1, ...p_k\}$.

Background on question semantics

- Questions are usually seen as **partitions** of the Context Set (**CS**, Stalnaker, 1974).
- For any set of worlds S, a partition of S can be generated from a set of propositions by simply grouping together the worlds of S that "agree" on all those propositions (Hamblin, 1973). Let's call that operation $\textsc{Partition}(S,\ p_1, ...p_k)$. Special cases:
  - You only consider one proposition p that is not settled by the CS; the partition obtained intuitively corresponds to the **polar question** of *whether p* ($\{p, \neg p\}$).
  - You consider a set of propositions corresponding to focus alternatives; the partition obtained intuitively corresponds to a **wh-question** inquiring about the focused material.
    - **Special subcase**: if the propositions are all possible and mutually exclusive in S, the corresponding question partition is just the set of those propositions: $\textsc{Partition}(S,\ p_1, ...p_k) = \{p_1, ...p_k\}$.

Background on question semantics

- Questions are usually seen as **partitions** of the Context Set (**CS**, Stalnaker, 1974).
- For any set of worlds S, a partition of S can be generated from a set of propositions by simply grouping together the worlds of S that "agree" on all those propositions (Hamblin, 1973). Let's call that operation $\textsc{Partition}(S,\ p_1, ...p_k)$. Special cases:
  - You only consider one proposition p that is not settled by the CS; the partition obtained intuitively corresponds to the **polar question** of *whether p* ($\{p, \neg p\}$).
  - You consider a set of propositions corresponding to focus alternatives; the partition obtained intuitively corresponds to a **wh-question** inquiring about the focused material.
    - **Special subcase**: if the propositions are all possible and mutually exclusive in S, the corresponding question partition is just the set of those propositions: $\textsc{Partition}(S,\ p_1, ...p_k) = \{p_1, ...p_k\}$.

One step forward: questions as trees

- The idea is not new (Büring, 2003; Riester, 2019; Zhang, 2024) but I want to give it a slightly different flavor, defining questions **trees as possible parse trees of the CS**.
- A **Q-tree** is a trees whose nodes all denote sets of worlds (i.e. propositions) and s.t.:
  - the **root** node denotes the CS;
  - **other nodes** are understood as possible answers (maximal or not) to the question;
  - the **children** of any node *N* partition *N* and can be seen as a "restricted" question defined in the domain of the CS where *N* holds.

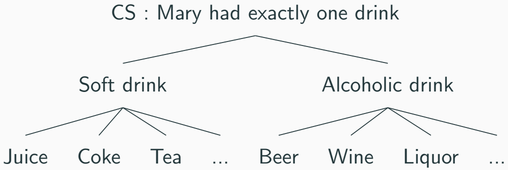


Figure 1: An intuitive Q-tree for the question *Which drink did Mary have?*

One step forward: questions as trees

- The idea is not new (Büring, 2003; Riester, 2019; Zhang, 2024) but I want to give it a slightly different flavor, defining questions **trees as possible parse trees of the CS**.
- A **Q-tree** is a trees whose nodes all denote sets of worlds (i.e. propositions) and s.t.:
  - the **root** node denotes the CS;
  - **other nodes** are understood as possible answers (maximal or not) to the question;
  - the **children** of any node N partition N and can be seen as a "restricted" question defined in the domain of the CS where N holds.

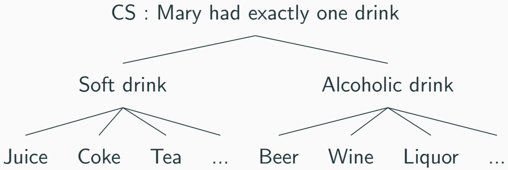


Figure 1: An intuitive Q-tree for the question *Which drink did Mary have?*

One step forward: questions as trees

- The idea is not new (Büring, 2003; Riester, 2019; Zhang, 2024) but I want to give it a slightly different flavor, defining questions **trees as possible parse trees of the CS**.
- A **Q-tree** is a trees whose nodes all denote sets of worlds (i.e. propositions) and s.t.:
  - the **root** node denotes the CS;
  - **other nodes** are understood as possible answers (maximal or not) to the question;
  - the **children** of any node N partition N and can be seen as a "restricted" question defined in the domain of the CS where N holds.

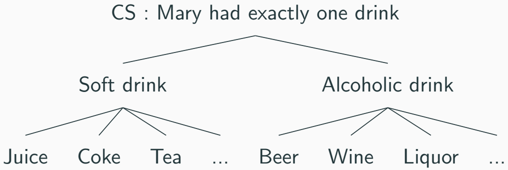


Figure 1: An intuitive Q-tree for the question *Which drink did Mary have?*

One step forward: questions as trees

- The idea is not new (Büring, 2003; Riester, 2019; Zhang, 2024) but I want to give it a slightly different flavor, defining questions **trees as possible parse trees of the CS**.
- A **Q-tree** is a trees whose nodes all denote sets of worlds (i.e. propositions) and s.t.:
  - the **root** node denotes the CS;
  - **other nodes** are understood as possible answers (maximal or not) to the question;
  - the **children** of any node N partition N and can be seen as a "restricted" question defined in the domain of the CS where N holds.

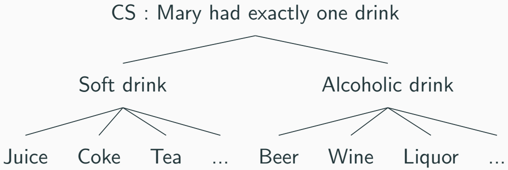


Figure 1: An intuitive Q-tree for the question *Which drink did Mary have?*

One step forward: questions as trees

- The idea is not new (Büring, 2003; Riester, 2019; Zhang, 2024) but I want to give it a slightly different flavor, defining questions **trees as possible parse trees of the CS**.
- A **Q-tree** is a trees whose nodes all denote sets of worlds (i.e. propositions) and s.t.:
  - the **root** node denotes the CS;
  - **other nodes** are understood as possible answers (maximal or not) to the question;
  - the **children** of any node *N* partition *N* and can be seen as a "restricted" question defined in the domain of the CS where *N* holds.

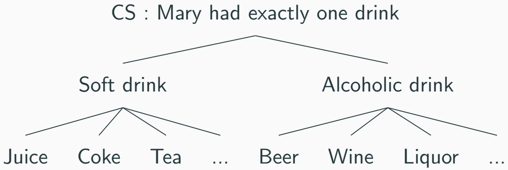


Figure 1: An intuitive Q-tree for the question *Which drink did Mary have?*

Interaction between assertive sentences and questions

- A recent line of research (Katzir and Singh, 2015 a.o.) develops the idea that **felicitous sentences should be possible answers to a "good" QuD**. What's the connection between assertive sentences and Q-trees then?
- Let's call $\widehat{Qs}(X)$ the set of Q-trees a Logical Form X can be can be seen as the answer to. We'd like some inductive algorithm allowing to **"retro-engineer" $\widehat{Qs}(X)$ starting from X's simplex parts and following its structure from the bottom up**.
- This algorithm will interact with Q-tree well-formedness constraints (Relevance, Redundancy, TBD) in such a way that **certain sentences will end up with an empty $\widehat{Qs}(X)$. Those will be deemed odd.**

Interaction between assertive sentences and questions

- A recent line of research (Katzir and Singh, 2015 a.o.) develops the idea that **felicitous sentences should be possible answers to a "good" QuD**. What's the connection between assertive sentences and Q-trees then?
- Let's call $\widehat{Qs}(X)$ the set of Q-trees a Logical Form X can be can be seen as the answer to. We'd like some inductive algorithm allowing to **"retro-engineer"** $\widehat{Qs}(X)$ **starting from X's simplex parts and following its structure from the bottom up**.
- This algorithm will interact with Q-tree well-formedness constraints (RELEVANCE, REDUNDANCY, TBD) in such a way that **certain sentences will end up with an empty $\widehat{Qs}(X)$. Those will be deemed odd.**

Interaction between assertive sentences and questions

- A recent line of research (Katzir and Singh, 2015 a.o.) develops the idea that **felicitous sentences should be possible answers to a "good" QuD**. What's the connection between assertive sentences and Q-trees then?
- Let's call $\widehat{Qs}(X)$ the set of Q-trees a Logical Form X can be can be seen as the answer to. We'd like some inductive algorithm allowing to **"retro-engineer"** $\widehat{Qs}(X)$ **starting from** X**'s simplex parts and following its structure from the bottom up**.
- This algorithm will interact with Q-tree well-formedness constraints (Relevance, Redundancy, TBD) in such a way that **certain sentences will end up with an empty** $\widehat{Qs}(X)$**. Those will be deemed odd.**

The machinery

$\widehat{Qs}$ of simplex LFs (no operator, connective or quantifier)

- Let's first consider a simplex LF X denoting a proposition p. We'd like $T \in \widehat{Qs}(X)$, to be s.t. its leaves denote the kind of traditional question-partition derived from p...
  - either the **polar partition**: $\textsc{Partition}(\text{CS}, \{p\}) = \{p, \neg p\}$;
  - or, the **same granularity wh-partition**: $\textsc{Partition}(\text{CS}, \mathscr{A}_p^g) = \mathscr{A}_p^g$, assuming $\mathscr{A}_p^g$ is the set of exclusive same-granularity focus alternatives to p (cf. Appendix).
- We also want to allow for multiple layers of increasing granularity (top-down), and s.t. **each layer is defined by same-granularity alternatives to an alternative of p entailed by p**: $\textsc{Partition}(\text{CS}, \mathscr{A}_q^g)$, with $q \in \mathscr{A}_p \wedge p \Rightarrow q$
- Finally, we secure a way to keep track of what is being asserted by X: we associate $T \in \widehat{Qs}(X)$ with a **set of verifying nodes** $\mathbb{N}_T^+$. In the simplex case, $\mathbb{N}_T^+ = \{p\}$ (=the p-leaf).

$\widehat{Qs}$ of simplex LFs (no operator, connective or quantifier)

- Let's first consider a simplex LF X denoting a proposition p. We'd like $T \in \widehat{Qs}(X)$, to be s.t. its leaves denote the kind of traditional question-partition derived from p...
  - either the **polar partition**: $\textsc{Partition}(\text{CS}, \{p\}) = \{p, \neg p\}$;
  - or, the **same granularity wh-partition**: $\textsc{Partition}(\text{CS}, \mathscr{A}^g_p) = \mathscr{A}^g_p$, assuming $\mathscr{A}^g_p$ is the set of exclusive same-granularity focus alternatives to p (cf. Appendix).
- We also want to allow for multiple layers of increasing granularity (top-down), and s.t. **each layer is defined by same-granularity alternatives to an alternative of p entailed by p**: $\textsc{Partition}(\text{CS}, \mathscr{A}^g_q)$, with $q \in \mathscr{A}_p \wedge p \Rightarrow q$
- Finally, we secure a way to keep track of what is being asserted by X: we associate $T \in \widehat{Qs}(X)$ with a **set of verifying nodes** $\mathbb{N}^+_T$. In the simplex case, $\mathbb{N}^+_T = \{p\}$ (=the p-leaf).

$\widehat{Qs}$ of simplex LFs (no operator, connective or quantifier)

- Let's first consider a simplex LF X denoting a proposition p. We'd like $T \in \widehat{Qs}(X)$, to be s.t. its leaves denote the kind of traditional question-partition derived from p...
  - either the **polar partition**: $\textsc{Partition}(\text{CS}, \{p\}) = \{p, \neg p\}$;
  - or, the **same granularity wh-partition**: $\textsc{Partition}(\text{CS}, \mathscr{A}_p^g) = \mathscr{A}_p^g$, assuming $\mathscr{A}_p^g$ is the set of exclusive same-granularity focus alternatives to p (cf. Appendix).
- We also want to allow for multiple layers of increasing granularity (top-down), and s.t. **each layer is defined by same-granularity alternatives to an alternative of p entailed by p**: $\textsc{Partition}(\text{CS}, \mathscr{A}_q^g)$, with $q \in \mathscr{A}_p \wedge p \Rightarrow q$
- Finally, we secure a way to keep track of what is being asserted by X: we associate $T \in \widehat{Qs}(X)$ with a **set of verifying nodes** $\mathbb{N}_T^+$. In the simplex case, $\mathbb{N}_T^+ = \{p\}$ (=the p-leaf).

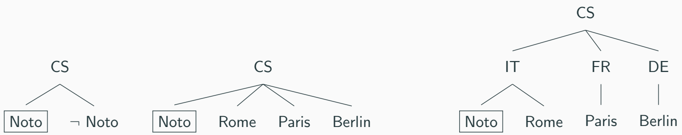
$\widehat{Qs}$ of simplex LFs (no operator, connective or quantifier)

- Let's first consider a simplex LF X denoting a proposition p. We'd like $T \in \widehat{Qs}(X)$, to be s.t. its leaves denote the kind of traditional question-partition derived from p...
  - either the **polar partition**: $\textsc{Partition}(\text{CS}, \{p\}) = \{p, \neg p\}$;
  - or, the **same granularity wh-partition**: $\textsc{Partition}(\text{CS}, \mathscr{A}_p^g) = \mathscr{A}_p^g$, assuming $\mathscr{A}_p^g$ is the set of exclusive same-granularity focus alternatives to p (cf. Appendix).
- We also want to allow for multiple layers of increasing granularity (top-down), and s.t. **each layer is defined by same-granularity alternatives to an alternative of** p **entailed by** p: $\textsc{Partition}(\text{CS}, \mathscr{A}_q^g)$, with $q \in \mathscr{A}_p \wedge p \Rightarrow q$
- Finally, we secure a way to keep track of what is being asserted by X: we associate $T \in \widehat{Qs}(X)$ with a **set of** **verifying nodes** $\mathbb{N}_T^+$. In the simplex case, $\mathbb{N}_T^+ = \{p\}$ (=the p-leaf).

$\widehat{Qs}$ of simplex LFs (no operator, connective or quantifier)

- Let's first consider a simplex LF X denoting a proposition p. We'd like $T \in \widehat{Qs}(X)$, to be s.t. its leaves denote the kind of traditional question-partition derived from p...
  - either the **polar partition**: $\textsc{Partition}(\text{CS}, \{p\}) = \{p, \neg p\}$;
  - or, the **same granularity wh-partition**: $\textsc{Partition}(\text{CS}, \mathscr{A}_p^g) = \mathscr{A}_p^g$, assuming $\mathscr{A}_p^g$ is the set of exclusive same-granularity focus alternatives to p (cf. Appendix).
- We also want to allow for multiple layers of increasing granularity (top-down), and s.t. **each layer is defined by same-granularity alternatives to an alternative of** p **entailed by** p: $\textsc{Partition}(\text{CS}, \mathscr{A}_q^g)$, with $q \in \mathscr{A}_p \wedge p \Rightarrow q$
- Finally, we secure a way to keep track of what is being asserted by X: we associate $T \in \widehat{Qs}(X)$ with a **set of verifying nodes** $\mathbb{N}_T^+$. In the simplex case, $\mathbb{N}_T^+ = \{p\}$ (=the p-leaf).

Q-trees for **p** and **p**$^{+}$



(a) “Polar” **(b)** “Wh” **(c)** 2-level “wh”

Figure 2: Some schematic Q-trees compatible with the simplex proposition **p**$^{+}$=*Mary lives in Noto*. Boxed cells denote verifying nodes $\mathbb{N}^{+}_{T}$.

(a) “Polar” **(b)** “Wh”

Figure 3: Some schematic Q-trees compatible with **p**=*Mary lives in Italy*.

$\widehat{Qs}$ of negated LFs

- Q-trees for a negated LF $\neg X$ are structurally similar to those of X, modulo the sets of verifying nodes, that are flipped into their non-verifying sisters.

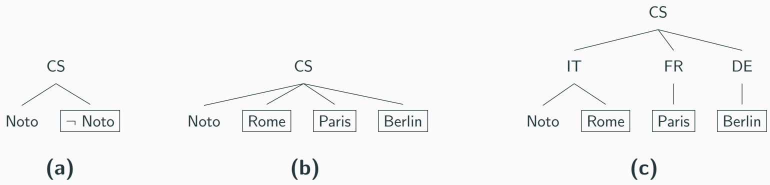


Figure 4: Some schematic Q-trees compatible with $\neg\mathbf{p}^{+}$=*Mary does not live in Noto*.

The conditional case

$\widehat{Qs}$ of conditional LFs (if X then Y)

- Intuitively, a Q-tree for $X \rightarrow Y$ **focuses on the question raised by Y in the sub-domain(s) of the CS where X holds**.
- To get a Q-tree T for $X \rightarrow Y$:
  - take a Q-tree $T_X \in \widehat{Qs}(X)$ and a Q-tree $T_Y \in \widehat{Qs}(Y)$;
  - for each verifying node of T_X, **replace it by its intersection with** T_Y (="plug in" T_Y).
- Intersecting a node N with a tree T amounts to intersecting each node of T with N, and pruning any resulting empty node. Verifying nodes are preserved: if M was verifying in T, then $M \cap N$ will be verifying in $T \cap N$.

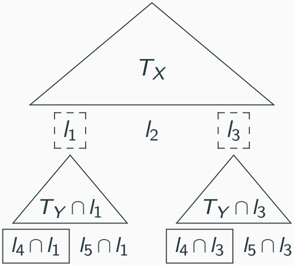


Figure 5: General form of a Q-tree for $X \rightarrow Y$. Nodes in dashed boxes are assumed to be verifying for X, and are thus further partitioned according to a Q-tree for Y. Boxed leaves are assumed to support Y, and thus also support $X \rightarrow Y$.

- Intuitively, a Q-tree for $X \rightarrow Y$ **focuses on the question raised by Y in the sub-domain(s) of the CS where X holds**.
- To get a Q-tree T for $X \rightarrow Y$:
  - take a Q-tree $T_X \in \widehat{Qs}(X)$ and a Q-tree $T_Y \in \widehat{Qs}(Y)$;
  - for each verifying node of T_X, **replace it by its intersection with** T_Y (="plug in" T_Y).
- Intersecting a node N with a tree T amounts to intersecting each node of T with N, and pruning any resulting empty node. Verifying nodes are preserved: if M was verifying in T, then $M \cap N$ will be verifying in $T \cap N$.

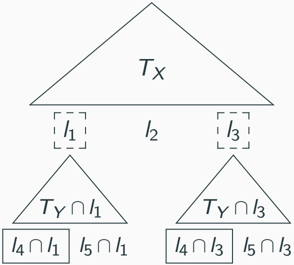


Figure 5: General form of a Q-tree for $X \rightarrow Y$. Nodes in dashed boxes are assumed to be verifying for X, and are thus further partitioned according to a Q-tree for Y. Boxed leaves are assumed to support Y, and thus also support $X \rightarrow Y$.

$\widehat{Qs}$ of conditional LFs (if X then Y)

- Intuitively, a Q-tree for $X \to Y$ **focuses on the question raised by Y in the sub-domain(s) of the CS where X holds**.
- To get a Q-tree T for $X \to Y$:
  - take a Q-tree $T_X \in \widehat{Qs}(X)$ and a Q-tree $T_Y \in \widehat{Qs}(Y)$;
  - for each verifying node of T_X, **replace it by its intersection with** T_Y (="plug in" T_Y).
- Intersecting a node N with a tree T amounts to intersecting each node of T with N, and pruning any resulting empty node. Verifying nodes are preserved: if M was verifying in T, then $M \cap N$ will be verifying in $T \cap N$.

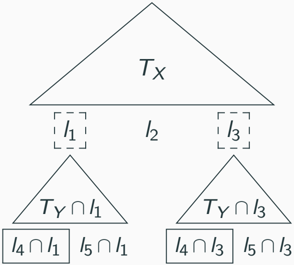


Figure 5: General form of a Q-tree for $X \to Y$. Nodes in dashed boxes are assumed to be verifying for X, and are thus further partitioned according to a Q-tree for Y. Boxed leaves are assumed to support Y, and thus also support $X \to Y$.

$\widehat{Qs}$ of conditional LFs (if X then Y)

- Intuitively, a Q-tree for $X \to Y$ **focuses on the question raised by Y in the sub-domain(s) of the CS where X holds**.
- To get a Q-tree T for $X \to Y$:
  - take a Q-tree $T_X \in \widehat{Qs}(X)$ and a Q-tree $T_Y \in \widehat{Qs}(Y)$;
  - for each verifying node of T_X, **replace it by its intersection with** T_Y (="plug in" T_Y).
- Intersecting a node N with a tree T amounts to intersecting each node of T with N, and pruning any resulting empty node. Verifying nodes are preserved: if M was verifying in T, then $M \cap N$ will be verifying in $T \cap N$.

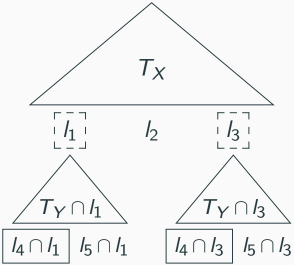


Figure 5: General form of a Q-tree for $X \to Y$. Nodes in dashed boxes are assumed to be verifying for X, and are thus further partitioned according to a Q-tree for Y. Boxed leaves are assumed to support Y, and thus also support $X \to Y$.

$\widehat{Qs}$ of conditional LFs (if X then Y)

- Intuitively, a Q-tree for $X \to Y$ **focuses on the question raised by Y in the sub-domain(s) of the CS where X holds**.
- To get a Q-tree T for $X \to Y$:
  - take a Q-tree $T_X \in \widehat{Qs}(X)$ and a Q-tree $T_Y \in \widehat{Qs}(Y)$;
  - for each verifying node of T_X, **replace it by its intersection with** T_Y (="plug in" T_Y).
- Intersecting a node N with a tree T amounts to intersecting each node of T with N, and pruning any resulting empty node. Verifying nodes are preserved: if M was verifying in T, then $M \cap N$ will be verifying in $T \cap N$.

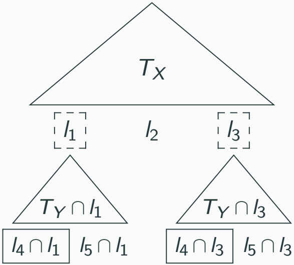


Figure 5: General form of a Q-tree for $X \to Y$. Nodes in dashed boxes are assumed to be verifying for X, and are thus further partitioned according to a Q-tree for Y. Boxed leaves are assumed to support Y, and thus also support $X \to Y$.

Q-trees for $\#\neg \mathbf{p}^{+} \rightarrow \mathbf{p}$ and $\mathbf{p} \rightarrow \neg \mathbf{p}^{+}$

(a) T_X=2a, T_Y=3b **(b)** T_X=2b, T_Y=3b

Figure 6: Q-trees for $\#\neg \mathbf{p}^{+} \rightarrow \mathbf{p}$ = *If Mary does not live in Noto, she lives in Italy*. More combinations possible but they all lead to the same result.

(a) T_X=3a, T_Y=2a **(b)** T_X=3a, T_Y=2b

Figure 7: Q-trees for $\mathbf{p} \rightarrow \neg \mathbf{p}^{+}$= *If Mary lives in Italy she does not live in Noto*. More combinations possible but they all lead to the same result.

Rephrasing Relevance

- Under the partition-based view of questions, a proposition p is relevant given a question, if it **does not cut across cells**. We want some generalization of this to apply as a filter during Q-tree derivation.

(5) Q-RELEVANCE: If T'' is derived from T and T' *via* Q-tree composition, then $\mathbb{N}^+_{T''} \subseteq \mathbb{N}^+_T \cup \mathbb{N}^+_{T'}$.

- This means that verifying nodes coming from the Q-trees passed as input to a binary Q-tree composition rule should be either fully ruled-out, or fully preserved in the output Q-tree, i.e., they should not be cut-across.
- A correlate in our $\{\neg, \vee, \rightarrow\}$-fragment (trust me on the meaning of $\vee$ at that point):

(6) Q-RELEVANCE (correlate): If tree T gets intersected with node N, $\mathbb{N}^+_{T\cap N} \subseteq \mathbb{N}^+_T$

Rephrasing Relevance

- Under the partition-based view of questions, a proposition p is relevant given a question, if it **does not cut across cells**. We want some generalization of this to apply as a filter during Q-tree derivation.

(5) Q-RELEVANCE: If T'' is derived from T and T' *via* Q-tree composition, then $\mathbb{N}^+_{T''} \subseteq \mathbb{N}^+_{T} \cup \mathbb{N}^+_{T'}$.

- This means that verifying nodes coming from the Q-trees passed as input to a binary Q-tree composition rule should be either fully ruled-out, or fully preserved in the output Q-tree, i.e., they should not be cut-across.
- A correlate in our $\{\neg, \vee, \rightarrow\}$-fragment (trust me on the meaning of $\vee$ at that point):

(6) Q-RELEVANCE (correlate): If tree T gets intersected with node N, $\mathbb{N}^+_{T \cap N} \subseteq \mathbb{N}^+_{T}$

Rephrasing Relevance

- Under the partition-based view of questions, a proposition p is relevant given a question, if it **does not cut across cells**. We want some generalization of this to apply as a filter during Q-tree derivation.

(5) Q-Relevance: If T'' is derived from T and T' *via* Q-tree composition, then $\mathbb{N}^+_{T''} \subseteq \mathbb{N}^+_{T} \cup \mathbb{N}^+_{T'}$.

- This means that verifying nodes coming from the Q-trees passed as input to a binary Q-tree composition rule should be either fully ruled-out, or fully preserved in the output Q-tree, i.e., they should not be cut-across.
- A correlate in our $\{\neg, \vee, \rightarrow\}$-fragment (trust me on the meaning of $\vee$ at that point):

(6) Q-Relevance (correlate): If tree T gets intersected with node N, $\mathbb{N}^+_{T \cap N} \subseteq \mathbb{N}^+_{T}$

Rephrasing Relevance

- Under the partition-based view of questions, a proposition p is relevant given a question, if it **does not cut across cells**. We want some generalization of this to apply as a filter during Q-tree derivation.

(5) Q-Relevance: If T'' is derived from T and T' *via* Q-tree composition, then $\mathbb{N}^+_{T''} \subseteq \mathbb{N}^+_{T} \cup \mathbb{N}^+_{T'}$.

- This means that verifying nodes coming from the Q-trees passed as input to a binary Q-tree composition rule should be either fully ruled-out, or fully preserved in the output Q-tree, i.e., they should not be cut-across.
- A correlate in our $\{\neg, \vee, \rightarrow\}$-fragment (trust me on the meaning of $\vee$ at that point):

(6) Q-Relevance (correlate): If tree T gets intersected with node N, $\mathbb{N}^+_{T \cap N} \subseteq \mathbb{N}^+_{T}$

Rephrasing Relevance

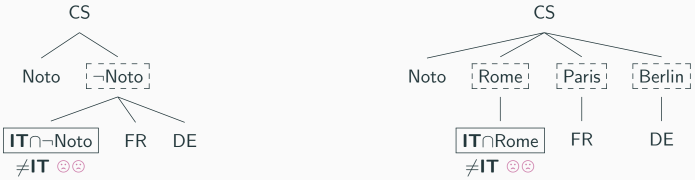
- Under the partition-based view of questions, a proposition p is relevant given a question, if it **does not cut across cells**. We want some generalization of this to apply as a filter during Q-tree derivation.

(5) Q-Relevance: If T'' is derived from T and T' *via* Q-tree composition, then $\mathbb{N}^+_{T''} \subseteq \mathbb{N}^+_T \cup \mathbb{N}^+_{T'}$.

- This means that verifying nodes coming from the Q-trees passed as input to a binary Q-tree composition rule should be either fully ruled-out, or fully preserved in the output Q-tree, i.e., they should not be cut-across.
- A correlate in our $\{\neg, \vee, \rightarrow\}$-fragment (trust me on the meaning of $\vee$ at that point):

(6) Q-Relevance (correlate): If tree T gets intersected with node N, $\mathbb{N}^+_{T\cap N} \subseteq \mathbb{N}^+_T$

- Q-RELEVANCE is violated in trees 6a & 6b, due to the impossibility for a verifying *Italy* node to be fully contained within city-level nodes (as introduced by the antecedent Q-tree).
- This implies that $\widehat{Qs}(\neg\mathbf{p}^{+} \rightarrow \mathbf{p}) = \emptyset$, and captures the infelicity of the HC (3).



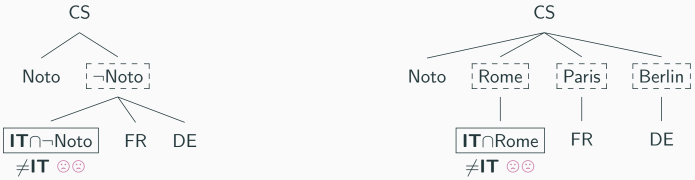
(a) T_X=2a, T_Y=3b

(b) T_X=2b, T_Y=3b

Figure 6 (repeated): Potential Q-trees obtained for $\#\neg\mathbf{p}^{+}\rightarrow\mathbf{p}$

Filtering out Q-trees via Q-Relevance: $\#\neg\mathbf{p}^{+}\rightarrow\mathbf{p}$

- Q-Relevance is violated in trees 6a & 6b, due to the impossibility for a verifying *Italy* node to be fully contained within city-level nodes (as introduced by the antecedent Q-tree).
- This implies that $\widehat{Qs}(\neg\mathbf{p}^{+} \rightarrow \mathbf{p}) = \emptyset$, and captures the infelicity of the HC (3).



(a) T_X=2a, T_Y=3b

(b) T_X=2b, T_Y=3b

Figure 6 (repeated): Potential Q-trees obtained for $\#\neg\mathbf{p}^{+}\rightarrow\mathbf{p}$

Filtering out Q-trees via Q-Relevance: $\mathbf{p} \to \neg \mathbf{p}^{+}$

- Tree 7b (although not Tree 7a[1]) satisfies Q-RELEVANCE, because it allows to fully map each verifying *not Noto*-node (city-level) to a particular country-level node.
- This implies that $\widehat{Qs}(\mathbf{p} \to \neg \mathbf{p}^{+}) \neq \emptyset$ and captures the felicity of the HC (4).

(a) T_X=3a, T_Y=2a

(b) T_X=3a, T_Y=2b

Figure 7 (repeated): Potential Q-trees obtained for $\mathbf{p} \to \neg \mathbf{p}^{+}$

[1] Tree 7a runs into the same issue as trees 6a & 6b

Filtering out Q-trees via Q-Relevance: $\mathbf{p}\to\neg\mathbf{p}^{+}$

- Tree 7b (although not Tree 7a[1]) satisfies Q-RELEVANCE, because it allows to fully map each verifying *not Noto*-node (city-level) to a particular country-level node.
- This implies that $\widehat{Qs}(\mathbf{p} \to \neg\mathbf{p}^{+}) \neq \emptyset$ and captures the felicity of the HC (4).

(a) T_X=3a, T_Y=2a

(b) T_X=3a, T_Y=2b

Figure 7 (repeated): Potential Q-trees obtained for $\mathbf{p}\to\neg\mathbf{p}^{+}$

[1] Tree 7a runs into the same issue as trees 6a & 6b

The disjunctive case

$\widehat{Qs}$ of disjunctive LFs (X or Y)

- Intuitively, a Q-tree for $X \vee Y$ raises a question pertaining to X and Y, simultaneously (Simons, 2001; Zhang, 2024). So, instead of plugging one tree into another as we did with conditionals, we want to **properly fuse them**.
- To get a Q-tree for $X \vee Y$:
  - take a Q-tree $T_X \in \widehat{Qs}(X)$ and a Q-tree $T_Y \in \widehat{Qs}(Y)$;
  - Graph-union T_X and T_Y by unioning the 2 sets of their nodes, the 2 sets of their verifying nodes, and the 2 sets of their edges (=all parent-child pairs).
  - Check that the resulting tree is a Q-tree; if it is, return it; if it's not, then it means we had a clash between the partitionings introduced by resp. T_X and T_Y somewhere, so, return nothing (cf. Appendix).[2]
- Note that the above **Q-tree-union operation is symmetric**, so whatever we predict for LF $X \vee Y$, we predict for $Y \vee X$.

[2] I thank Amir Anvari for helping me come up with this definition.

$\widehat{Qs}$ of disjunctive LFs (X or Y)

- Intuitively, a Q-tree for $X \vee Y$ raises a question pertaining to X and Y, simultaneously (Simons, 2001; Zhang, 2024). So, instead of plugging one tree into another as we did with conditionals, we want to **properly fuse them**.
- To get a Q-tree for $X \vee Y$:
  - take a Q-tree $T_X \in \widehat{Qs}(X)$ and a Q-tree $T_Y \in \widehat{Qs}(Y)$;
  - Graph-union T_X and T_Y by unioning the 2 sets of their nodes, the 2 sets of their verifying nodes, and the 2 sets of their edges (=all parent-child pairs).
  - Check that the resulting tree is a Q-tree; if it is, return it; if it's not, then it means we had a clash between the partitionings introduced by resp. T_X and T_Y somewhere, so, return nothing (cf. Appendix).[2]
- Note that the above **Q-tree-union operation is symmetric**, so whatever we predict for LF $X \vee Y$, we predict for $Y \vee X$.

[2] I thank Amir Anvari for helping me come up with this definition.

$\widehat{Qs}$ of disjunctive LFs (X or Y)

- Intuitively, a Q-tree for $X \vee Y$ raises a question pertaining to X and Y, simultaneously (Simons, 2001; Zhang, 2024). So, instead of plugging one tree into another as we did with conditionals, we want to **properly fuse them**.
- To get a Q-tree for $X \vee Y$:
  - take a Q-tree $T_X \in \widehat{Qs}(X)$ and a Q-tree $T_Y \in \widehat{Qs}(Y)$;
  - Graph-union T_X and T_Y by unioning the 2 sets of their nodes, the 2 sets of their verifying nodes, and the 2 sets of their edges (=all parent-child pairs).
  - Check that the resulting tree is a Q-tree; if it is, return it; if it's not, then it means we had a clash between the partitionings introduced by resp. T_X and T_Y somewhere, so, return nothing (cf. Appendix).[2]
- Note that the above **Q-tree-union operation is symmetric**, so whatever we predict for LF $X \vee Y$, we predict for $Y \vee X$.

[2] I thank Amir Anvari for helping me come up with this definition.

$\widehat{Qs}$ of disjunctive LFs (X or Y)

- Intuitively, a Q-tree for $X \vee Y$ raises a question pertaining to X and Y, simultaneously (Simons, 2001; Zhang, 2024). So, instead of plugging one tree into another as we did with conditionals, we want to **properly fuse them**.
- To get a Q-tree for $X \vee Y$:
  - take a Q-tree $T_X \in \widehat{Qs}(X)$ and a Q-tree $T_Y \in \widehat{Qs}(Y)$;
  - Graph-union T_X and T_Y by unioning the 2 sets of their nodes, the 2 sets of their verifying nodes, and the 2 sets of their edges (=all parent-child pairs).
  - Check that the resulting tree is a Q-tree; if it is, return it; if it's not, then it means we had a clash between the partitionings introduced by resp. T_X and T_Y somewhere, so, return nothing (cf. Appendix).[2]
- Note that the above **Q-tree-union operation is symmetric**, so whatever we predict for LF $X \vee Y$, we predict for $Y \vee X$.

[2] I thank Amir Anvari for helping me come up with this definition.

$\widehat{Qs}$ of disjunctive LFs (X or Y)

- Intuitively, a Q-tree for $X \vee Y$ raises a question pertaining to X and Y, simultaneously (Simons, 2001; Zhang, 2024). So, instead of plugging one tree into another as we did with conditionals, we want to **properly fuse them**.
- To get a Q-tree for $X \vee Y$:
  - take a Q-tree $T_X \in \widehat{Qs}(X)$ and a Q-tree $T_Y \in \widehat{Qs}(Y)$;
  - Graph-union T_X and T_Y by unioning the 2 sets of their nodes, the 2 sets of their verifying nodes, and the 2 sets of their edges (=all parent-child pairs).
  - Check that the resulting tree is a Q-tree; if it is, return it; if it's not, then it means we had a clash between the partitionings introduced by resp. T_X and T_Y somewhere, so, return nothing (cf. Appendix).[2]
- Note that the above **Q-tree-union operation is symmetric**, so whatever we predict for LF $X \vee Y$, we predict for $Y \vee X$.

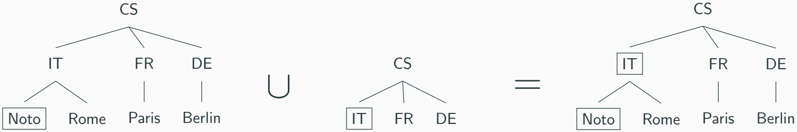
[2] I thank Amir Anvari for helping me come up with this definition.

$\widehat{Qs}$ of disjunctive LFs (X or Y)

- Intuitively, a Q-tree for $X \vee Y$ raises a question pertaining to X and Y, simultaneously (Simons, 2001; Zhang, 2024). So, instead of plugging one tree into another as we did with conditionals, we want to **properly fuse them**.
- To get a Q-tree for $X \vee Y$:
  - take a Q-tree $T_X \in \widehat{Qs}(X)$ and a Q-tree $T_Y \in \widehat{Qs}(Y)$;
  - Graph-union T_X and T_Y by unioning the 2 sets of their nodes, the 2 sets of their verifying nodes, and the 2 sets of their edges (=all parent-child pairs).
  - Check that the resulting tree is a Q-tree; if it is, return it; if it's not, then it means we had a clash between the partitionings introduced by resp. T_X and T_Y somewhere, so, return nothing (cf. Appendix).[2]
- Note that the above **Q-tree-union operation is symmetric**, so whatever we predict for LF $X \vee Y$, we predict for $Y \vee X$.

[2] I thank Amir Anvari for helping me come up with this definition.

Q-tree for $\mathbf{p}^+ \vee \mathbf{p}$ and $\mathbf{p} \vee \mathbf{p}^+$



(a) Q-tree for $\mathbf{p}^+$ **(b)** Q-tree for $\mathbf{p}$ **(c)** Q-tree for either $\mathbf{p} \vee \mathbf{p}^+$ or $\mathbf{p}^+ \vee \mathbf{p}$

Figure 8: Deriving the only possible Q-tree for $\mathbf{p} \vee \mathbf{p}^+$/ $\mathbf{p}^+ \vee \mathbf{p}$

- What's wrong with the resulting disjunctive Q-tree? If you see a path in a Q-tree as a **strategy of inquiry to converge to a maximal true answer**, then there's something suboptimal in Tree 8c: if you reach the *Noto*-node, then you've also reached the *Italy*-node along the way!

Rephrasing Redundancy

- There are many different view on what REDUNDANCY should be like, but one of those views states that a sentence is redundant if it has a formal simplification that ends up being contextually equivalent (Katzir & Singh, 2014). We want some generalization of this to apply to Q-trees.

(7) **Q-Redundancy**: LF X is Q-REDUNDANT iff there is a formal simplification X' of X obtained *via* constituent-to-subconstituent substitution, s.t. Qtrees(X) $\leq$ Qtrees(X').

Rephrasing Redundancy

- There are many different view on what Redundancy should be like, but one of those views states that a sentence is redundant if it has a formal simplification that ends up being contextually equivalent (Katzir & Singh, 2014). We want some generalization of this to apply to Q-trees.

(7) **Q-Redundancy**: LF X is Q-Redundant iff there is a formal simplification X' of X obtained *via* constituent-to-subconstituent substitution, s.t. Qtrees(X) $\leqq$ Qtrees(X').

Unpacking Q-Redundancy

(8) **Equivalent Sets of Qtrees**: $S \leqq S'$ iff $\forall T \in S.\ \exists T' \in S'.\ T \equiv T'$ (note: it is an asymmetric relation!)

(9) **Equivalent Qtrees**: $T \equiv T'$ iff T and T' have same structure and same set of maximal verifying paths.

(10) **Verifying paths**: set of paths (=ordered list of nodes) from the root to each verifying node.

(11) **Path containment**: $p \subseteq p'$ iff p is a prefix of p'.

(12) **Maximal Verifying Paths (P^*)**: if P is a set of verifying paths, P^* is the set of maximal elements of P w.r.t. path containment.

Filtering out Q-trees via Q-Redundancy

- Q-REDUNDANCY rules out the 2 HDs (1) and (2), more trivial cases such as **p** ∨ **p**, and more complex cases such as Long-Distance HDs (Marty & Romoli, 2022) (cf. Appendix).

(a) Q-tree for either **p** ∨ $\mathbf{p}^+$ or $\mathbf{p}^+$ ∨ **p**; P^*={[CS, IT, Noto]}

(b) Q-tree for $\mathbf{p}^+$; P^*={[CS, IT, Noto]}

Figure 9: Equivalence between the only Q-tree compatible with (1) or (2) and one Q-tree (Tree 2c) compatible with the simplification $\mathbf{p}^+$

Conclusion

- I don't really want to sell this as better than the other accounts, because obviously it's full of stipulations, but maybe this gives us a more integrated framework to think about how sentences relate to questions from a compositional perspective.
- A couple topics I wish to explore further:
  - **Coordination/Accommodation**; how does Q-tree derivation interact with updates of the CS? Should we e.g. trim the Q-tree from the top-down?
  - **Scalar implicatures**: the presence of scalar items in HDs creates a new asymmetry (Singh, 2008), possibly due to licensing constraints on Exh (Fox & Spector, 2018), or on how alternatives passed as argument to Exh are being pruned (Hénot-Mortier, 2023). Could the latter constraint be better motivated by the current framework?

Conclusion

- I don't really want to sell this as better than the other accounts, because obviously it's full of stipulations, but maybe this gives us a more integrated framework to think about how sentences relate to questions from a compositional perspective.
- A couple topics I wish to explore further:
  - **Coordination/Accommodation**; how does Q-tree derivation interact with updates of the CS? Should we e.g. trim the Q-tree from the top-down?
  - **Scalar implicatures**: the presence of scalar items in HDs creates a new asymmetry (Singh, 2008), possibly due to licensing constraints on EXH (Fox & Spector, 2018), or on how alternatives passed as argument to EXH are being pruned (Hénot-Mortier, 2023). Could the latter constraint be better motivated by the current framework?

Conclusion

- I don't really want to sell this as better than the other accounts, because obviously it's full of stipulations, but maybe this gives us a more integrated framework to think about how sentences relate to questions from a compositional perspective.
- A couple topics I wish to explore further:
  - **Coordination/Accommodation**; how does Q-tree derivation interact with updates of the CS? Should we e.g. trim the Q-tree from the top-down?
  - **Scalar implicatures**: the presence of scalar items in HDs creates a new asymmetry (Singh, 2008), possibly due to licensing constraints on Exh (Fox & Spector, 2018), or on how alternatives passed as argument to Exh are being pruned (Hénot-Mortier, 2023). Could the latter constraint be better motivated by the current framework?

Conclusion

- I don't really want to sell this as better than the other accounts, because obviously it's full of stipulations, but maybe this gives us a more integrated framework to think about how sentences relate to questions from a compositional perspective.
- A couple topics I wish to explore further:
  - **Coordination/Accommodation**; how does Q-tree derivation interact with updates of the CS? Should we e.g. trim the Q-tree from the top-down?
  - **Scalar implicatures**: the presence of scalar items in HDs creates a new asymmetry (Singh, 2008), possibly due to licensing constraints on $\textsc{Exh}$ (Fox & Spector, 2018), or on how alternatives passed as argument to $\textsc{Exh}$ are being pruned (Hénot-Mortier, 2023). Could the latter constraint be better motivated by the current framework?

Thank you very much for your attention !

Selected references i

Wason, P. C. (1968). Reasoning about a rule. Quarterly Journal of Experimental Psychology, 20(3), 273–281. https://doi.org/10.1080/14640746808400161

Hamblin, C. L. (1973). Questions in montague english. Foundations of Language, 10(1), 41–53.

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

Stalnaker, R. (1974). Pragmatic presuppositions. In Context and content (pp. 47–62). Oxford University Press.

Karttunen, L. (1977). Syntax and semantics of questions. Linguistics and Philosophy, 1(1), 3–44. https://doi.org/10.1007/bf00351935

Groenendijk, J., & Stokhof, M. (1984). Studies in the semantics of questions and the pragmatics of answers (Doctoral dissertation) [(Unpublished doctoral dissertation)]. University of Amsterdam.

Heim, I. (1991). Artikel und definitheit. In A. Von Stechow & D. Wunderlich (Eds.), Semantics: An international handbook of contemporary research (pp. 487–535). Mouton de Gruyter.

Rooth, M. (1992). A theory of focus interpretation. Natural Language Semantics, 1(1), 75–116. https://doi.org/10.1007/bf02342617

Selected references ii

Van Kuppevelt, J. (1995). Main structure and side structure in discourse. Linguistics, 33(4), 809–833. https://doi.org/10.1515/ling.1995.33.4.809

Roberts, C. (1996). Information structure in discourse: Towards an integrated formal theory of pragmatics. Semantics and Pragmatics, 5. https://doi.org/10.3765/sp.5.6

Simons, M. (2001). Disjunction and alternativeness. Linguistics and Philosophy, 24(5), 597–619. https://doi.org/10.1023/a:1017597811833

von Fintel, K. (2001). Conditional strengthening: A case study in implicature [Unpublished manuscript, MIT].

Büring, D. (2003). On d-trees, beans, and b-accents. Linguistics & Philosophy, 26(5), 511–545.

Katzir, R. (2007). Structurally-defined alternatives. Linguistics and Philosophy, 30(6), 669–690. https://doi.org/10.1007/s10988-008-9029-y

Singh, R. (2008). On the interpretation of disjunction: Asymmetric, incremental, and eager for inconsistency. Linguistics and Philosophy, 31(2), 245–260. https://doi.org/10.1007/s10988-008-9038-x

Zondervan, A., & Meroni, A., L.and Gualmini. (2008). Experiments on the role of the question under discussion for ambiguity resolution and implicature computation in adults. Semantics and linguistic theory, 18, 765–777.

Selected references iii

Magri, G. (2009). A theory of individual-level predicates based on blind mandatory scalar implicatures. Natural Language Semantics, 17(3), 245–297. https://doi.org/10.1007/s11050-009-9042-x

Schlenker, P. (2009). Local contexts. Semantics and Pragmatics, 2. https://doi.org/10.3765/sp.2.3

Schoubye, A. J. (2009). Descriptions, truth value intuitions, and questions. Linguistics and Philosophy, 32(6), 583–617. https://doi.org/10.1007/s10988-010-9069-y

Simons, M., Tonhauser, D., J.and Beaver, & Roberts, C. (2010). What projects and why. In D. Lutz & N. Li (Eds.).

Katzir, R., & Singh, R. (2014). Hurford disjunctions: Embedded exhaustification and structural economy. Proceedings of Sinn und Bedeutung, 18, 201–216. https://ojs.ub.uni-konstanz.de/sub/index.php/sub/article/view/313

Heim, I. (2015). Unpublished lecture notes.

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

Kaufmann, M. (2016). Free choice is a form of dependence. Natural Language Semantics, 24(3), 247–290. https://doi.org/10.1007/s11050-016-9125-4

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

Fox, D., & Spector, B. (2018). Economy and embedded exhaustification. Natural Language Semantics.

Mandelkern, M., & Romoli, J. (2018). Hurford conditionals. Journal of Semantics, 35(2), 357–367. https://doi.org/10.1093/jos/ffx022

Westera, M. (2018). A pragmatic approach to hurford disjunctions.

Onea, E. (2019). Underneath rhetorical relations: The case of result. In Questions in discourse (pp. 194–250). BRILL. https://doi.org/10.1163/9789004378322_008

Riester, A. (2019). Constructing qud trees. In Questions in discourse (pp. 164–193). BRILL. https://doi.org/10.1163/9789004378322_007

Marty, P., & Romoli, J. (2022). Varieties of hurford disjunctions. Semantics and Pragmatics, 15(3), 1–25. https://doi.org/10.3765/sp.15.3

Haslinger, N. (2023). Pragmatic constraints on imprecision and homogeneity (Doctoral dissertation). Georg-August-Universität Göttingen.

Hénot-Mortier, A. (2023). Alternatives are blind to some but not all kinds of context: The view from hurford disjunctions. Proceedings of Sinn und Bedeutung 27, 291–308.

Selected references v

Benbaji-Elhadad, I., & Doron, O. (2024). Relevance as a dynamic constraint. Proceedings of the 42nd West Coast Conference on Formal Linguistics (WCCFL42.

Kalomoiros, A. (2024). An approach to hurford conditionals. Semantics and Linguistic Theory, 724–743. https://doi.org/10.3765/68bn3095

Zhang, Y. (2024). A questions-under-discussion-based account of redundancy. Unpublished manuscript.

Ciardelli, I., Groenendijk, J., & Roelofsen, F. (n.d.). Inquisitive semantics. Oxford University Press.

Appendix

Some evidence that $\rightarrow$ and $\vee$ package information differently

- Assuming the structure *Depending on Q, p* (Karttunen, 1977; Kaufmann, 2016), where Q is a question and p a proposition, has to match the cells of Q to the maximal answers of any QuD evoked by p, the contrast (13a) vs. (13b) suggests *France* and *Belgium* can be matched against Q in the disjunctive, but not the conditional case.
- The improvement between (13b) and (13c) also implies that **the answers targeted by depending on Q, when p is conditional, are the ones made available by the consequent of p** (which is appropriately disjunctive in (13c)).

(13) Depending on [how her accent sounds]$_Q$...

a. Mary comes from France **or** Belgium. $\mathbf{p}\vee\mathbf{q}$

b. ?? **if** Mary doesn't come from France she comes from Belgium. $\neg\mathbf{p}\rightarrow\mathbf{q}$

c. ? **if** Mary doesn't come from France she comes from Belgium **or** Québec. $\neg\mathbf{p}\rightarrow(\mathbf{q}\vee\mathbf{r})$

How $\vee$ prevents Q-tree bracketing clashes

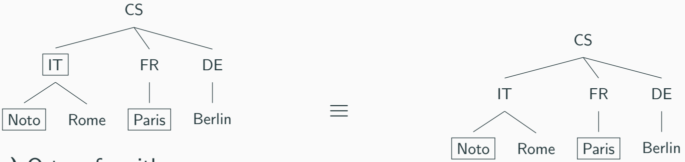
- Two Q-trees T and T' have a bracketing clash if there is $N \in T$ and $N' \in T'$ s.t. $N = N'$ but the sets of children of N and N' differ.
- **If T and T' have such a clash, their disjunction is not a Q-tree:**
  - Let's call C and C' the sets of nodes of resp. T and T' that induce a bracketing clash. C and C' are s.t. $C \neq C'$, and have mothers N and N' s.t. $N = N'$.
  - Because $\vee$ achieves graph-union, $T \vee T'$ will have a node N with $C \cup C'$ as children, and because $C \neq C'$, $C \cup C' \supset C, C'$.
  - And given that both C and C' are partitions of N, $C \cup C'$ cannot be a partition of N.
- A generalization of this property is that, **to be disjoined, T and T' must have parallel structures at least up to a certain point, and any partitionings T and T' independently introduce to dot induce bracketing clashes.**

Q-Redundancy in Long-Distance Hurford Disjunctions

- Long-Distance Hurford Disjunctions differ from standard HDs in that the strong disjunct (e.g. **Noto**) gets further disjoined with an element (e.g. **Paris**) that is incompatible with the weak disjunct (e.g. **Italy**).

(14) # Either Mary lives in **Italy** or she lives in **Noto** or **Paris**. $\mathbf{p} \vee (\mathbf{p}^{+} \vee \mathbf{q})$

(15) # Either Mary lives in **Noto** or **Paris** or she lives in **Italy**. $(\mathbf{p}^{+} \vee \mathbf{q}) \vee \mathbf{p}$



(a) Q-tree for either $\mathbf{p} \vee (\mathbf{p}^{+} \vee \mathbf{q})$ or $(\mathbf{p}^{+} \vee \mathbf{q}) \vee \mathbf{p}$; P^{*}={[CS, IT, Noto], [CS, FR, Paris]}

(b) Q-tree for $\mathbf{p}^{+} \vee \mathbf{q}$; P^{*}={[CS, IT, Noto], [CS, FR, Paris]}

Figure 10: Equivalence between the only Q-tree compatible with (14) or (15) and one Q-tree compatible with the simplification $\mathbf{p}^{+} \vee \mathbf{q}$

Repairing Hurford sentences

- The infelicitous HC (3) is repairable by *at least* (cf. Singh, 2008 & (16)) and *else*-periphrasis (cf. Katzir and Singh, 2014 & (17)).
- Yet, (18), which appears truth-conditionally equivalent to (17), remains redundant. Interestingly, all 3 strategies can fix (1).

(16) If Mary does **not** live in **Noto**, **at least** she lives in **Italy**.

(17) If Mary does **not** live in **Noto**, she lives **somewhere in Italy that is not Noto**.

(18) ?? If Mary does **not** live in **Noto**, she lives in **Italy but** does **not** live in **Noto**.

- We argue the repairs in (16) & (17) modify the consequent Q-tree to make it fine-grained enough to satisfy Q-RELEVANCE.

At least

(16) If Mary does **not** live in **Noto**, **at least** she lives in **Italy**.

- We assume that *at least* takes an antecedent proposition (here, $\neg\mathbf{p}^{+}$) as an extra argument, and returns Q-trees that are "at least" as fine-grained as those of its prejacent (here, $\mathbf{p}$), i.e., trees structurally equal to a Q-tree of $\mathbf{p}$ up to a certain depth, and s.t. any deeper layer corresponds to a partition induced by a set of same-granularity alternatives to $\neg\mathbf{p}^{+}$.
- We also assume *at least* updates the verifying nodes of its input Q-tree s.t. the "verifying" property is recursively passed from a verifying mother node to all its children.
- Q-trees for *at least Italy* then end up looking like Tree 11, and are thus fine-grained enough to satisfy RELEVANCE in conditionals like (16).

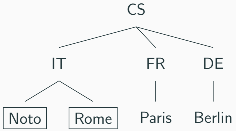


Figure 11: Q-tree for *at least Italy*

Else-periphrasis

(17) If Mary does **not** live in **Noto**, she lives **somewhere in Italy that is not Noto**.

- *somewhere in ... that is not Noto* clearly introduces a sub-country granularity, by e.g. taking Q-trees of $\neg\mathbf{p}^{+}$(cf. Fig. 4), and intersecting the resulting verifying nodes with *Italy*. This again yields city-level Q-trees that are fine-grained enough to satisfy Q-RELEVANCE.

But

(18) ?? If Mary does **not** live in **Noto**, she lives in **Italy** **but** does **not** live in **Noto**.

- We suggest *but not Noto* affects the structure of the candidate Q-tree introduced by *Italy* by adding city-level partitions and marking the *not Noto* leaves as verifying, but crucially, retains the verifying nodes of the original Q-tree (i.e., the *Italy* node).
- This is justified by the idea that a statement of the form *q but q'* can still be felt to answer a question of granularity *q* (e.g., *which country does Mary live in?*), with *but q'* introducing an optional precisification of the answer.
- Because *Italy* is assumed to remain a verifying node in the consequent of (18), RELEVANCE is still violated.