Contents

Chapter 1

Assertions and Questions

Assertions and questions can be seen as the two sides of the same coin, as they form the two core building blocks of any given conversation. Question typically request information, while assertions typically provide information. (1a) for instance, is a question that requests information about the country where Jo grew up (presupposing there is one such country). (1b) can be seen as a good (assertive) answer to this question, providing the piece of information that Jo grew up in France. Semanticists have observed that the pairs formed by questions and answers are restricted: some are obviously good, while some others are (sometimes surprisingly) odd. So, questions and answers have to be somewhat *congruent*. For instance, (1c) cannot be seen as a suitable answer to (1a), even if it seems to indicate something about Jo's nationality.

(1) a. In which country did Jo grow up?
    b. –Jo grew up in France.
    c. # –Jo speaks French natively.

This Chapter motivates and lays the foundation of the main contribution of this dissertation: a constrained machinery "retro-engineering" questions out of assertions, allowing to capture intricate patterns in the domain of pragmatic oddness, that were not previously seen as an issue of question-answer congruence. This Chapter is organized as follows. Section 1.1 provides a broad overview of the semantics of assertions, and discusses to what extent they can meaningfully contribute to a conversation. Section 1.2 turns to the semantics and pragmatics of questions and highlights how questions relate to alternative assertions, and their possible answers. Section 1.3 bridges Sections 1.1 and 1.2, by discussing how questions further constrain which assertions should matter in a given conversation. It also points out a few cases in which question-answer is (seemingly) unhelpful. Section 1.5 constitutes a more technical appendix sketching how the semantics of questions is standardly derived. This whole Chapter heavily builds on the section of von Fintel and Heim (2023) dedicated to Questions.

1.1 Assertions provide information in the form of propositions

1.1.1 Extension and intension of assertions

When studying the semantics of natural language expressions, one usually starts with assertions, because they appear intuitively simpler. We will use the simple assertion in (1b), as a running example. At the most basic level, assertions are truth-conditional, i.e. their meaning corresponds to the set of conditions under which they hold. For instance, *Jo grew up in France* will be true if and only if whoever *Jo* is, grew up in whatever geographical entity *France* is. The *extension* of an assertion is therefore of type $\mathtt{t}$, the type of truth-values.

Additionally, the truth-conditions of a sentence are parametrized by (at least) a world variable.[1]. So, *Jo grew up in France* will be true as evaluated against a world w_0 if and only if whoever *Jo* is in w_0, grew up in w_0 in whatever geographical entity *France* is in w_0. One can then abstract over this world-parameter, and define the *intension* of an assertion as a function from worlds to truth-values. Such functions are called *propositions*, and have type $\langle \mathtt{s}, \mathtt{t} \rangle$, where $\mathtt{s}$ is the type of world-variables. So, the intension, or propositional content of *Jo grew up in France*, will be a function mapping any world variable w, to true if and only if, whoever *Jo* is in w, grew up in w in whatever geographical entity *France* is in w. This is formalized (with some simplifications) in (2).

(2) $$[\![\text{ Jo grew up in France }]\!] = \lambda w.\ \text{Jo grew up in France in } w \\ : \langle \mathtt{s}, \mathtt{t} \rangle$$

Propositions can receive an alternative, equivalent interpretation in terms of sets, based on the idea that any function with domain D and range R is just a (potentially infinite) set of pairs of elements in $D \times R$. A proposition is then simply the set of worlds in which it holds. This interpretation of propositions will be heavily used throughout the dissertation, and is outlined in (3).

(3) $$[\![\text{ Jo grew up in France }]\!] = \lambda w.\ \text{Jo grew up in France in } w \\ \simeq \{ w \mid \text{ Jo grew up in France in } w \ \}$$

[1]Other parameters can also be relevant, like times, and assignments. But we choose to keep things simple here.

1.1.2 Assertions in conversation

Propositions either denote functions of type $\langle \texttt{s}, \texttt{t} \rangle$, or subsets of the set of elements of type s. Should all elements of type s be considered when evaluating such functions, or computing such subsets? It is commonly assumed that the worlds under consideration at any point of a conversation, are the ones that are compatible with the premises of the said conversation (Stalnaker, 1974, 1978). For instance, if two people have a discussion about *France*, it is often reasonable to assume that they agree on what geographical area *France* encompasses, and more generally about the topology of Earth. Moreover, they agree that they agree on this; and agree that they agree that they agree on this; etc. Propositions subject to this recursive, mutual, tacit agreement pattern, form what is called a Common Ground (henceforth **CG**, (Stalnaker, 1978)). Each conversation has its own CG, as defined in (4). The set of worlds in which all the propositions of the CG hold, is called the Context Set (henceforth **CS**). The CS associated with a conversation is therefore a subset of the set of all possible worlds; and can also be seen (under the set interpretation of propositions) as the grand intersection of the propositions in the CG. This is defined in (5).

(4) *Common Ground* (***CG***). Let $\mathcal{C}$ be a conversation between participants $\{P_1, ..., P_k\}$. Let $K(x, p)$ is a proposition meaning that individual x knows p, and p is a proposition. The Common Ground of $\mathcal{C}$ is the set of propositions that are recursively taken for granted by all the participants in $\mathcal{C}$:
$p \in CG(\mathcal{C}) \iff \forall n \in \mathbb{N}^*.\ \forall\{k_1, ...k_n\} \in [1; k]^n.\ K(P_{k_1}, K(P_{k_2}, ...K(P_{k_n}, p)...)$

(5) *Context Set* (***CS***). Let $\mathcal{C}$ be a conversation between participants $\{P_1, ..., P_k\}$. Let CG $CG(\mathcal{C})$ be the Common Ground of this conversation. Under a set interpretation of propositions, the resulting Context Set $CS(\mathcal{C})$ is the set of worlds verifying all propositions of the CG, i.e.:
$CS(\mathcal{C}) = \bigcap\{p \mid p \in CG(\mathcal{C})\}$.

The concepts of CG and CS help delineate which worlds to focus on when evaluating an assertion in context, and determining to what extent this assertion is informative. If uttering an assertion is akin to *adding* it to the CG, then, it also amounts to *intersecting* this assertion with the CS.

(6) *Updating the Common Ground.* Let $\mathcal{C}$ be a conversation, and $CG(\mathcal{C})$ its Common Ground. If a sentence S denoting p is uttered, then p is added to $CG(\mathcal{C})$ to form a new Common Ground $CG'(\mathcal{C})$:
$CG'(\mathcal{C}) = CG(\mathcal{C}) \cup \{p\}$

(7) *Updating the Context Set.* Let $\mathcal{C}$ be a conversation and $CS(\mathcal{C})$ its Context Set. If a sentence S denoting p is uttered, then a new Context Set $CS'(\mathcal{C})$ is derived by intersecting $CS(\mathcal{C})$ with p:
$CS'(\mathcal{C}) = CS(\mathcal{C}) \cap p$

(8) *Link between the two updates.* (7) can be derived from (6) and the definition of the CG in (5):

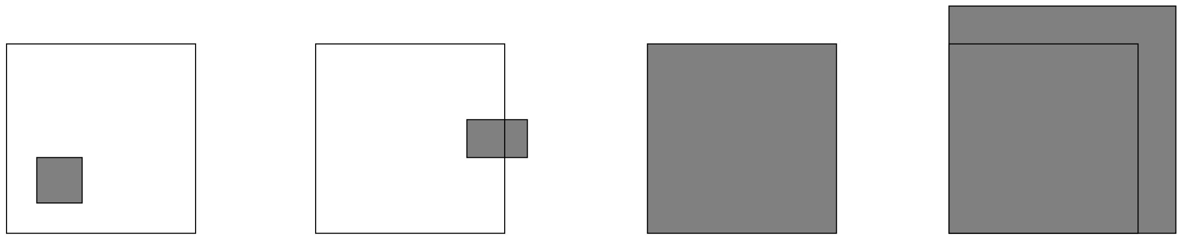
$$\begin{aligned} CS'(\mathcal{C}) &= \bigcap\{q \mid q \in CG'(\mathcal{C})\} \\ &= \bigcap\{q \mid q \in CG(\mathcal{C}) \cup \{p\}\} \\ &= \bigcap\{q \mid q \in CG(\mathcal{C})\} \cap p \\ &= CS(\mathcal{C}) \cap p \end{aligned}$$

Note that updating the CG will always create a bigger set, because the CG is simply a collection of propositions. For instance, if *Jo grew up in Paris* is already in the CG, then, adding the proposition denoted by *Jo grew up in France* to the CG will mechanically expand it. Updating the CS however, does not always lead to a different, smaller CS. For instance, taking for granted that Paris is in France (i.e., all the *Paris*-worlds of the CS are *France*-worlds), and assuming that *Jo grew up in Paris* is already common ground, intersecting the CS with the proposition that *Jo lives in France* will not have any effect. This seems to capture the idea that a proposition like *Jo lives in France* is *uninformative* once it is already known by all participants that *Jo lives in Paris*.

More generally, if it is Common Ground that p, and a sentence S denoting p^- s.t. $p \vDash p^-$ is uttered, then S will feel uninformative. An informative assertion should lead to a non-vacuous update of the CS, i.e. it should properly *shrink* the CS. This is spelled out in (9).

(9) *Informativity (propositional view).* A sentence S denoting a proposition p is informative in a conversation $\mathcal{C}$, iff $CS(\mathcal{C}) \cap p \subset CS(\mathcal{C})$.

In that framework, an assertion provides information in the sense that it reduces the set of live possibilities, and allows to better guess which world is the "real" one. Figure 1.1 illustrates how an asserted proposition can be informative or uninformative, depending on its set-theoretic relationship to the CS.



(a) Informative update (p strictly contained in the CS)

(b) Informative update (p strictly overlaps with the CS)

(c) Uninformative update (p equal to the CS)

(d) Uninformative update (p containing the CS)

Figure 1.1: A few examples of informative and uninformative updates of the CS. The big squares represent the CS. The grey shapes refer to p, the proposition added to the CG (and intersected with the CS to update it).

1.1.3 Dynamic Semantics

So far, we have mainly considered "simplex" assertions that did not make use of operators, connectives or quantifiers. But what about sentences like those in (10)? How should they interact with the Context Set?

(10) a. Jo did not grow up in France.
b. Jo grew up in France or Belgium.
c. Jo grew up in France and Ed in Belgium.

The simplest way to deal with these sentences, would be to compute their intension (the proposition they denote) based on the semantics of negation, disjunction, and conjunction, and then, intersect the resulting proposition with the Context Set. We will call this approach the naive "bulk" CS update. There is evidence, coming from the behavior of presuppositions, that this might not be the way to go, and that complex assertions should be added to the Context Set "bit by bit" (Heim, 1982, 1983; Hei, 1983).

To see this, let us consider the pair in (11). The sentences in (11) are conjunctive and only vary in the order of their conjuncts. Additionally, one of their conjuncts contains the presupposition trigger *too*, associated with the predicate *grew up in France*. In the felicitous variant (11a), *too* occurs in the second conjunct; in the the infelicitous variant (11a), *too* occurs in the first conjunct. Intuitively, *X too VP* imposes that whatever predicate *VP* denotes be true of at least one individual different from the one *X* denotes. This presupposition can be seen as a precondition on the Context Set (as defined prior to the update step). In the case of (11a) and (11a), *Ed too grew up in France* then imposes that the Context Set at the time of the update entail that somebody other than Ed (e.g., Jo) grew up in France.

(11) a. Jo grew up in France, and Ed too grew up in France.

b. # Ed too grew up in France, and Jo grew up in France.

Let us attempt a naive "bulk" CS update with sentences (11a)/(11b). The first step is to compute (11a)/(11b)'s presuppositions and (propositional) assertions. The CS, as defined prior to the utterance of (11a)/(11b), then gets updated, provided that it verifies (11a)/(11b)'s presupposition. Let us start with (11a) and (11b)'s presuppositional component. We can assume that the presupposition that somebody other than Ed grew up in France projects from inside the conjunctive operator. Under this assumption, both (11a) and (11b) end up imposing that the CS prior to their utterance entail that somebody other than Ed grew up in France. This will in principle *not* be verified. So, the naive "bulk" Context Set update correctly predicts the infelicity of (11b), but, also, incorrectly predicts (11a) to be odd. Assuming the presupposition does not project does not address the issue. Under this assumption, both (11a) and (11b) end up being presuppositionless, and the naive "bulk" CS update correctly predicts (11a)'s felicity, but also incorrectly predicts (11b) to be just as felicitous. So, regardless of how presupposition should exactly behave in complex sentences, the asymmetry between (11a) and (11b) does not seem to be captured by the naive "bulk" CS update.

The linear asymmetry in (11) in fact suggests an alternative, "bit by bit" update strategy for complex sentences like conjunctions. If each conjunct were to update the CS one at a time, following the linear order of the sentence, then, the first conjunct of (11a) would create an updated CS that would incorporate the information that *Jo grew up in France*, and as such verify the presupposition of (11a)'s second conjunct (that somebody other than Ed grew up in France). This would allow (11a)'s second conjunct to be subsequently intersected to the CS, and would predict the whole conjunction in (11a) to be felicitous. By contrast, (11b)'s first conjunct would still be problematic in this framework, because its presupposition would not be satisfied by the original CS.

In this toy example, a presupposition was used as a diagnostic to better determine the nature of the CS update triggered by a conjunctive sentence. The conclusion is that the update should be dynamic: the two conjuncts should be intersected with the CS one by one, in the order in which they appear. This should apply to presuppositionless sentences as well; and is summarized in (12).

(12) *Conjunctive update of the CS.* Let $\mathcal{C}$ be a conversation and $CS(\mathcal{C})$ its Context Set. If a sentence S of the form $X \wedge Y$, with $[\![X]\!] = p$ and $[\![Y]\!] = q$ is uttered, then a new Context Set $CS''(\mathcal{C})$ is derived by, first intersecting $CS(\mathcal{C})$ with p to create $CS'(\mathcal{C})$, and second, intersecting $CS'(\mathcal{C})$ with q to create $CS''(\mathcal{C})$:
$CS''(\mathcal{C}) = (CS(\mathcal{C}) \cap p) \cap q = CS'(\mathcal{C}) \cap q$

The potential presuppositions of X and Y are tested on the CS at the time of their respective update, i.e. on $CS(\mathcal{C})$ and $CS'(\mathcal{C})$ respectively.

Dynamic Semantics is a framework that proposes to extend this view to other kinds of complex sentences, e.g. disjunctive and conditional sentences. In Dynamic Semantics, sentences give rise to different kinds of CS updates, depending on how they are constructed. More fundamentally, Dynamic Semantics proposes a shift of perspective when it comes to the meaning of assertions: assertions no longer denote propositions, instead they denote proposals to update the CS in specific ways. In that sense, assertions can be seen as functions from an input CS, to an output CS–sometimes called Context-Change Potentials (**CCP**). CCPs for disjunctive and conditional sentences are spelled out in (13) and (14) respectively.

(13) *Disjunctive update of the CS.* Let $\mathcal{C}$ be a conversation and $CS(\mathcal{C})$ its Context Set. If a sentence S of the form $X \vee Y$, with $[\![X]\!] = p$ and $[\![Y]\!] = q$ is uttered, then a new Context Set $CS'(\mathcal{C})$ is derived by intersecting $CS(\mathcal{C})$ with $p \cup q$:
$CS'(\mathcal{C}) = CS(\mathcal{C}) \cap (p \cup q)$
The potential presuppositions of X and Y are tested on, respectively, $CS(\mathcal{C})$ and $CS(\mathcal{C}) \cap \neg p$.[2]

(14) *Conditional update of the CS.* Let $\mathcal{C}$ be a conversation and $CS(\mathcal{C})$ its Context Set. If a sentence S of the form *if X then Y*, with $[\![X]\!] = p$ and $[\![Y]\!] = q$ is uttered, then a new Context Set $CS''(\mathcal{C})$ is derived by, first intersecting $CS(\mathcal{C})$ with p to create $CS'(\mathcal{C})$, and second, intersecting $CS'(\mathcal{C})$ with q to create $CS''(\mathcal{C})$:
$CS''(\mathcal{C}) = (CS(\mathcal{C}) \cap p) \cap q = CS'(\mathcal{C}) \cap q$
The potential presuppositions of X and Y are tested on the CS at the time of their respective update, i.e. on $CS(\mathcal{C})$ and $CS'(\mathcal{C})$ respectively.

This incremental view of assertions leads to a revised, incremental definition of informativity, given in (15).

(15) *Informativity (CCP view).* A sentence S is informative in a conversation $\mathcal{C}$, iff all the updates of $CS(\mathcal{C})$ it gives rise to are non-vacuous.

In sum, assertions can be seen as proposals to update (shrink) the CS. The specific update they give rise to is compositionally derived, and incrementally performed, following the structure of the sentence. We will use a similar approach in Chapter **??** when defining

[2]There is a debate on whether or not disjunctions should behave symmetrically w.r.t. the presupposition(s) carried by their disjuncts. An alternative, symmetric way to evaluate X and Y's potential presuppositions, would be to test them against $CS(\mathcal{C}) \cap \neg q$ and $CS(\mathcal{C}) \cap \neg p$ respectively.

questions *evoked* by assertions. But this first requires to define what questions mean. This is what we do in the next section, in which we show that questions influence, not the size, but rather, the topology of the CS.

1.2 Questions indicate which kind of information is worth providing

1.2.1 Questions as answerhood conditions

Participants in a conversation utter assertions to shrink the CS, and hopefully, jointly figure out which world they are in. But this allows for very unnatural interactions like (16), taking the forms of sequences of intuitively unrelated sentences–as long as each of them denotes propositions shrinking the CS!

(16) –Jo grew up in France.
–I like cheese.
–Al is arriving tomorrow.

This is where questions enter the game. Intuitively, a question indicates an interest in *which* proposition(s) hold, among a restricted set. The proposition at stake are typically possible answers to the question. Questions therefore denote sets of sets of worlds (equivalent to a type $\langle\langle \mathtt{s}, \mathtt{t}\rangle, \mathtt{t}\rangle$), and constrain which kind of (informative) propositions can be uttered as a follow-up. For instance, a polar question such as *Is it raining?* will typically request information of the form *It is raining*, or *It is not raining*, see (17).

(17) –Is it raining?
–Yes, it is raining. / No, it is not raining.

The question *Is it raining?* can thus be represented as a set made of two propositions, namely, the proposition that *it is raining*, and the proposition that *it is not raining*.

(18)
$$\begin{aligned} [\![\text{ Is it raining? }]\!] &= \{ [\![\text{ It is raining }]\!], [\![\text{ It is not raining }]\!] \} \\ &= \{\lambda w.\ \text{it is raining in } w,\ \lambda w.\ \text{it is not raining in } w\} \\ &: \langle\langle \mathtt{s}, \mathtt{t}\rangle, \mathtt{t}\rangle \end{aligned}$$

In the case of the question *is it raining?*, the set of possible answers is fairly simple: it only contains two elements.These two elements cover the space of all possibilities,[3] and

[3]This is the case assuming there is no vagueness-induced "grey area", i.e. any salient situation is either a *raining*-situation, or a *not raining*-situation

are *exclusive*: if it's the case that it's raining (at a salient place, at a salient time) in w, then, it's not the case that it is not raining (at the same place, at the same time), in w. We will see in the next section that this configuration amounts to a partition of the CS. A definition of exclusivity under the set interpretation of propositions is given in (19).

(19) *Exclusive propositions.* $p : \langle \mathtt{s}, \mathtt{t} \rangle$ and $q : \langle \mathtt{s}, \mathtt{t} \rangle$ are exclusive if $p \cap q = \emptyset$.

But questions may not always intuitively request information about exclusive propositions. For instance, a *wh*-question like *Which students passed the class?* expects answers that convey a subset of students who passed the class, see (20). But there are many possible, overlapping subsets of students, so, the corresponding propositions will be overlapping as well. For instance, the proposition that *Jo passed the class*, denotes the set of worlds in which Jo passed the class, and this set happens to contain the set of worlds where both Jo and Al passed the class. It also overlaps with the set of worlds in which Al passed the class.

(20) Which students passed the class?
–Jo did.
–Al did.
–Jo and Al did.

We will call propositions like *Jo passed the class*, and *Jo and Al passed the class*, alternatives associated to the question *Which students passed the class?* Alternatives may be overlapping; and, as we will see, can be obtained from the original question by substituting its *wh*-component (e.g., *which students*), with relevant, same-type material (e.g., students or groups of students).[4]

(21) Question : $[\![$ Which students passed the class? $]\!]$
Alternatives: $\{[\![$ Jo passed $]\!]$, $[\![$ Al passed $]\!]$, $[\![$ Jo and Al passed $]\!]$ ... $\}$

Why would this overlap between alternative answers be an issue in modeling the meaning of questions? The fact that entailing or merely overlapping propositions should be considered equally good answers does not capture the idea that more specific propositions constitute more exhaustive answers than less specific ones. For instance, answering that *Jo passed*, in theory leaves the fate of the other students undecided–for instance, it does

[4]It is worth mentioning that the set $\{\lambda w.$ it is raining in $w,\ \lambda w.$ it is not raining in $w\}$ does not strictly speaking correspond to the set of alternatives raised by *Is it raining?* Section 1.5 further specifies how alternatives get compositionally derived, and predicts that *Is it raining?* should only give rise to one alternative: $\lambda w.$ it is raining in w. The set $\{\lambda w.$ it is raining in $w,\ \lambda w.$ it is not raining in $w\}$ is derived from this singleton alternative *via* the "pragmatic" process presented in (24), in the next Section.

not settle if *Al passed*, or not. Answering that *Jo and Al passed* by contrast, settles Al's fate, in addition to Jo's. Ideally, an answer to *Which students passed?* should explicitly address whether *each* student of the class passed, or not. That would be an exhaustive answer.

1.2.2 Questions as partitions of the Context Set

We have just discussed that, at the semantic level, questions characterize the conditions under which they are answered, i.e. denote a set of potentially overlapping propositions. But, just like we did with assertions, the effect of this semantics on the Context Set has to be defined. There is in fact a deterministic way to change a set of overlapping propositions P (i.e. a set of subsets of the CS), into a set of exclusive subsets of the CS (called *cells*, for reasons made clear in (24)). To do so, one can group in the same cell the worlds of the Context Set that all "agree" on all propositions in P. This "agreement" property amounts to the same-cell relation in (22). This relation is reflexive, symmetric and transitive, i.e. is an equivalence relation (see proof in (23)). From this, we can conclude that the set of subsets of the CS induced by P, obtained by grouping worlds of the CS according to the same-cell relation, forms a partition of the Context Set (see proof in (24)).[5] So, on top of being exclusive, cells are non-empty and together cover the CS. We assume that the process changing the set of alternative propositions raised by a question, to a partition of the CS, belongs to pragmatics. So, questions *denote* sets of alternative propositions, and this set *pragmatically induces* a partition structure on the CS.

(22) *Same-cell relation $\equiv_P$.* Let P be a set of propositions, i.e. a set of subsets of the Context Set ($P \in \mathcal{P}(\mathcal{P}(CS))$, with $\mathcal{P}$ the powerset operation). Let w and w' be two worlds of the Context Set. $w \equiv_P w'$ iff, $\forall p \in P.\ p(w) = p(w')$.

(23) $\equiv_P$ is an equivalence relation, no matter what P is. Let $\forall P \in \mathcal{P}(\mathcal{P}(CS))$.

- $\equiv_P$ is reflexive: $\forall w \in CS.\ \forall p \in P.\ p(w) = p(w)$.
- $\equiv_P$ is symmetric. Let $\forall (w, w') \in CS^2$.
  $\forall p \in P.\ p(w) = p(w')$ iff $\forall p \in P.\ p(w') = p(w)$.
- $\equiv_P$ is transitive. Let $\forall (w, w', w'') \in CS^3$.
  We assume $\forall p \in P.\ p(w) = p(w')$ and $\forall p \in P.\ p(w') = p(w'')$.
  Let $\forall p \in P$. We have $p(w) = p(w')$ and $p(w') = p(w'')$, so $p(w') = p(w'')$.
  So, $\forall p \in P.\ p(w) = p(w'')$

[5] Cells as we defined them are also called equivalence classes. It's a general property that equivalence classes induced by an equivalence relation on a certain set on which this relation is defined, will create a partition of the set.

(24) *Partition of the CS induced by P.*[6] Let P be a set of propositions. The partition induced by P in the Context Set is the set of subsets of the CS (cells): $\mathfrak{P}_{P,CS} = \{\{w' \mid w' \in CS \wedge w' \equiv_P w\} \mid w \in CS\}$. This set partitions the CS.

- No cell c of $\mathfrak{P}_{P,CS}$ is empty. Let $c \in \mathfrak{P}_{P,CS}$. There is a $w \in CS$ s.t. $c = \{w' \mid w' \in CS \wedge w' \equiv_P w\}$. Then at least $w \in c$, because $w \equiv_P w$.
- Cells cover the CS. Let $w \in CS$. $\mathfrak{P}_{P,CS}$ contains a cell $c = \{w' \mid w' \in CS \wedge w' \equiv_P w\}$. Then $w \in c$ because $w \equiv_P w$.
- Cells are disjoint. Let $(c, c') \in \mathfrak{P}_{P,CS}$, s.t. $c \cap c' \neq \emptyset$. We show $c = c'$. c and c' have resp. the form $c = \{w'' \mid w'' \in CS \wedge w'' \equiv_P w\}$ and $c = \{w'' \mid w'' \in CS \wedge w'' \equiv_P w'\}$, for $(w, w') \in CS^2$. Let $w''' \in c \cap c'$. Then $w''' \equiv_P w$ and $w''' \equiv_P w'$, and so by symmetry and transitivity, $w \equiv_P w'$, and $c = c'$.

It is easy to show that, in the polar example (17), the subsets of the CS defined by *It is raining* and *It is not raining*, which we said were intuitive answers to the question, form a partition of the CS. Section 1.5 will in fact show that polar questions of the form $p?$ *denote* the singleton set formed by p, and *induce* a 2-cell partition of the form $\{p, \neg p\}$.

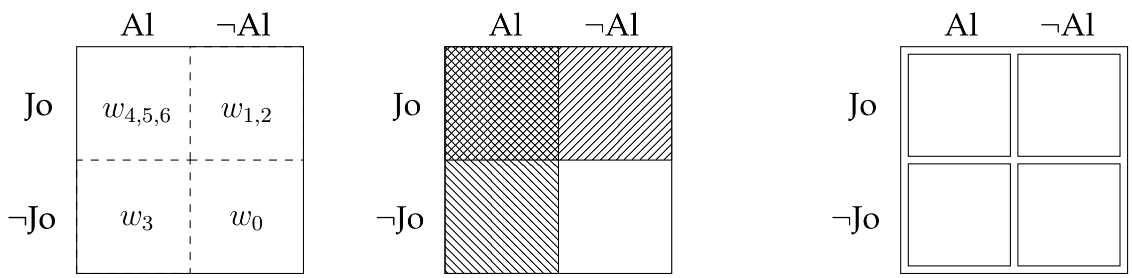
Let us now see how the above definitions apply to a *wh*-question like *Which students passed?* in (20). Let's assume there are only two salient students, Jo and Al. We assume that the alternatives the question raises (labeled P), are the proposition that *Jo passed*, and the proposition that *Al passed*. We assume that the CS contains six possible worlds, which vary according to whether Jo, Al, both, or none passed the class. The worlds may vary in other respects, that are not relevant to us here. The alternatives and cells associated with this question are given in (25). The alternative set P then corresponds to two subsets of the CS, which do not cover it. In particular, the world in which nobody passed (w_0) is included in none of the two alternatives. Moreover, the two subsets are overlapping: both *Jo passed* and *Al passed* contain w_4, w_5, and w_6. Now turning to the cells induced by P on the CS, we notice that there are four of them, which correspond to worlds where nobody, only Jo, only Al, or both Jo and Al passed the class. Such cells cover the CS, are disjoint, and non-empty, so correctly form a partition of the CS. They also fully specify, for *both* Jo and Al, if they passed the class; and as such constitute exhaustive answers to the original question.

(25) Question : *Which students passed the class?*
Context Set: $\{w_0, w_1, w_2, w_3, w_4, w_5, w_6\}$, s.t.:

[6] *Fox (2018) proposes an alternative way to derive a partition of the CS from a set of alternative propositions, leveraging the covert operator exh*

- Nobody passed in w_0;
- Only Jo passed in w_1 and w_2;
- Only Al passed in w_3;
- Both Jo and Al passed in w_4, w_5, and w_6.

Alternatives (P): $\{[\![\text{Jo passed}]\!], [\![\text{Al passed}]\!]\} = \{\{w_1, w_2, w_4, w_5, w_6\}, \{w_3, w_4, w_5, w_6\}\}$
Cells induced by $\equiv_P$: $\{\{w_0\}, \{w_1, w_2\}, \{w_3\}, \{w_4, w_5, w_6\}\}$



(a) Distribution of $w_0 ... w_6$ in the CS.

(b) Alternatives: defines *Al passed* and defines *Jo passed*

(c) Partition induced by and . Cells correspond to the subsets of Figure 1.2b featuring the same overall pattern.

Figure 1.2: Partitioning of the CS defined in (25) according to the alternatives *Jo passed* and *Al passed*. The CS is organized as follows: counter-clockwise, quadrant I is made of *Jo but not Al*-worlds; quadrant II, *Jo and Al*, quadrant III, *Al but not Jo*, and quadrant IV, *neither Jo nor Al*.

To summarize, at the pragmatic level questions are partitions of the Context Set, as formalized in (26).[7] The cells of such partitions constitute maximal answers to the questions. Unions of two or more cells constitute non-maximal answers, as defined in (27).

(26) *Standard semantics for questions* (*Jäger, 1996; Hulstijn, 1997; Groenendijk and Stokhof, 1984; Groenendijk, 1999*). Given a conversation $\mathcal{C}$ and a Context Set $CS(\mathcal{C})$, a question on $CS(\mathcal{C})$ is a partition of $CS(\mathcal{C})$, i.e. a set of subsets of $CS(\mathcal{C})$ ("cells") $\{c_1, ..., c_k\}$ s.t.:

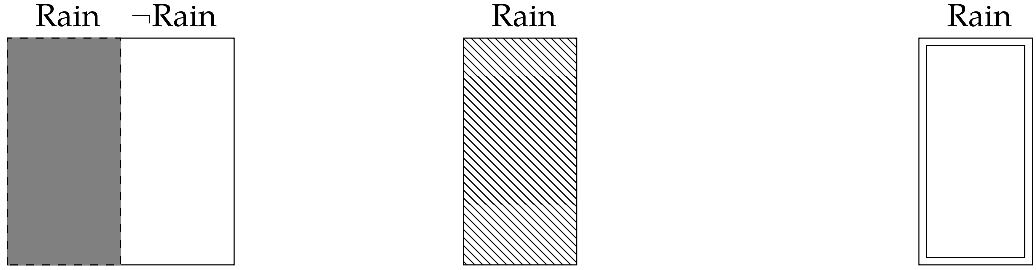
- "No empty cell": $\forall i \in [1;k].\ c_i \neq \emptyset$
- "Full cover": $\bigcup_{i \in [1;k]} c_i = CS(\mathcal{C})$
- "Pairwise disjointness": $\forall (i,j) \in [1;k]^2.\ i \neq j \Rightarrow c_i \cap c_j = \emptyset$

[7] It is important to note that questions may be taken to have a partition *semantics*. But we do not cover this here.

(27) *Maximal and non-maximal answers to a question.* Given a conversation $\mathcal{C}$, a Context Set $CS(\mathcal{C})$, and a question Q forming a partition $\{c_1, ..., c_k\}$ of $CS(\mathcal{C})$:

- Any $c \in \{c_1, ..., c_k\}$ constitutes a maximal answer to Q;
- Any c' s.t. $\exists C \subseteq \{c_1, ..., c_k\}.\ |C| > 1 \wedge c' = \bigcup C$ is a non-maximal answer to Q.

Just like we did with assertions, let us clarify further what it means to be a good question. We have established that the idea of a partition is a good candidate to model the effect of questions on a given CS. But what if the CS is already such that the partition induced by the question's alternatives is just made of one big cell? Such a configuration suggests that the question is already *settled*, meaning, the CS already makes one maximal answer trivial. For instance, if it is already common ground between the conversation's participants that *it is raining* (at the salient place and time) in (17), then, the question *Is it raining?* appears completely trivial. This is illustrated in Figure 1.3 and generalized in (28).



(a) Interecting the proposition that *It's raining* ( ) with a CS that is agnostic about the weather.

(b) Alternatives on the restricted CS: defines *It's raining*.

(c) Partition induced by on the CS. Cells correspond to the subsets of Figure 1.3b featuring the same overall pattern (1 such subset).

Figure 1.3: Updating the CS with the proposition that *It's raining*, and then computing the partition induced by *Is it raining* on the resulting "shrunk" CS. The outcome is a single-cell partition, i.e., the question has a trivial pragmatics.

(28) *Trivial question.* Let $\mathcal{C}$ be a conversation, $CS(\mathcal{C})$ its associated Context Set, and Q a question. Q is trivial given $CS(\mathcal{C})$ iff the partition induced by Q on $CS(\mathcal{C})$ is made of a singleton cell, i.e. has cardinal 1.

We now have a basic notion of what it mean to be a good assertion, given a CS, and a good question, given a CS. A good assertion has to be informative, i.e. properly shrink the CS (as per (9)/(15)). A good question has to induce a non-trivial, multiple-cell partition on the CS (as per (28)). But being a good question or a good assertion, does not *only* depend on the state of the CS! In particular, good assertions also have to be good answers to good questions. This principle, dubbed *Question-Answer Congruence*, is given in (29).

(29) *Question-Answer Congruence* (*Katzir and Singh, 2015*). A felicitous assertion has to be a good answer to a good question.

The next Section presents what can be seen as a partial implementation of this principle, in the form of a general principle dubbed RELEVANCE. It also points out the limitations of this principle.

1.3 Assertions as good answers to questions

1.3.1 Relevance mediate questions and assertions

Now that we precisified what assertions and questions are, it becomes possible to (at least partially) define what a good assertion should be, given a question. The principles we introduce in this Section are based on the general concept of RELEVANCE. They will eventually rule out informative but "unnatural" sequences of assertions like (16), but also, more generally, a wide range of odd question-answer pairs.

Following much previous literature (van Kuppevelt, 1995a,b; Roberts, 1996, 2012; Ginzburg, 1996; Büring, 2003), we call the question against which assertions are evaluated, *Question under Discussion* (henceforth **QuD**). QuDs are typically seen as partitions of the CS. In (27), we defined cells an unions of cells as respectively maximal and non-maximal answers to a question. Very broadly, RELEVANCE constrains what a proposition should do to the cells of the QuD. Let us now unpack this with an example.

If for instance the QuD is about which country Jo grew up in (as in (30)), the CS will be partitioned according to propositions of the form *Jo grew up in c*, with *c* a country. Utterances such as (30a) or (30b), both seem relevant to that kind of QuD, and both constitute answers to the QuD–maximal, or not. By contrast, utterances such as (30c), (30d) or (30e), do not appear relevant, and do *not* constitute answers to the QuD: there are native and non-native French speakers in virtually all countries; same holds for wine-lovers and wine-haters; as for (30e) it seems completely independent from the subject matter.[8] These various configurations are sketched in Figure 1.4.

(30) QuD: In which country did Jo grow up?

a. Jo grew up in France.

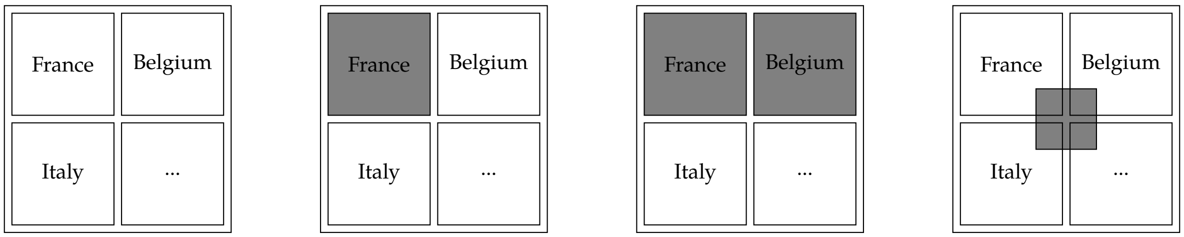
[8]It is interesting to note that (30c) and (30d) can be more easily coerced into relevance than (30e). For instance with (30c), one might consider that France is the country which, in proportion, comprises the most native French speakers, and so (30c) may be understood as *It is likely that Jo grew up in France*–which constitutes a modalized answer to the QuD. This kind of reasoning is harder (if not impossible) to perform when facing an utterance like (30e).

b.   Jo grew up in France or Belgium.

c. ?? Jo speaks French natively.

d. ?? Jo enjoys wine.

e. # The cat went outside.



(a) QuD for *In which city did Jo grow up?* (b) Utterance: *Jo grew up in France.* (c) Utterance: *Jo grew up in France or Belgium.* (d) Utterance: (30c), (30d) or (30e).

Figure 1.4: QuD-utterance configurations for a QuD like *In which country did Jo grow up*, and possible follow-up utterance.

From this, we can conclude that a proposition is "relevant" to a question, if it constitutes a maximal or a non-maximal answer to the question. This is similar in spirit to the notion of *Aboutness* developed Lewis (1988), according to which a proposition p is about a subject matter (in modern terms, a QuD), if and only if the truth value of that proposition supervenes on that subject matter (i.e. p should not introduce truth-conditional distinctions between cellmates, i.e. p does not "cut across" cells). This is rephrased in (31). A typical Lewis-relevant configuration is exemplified in Figure 1.6a. It is also worth mentioning that (31) deems propositions contradicting the CS irrelevant (see Figure 1.6b), but in principle does not rule out propositions covering the whole CS (see Figure 1.6c), i.e. uninformative propositions as per (9).

(31) Lewis's Relevance (rephrased in the QuD framework). Let $\mathcal{C}$ be a conversation, Q a QuD defined as a partition of $CS(\mathcal{C})$. Let p be a proposition. p is Lewis-relevant to Q, iff $\exists C \subseteq Q.\ p \cap CS(\mathcal{C}) = C$

But, coming back to the QuD *In which country did Jo grow up?*, what about an utterance of the form *Jo grew up in Paris?* Although overinformative (the QuD was only asking about countries, not cities!), this utterance appears relevant, because it allows to infer that Jo grew up in France, and not, say, Belgium. This kind of configuration is sketched in Figure 1.5.

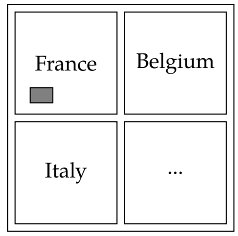
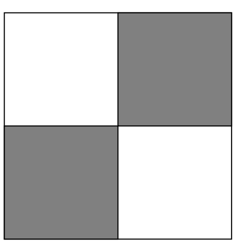


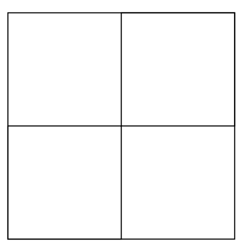
Figure 1.5: QuD-utterance configuration for a QuD like *In which country did Jo grow up?* and an utterance like *Jo grew up in Paris*.

The view of relevance, developed by Roberts (2012), captures this intuition, by stating that a relevant proposition has to rule out at least one maximal answer conveyed by the QuD. In other words, a relevant proposition has to be incompatible with at least one cell of the QuD. This is summarized in (32). This definition makes uninformative propositions irrelevant (see Figure 1.6c), but allows certain propositions that do not coincide with the grand union of a subset of the QuD's cells, to be relevant (see Figures 1.6d and 1.6e). In other words, relevant propositions in the sense of Roberts may introduce truth-conditional distinctions between cellmates–as long as they rule out a cell. A particular case is that of propositions like *Jo grew up in Paris*, when the QuD is about countries, which strictly entail a specific cell of the QuD, i.e. are strictly contained in one single cell (see Figure 1.6d).

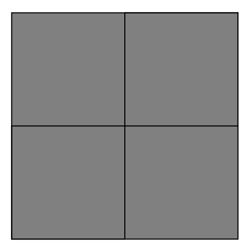
(32) Roberts's Relevance (Roberts, 2012). Let $\mathcal{C}$ be a conversation, Q a (non-trivial) QuD defined as a partition of $CS(\mathcal{C})$. Let p be a proposition. p is Roberts-relevant to Q, if $\exists c \in Q.\, p \cap c = \emptyset$.



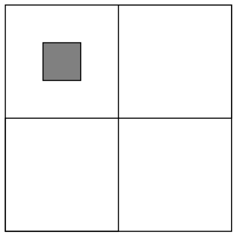
(a) Informative, Lewis-relevant, Roberts-relevant



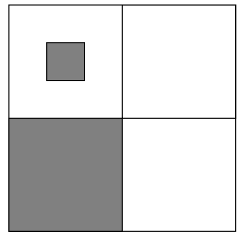
(b) Informative, not Lewis-relevant, Roberts-relevant



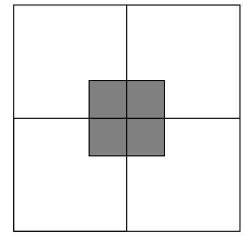
(c) Uninformative, Lewis-relevant, not Roberts-relevant.



(d) Informative, not Lewis-relevant, Roberts-relevant



(e) Informative, not Lewis-relevant, Roberts-relevant



(f) Informative, not Lewis-relevant, not Roberts-relevant

In sum, the concept of Relevance (whether it follows Lewis's or Roberts's implementation) allows to rule-out a wide range of QuD-utterance pairs, by stating that propositions should properly relate to an existing question. We will not discuss which approach between Lewis's and Roberts's is best here, and will propose an incremental variant of this core concept in Chapter **??**, to deal with certain complex, out-of-the blue sentences. The next two section outline a few limitations of relevance.

1.3.2 A few conceptual shortcomings of Relevance

Regardless on which view of Relevance is adopted, relevant propositions can be added to the Common Ground, and as such, trigger an update of the CS. This, in turn, updates the QuD, which must remain a partition of the CS. It is easy to show, given how partition are "induced" on a set (see definition (24)), that the updated QuD on the smaller CS corresponds to the previous QuD, whose cells are pointwise intersected with the newly added proposition, and such that empty cells are filtered. This is formalized in (33).

(33) *Updating the partitioned Context Set.* Let $\mathcal{C}$ be a conversation, $CS(\mathcal{C})$ its Context Set, and let Q be a partition of $CS(\mathcal{C})$. If a sentence S denoting p is uttered and relevant given Q (as per (31) or (32)), then a new Context Set $CS'(\mathcal{C})$ is derived by intersecting $CS(\mathcal{C})$ with p, and this new context set is partitioned by Q', s.t.:
$Q' = \{c' \mid \exists c \in Q.\, c' = c \cap p \wedge c' \neq \emptyset\}$

There are two shortcomings to the current framework. First, adding a proposition to the CG "mechanically" leads to an update of the CS and of the QuD, but does not directly affect the *structure* of this QuD: even if some cells should shrink, the *limits* of each cell remain the same. This goes against the intuition that sometimes, sentences give rise rise to brand new QuDs, as exemplified by the exchange in (34).

(34) –Is it raining?
–Yes, I think so. I just so Ed come in with this very pretty umbrella.
(Likely follow-up: Where did Ed find this umbrella?)

Second, and relatedly, one can wonder what is supposed to happen in the case of out-of-the-blue sentences, i.e. sentences for which there is no explicit QuD. In such cases, it is generally assumed that a reasonable QuD is somehow inferred. But, given the fact that a QuD is merely a partition of the current CS, there exists many options. This dissertation will focus on how exactly QuDs are inferred, what additional constraints hold between an assertion and a QuD, and what the consequences are for pragmatic theory.

1.3.3 Relevance and the packaging of information

We start by showing that the felicity of disjunctions and conditionals is sensitive to *overt* QuDs – but in different ways. We take this as evidence that out-of-the-blue disjunctions and conditionals accommodate different kinds of implicit QuDs.

If a context contrasting *Paris* and *France but not Paris* is set as in (35), (**??**) and (**??**) improve (see Haslinger (2023) for similar effects on disjunctions and conjunctions). This is strange: even if the context and question made *Paris* (but no other French city) a relevant alternative to *France*, *exh* would remain IW in the consequent of (**??**): *if Jo did not grow up in Paris, she grew up in France but not Paris*, is equivalent to *if Jo did not grow up in Paris, she grew up in France*. In other words, *exh* (as constrained by IW) cannot leverage the contextually provided alternatives to make (**??**) escape SR in (35). The same applies to (**??**).

(35) *Context: French accents vary across countries and between Paris the rest of France.*
Al: I'm wondering where Jo learned French.
Lu: I'm not completely sure but... (**??**) ✓(**??**) ✓

This suggests that a purely LF-based view of redundancy such as SR, may be insufficient to capture the interaction between HCs and how their context of utterance packages information. Rather, it seems that the context of (35) makes a specific partition of the CS salient, and that this partition can be used to make otherwise infelicitous assertions accommodate a different question than the one they would evoke out-of-the-blue.

Additionally, conditionals and disjunctions seem to accommodate distinct QuDs. To show this, we use the construction *depending on Q, p* (Karttunen (1977); Kaufmann (2016)), where Q is a question and p a proposition. This construction has been argued to force the partition conveyed by Q to match specific live issues raised by p. We understand such "live issues" as the maximal true answers of the QuD evoked by p. The contrast between (36a) and (36b) then suggests that the *France* and *Belgium* answers can be matched against Q in the disjunctive, but not in the conditional case. This in turn means that a disjunction introduces a QuD making both disjuncts maximal true answers, while a conditional does not do the same with its consequent and the negation of its antecedent.

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

a. Jo grew up in France **or** in Belgium. $\mathbf{p}\vee\mathbf{q}$

b. ?? **if** Jo didn't grow up in France she grew up in Belgium. $\neg\mathbf{p}\to\mathbf{q}$

c. ? **if** Jo didn't grow up in France, she grew up in Belgium **or** in Québec. $\neg\mathbf{p}\to(\mathbf{q}\vee\mathbf{r})$

d. ?? **if** Jo didn't grow up in France **or** Belgium, she grew up in Québec. $\neg(\mathbf{p}\vee\mathbf{q})\to\mathbf{r}$

The existence of an improvement between (36b) and (36c), and the absence of a similar improvement in between (36b) and (36d), 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 (36c), but not (36d)).

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.

1.4 Roadmap of the dissertation

Specifically, we will claim that instead of being a "good" answer to *some* QuD, an out-of-the-blue sentence must be a good answer to a *good* QuD, following insight by Katzir and Singh (2015). We will show that operationalizing this principle allows to account for a wider range of oddness phenomena, that previous approaches struggled to capture under the same umbrella.

"Good" QuDs are determined from the shape of the assertive sentence itself. This is pushing the idea that assertions evoke alternatives one step further, in the sense that sentences will be taken to evoke questions (themselves derived from alternatives). These evoked questions will have a structure that consists in a generalization of the partition structure, namely, they will take the form of parse tree of the CS. We will additionally claim the process deriving good questions from good answers, is subject to constraints that go beyond relevance, and cover concepts such as redundancy. These constraints will make way for a "lifted" view of pragmatic oddness, under which an assertion is not odd *per se*, but rather, is odd due to its interaction with the QuDs it evokes. Before presenting the core components of our model, we will briefly present two recent accounts of oddness based on similar ideas.

1.5 Appendix: computing questions from propositions

So far, we have described what could be a reasonable model for questions, in the form of partitions of the CS. But this was done without explaining how exactly such partitions are derived from the Logical Form of questions. This sketches how this is done, while further clarifying the distinction between propositions, alternatives, and questions. We will show that questions are standardly derived from closely related propositions, by abstracting over specific variables.

We will use the question *In which country did Jo grow up?* as an example. The LF associated with this question is given in Figure 1.7.

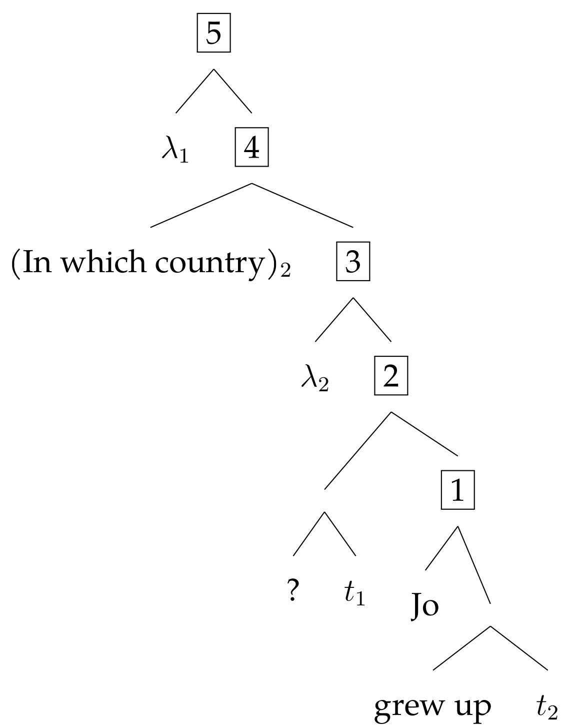


Figure 1.7: LF of the question *In which country did Jo grow up?*

This question involves a *wh*-phrase (*in which country*), which syntactically originates in an adjunct of *grow up*. It is assumed that the *wh*-phrase leaves a trace t_2 in this position. The semantics assigned to the *wh*-phrase is existential, and akin to *some country*. Specifically, *in which country* takes a predicate of type $\langle \mathtt{e}, \mathtt{t} \rangle$ as argument, and returns the quantified statement that *some country* verifies the predicate.

(37) $[\![\text{In which country}]\!]^w = \lambda P.\ \exists l.\ l \text{ is a country in } w \wedge P(l) = 1$

The *wh*-phrase outscopes another "proto-question" operator (Karttunen, 1977). This operator takes two propositions (here, the trace t_1 and the proposition that *Jo grew up in* t_2), and simply equates them.

(38) $[\![?]\!]^w = \lambda p.\ \lambda q.\ p = q$

Applying this operator successively to t_1 and the intension of $\boxed{1}$, yields the following.

(39) $\boxed{1} = [\![\text{Jo grew up } t_2]\!]^w = 1$ iff Jo grew up in t_2 in w

(40) $\boxed{2} = [\![?\ t_1 \text{ Jo grew up } t_2]\!]^w = 1$ iff $t_1 = \lambda w'.$ Jo grew up in t_2 in w'

Abstraction then applies to $\boxed{2}$, binds t_2 and yields a predicate that can then serve as an argument of the *wh*-phrase. The *wh*-phrase then turns this predicate into an existentially quantified expression targeting the element being questioned (here, a country).

(41) $\boxed{3} = [\![\ \lambda_2\ ?\ t_1 \text{ Jo grew up } t_2]\!]^w = \lambda l.\ t_1 = \lambda w'.$ Jo grew up in l in w'

(42) $\boxed{4} = [\![$ In which country ... Jo grew up $t_2]\!]^w$
$= \exists l.\ l$ is a country in $w \wedge t_1 = \lambda w'.$ Jo grew up in l in w'

Lastly, a t_1 gets bound to produce a set of propositions, namely, the set of propositions that coincide with the proposition that *Jo grew up in l*, for some country l.

(43) $\boxed{5} = [\![\ \lambda_1$ In which country ... Jo grew up $t_2]\!]^w$
$= \lambda p.\ \exists l.\ l$ is a country in $w \wedge p = \lambda w'.$ Jo grew up in l in w'
$\simeq \{p \mid \exists l.\ l$ is a country in $w \wedge p = \lambda w'.$ Jo grew up in l in $w'\}$

This example showed that the semantics of a question is derived from that of its "assertive counterpart", where the *wh*-phrase is replaced by a quantified variable. Combined with the proto-question operator and λ-abstraction, this allows to generate a set of propositions, which only vary in terms of the variable being questioned. This set of propositions (alternatives) can then be used to induce a partition of the CS, as per (**??**).

Chapter 2

Accommodating QuDs: Qtrees

This Chapter introduces a model of questions that is more sophisticated than standardly assumed (cf. Chapter 1). Questions are defined as recursive partitions, or parse trees of the Context Set. This model is shown to capture fine-grained information about how questions relate to each other in terms of specificity, and what it means to answer a question. The Chapter then describes how such questions can be "retro-engineered" from assertions, in a compositional way. Lastly, we suggest how this model of questions can eventually make novel predictions in the domain of pragmatic oddness.

2.1 Making sense

2.1.1 Oddness despite relevance and informativeness

In Chapter 1, we have seen that assertive sentences should be informative, i.e. lead to an incremental shrinkage of the Context Set (**CS**) (Stalnaker, 1978; Heim, 1982). We have also seen that they should be relevant, i.e. shrink the CS in a way consistent with the Question under Discussion (**QuD**) (Lewis, 1988; Roberts, 2012). But sometimes, it is unclear what the QuD should be. For instance, the exchange in (44) already settles the overt QuD (*Have you seen Jo today?*), and intersects the CS with the set of worlds in which Ed has not seen Jo on the day that *today* refers to. But one could imagine many possible continuations to Ed's utterance. Any such continuation should be informative and relevant to *some* QuD, but it is unclear how this QuD should be determined. In principle, it could be any non-vacuous partition of the newly updated CS. But there are many such partitions. How to know which one to pick?

(44) Al: Have you seen Jo today?
Ed: No I haven't...

Let us consider the following felicitous follow-up to (44). This continuation is felicitous, so, should be both informative and relevant. To be relevant, the sentence has to relate to a QuD. But, as mentioned earlier, the overt QuD *Have you seen Jo?* is at that point already settled. This suggests that, when no overt QuD is on the table, a "reasonable" QuD is chosen among all the possible non-vacuous partitions of the CS, and is such that the sentence under consideration properly answers it. This is motivated by the idea that sentences are never uttered in and of themselves; their purpose is to answer a question, overt or not, and to induce further questions Roberts (1996). A pragmatic model of assertion therefore needs to integrate what sentences mean, but also what kind of information structure they evoke. Assuming such a "reasonable" QuD is along the lines of *Where is Jo?*, then, the continuation in (45) is predicted to be both informative (it says that Jo is sick or at a conference), and relevant.

(45) –Have you seen Jo today?
–No I haven't... Either she is sick, or if she's not sick, she is at a conference.

But even if some implicit "reasonable" QuD can be inferred in the absence of an overt one, some cases of oddness remain mysterious. The follow-up sentence in (46) for instance, is equivalent to the one in (45) assuming implication is material, and so should in principle evoke the same QuD. (45) is thus predicted to be both informative and relevant, just like (45). Yet, this follow-up is sharply odd.

(46) –Have you seen Jo today?
–No I haven't... # Either she is sick, or if she's not at a conference, she is sick.

These two datapoints outline the following desideratum: if the contrast between (45) and (46) is due to the nature of the "reasonable" QuDs inferred from these sentences, then one must devise a way to systematically derive QuDs from out-of-the-blue assertions, in such a way that semantically similar, yet structurally distinct assertions, sometimes give rise to distinct QuDs.

This Chapter will address this desideratum and introduce a pragmatic model of these sentences in which (i) they package information differently in terms of their evoked QuDs, and (ii) unlike (46), (45), packages information in a way that is pragmatically optimal.

2.1.2 Overview and motivation of the Chapter

The machinery we introduce in this Chapter aims to account for the above datapoints (among others), by relating their felicity or oddness to the QuD(s) inferred from them.

The fundamental principle we want to operationalize is *Question-Answer Congruence* (henceforth **QAC**), as formalized by Katzir and Singh (2015),[1] and given in (47).

(47) *Question-Answer Congruence* (***QAC***). A felicitous assertion has to be a good answer to a good question.

This take on QAC is interesting because it roots this principle in pragmatics, and is broad enough to encompass a variety of constraints that were previously not grouped under the same umbrella. Chapter 1 for instance, showed that RELEVANCE could rule out a wide range of question-answer pairs, and as such could constitute a partial implementation of QAC. But QAC may in principle involve other constraints applying to question-answer pairs. This dissertation will show that, under a certain interpretation of "good answer" and "good question", many more cases of pragmatic oddness can be understood as an accross-the-board failure of QAC.

In this Chapter, we will lay out the groundwork for this more general pragmatic theory of question-answer well-formedness. We begin by introducing a more new model of questions, based on nested partitions, instead of mere partitions of the CS (as discussed in Chapter 1). This model is building on Büring (2003); Ippolito (2019); Zhang (2022), among many others. Next, equipped with this model of questions, we will show that questions can be evoked by assertions in a compositional way. As a result, sentences involving different operators (specifically, disjunctions and conditionals), give rise to different kinds of questions. Crucially in this model, each sentence may be associated with multiple potential questions. Finally, we will sketch what a pragmatics for question-answer pairs should look like in that framework. In line with QAC, a sentence which cannot be felicitously paired with *any* question will be deemed odd. This can happen if *all* the pairs formed by a sentence and a question it evokes, are themselves ill-formed.

We now proceed to define questions, not just as partition, but rather, as parse trees of the Context Set, that we will call Qtrees.

2.2 Structure of Question Trees

2.2.1 From partitions to recursive partitions, to parse trees

Building on the standard model presented in Chapter 1, we introduce a more elaborate view of the pragmatics of questions. This model will incorporate the idea that questions

[1] This principle has been discussed in several forms for many years, within and outside the field of generative linguistics. See for instance Rooth (1992) for a discussion on how focused assertions and questions can be systematically related in terms of their *semantics*.

have internal structure, and specifically, are hierarchically organized. This hierarchical organization is meant to capture the intuition that a question such as (48a) for instance, appears more *fine-grained*, than a question like (48b). Alternatively, whatever proposition identifies a cell in (48a), also identifies a cell in (48b). Crucially, this intuition will be incorporated in the pragmatics of questions, and so will be made directly accessible to the grammar.

(48) a. In which city did Jo grow up?

b. In which country did Jo grow up?

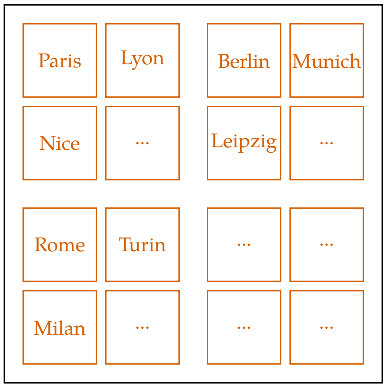
First, let us observe that these intuitions about question-specificity are *not* readily cashed out by standard partitions or alternative sets associated with questions. (48a)'s and (48b)'s sets of alternatives, given in (49a) and (49b) respectively, are made of disjoint, non empty propositions which, at a certain level of approximation, cover the space of all possibilities.[2] In other words, these alternatives already partition the set of *all* worlds. The partition that (49a) (resp. (49b)) induces on the CS is therefore obtained from (49a) (resp. (49b)) by simply intersecting each of its elements (a proposition/cell) with the CS–discarding empty sets.

(49) a. $[\![$ In which city did Jo grow up?$]\!]^w =$
$\{p \mid \exists l.\ l$ is a city $\wedge\ p = \lambda w'.$ Jo grew up in l in $w'\}$

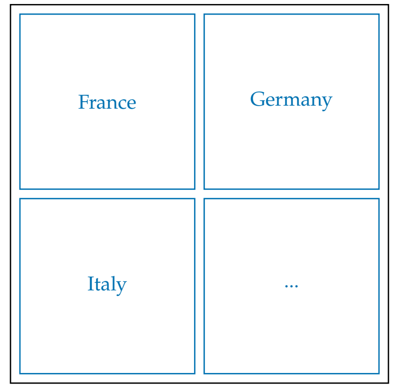
b. $[\![$ In which country did Jo grow up?$]\!]^w =$
$\{p \mid \exists l.\ l$ is a country $\wedge\ p = \lambda w'.$ Jo grew up in l in $w'\}$

(48a) therefore induces a by-city partition of the CS (see Figure 2.1a), while (48b) induces a by-country partition (see Figure 2.1b). But nothing in (48a)'s partition signals that each of its cells is properly contained in a cell of (48b)'s partition. This property can de derived from the two structures, but is not readily *encoded* by them.

[2]We will assume here, that any point on Earth is associated with one single country, and one single city, in a Voronoi fashion. At this level of approximation, there is no countryless or cityless area. Alternatively, one could assume that there are cityless areas, but that the possibility of Jo growing up in such areas is ruled-out by the presupposition carried by *which*-questions like (48a). Under this assumption, *where*-questions may require more work.



(a) By-city partition associated with (49a). Cells are ordered on a grid for clarity only.



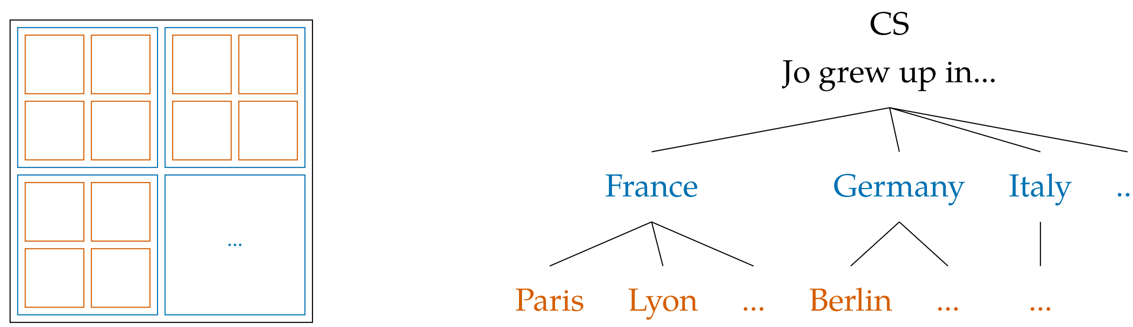
(b) By-country partition associated with (49b). Cells are ordered on a grid for clarity only.

Figure 2.1: Standard partitions induced by a fine-grained (49a) and a coarser-grained question (49b).

Intuitively, grouping together the propositions listed in (49a) talking about cities belonging to the same country, would help capture the desired property. This is done in (50). (50) then defines a set of sets of propositions.

(50) $[\![$ In which city did Jo grow up?$]\!]^w$ =
$\{\{p \mid \exists l.\ l$ is a city in $l' \wedge\ p = \lambda w'.$ Jo grew up in l in $w'\} \mid l'$ is a country$\}$

Grouping together cells within bigger sets (which are cells themselves), amounts to building a *nested* partition of the CS. In our example, the "outer" partition is by-country, and the "inner" partition, is by-city. Graphically, this is equivalent to adding the "blue rectangles" from Figure 2.1b, to Figure 2.1a. This operation is performed in Figure 2.2a. The tree in Figure 2.2b is yet another, more readable way to represent the same thing. In this tree, each node refers to a proposition of the form *Jo grew up in l*, l denoting a city or a country. Each node is understood as intersected with the CS, which corresponds to the root of the tree. Therefore, each node forms a proper subset of the CS. Nodes appearing at the same level (forming a "layer"), partition the CS. Deeper layers, correspond to finer-grained partitions. Tree like Figure 2.2b will be used throughout the dissertation to represent nested partitions like Figure 2.2a. One must always keep in mind that the two representations are equivalent. (51) formally defines the bijective mapping between nested sets of propositions (dubbed *inductive propositions*) like (50), and tree structures like Figure 2.2b.



(a) Recursive partition view

(b) Tree view

Figure 2.2: Alternative representations of the CS corresponding to the nested sets of (50).

(51) *Set-to-tree bijection.* To define this bijection, we first define inductive propositions, and their propositional content. S is an inductive proposition if either:

- S is a set of worlds (i.e. a proposition);
- S is a set of inductive propositions.

The propositional content of an inductive proposition is then defined as:

- If S is a proposition: S;
- If S is a set of inductive propositions: the grand union of the propositional contents of S's elements.

Any inductive proposition S is in a bijection with a tree structure whose nodes are propositions, and defined as:

- If S is a proposition: the tree node denoting S;
- If S is a set of inductive propositions: the tree whose root denotes S's propositional content, and whose children are the tree structures induced by each of S's elements.

So far, we have shown that the standard view linking questions to partitions, fails to account for the intuition that questions differing in terms of specificity, stand in some kind of inclusion relation encoded in their structure. We proposed a way to cash out this intuition, by appealing to recursive partitions, that we represent as trees for clarity.

We now proceed to generalize these observations about the structure of questions. Building on Büring (2003); Riester (2019); Onea (2016); Ippolito (2019); Zhang (2022) (among others), we take questions to denote *parse trees* of the CS, i.e. structures that hierarchically organize the worlds of the CS. Such trees (abbreviated **Qtrees**) are defined in (52).

(52) *Structure of Question-trees* (***Qtrees***). Qtrees are rooted trees whose nodes are all subsets of the CS and s.t.:

- Their root generally[3] refers to the CS;
- Any intermediate node is a proposition, which is partitioned by the set of its children.

A Qtree can be bijectively mapped to a nested partition of the CS as defined in (53). Due to this equivalence, we will mostly use Qtrees in the rest of this dissertation.

(53) *Nested partition.* A nested partition P of a set S is a kind of inductive proposition, s.t.:

- If P is a set of inductive propositions, then the propositional contents of P's elements partition P's propositional content. Additionally, P's elements are nested partitions of their own propositional content.

Before investigating the interpretation and the structural properties of model of questions, the next Section covers a few core concepts from graph theory that will be useful in the rest of the Chapter and beyond.

2.2.2 A brief refresher on graph theory (and a few useful concepts for Qtrees)

(52) defines Qtres as rooted trees. Linguists typically understand trees as relations between parent nodes and their children, along the lines of (65).

(54) *Rooted tree* (*inductive version*). A tree rooted in N is either:

- N (single, childless node);
- N, along with N's children, which are all rooted trees.

But we will see throughout this dissertation that it is also useful to see a tree as a specific kind of graph. We will first define graphs, then define trees as a subkind of graph, and lastly, show the importance of defining a root in such trees. The definition of a graph is given in (55). A graph is a way to represent a binary relation, which by default will be

[3] In the case of sentences carrying presuppositions, the root will be assumed to correspond to the intersection between the CS and the sentence's presupposition. In fact, the whole Qtree will be nodewise intersected with the presupposition. This will be put to use in Chapters **??** and **??**. But the examples we will see before this, will all involve Qtree rooted in the CS.

symmetric[4]. Elements in the domain of the relation are modeled as nodes, and unordered pairs of nodes are connected with an edge, iff they verify the relation. A graph therefore amounts to a set of nodes, and a set of edges between these nodes. This is illustrated in Figure 2.3.

(55) *Graph.* A graph is defined by a set of nodes $\mathcal{N}$ and by a set of edges $\mathcal{E}$ between elements of $\mathcal{N}$. Edges are defined as unordered pairs of nodes: $\mathcal{E} \subseteq \{\{N_1, N_2\} \mid (N_1, N_2) \in \mathcal{N}^2\}$

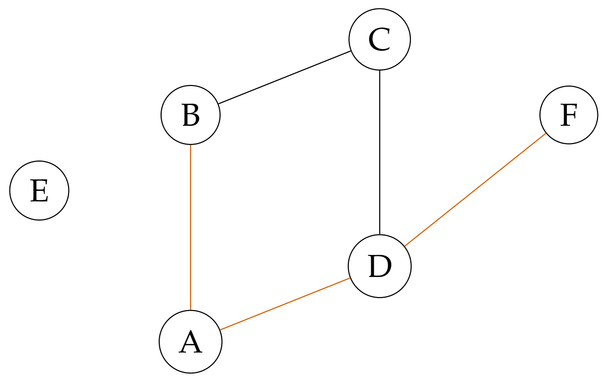


Figure 2.3: A graph $G = (\mathcal{N}, \mathcal{E})$, with $\mathcal{N} = \{A, B, C, D, E, F\}$ and $\mathcal{E} = \{\{A, B\}, \{A, D\}, \{B, C\}, \{C, D\}, \{D, F\}\}$.

This definition allows to define rooted trees as a kind of graph with a few extra properties: connectivity, acyclicity, and rootedness; see (56). We now unpack what these three extra properties mean for graphs. This will lead us to define a few useful concepts applying to trees, namely paths, ancestry, and depth.

(56) *Rooted tree* (*graph version*). A rooted tree is a graph that is connected and acyclic, and features a distinguished node called root.

In graphs, sequences of adjacent edges form paths. For instance, in Figure 2.3, the ordered sequence $[\{A, B\}, \{A, D\}, \{D, F\}]$ forms a path, between node A and node F. This is generalized in (57).

(57) *Path.* Let $G = (\mathcal{N}, \mathcal{E})$ be a graph. Let $(N_1, N_2) \in \mathcal{N}^2$ be two nodes of G. There is a path in G between N_1 and N_2 (abbreviated $N_1 \overset{G}{\leadsto} N_2$) iff N_1 and N_2 can be connected by a series of edges in G, i.e. $\exists (e_1, ...e_k) \in \mathcal{E}^k.\ N_1 \in e_1 \wedge N_2 \in e_k \wedge \forall i \in [1; k-1].\ |e_i \cap e_{i+1}| = 1$, where $|.|$ is the cardinality operator.

[4] *Undirected* graphs, that we will simply call graphs, implement symmetric relations, while *directed* graphs implement asymmetric relations.

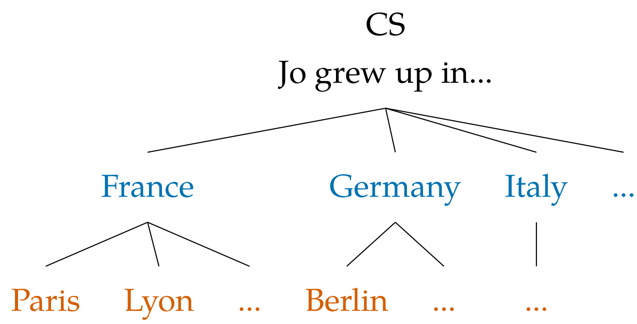
In Figure 2.3, it is easy to see that nodes A, B, C, D and F are all connected to each other by at least one path (in fact, infinitely many of them that cycle through these nodes). Node E on the other hand, is isolated. So, Figure 2.3 represents a graph that is *not* connected. If E were removed from the set of nodes, and the edges remained the same, the resulting graph would be connected. This concept of connectivity is generalized in (58). If a graph is a tree, then, it is connected.

(58) *Connectivity.* Let $G = (\mathcal{N}, \mathcal{E})$ be a graph. G is connected, iff there is a path in G between any pair of nodes in $\mathcal{N}$, i.e. $\forall(N_1, N_2) \in \mathcal{N}^2.\ N_1 \overset{G}{\rightsquigarrow} N_2$.

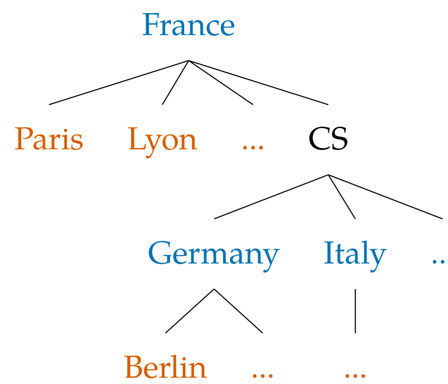
Another thing to note about Figure 2.3, is that nodes A, B, C, and D form a "cycle", there is a path that starts at one of these nodes (e.g., C), and ends at this very same node, *via* B, A, and D. Because of this cycle, there are infinitely many paths between A, B, C, and D, and also between each of these nodes, and F. Removing the edge between, say, A and B, would break the cycle (yet, interestingly, maintain connectivity between A, B, C, and D). The resulting graph would be acyclic. The general definition of an acyclic graph, is given in (59). If a graph is a tree, then, it is acyclic. Moreover, connectivity and acyclicity, are necessary and sufficient for a graph to be a tree.

(59) *Acyclicity.* Let $G = (\mathcal{N}, \mathcal{E})$ be a graph. G is acyclic, iff no node N of $\mathcal{N}$ is s.t. there is a path starting and ending at N in G, i.e. $\neg\exists N \in \mathcal{N}.\ N \overset{G}{\rightsquigarrow} N$.

We now have a definition of what kind of data structure a tree is. But why do we need Qtrees to be "rooted"? To understand why, let us go back to the tree in Figure 2.2b, repeated in Figure 2.4a below. The way this tree is represented on paper, is somehow misleading. Recall that, from the point of view of graph theory, a tree is just an undirected graph, with a few extra properties constraining its edges. If the tree represented in Figure 2.4a were not "rooted", nothing would prevent us from representing it in the form of Figure 2.4b: the nodes and edges are strictly the same, but in Figure 2.4b, *France* "appears" to be the root of the tree, because visually, it is represented at the top. To avoid this confusion, the fact that the CS node should be "at the top" is made part of the representation of the tree–which then becomes a *rooted* tree. So, a rooted tree is just a tree, plus one distinguished node that serves as root.



(a) The "intuitive" view, in which the CS appears at the top.



(b) An alternative "counter-intuitive" view, in which the CS is *not* at the top, yet all edges and nodes are the same.

Figure 2.4: Two equivalent ways to represent the tree corresponding to the question in (48a); assuming trees were connected, acyclic graphs, but not rooted.

The notion of a distinguished root in fact allows to define a few interesting properties on trees that linguist may be more familiar with, and that will be used throughout this dissertation. First, once a tree is rooted, it is possible to define a measure of distance between each node of the tree, and the root. This corresponds to the concept of depth, defined in (60a). In Figure 2.4a for instance, the CS has depth 0, *Germany* depth 1, and *Lyon* depth 2. This also allows to define the global "size" of the tree, in the form of its maximal depth; see (60b). Figure 2.4a for instance, is a tree of depth 2.

(60) a. *Depth of a node in a rooted tree.* Let $T = (\mathcal{N}, \mathcal{E}, R)$ be a rooted tree, with root R. Let $N \in \mathcal{N}$. The depth of N in T ($d(N,T)$) corresponds to the length of the minimal path between R and N if $N \neq R$,[5] and is set to 0 if $N = R$.

b. *Depth of a rooted tree.* Let $T = (\mathcal{N}, \mathcal{E}, R)$ be a rooted tree, with root R. The depth of T ($d(T)$) is the maximal depth of a node in T: $d(T) = max_{N \in \mathcal{N}}(d(N,T))$.

Having a distinguished root, and the derived concepts of depth, gives us the parent-child relation between nodes for free.[6] This relation is defined based on depth and edges in (61), and its transitive closure (the ancestor relation) is defined in (62), in two possible ways.

(61) *Parent-child relation in a rooted tree.* Let $T = (\mathcal{N}, \mathcal{E}, R)$ be a rooted tree. Let $(N_1, N_2) \in \mathcal{N}^2$. N_1 is the parent of N_2 (and N_2 is the child of N_1), iff $\{N_1, N_2\} \in \mathcal{E}$ and $d(N_1, T) < d(N_2, T)$.

[5]This path can be determined using a simple Depth-First Search algorithm starting from the root.

[6]in the next Section, we will introduce another definition of tree, that takes this relation as a primitive

(62) a. *Ancestor relation* (*recursive version*). Let $T = (\mathcal{N}, \mathcal{E}, R)$ be a rooted tree. Let $(N_1, N_2) \in \mathcal{N}^2$. N_1 is an ancestor of N_2 iff either:

- N_1 is the parent of N_2;
- or N_1 is the parent of an ancestor of N_2.

b. *Ancestor relation* (*path version*). Let $T = (\mathcal{N}, \mathcal{E}, R)$ be a rooted tree. Let $(N_1, N_2) \in \mathcal{N}^2$. N_1 is an ancestor of N_2 iff $N_1 \overset{T}{\rightsquigarrow} N_2$ and $d(N_1, T) < d(N_2, T)$.

Lastly, in the rest of this dissertation, we will extensively use the concept of *layer*, that we define as a the maximal set of same-depth nodes in a rooted tree; see (63). Figure 2.4a features a country-layer at depth 1, and a city-layer at depth 2. Layers therefore reflect an intuitive notion of granularity.

(63) *Depth-k layer of a rooted tree.* Let $T = (\mathcal{N}, \mathcal{E}, R)$ be a rooted tree, with root R. Let k be an integer s.t. $0 \geq k < d(T)$. The depth-k layer of T is the set of nodes in $\mathcal{N}$ whose depth is k, i.e. $\{N \in \mathcal{N} \mid d(N, T) = k\}$.

Now that we have defined the core structure of Qtrees along with a few related properties and metrics, we proceed to assign an interpretation to this kind of structure.

2.2.3 Interpreting Qtrees

At the end of Section 2.2.1, we showed that question should better be represented as nested partition, in order to encode their degree of specificity, and the grammar sensitive to how more or less fine-grained questions relate to each other. We also discussed how nested partitions could be unequivocally represented as Qtrees. It is easy to see that the tree in Figure 2.2b/2.4a, repeated again in Figure 2.5, is a Qtree according to (52). We saw that this Qtree intuitively capture the idea that a *Which country?* kind of question, is contained in a *Which city?* kind of question. We now investigate how to exploit this hierarchy in a meaningful way. We will use Figure 2.5 as an example, and now assign an interpretation to nodes and paths in such structures. We will focus on three meaningful aspects of Qtrees: answer-granularity (understood as node depth), strategies of inquiry (understood as paths), and question refinement (understood as tree inclusion)

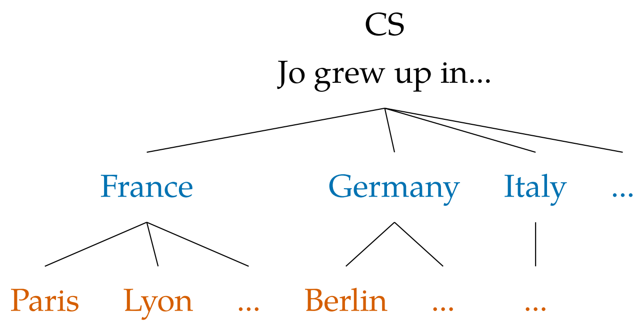


Figure 2.5: "Intuitive" Qtree for *Which city did Jo grow up in?*

We start with the interpretation of nodes as possible answers, with different granularities. The root of Figure 2.5 for instance, which corresponds to the entire CS, defines a tautology: it is a proposition which is true of all worlds of the CS, because it simply coincides with it.[7] It can be understood as identifying the unique cell of coarsest-grained partition of the CS, that is, the CS itself. By contrast, leaves like *Paris*, *Lyon*, *Berlin* in Figure 2.5, correspond to the "smallest" cells of the recursive partition that the Qtree defines. They can be seen as maximal answer to the underlying question, e.g., *In which city did Jo grow up?*. Intermediate nodes like *France* or *Germany* in Figure 2.5, form cells of "intermediate" size, and can always be seen as unions of leaves. They appear to correspond to non-maximal answers. Because Qtrees can be made of many layers, they induce a hierarchy between non-maximal answers: an non-maximal answer p is "more maximal" than another non-maximal answer q, iff the node corresponding to p is located deeper in the Qtree than the node corresponding to q. This is formalized in (64).

(64) *Answer granularity.* Let T be a Qtree and (N_1, N_2) be two nodes in T. N_1 constitutes a finer-grained answer than N_2 iff $d(N_1, T) > d(N_2, T)$. This implies that leaves of T correspond to the finest-grained answers (maximal answers) to the question T represents.

Next, we discuss how Qtree encapsulate Roberts's notion of *Strategy of Inquiry*. To this end, we observe that nodes in a tree can receive a "recursive" interpretation, that incorporates everything the node dominates. Under this interpretation, a node N in a Qtree is not only what N denotes; it is the whole subtree ($\sim$subquestion) rooted in N, as defined in (65)

(65) *Recursive interpretation of tree nodes.* Let $T = (\mathcal{N}, \mathcal{E}, R)$ be a rooted tree. Let $N \in \mathcal{N}$ be a node of T. N's recursive interpretation corresponds to:

[7] Chapter 1 moreover identifies it as an uninformative proposition that is Lewis-relevant but not Roberts-relevant.

- N, if N is a leaf;
- the subtree of T rooted in N, otherwise.

This point of view originates from the inductive definition of a rooted tree given in (54) and repeated below.

(54) *Rooted tree* (*inductive version*). A tree rooted in N is either:

- N (single, childless node);
- N, along with N's children, which are all rooted trees.

If T is a Qtree, then N's recursive interpretation will be the Qtree rooted in N. This Qtree's root can be seen as a "local" CS, which is equal to the global CS, updated with N. For instance, the recursive interpretation of the *France*-node in Figure 2.5, corresponds to the subtree of Figure 2.5 rooted in *France*. This subtree, given in Figure 2.6, amounts to the question *In which city did Jo grow up?*, granted that *Jo lives in France*, since its root corresponds to the CS intersected with the proposition that *Jo lives in France*.

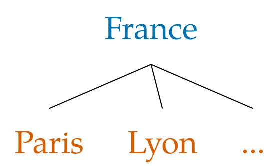
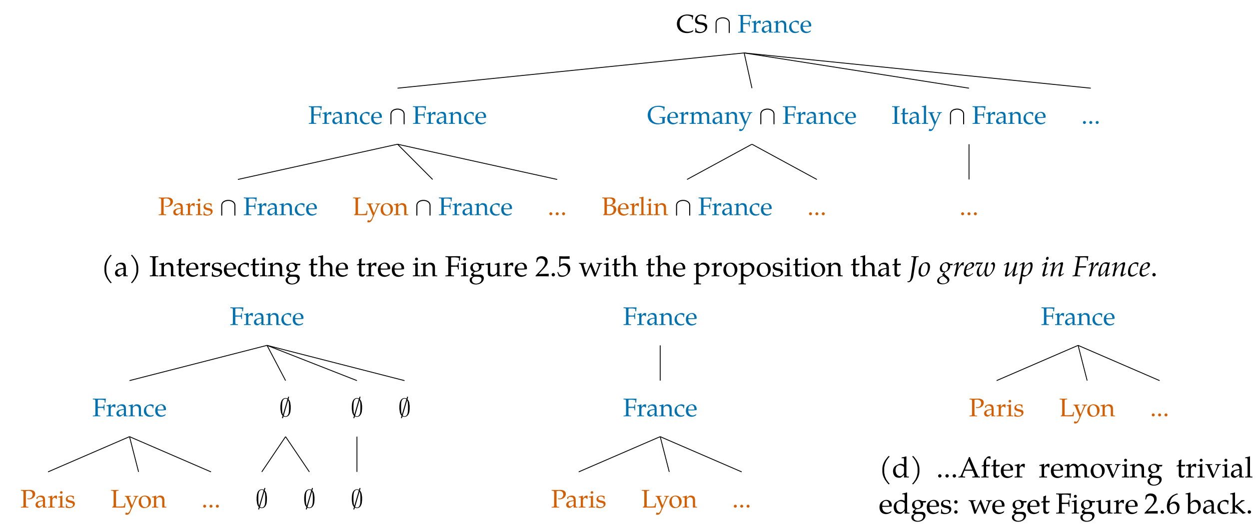


Figure 2.6: "Recursive" interpretation of the *France*-node in Figure 2.5.

In fact, this subtree as a whole, can be understood as the nodewise intersection of Figure 2.5 and the proposition that *Jo grew up in France*. Nodewise intersection is defined in (66). This operation takes a Qtree and a proposition p, and creates a Qtree whose nodes are each intersected with p, and resulting empty nodes are removed. Edges from the original Qtree are retained, as long as the nodes they connect are still part of the newly formed Qtree. Note that, because the nodes and edges of a tree form *sets* (and not *multisets*), nodewise intersection automatically collapses nodes from the original tree whose intersections with p yield the same result; and it also collapses the edges between such nodes. Figure 2.7 provides a decomposition of this procedure, computing the nodewise intersection between Figure 2.5 and the proposition that *Jo grew up in France*, and illustrating how nodes and edges may "collapse". (67) generalizes this point, by stating that the subtree of a Qtree rooted in a node N, can be reconstructed by nodewise intersecting the entire Qtree with the proposition N corresponds to. In other words, the subquestion corresponding to a node N, can be seen as a *restriction* of the entire Qtree, taking N for granted. This is proved in (68).

(66) *Nodewise intersection.* Let $T = (\mathcal{N}, \mathcal{E}, R)$ be a Qtree. Let p be a proposition. The nodewise intersection between T and p, noted $T \cap p$, is defined iff $R \cap p \neq \emptyset$ and, if so, is the Qtree $T' = (\mathcal{N}', \mathcal{E}', R')$ s.t.:

- $\mathcal{N}' = \{N \cap p \mid N \in \mathcal{N} \wedge N \cap p \neq \emptyset\}$
- $\mathcal{E}' = \{\{N_1 \cap p, N_2 \cap p\} \mid \{N_1, N_2\} \in \mathcal{E} \wedge (N_1 \cap p) \neq (N_2 \cap p) \wedge N_1 \cap p \neq \emptyset \wedge N_2 \cap p \neq \emptyset\}$
- $R' = R \cap p$



(a) Intersecting the tree in Figure 2.5 with the proposition that *Jo grew up in France*.

(b) ...After computing nodewise intersections.

(c) ...After removing empty nodes and resulting dangling edges.

(d) ...After removing trivial edges: we get Figure 2.6 back.

Figure 2.7: The recursive interpretation of a node, can be obtained by intersecting the whole Qtree with that node, removing empty nodes and trivial edges (formed by a parent node and its only child).

(67) *Recursive interpretation and CS update.* Let T be a Qtree. Let N be a node of T. N's recursive interpretation corresponds to the nodewise intersection of T with N.

(68) *Proof of* (67). Let T be a Qtree. Let N be a node of T. Because T is a Qtree, any node N dominates is a subset of N; any node dominating N, is a superset of N, and any node that is neither dominated nor dominating N, is disjoint from N. By definition, N's recursive interpretation is the subtree of T rooted in N, noted T'. We show that T' corresponds to the nodewise intersection between T and N, $T \cap N$. Let N' be N or a node dominated by N. $N' \subseteq N$, so $N' \cap N = N'$. This holds for any N' dominated by N or equal to N. So $T \cap N$ preserves T'. Let N' be an ancestor of N. $N \subseteq N'$ so $N' \cap N = N$. So any ancestor of N in T, is reduced

to N in $T \cap N$. Let N' be a node in T that is neither dominated nor dominating N. $N \cap N' = \emptyset$, and so any sibling/uncle/cousin of N in T is absent in $T \cap N$, along with any incident edges. Therefore, $T \cap N$ ends up being just T'.

We have just seen that under the recursive interpretation of nodes, each node N can be seen as a subquestion of the whole Qtree, which takes N's propositional content for granted. Under this interpretation, a path from the root (CS) to any node N, can then be seen as a series of subquestions, taking for granted increasingly strong propositions. In Figure 2.5 for instance, a path of the form [*CS*, *France*, *Paris*], can be interpreted as a series of inquiries of the form: [*In which city did Jo grow up* (*I have no idea*)?, *In which city did Jo grow up* (*given Jo grew up in France*)?, *Jo grew up in Paris*]. If the path terminates on a leaf, then the series of inquiries converges to a maximal answer. We will call such paths complete strategies of inquiry.

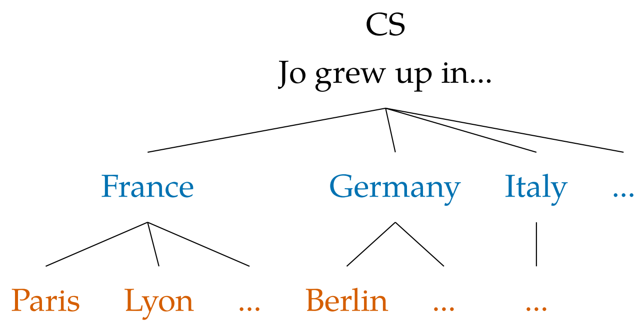
(69) *Complete Strategy of Inquiry.* Let T be a Qtree. A complete strategy of inquiry on T is a path from T's root to one of T's leaves.

This model is very close to what the previous literature had posited at the conversational level, whereby sentences answer questions and sometimes evoke new, finer-grained questions. The key difference here, is that individual questions are assumed to encapsulate the same kind of dynamic, hierarchical information. How does this relate to question granularity? Note that intuitively finer-grained questions yield deeper Qtrees than intuitively coarser-grained ones. Additionally, (60b) defined Qtree depth as the maximal length of a path from the root to a leaf in the tree. This leads to the equivalence in (70).

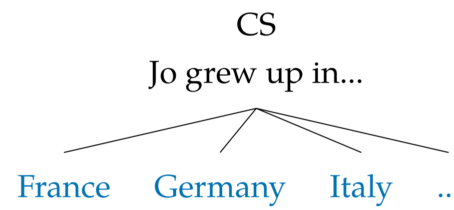
(70) *Depth and Complete Strategies of Inquiry.* Let T be a Qtree. T's depth can be recovered by finding the length of its longest complete strategy of inquiry, dubbed maximal complete strategy of inquiry.

In other words, finer-grained questions are linked to deeper Qtrees, which are characterized by a longer, maximal complete strategy of inquiry. In sum, a fine-grained question is a question for which converging to a maximal answer may require a lot of intermediate steps, or subquestions. This is useful as an absolute measure of question-complexity, but probably not enough to determine if a question is finer-grained than another question. For instance, this incorrectly predicts two completely independent questions to be comparable in terms of granularity, just because they give rise to Qtree of different depths.

There is in fact another way in which the recursive interpretation of nodes can help clarify in what sense a *Which city* kind of question, is more fine-grained than a *Which country* kind of question, in the current framework. Figure 2.8 shows what a Qtree for (49a) and a Qtree for (49b) should intuitively look like.



(a) "Intuitive" Qtree for (49a) = *Which city did Jo grow up in?*



(b) "Intuitive" Qtree for (49b) = *Which country did Jo grow up in?*

Figure 2.8: Comparing *Which city* and *Which country* Qtrees.

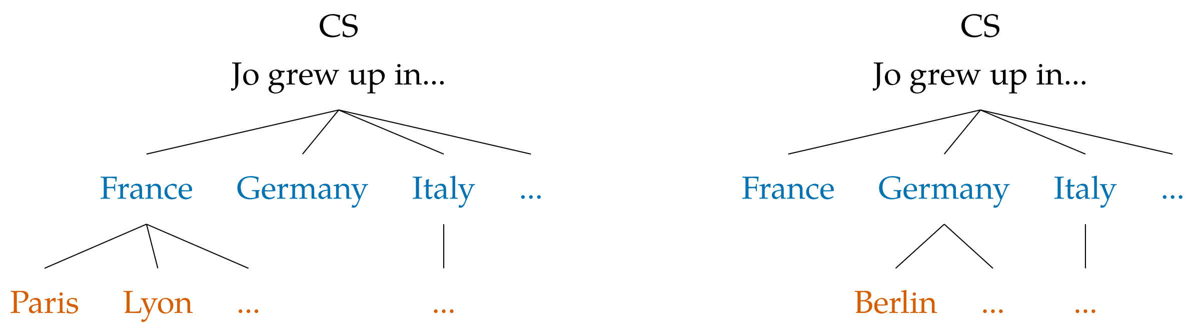
The Qtree for (49a) stops at the city-level, because cities should constitute maximal answer to that kind of question; the Qtree for (49b) on the other hand, stops at the country level, for similar reasons. And it is easy to notice that the Qtree for (49b) somehow forms a "subset" of the Qtree for (49a): it forms a subset of the nodes, and a subsets of the edges, of the Qtree for (49a). Additionally, it is not a random subgraph of Figure 2.8a (as defined in (71)). It remains a Qtree, that constitues a refinement of Figure 2.8a, as defined in (72). It can also be shown, that all the possible refinements of a Qtree T, correspond to all the possible subgraphs of T that have the Qtree property.[8]

(71) *Subgraph.* Let $G = (\mathcal{N}, \mathcal{E})$ and $G' = (\mathcal{N}', \mathcal{E}')$ be two graphs. $G' \subseteq G$, iff $\mathcal{N}' \subseteq \mathcal{N}$ and $\mathcal{E}' \subseteq \mathcal{E}$.

[8]We identify the refinement operation between T and T', as a set $\mathcal{T}$ of subtrees of T, that is closed under root-sisterhood. We assume T is a refinement of a Qtree T' and show T' is a subgraph of T with the Qtree property. T' is obtained from T by removing the subtrees in $\mathcal{T}$ from T. So it is obviously a subgraph of T. We now show T' is a Qtree. $\mathcal{T}$ cannot contain the tree rooted in T, otherwise T' would be empty. So T' has same root as T, and this root is the CS. Let N be an intermediate node in T'. N has at least one child N', which means that $\mathcal{T}$ cannot contain the subtree of T rooted in N'. To be partitioned by its children, N in T' must have the same children as N in T, i.e. $\mathcal{T}$ should not contain any tree rooted in a child of N. If $\mathcal{T}$ did, then $\mathcal{T}$ would also contain the subtree of T rooted in N', du to its closure property. Contradiction. So N retained all its children from T, and is partitioned by them, given that T is a Qtree.
We assume T' is a subgraph of T with the Qtree property and show T is a refinement of a Qtree T'. T' is a subgraph of T, so we can define S the set of nodes of T not in T'. We then define $\mathcal{T}$ as the set of subtrees of T rooted in a maximal element of S w.r.t. the ancestor relation, as induced by T's edges. We show this set is closed under root-sisterhood. Let $T'' \in \mathcal{T}$. It is subtree of T rooted in some node N, and N is maximal in S w.r.t. the ancestor relation. If N has no sister in T, then the closure property is trivially verified. If N has a sister N' in T, we show the closure property by contradiction. If the subtree of T rooted in N' did not belong to $\mathcal{T}$, then, either N' would be part of a subtree of $\mathcal{T}$ (but not as root), or, N' would be a node in T'. The former option would imply that some common ancestor of N and N' would be the root of a subtree in $\mathcal{T}$. But then both N and some ancestor of N, would be maximal in S w.r.t. the ancestor relation. Contradiction. The former option would mean that N''s parent in T', would have N', but not N has child, and so would not be partitioned by its children. Therefore, T' would not be a Qtree. Contradiction.

(72) *Qtree refinement.* Let T and T' be Qtrees. T is a refinement of T' (or: T is finer-grained than T'), iff T' can be obtained from T by removing a subset $\mathcal{T}$ of T's subtrees, s.t., if $\mathcal{T}$ contains a subtree rooted in N, then, for each node N' that is a sibling of N in T, the subtree of T rooted in N', is also in $\mathcal{T}$.

Two other possible refinements of Figure 2.8a are given below. It is worth noting that the process deriving a refinement from a Qtree need not remove entire layers.



(a) A refinement where the children of the *Germany*-node are deleted.

(b) A refinement where the children of the *France*-node are deleted.

Figure 2.9: Possible refinements of the Qtree for *Which city did Jo grow up in?* in Figure 2.8a.

2.2.4 Flagging Qtrees

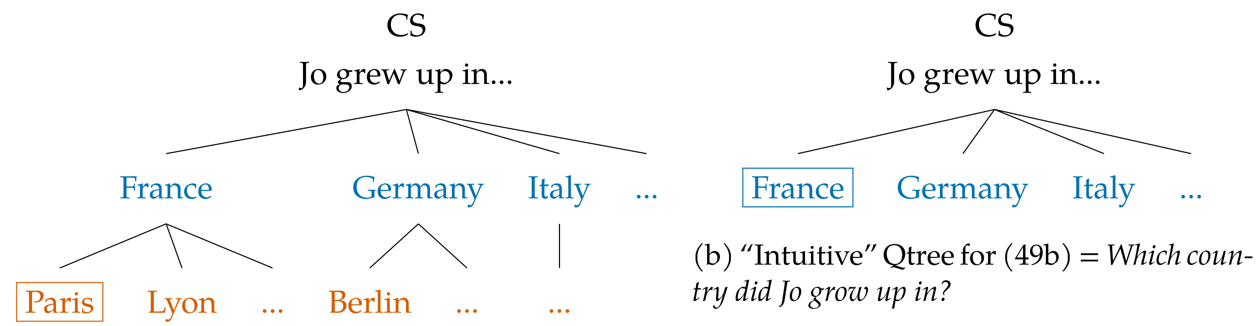
So far, we have considered Qtrees directly associated with questions, like (48a) and (48b). But, as suggested in the introduction to this Chapter, we want to go one step further, and posit that assertive sentences, like (73a) and (73b), also evoke questions in the form of Qtrees. Such questions will correspond to the ones a given assertion could be a good answer to.

(73) a. Jo grew up in Paris.
b. Jo grew up in France.

So (73a) and (73b) for instance, should evoke Qtrees associated with questions like (48a) and (48b), respectively. We sketched an intuitive representation of these Qtrees in Figure 2.8. Are these Qtrees representing everything that the assertions in (73a) and (73b) convey though? One major difference between questions and assertions, is that questions are ignorant of the answer, while assertions provide such an answer. So, if (73a) were directly mapped to the Qtree in Figure 2.8a, the information that (73a) actually answers the question by identifying the *Paris*-node, would be lost. Another way to see the issue, is to observe that (73a) and (74) would then be associated with the exact same Qtree.

(74) Jo grew up in Berlin.

To avoid such collisions in the case of assertive sentences, we define an extra piece of machinery on top of the Qtree architecture, that consists in a set of "verifying" nodes keeping track of *how* the assertion answers the question it evokes. I Figures, these nodes will be represented in boxes; given a Qtree T, T's set of verifying nodes will be referred to as $\mathcal{N}^+(T)$. (73a) and (73b) for instance, will intuitively evoke Qtree that *structurally* match those in Figure 2.8, but whose *Paris* and *France* noes respectively, are "boxed", i.e. flagged as verifying.



(a) "Intuitive" Qtree for (49a) = *Which city did Jo grow up in?*

(b) "Intuitive" Qtree for (49b) = *Which country did Jo grow up in?*

Figure 2.10: Comparing *Which city* and *Which country* Qtrees.

In that particular case, the nodes that are flagged as verifying in both Qtrees, strictly coincide with the proposition conveyed by the assertions, namely, that *Jo grew up in Paris*, and that *Jo grew up in France*. For an assertion like (74), the only flagged node would be *Berlin*. But we will not take this strict equivalence between prejacent proposition and verifying nodes to be a generality. In the model laid out in the next Section, we will assume that verifying nodes, just like Qtree structure, are compositionally "retro-engineered" from the structure and meaning of the sentence. As a result, there may be more than on verifying node in a given Qtree, and, the grand union of a Qtree's verifying nodes, may not always coincide with the proposition denoted by the assertion.[9] The next Section therefore introduces a more systematic way to derive Qtrees and their verifying nodes from simplex sentences.

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.(see (75)). More generally, we as-

[9]This will in particular be true of conditional assertions.

sume that oddness results from the fact that a given sentence, through its LF, cannot give rise to any well-formed Qtree. This is summarized in (76) and (77).

(75) *Empty 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.

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

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

2.3 Compositional Qtrees: base case

So far, we have established that assertive sentences can evoke the questions they are good answers to, in the form of Qtrees. And we used "intuitive" Qtrees like the ones in Figures 2.10a and 2.10b to show how such structures could capture a wide range of properties connecting questions and answers, and questions to each other. We now introduce a principled algorithm to derive Qtrees out of assertive sentences. This will heavily build on the notion of alternatives and how they can be ordered in terms of specificity. We will show that this ordering of alternatives induces a specific "layering" on Qtrees.

2.3.1 Alternatives

It is quite uncontroversial that assertive sentences evoke alternatives (Rooth, 1992; Katzir, 2007; Fox and Katzir, 2011). An alternative is a sentence that is sufficiently "similar" to the sentence it is evoked by, but may have a different meaning. For instance, (78a) is felt to have (78b) as alternative, and vice versa. Pre-theoretically, this is because, in many contexts, (78a) is utterable iff (78b) is, too. The same holds for (79a) and (79b).

(78) a. Jo ate all of the cookies.
b. Jo ate some of the cookies.

(79) a. Jo grew up in Paris.
b. Jo grew up in Lyon.

Moreover, the computation of alternatives is driven by focus. In (78) and (79), it was implicitely assumed that quantifiers and city names respectively, were focused, and so gave rise to alternative varying in terms of the focused element. But note that, if the object of (78a) and the verb of (79a) had been focused instead, the alternatives to these sentences would have been different, along the lines of (80b) and (81b), respectively.

(80) a. Jo ate all of the COOKIES.
b. Jo ate all of the muffins.

(81) a. Jo GREW UP in Paris.
b. Jo resides in Paris.

We define what counts as an alternative to a given sentence (or more generally an LF) in (82), based on Rooth (1992) and Katzir (2007).

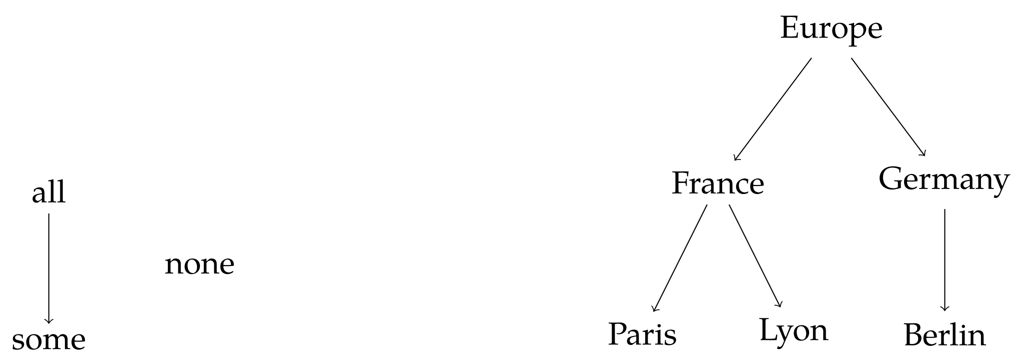
(82) *Structural alternatives.* Let X be an LF containing a focused constituent. The set of X's alternatives is the set $\mathcal{A}_X$ of LFs Y, obtained by substituting X's focused constituent with an element that is at most as complex.

(83) *Structural complexity.* Let X and Y be two LFs. X is at most as complex as Y iff X can be obtained from Y *via* substitutions of lexical items with other lexical items, or constituent-to-subconstituent substitutions.

Structural alternatives to a given LF (which are also LFs), induce a set of propositional alternatives, as defined in (84).

(84) *Propositional alternatives.* Let X be an LF denoting a proposition p. The set of X's propositional alternatives is the set $\mathcal{A}_{p,X}$ of propositions denoted by X's structural alternatives: $\mathcal{A}_{p,X} = \{q \mid \exists Y \in \mathcal{A}_X.\, [\![Y]\!] = q\}$

Propositional alternatives can be related to each other by entailment–forming a partial order. This partial order between propositional alternatives can be graphically represented in the form of a Hasse diagram. Hasse diagrams for two possible sets of propositional alternatives, roughly corresponding to (80) and (79), are given in Figure 2.11.



(a) Hasse diagram for $\{all, some, none\}$

(b) Hasse diagram for $\{Europe, France, Germany, Paris, Lyon, Berlin\}$

Figure 2.11: Hasse diagrams generated by $\vDash$ on two possible sets of propositional alternatives.

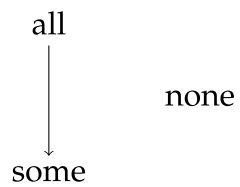
How are these diagrams obtained from propositional alternatives, and the entailment relations between them? Formally, a Hasse diagram is a directed graph, as defined in (85). The only difference between a graph and a directed graph, is that the edges of a directed graph have a direction, i.e. they correspond to ordered pairs instead of sets of cardinality 2. If $[N_1, N_2]$ is a directed edge, $[N_1, N_2]$ is visually represented as $N_1 \rightarrow N_2$. Paths are also directed, and so is the ancestor relation (see (86) and (87)). Directed graphs are designed to model *asymmetric* relations, like $\vDash$.

(85) *Directed graph.* A directed graph is defined by a set of nodes $\mathcal{N}$ and by a set of directed edges $\mathcal{E}$ between elements of $\mathcal{N}$. Directed edges are defined as ordered pairs of nodes: $\mathcal{E} \subseteq \{[N_1, N_2] \mid (N_1, N_2) \in \mathcal{N}^2\}$

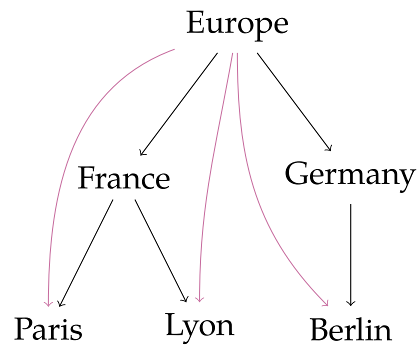
(86) *Directed Path.* Let $G = (\mathcal{N}, \mathcal{E})$ be a directed graph. Let $(N_1, N_2) \in \mathcal{N}^2$ be two nodes of G. There is a path in G between N_1 and N_2 (abbreviated $N_1 \overset{G}{\rightsquigarrow} N_2$) iff N_1 can be connected to N_2 by a series of directed edges in G, i.e. $\exists(e_1, ...e_k) \in \mathcal{E}^k.\ e_1^{(0)} = N_1 \wedge e_k^{(1)} = N_2 \wedge \forall i \in [1; k-1].\ e_i^{(1)} = e_{i+1}^{(0)}$, where, for any edge e, $e = [e^{(0)}, e^{(1)}]$.

(87) *Ancestor relation* (*directed path version*). Let $G = (\mathcal{N}, \mathcal{E})$ be a directed graph. Let $(N_1, N_2) \in \mathcal{N}^2$. N_1 is an ancestor of N_2 iff $N_1 \overset{G}{\rightsquigarrow} N_2$.

The directed graphs (not yet the Hasse diagrams) induced by $\vDash$ on the sets of alternatives from Figure 2.11, are given in Figure 2.12. In these graphs, there is a directed edge $[p, q]$ between two nodes corresponding to propositions p and q, iff $p \vDash q$. Figure 2.12a already looks like the corresponding Hasse diagram in Figure 2.11a, but Figure 2.12b does not: it features a few more directed edges (in red) than its Hasse counterpart in Figure 2.11b.



(a) Directed graph induced by $\vDash$ on $\{all, some, none\}$



(b) Directed graph induced by $\vDash$ on $\{Europe, France, Germany, Paris, Lyon, Berlin\}$

Figure 2.12: Directed graphs generated by $\vDash$ on two possible sets of propositional alternatives.

How do Hasse diagrams eliminate these few superfluous edges? The Hasse diagrams we are interested in correspond to the transitive reduction of the graphs in Figure 2.12, which were induced by $\vDash$ on sets of propositional alternatives. The transitive reduction operation precisely gets rid of the red edges in Figure 2.11b, based on the idea that such edges correspond to paths formed by the black ones. The formal (though, non constructive) definition of a transitive reduction, is given in (88). This definition maps the graphs in Figure 2.12, to the Hasse diagrams in Figure 2.11.

(88) *Transitive reduction of a graph.* Let $G = (\mathcal{N}, \mathcal{E})$ be a graph. The transitive reduction G' of G is the graph:

- Whose set of nodes is $\mathcal{N}$;
- Whose edges are the smallest set $\mathcal{E}'$ s.t. $\forall (N_1, N_2) \in \mathcal{N}.\ N_1 \overset{G}{\rightsquigarrow} N_2 \iff N_1 \overset{G'}{\rightsquigarrow} N_2$

The Hasse diagram in Figure 2.11b is basically a rooted tree, and may look like a Qtree, but this is not a generality. For instance, the Hasse diagram in Figure 2.11a is not connected, so is not even a tree. Additionally, not all sets of propositional alternatives, even if their Hasse diagram is tree-like, are guaranteed to verify the partition property of Qtrees. The next Section focuses on how Hasse diagrams can be used to determine how alternatives relate to each other in terms of granularity. This will eventually allow us to encode granularity in the structure of Qtree.

2.3.2 Alternatives and granularity

In this section, we use a notion of granularity to constrain what kind of Qtree can be evoked by a simplex assertive LF. The goal is to organize the layers of a Qtree in terms of how specific the nodes in this layer are. Why is an external notion of granularity needed to structure Qtree? In the Qtree sketched in e.g. Figure 2.10a, each layer corresponds to an intuitive degree of specificity: a by-country layer dominates a by-city layer. Though intuitive, this kind of configuration is not the only one to verify the Qtree property. The tree in Figure 2.13, where the *Germany*-node is replaced by its children, is also a Qtree: at our level of approximation, all countries but Germany, plus all the German cities, partition the set of all possible locations, and, each country represented in this tree is properly partitioned by the set of its cities.

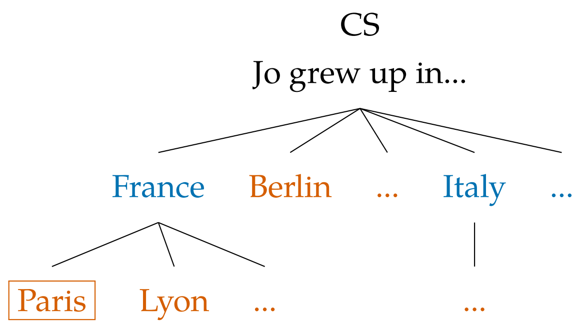


Figure 2.13: "Unintuitive" Qtree for (49a) = *Which city did Jo grow up in?*, where layers exhibit "mixed granularity".

To derive Qtrees like Figure 2.10a, and rule-out Qtrees like Figure 2.13, we need a notion of granularity that can transfer into the Qtree layers. We now show that a relation of same-granularity can be derived from Hasse diagrams induced by $\vDash$ on "complete" sets of propositional alternatives. Considering such diagrams, where *all* relevant alternatives are considered, we take that two nodes (two propositional alternatives) have same granularity if they are equidistant (in terms of path length) to a common ancestor. This relation, defined in (89) is close in spirit to Ippolito (2019)'s *Specificity Condition*.[10] Note that this definition is conditional, and not biconditional: nodes that do not have a common ancestor, may or may not be seen as same-granularity.[11]

[10] However, the structure on which this condition operates in Ippolito's model, appears slightly different (*Structured Sets of Alternatives*). Also, the *Specificity Condition* is not taken to be a relation, but a rather, a constraint defining which kind of alternative can be raised in e.g. disjunctive environments.

[11] This will be discussed in more detail when dealing with scalar alternatives such as $\langle$*some*, *all*$\rangle$, in Chapter **??**. It will be crucial that such alternatives, which do *not* have a common ancestor in their Hasse diagram (see Figure 2.11a), *can* be seen as same-granularity.

(89) *Same granularity relation* $\sim_g$. Let p and q be two propositions belonging to the same set of propositional alternatives. Let H be the Hasse diagram induced by $\vDash$ on this set of alternatives. If p and q have a common ancestor r in H, and the paths from r to p and r to q have same length, then $p \sim_g q$.

We now leverage this relation between propositions to define the layers of a Qtree as partitions induced by sets of same-granularity alternatives. We first observe that the relation $\sim_g$ defined in (89) can be used to divide the set of propositional alternatives to a given LF, into subsets sharing the same level of granularity. This gives rise to a "tiered" set of alternatives, as defined in (90).

(90) *Tiered set of propositional alternatives.* Let X be a sentence denoting a proposition p. A tiered set of propositional alternatives to X, is the set of sets of propositions, whose elements are the maximal sets of propositions related by the same-granularity relation. In other words, $\mathcal{A}_{p,X}^{\sim_g} = \{\{r \in \mathcal{A}_{p,X} \mid r \sim_g q\} \mid q \in \mathcal{A}_{p,X}\}$. If $\sim_g$ is an equivalence relation, $\mathcal{A}_{p,X}^{\sim_g}$ is a partition.

Tiered sets of alternatives are quite close to the *Structured Sets of Alternatives* defined in Ippolito (2019). One difference however, is that tiered sets of propositional alternatives are *not* assumed to include propositions corresponding to alternatives that are more complex than the original LF. The elements of a tiered set of propositional alternatives are sets of propositions and form same-granularity "tiers", as defined in (91). These tiers will be used to form Qtree layers. If $\sim_g$ is an equivalence relation when restricted to a specific set of propositional alternatives, then the resulting tiered set of alternatives will partition it, i.e. same-granularity tiers will be cells.

(91) *Same-granularity tier.* Let X be a sentence denoting a proposition p, and $\mathcal{A}_{p,X}$ its set of propositional alternatives. Let $q \in \mathcal{A}_{p,X}$. The set of same-granularity alternatives to q (in $\mathcal{A}_{p,X}$), is the set of propositions in $\mathcal{A}_{p,X}$ sharing same-granularity with q. We call this set $\mathcal{A}^q_{p,X}$. $\mathcal{A}^q_{p,X} = \{r \in \mathcal{A}_{p,X} \mid r \sim_g q\}$. $\mathcal{A}^q_{p,X}$ is a subset of the tiered set of propositional alternatives to X, $\mathcal{A}_{p,X}^{\sim_g}$. Moreover, if $\sim_g$ is an equivalence relation, then $\mathcal{A}^q_{p,X}$ constitutes a cells of $\mathcal{A}_{p,X}^{\sim_g}$.

2.3.3 Leveraging alternatives to generate Qtrees

We are now equipped to devise a recipe generating Qtrees out of simplex sentences, based on tiered sets of propositional alternatives. We start by considering the standard constraint on question-answer pairs, given in (92). This constraint establishes a connection between the standard set of alternatives derived from a sentence involving focus, and the kind of question this sentence answers.

(92) *Constraint on question-answer pairs* (*Rooth, 1992, to be revised*). A good question-answer pair (Q, A) is s.t. $[\![Q]\!] \subseteq [\![A]\!]^f$, where:

- $[\![Q]\!]$ corresponds to the alternative semantics of the question;
- $[\![A]\!]^f$ corresponds to the focus semantic value of the answer, i.e. the set of propositions denoted by LFs obtained from A *via* the substitution of A's focused material by a same-type element.

Let us show that this constraint is not sufficient (though, a good starter) for a model of questions evoked by assertions. We assume that A corresponds to the sentence *Jo grew up in PARIS*, where *PARIS* is focused. The focus semantic value of A then involves propositions denoted by LFs of the form *Jo grew up in l*, with *l* a location, e.g. *Paris*, *France*, or *Germany*. If the only constraint on the question Q accommodated from A was that $[\![Q]\!]$ should be a subset of $[\![A]\!]^f$, then, in principle, $[\![Q]\!]$ could be made of the three propositions that *Jo grew up in Paris*, *Jo grew up in France*, and *Jo grew up in Germany*. Granted that Paris is in France, and that France and Germany are disjoint, this set of alternatives would induce a partition of the CS of the form $\{\neg \textit{France} \wedge \neg \textit{Germany}, \textit{Germany}, \textit{France} \wedge \neg \textit{Paris}, \textit{Paris}\}$. This appears similar to the mixed-granularity layer that we said was problematic in Figure 2.13. So not all questions allowed by (92), given a fixed assertion, appear to make sense. There are two ways to alter (92) to avoid that kind of configuration: modify the relation between $[\![Q]\!]$ and $[\![A]\!]^f$, and/or, change $[\![A]\!]^f$ into something else.

We in fact opt for both options, and reuse the ideas presented in the previous Section. Specifically, we consider two subcases: the case in which $[\![Q]\!]$ simply corresponds to $\{[\![A]\!]\}$, and induces a partition of the CS of the form $\{[\![A]\!], \neg[\![A]\!]\}$; and the case foreshadowed in the previous section, in which $[\![Q]\!]$ corresponds to same-granularity alternatives to $[\![A]\!]$. These two cases are repeated in (93).

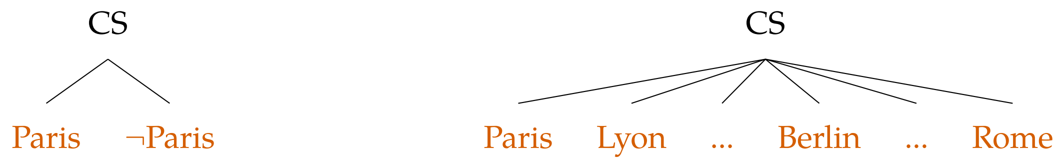
(93) *Constraint on question-answer pairs* (*first revision*). Let X be a LF denoting p. X evokes a question that is either:

(i) $[\![Q]\!] = \{p\}$;

(ii) $[\![Q]\!] = \mathcal{A}^p_{p,X}$, the set of same-granularity alternatives to p.

(93) allows assertions to evoke multiple potential Qtrees. According to (93), an assertion such as *Jo grew up in Paris*, will either evoke the question $[\![Q]\!] = \{\textit{Paris}\}$, inducing a

partition of the CS of the form $\{Paris, \neg Paris\}$, and corresponding to the polar question of whether or not Jo grew up in Paris; or, $[\![Q]\!] = \{Paris, Lyon, Nice, ..., Berlin, ..., Rome, ...\}$, inducing a similar partition of the CS, and corresponding to the *wh*-question *In which city did Jo grow up?*. These partitions are represented in Figure 2.14.



(a) A Qtree for (73a) assuming (93i)

(b) A Qtree for (73a) assuming (93ii)

Figure 2.14: One-layer Qtrees generable from (93) and the sentence (73a)=*Jo grew up in Paris*.

The above partitions seem more in line with intuitions than the pathological ones generable from (92). However, they still do not form layered Qtrees. Therefore, (93) is still not powerful enough to capture the specificity differences sketched in Figure 2.8 among others. Going one step further, we can assume that the Qtrees compatible with a sentence, are either generated by the proposition p denoted by the sentence (thus creating a Qtree like Figure 2.14a), or, by the sentence's tiered set of propositional alternatives, as defined in (90). Specifically in the latter case, it will be assumed that each layer of the Qtree corresponds to the partition induced on the CS by a same-granularity tier of propositional alternatives, and that layers are ordered in terms of granularity. Figure 2.14b constitutes the simplest subcase of this principle, in which only one layer gets generated out of same-granularity alternatives to p. In any case, verifying nodes are defined as the leaves of the tree entailing p (i.e. contained in p). This is formalized in (94). In this definition, (94ii) may be seen as a subcase of (94iii), in which the p-chain set to p only.

(94) *Qtrees for simplex LFs* (*to be further generalized in Chapter* **??**). Let X be a simplex LF denoting p, not settled in the CS. Let $\mathcal{A}_{p,X}$ be the set X's propositional alternatives. For any $q \in \mathcal{A}_{p,X}$, let $\mathcal{A}^q_{p,X} \subseteq \mathcal{A}_{p,X}$ be the set of alternatives from $\mathcal{A}_{p,X}$ sharing same granularity with q. We assume for simplicity that for any q, $\mathcal{A}^q_{p,X}$ partitions the CS. A Qtree for X is either:

(i) A depth-1 Qtree whose leaves denote $\mathfrak{P}_{\{p\},CS} = \{p, \neg p\}$

(ii) A depth-1 Qtree whose leaves denote $\mathfrak{P}_{\mathcal{A}^p_{p,X},CS} = \mathcal{A}^p_{p,X}$.

(iii) A depth-k Qtree ($k > 1$) constructed in the following way:

- Formation of a "p-chain" $p_0 = p \subset p_1 \subset ... \subset p_n$ where $p_0, ..., p_n$ are all in $\mathcal{A}_{p,X}$ but belong to different granularity tiers in $\mathcal{A}^{\sim g}_{p,X}$: $\mathcal{A}^{p_0}_{p,X} \neq \mathcal{A}^{p_1}_{p,X} \neq ... \neq \mathcal{A}^{p_n}_{p,X}$.
- Generation of the "layers" of the Qtree, based on the partitions induced by the granularity tiers corresponding to each element of the p-chain: $\left\{\mathfrak{P}_{\mathcal{A}^{p_i}_{p,X},CS} \mid i \in [0;n]\right\}$.
- Determination of the edges between nodes (cells) of adjacent layers (and between the highest layer and the root), based on the subset relation.[12]

In any case, $\boxed{\text{verifying nodes}}$ are defined as the set of leaves entailing p.

2.3.4 Applying the recipe to two simple sentences

We can now apply (94) to sentences like (73a) and (73b), repeated below.

(73) a. Jo grew up in Paris.
b. Jo grew up in France.

We start with (73a), and assume that its alternatives are of the form *Jo grew up in l*, with *l* a city or a country. Taking for granted that "city" propositions and "country" propositions form two distinct granularity tiers, the tiered set of propositional alternatives to (73a), will be as in (95).

(95) $$\begin{aligned}\mathcal{A}^{\sim g}_{\mathit{Paris},(73a)} &= \{\{\mathit{Paris}, \mathit{Lyon}, ..., \mathit{Berlin}, ...\}, \{\mathit{France}, \mathit{Germany}, ...\}\} \\ &= \{\{p \mid \exists l.\ l \text{ is a city} \wedge\ p = \lambda w. \text{ Jo grew up in } l \text{ in } w\}, \\ &\quad\ \{p \mid \exists l.\ l \text{ is a country} \wedge\ p = \lambda w. \text{ Jo grew up in } l \text{ in } w\}\}\end{aligned}$$

First, we can generate a Qtree for (73a) using principle (94i). This Qtree will have the CS as root, and two leaves corresponding to the propositions that *Jo grew up in Paris*, and *Jo did not grow up in Paris* (assuming this matter is not settled in the CS). This Qtree is depicted in Figure 2.15a. Intuitively, it corresponds to the question of whether or not Jo grew up in Paris.

Second, we can use principle (94ii). To do so, we must determine the set of same-granularity alternatives to the prejacent proposition that *Jo grew up in Paris*. This set, labeled $\mathcal{A}^{\mathit{Paris}}_{\mathit{Paris},(73a)}$, corresponds to the first element of the tiered set of propositional alternatives $\mathcal{A}^{\sim g}_{\mathit{Paris},(73a)}$ in (95). It is repeated in (96). The alternatives contained in $\mathcal{A}^{\mathit{Paris}}_{\mathit{Paris},(73a)}$ are all exclusive (cities are spatially disjoint), and moreover cover the space of possibilities.

[12] This may not always create well-formed Qtrees. Chapter **??** will explore such cases update (94) in consequence.

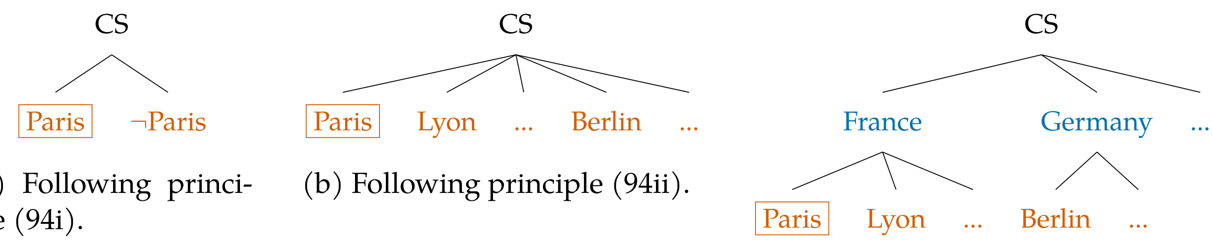
So, once intersected with the CS, they already form a partition of the CS. According to principle (94ii), this partition correspond to the leaves of the resulting Qtree. This Qtree is depicted in Figure 2.15b. Intuitively, it corresponds to the question of which city Jo grew up in.

(96) $$\begin{aligned} \mathcal{A}^{Paris}_{Paris,(73a)} &= \{\textit{Paris}, \textit{Lyon}, ..., \textit{Berlin}, ...\} \\ &= \{p \mid \exists l.\ l \text{ is a city} \wedge\ p = \lambda w. \text{ Jo grew up in } l \text{ in } w\} \\ &= \mathfrak{P}_{\{\textit{Paris},\textit{Lyon},...,\textit{Berlin},...\},CS} \end{aligned}$$

Third and lastly, we can use principle (94iii), which constitutes are multi-layer generalization of principle (94ii). To do so, we need to define a p-chain of propositions entailed $p = \lambda w$. *Jo grew up in Paris in* w. The tiered set of alternatives posited in (95) contains one such proposition, namely $p' = \lambda w$. *Jo grew up in France in* w. The resulting Qtree will therefore be made of three layers: the CS (root), the partition generated by the same granularity alternatives to p', and the partition generated by the same granularity alternatives to p, already defined in (96). The set of same-granularity alternatives to p', labeled $\mathcal{A}^{France}_{Paris,(73a)}$, corresponds to the second element of the tiered set of propositional alternatives $\mathcal{A}^{\sim g}_{Paris,(73a)}$ in (95). It is repeated in (97). The alternatives contained in $\mathcal{A}^{France}_{Paris,(73a)}$ are all exclusive (country are spatially disjoint), and moreover cover the space of possibilities. So, once intersected with the CS, they already form a partition of the CS.

(97) $$\begin{aligned} \mathcal{A}^{France}_{Paris,(73a)} &= \{\textit{France}, \textit{Germany}, ...\} \\ &= \{p \mid \exists l.\ l \text{ is a country} \wedge\ p = \lambda w. \text{ Jo grew up in } l \text{ in } w\} \\ &= \mathfrak{P}_{\{\textit{France},\textit{Germany},...\},CS} \end{aligned}$$

As per principle (94iii), a Qtree evoked by (73a) will then have the CS as top layer, the nodes corresponding to the partition in (97) as middle layer, and the nodes corresponding to the partition in (96) as bottom (leaf) layer. Connectivity between layers is straightforward: it corresponds to the inclusion relation between cities and countries, and between countries and "the whole world" ($\sim$CS). The resulting Qtree is given in Figure 2.15c. Intuitively, it corresponds to the question of which city Jo grew up in, but such that this question is decomposed into two subquestions: first, which country Jo grew up in; then, knowing the country, which city Jo grew up in, in that country.



(a) Following principle (94i).

(b) Following principle (94ii).

(c) Following principle (94iii).

Figure 2.15: Possible Qtrees evoked by the assertion (73a)=*Jo grew up in Paris*.

Of course, if more alternatives to (73a) had been posited in the first place, principle (94iii) would have produced more Qtrees. For instance, if continent alternative had been considered, the tiered set of propositional alternatives to (73a), $\mathcal{A}^{\sim g}_{Paris,(73a)}$, would have been as in (98), and the Qtrees generated by principle (94iii), would have been the one in Figure 2.15c, plus the one in Figure 2.16.

(98) $$\begin{aligned}\mathcal{A}^{\sim g}_{Paris,(73a)} &= \{\{Paris, Lyon, ..., Berlin, ...\}, \{France, Germany, ...\}, \{Europe, Asia, ...\}\}\\ &= \{\{p \mid \exists l.\ l \text{ is a city} \wedge\ p = \lambda w.\ \text{Jo grew up in } l \text{ in } w\},\\ &\qquad \{p \mid \exists l.\ l \text{ is a country} \wedge\ p = \lambda w.\ \text{Jo grew up in } l \text{ in } w\}\\ &\qquad \{p \mid \exists l.\ l \text{ is a continent} \wedge\ p = \lambda w.\ \text{Jo grew up in } l \text{ in } w\}\}\end{aligned}$$

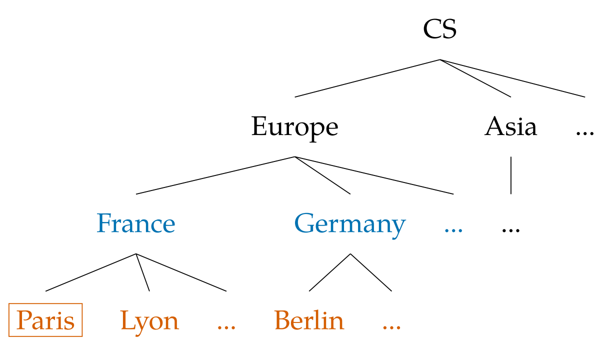


Figure 2.16: An extra Qtree for (73a), generated by principle (94iii), assuming that (73a)'s tiered set of propositional alternatives is as in (98). .

For simplicity and ease of comparison, we will stick to a tiered set of alternatives involving city- and country-tiers, as defined in (95). Similarly, we can derive Qtrees for (73b)=*Jo grew up in France*. This assertion will in fact require less work, because it appears coarser-grained than (73a). To ensure that (73a) and (73b) are analyzed at the same level of approximation, we assume that (73b)'s alternatives are also of the form *Jo grew up in l*, with *l* a city or a country. The tiered set of propositional alternatives to (73b) is therefore identical to that of (73a), and given in (99).

(99) $\mathcal{A}^{\sim g}_{France,(73b)} = \{\{Paris, Lyon, ..., Berlin, ...\}, \{France, Germany, ...\}\}$
$= \{\{p \mid \exists l.\ l \text{ is a city} \wedge\ p = \lambda w. \text{Jo grew up in } l \text{ in } w\},$
$\{p \mid \exists l.\ l \text{ is a country} \wedge\ p = \lambda w. \text{Jo grew up in } l \text{ in } w\}\}$
$= \mathcal{A}^{\sim g}_{Paris,(73a)}$

First, we can generate a Qtree for (73b) using principle (94i). This Qtree will have the CS as root, and two leaves corresponding to the propositions that *Jo grew up in France*, and *Jo did not grow up in France* (assuming this matter is not settled in the CS). This Qtree is depicted in Figure 2.17a. Intuitively, it corresponds to the question of whether or not Jo grew up in France.

Second, we can use principle (94ii). To do so, we must determine the set of same-granularity alternatives to the prejacent proposition that *Jo grew up in France*. This set, labeled $\mathcal{A}^{France}_{France,(73b)}$, corresponds to the second element of the tiered set of propositional alternatives $\mathcal{A}^{\sim g}_{France,(73b)}$. It is repeated in (100). This set is also equal to the set of same-granularity alternative to *France*, when the prejacent was *Paris* (see (97)). Thus, the alternatives in this set, once intersected with the CS, already form a partition of the CS. According to principle (94ii), this partition correspond to the leaves of the resulting Qtree. This Qtree is depicted in Figure 2.17b. Intuitively, it corresponds to the question of which country Jo grew up in.

(100) $\mathcal{A}^{France}_{France,(73b)} = \{France, Germany, ...\}$
$= \{p \mid \exists l.\ l \text{ is a city} \wedge\ p = \lambda w. \text{Jo grew up in } l \text{ in } w\}$
$= \mathfrak{P}_{\{France, Germany, ...\}, CS}$
$= \mathcal{A}^{France}_{Paris,(73a)}$

Third and lastly, we could use principle (94iii), but this principle would in fact give us nothing more than principle (94ii), given our assumptions about (73b)'s tiered set of propositional alternatives. This is because no proposition in $\mathcal{A}^{\sim g}_{France,(73b)}$ is weaker than $p{=}\lambda w.$ *Jo grew up in France in* w, and therefore, the only p-chain available in the case of (73b), is made of simply p. This p-chain would generate one single country-layer beyond the CS root, and the resulting Qtree, would simply be the one in Figure 2.17b.[13]

[13]Of course, if we had considered continent-level alternatives as well, principle (94iii) would have generated an extra Qtree for (73b), characterized by a continent-layer on top of a country-layer. But this would have led us to do the same move for (73a), and thus to generate the Qtree in Figure 2.16 for that sentence.

(a) Following principle (94i). (b) Following principle (94ii).

Figure 2.17: Possible Qtrees evoked by the assertion (73b)=*Jo grew up in France*.

Before moving on to "compositional" Qtrees, let us take stock.

First, the recipe in (94) defines way to determine which parse of the CS assertive sentences evoke. We have discussed in Chapter 1 that questions typically correspond to partitions of the CS *in the pragmatic domain*. Semantically, questions are taken to be sets of alternatives. In that sense, Qtrees evoked by sentences should be understood as a form of "inquisitive pragmatics" rather than "inquisitive semantics". Tiered sets of propositional alternatives may be closer to the latter concept.

Second, the recipe in (94) typically generates *multiple* Qtrees out of one assertion. Under our current assumptions, (73a) gives rise to three possible Qtrees, and (73b), to two. So there is some degree of uncertainty about which Qtree any given sentence actually answers. Very roughly, evoked Qtrees can be "polar" (principle (94i)), "*wh*" (principle (94ii)), or "*wh*-articulated" (principle (94iii)). This optionality contrasts with frameworks like inquisitive semantics, in which any given sentence is mapped to a single nonempty downward-closed set of propositions. Given this, our recipe generates more Qtrees than intuitively assumed in the previous Sections. Additionally, this leads us to define the oddness of a sentence as equivalent to the oddness of *all* sentence-Qtree pairs to sentences can generate. This was already defined in (76) and (77), both repeated below.

(76) *Oddness of a Qtree, given a sentence.* If a sentence S evokes a Qtree T and the pair (S, T) induces a vacuous labeling of verifying nodes, or violates other sentence-Qtree well-formedness constraints (tbd), then T is deemed odd given S.

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

Third, we mentioned that (73a) and (73b), beyond the fact that they are obviously in a relation of logical entailment, are such that (73a) feels more "fine-grained" than (73b). This is somehow cashed out by the kind of Qtrees these sentences evoke. Specifically, we observe that some Qtrees (73a) evokes (Figure 2.15c) constitute refinements of some Qtree (73b) evokes (Figure 2.17b), where refinement is defined as in (72), repeated below. This implication does not hold in the opposite direction: no Qtree (73b) evokes, constitutes a refinement of a Qtree (73a) evokes. So (in a very weak sense) finer-grained assertions evoke finer-grained Qtrees. This observation will be crucial in Chapter **??**.

(72) *Qtree refinement.* Let T and T' be Qtrees. T is a refinement of T' (or: T is finer-grained than T'), iff T' can be obtained from T by removing a subset $\mathcal{T}$ of T's subtrees, s.t.:

- if $\mathcal{T}$ contains a subtree rooted in N, then, for each node N' that is a sibling of N in T, the subtree of T rooted in N', is also in $\mathcal{T}$.

We now proceed to define Qtree for complex sentences belonging to the $\{\neg, \vee, \rightarrow\}$-fragment of the language. We will do so inductively, using our recipe for simplex sentences (94) as base case, along with specific combination rules corresponding to the inquisitive effect of each operator.

2.4 Compositional Qtrees: inductive step

In the previous Section, we have seen how to derive Qtrees from simplex sentences, containing no operator or connective. In this Section, we clarify how complex sentences, that may be equally informative, and may even have same propositional meaning, may end up packaging information differently from one another, in terms of their evoked Qtrees. This difference in information packaging, will allow us to derive different felicity profiles for these sentences. We start with Qtrees evoked by negated LFs, before moving on to Qtrees evoked by disjunctions and conditionals.

2.4.1 Questions evoked by negated LFs

We assume negated LFs evoke questions that are structurally similar to those evoked by their non-negated counterpart. The only difference resides in the set of verifying nodes, which is "flipped" by negation. This is formalized in (101).[14]

(101) *Qtrees for negated LFs.* Let T be a Qtree evoked by a LF X. A Qtree $T_\neg$ for $\neg X$ is obtained from T by:

- retaining T's structure; i.e. if $T = (\mathcal{N}, \mathcal{E}, R)$, then $T_\neg = (\mathcal{N}, \mathcal{E}, R)$, too;
- defining $T_\neg$'s set of verifying nodes $\mathcal{N}^+(T_\neg)$ as the set of $T_\neg$'s nodes N that are not verifying in T ($N \notin \mathcal{N}^+(T)$) but belong to a layer containing at least one verifying node N' in T ($N' \in \mathcal{N}^+(T)$). In other words:

[14]This approach is perhaps a bit naive; uttering p vs. $\neg p$, does not seem to preferentially answer the same kind of question. More specifically, it seems that uttering negative statements in general conveys the idea that the original question was more likely to be a polar question of the form *whether p?*–as opposed to a *wh* kind of question. We discuss this more in depth in Chapter **??**.

$\mathcal{N}^+(T_\neg) = \{N \mid N \notin \mathcal{N}^+(T) \wedge \exists N' \in \mathcal{N}^+(T).\ d(N, T_\neg) = d(N', T_\neg)\}$
With $d(N,T)$ the depth of a node N in a tree T (see (60b)).[15]

The recipe in (101) is exemplified in the abstract Qtrees in Figure 2.18.

(a) An abstract Qtree for X

(b) An abstract Qtree for $\neg X$, derived from Figure 2.18a.

Figure 2.18: An abstract Qtree for X and the abstract Qtree for $\neg X$ derived from it, *via* (101).

It may not seem obvious at this point why and how verifying nodes would occur at intermediate levels in a Qtree; after all, all the Qtrees we have seen so far (derived from simplex sentences) had their verifying nodes at the leaf level. But we will see that Qtrees derived from complex sentences (typically, involving disjunctions and conditionals) can in principle feature intermediate verifying nodes, because such nodes are also derived compositionally. Now, granted that verifying nodes may indeed occur at different levels, the intuition behind the "flipping" algorithm in (101) is the following. If a node N is verifying in a Qtree T corresponding to an LF X, and N is located at depth k in T, then somehow the k-layer of T is "addressed" by X. We aim for a pair $(\neg X, T_\neg)$ to address the same layers as (X, T), so $T_\neg$'s verifying nodes should have similar a similar depth distribution as T's verifying nodes. But of course, the two sets of nodes need to be distinct, because negation standardly flips truth values–hence the by-layer flipping.

It is additionally worth mentioning that, if all verifying nodes in the original Qtree T are leaves, (101) is simplified: $T_\neg$'s set of verifying nodes is simply the set of leaves in $T/T_\neg$ that are not verifying in T. This is summarized in (102).

[15]Because T and $T_\neg$ have same structure, it does not matter which Qtree among T and $T_\neg$ is passed as argument to the depth function; in that particular case, $\forall N \in \mathcal{N}.\ d(N,T) = d(N,T_\neg)$.

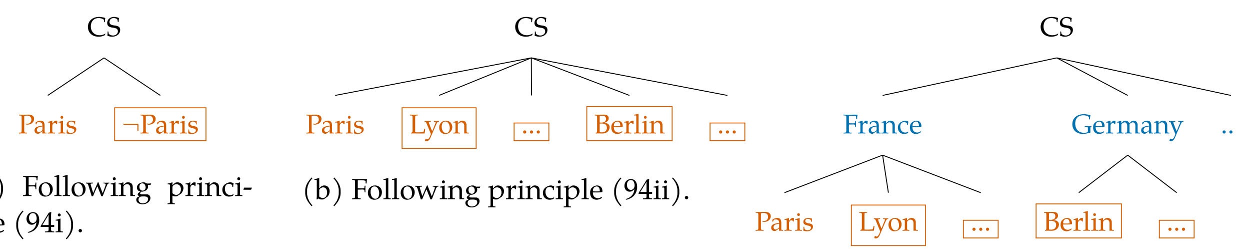
(102) *Qtrees for negated LFs (leaf-only version, subcase of (101)).* Let T be a Qtree evoked by a LF X s.t. $\mathcal{N}^+(T) \subseteq \mathcal{L}(T)$, where $\mathcal{L}(T)$ refers to T's leaves. A Qtree $T_\neg$ for $\neg X$ is obtained from T by:

- retaining T's structure;
- defining $T_\neg$'s set of verifying nodes as the complement set of $\mathcal{N}^+(T)$ within $\mathcal{L}(T)$: $\mathcal{N}^+(T_\neg) = \{N \in \mathcal{L}(T) \mid N \notin \mathcal{N}^+(T)\}$

Following this simplified recipe, Qtrees for (103), which correspond to the negation of (73a), are given below. They are obtained from Figure 2.15, by simply flipping boxed nodes at the leaf level. These new Qtree capture the intuition that (103) can answer three kinds of question: a question about whether or not Jo grew up in Paris; a question about which city Jo grew up in; and a question about which city Jo grew up in question of which city Jo grew up in, but such that this question is decomposed into two subquestions: first, which country Jo grew up in; then, knowing the country, which city Jo grew up in, in that country. The nodes that get flagged as verifying, correspond to sets of worlds disjoint from $\lambda w.$ *Jo grew up in Paris in* w. Interestingly, negation preserves Qtree granularity, simply because it preserves Qtree structure.

(73a) Jo grew up in Paris.

(103) Jo did not grow up in Paris.



(a) Following principle (94i).

(b) Following principle (94ii).

(c) Following principle (94iii).

Figure 2.19: Possible Qtrees evoked by the assertion (103)=*Jo did not grow up in Paris*.

2.4.2 Questions evoked by disjunctive LFs

Let us consider the disjunction in (104). Intuitively, this sentence is a good, non-maximal answer to a question like (48a), repeated below. It identifies two cities in which Jo could have grown up, and conveys ignorance about which city Jo actually grew up in. Note that either disjunct taken in isolation, *Jo grew up in Paris*, or *Jo grew up in Lyon*, constitutes a *maximal* answer to (48a).

(48a) In which city did Jo grow up?

(104) Jo grew up in Paris or Lyon.

This observation is consistent with the idea that, in a felicitous disjunction, both disjuncts must answer the same kind of question (Simons, 2001; Zhang, 2022). Ou rephrasing of this observation is spelled out in (105).

(105) *Disjunctive answer.* Let $X = Y \vee Z$ be a disjunctive LF. If X is a felicitous assertion, then the set of questions Y answers is equal to the set of questions Z answers. Additionally, if Y/Z answer a question, then X answers it too.

A way to further specify this intuition in our model, is to assume that a Qtree for $X = Y \vee Z$, must contain a Qtree for Y and a Qtree for Z. Containment is understood as the subgraph relation (defined in (71)). This ensures that any node in Y's Qtree is also in X's Qtree, and any node in Z's Qtree, is also in X's Qtree. So, whatever answers Y or Z, also answers X. This is modeled by assuming that the Qtrees evoked by a disjunction are all the possible well-formed unions of Qtrees evoked by each disjunct. This is spelled out in (106). In this definition, Qtree union builds on the notion of graph-union, as formalized in (107).[16] On top of this, Qtree union involves the union of verifying nodes, and the determination of a root node for the output Qtree, defined as the maximum between the two roots of the input Qtrees.

(106) *Qtrees for disjunctive LFs.* A Qtree $T_\vee$ for $X \vee Y$, if defined, is obtained from a Qtree T_X for X and a Qtree T_Y for Y by:

- graph-unioning T_X and T_Y;
- defining $T_\vee$'s root as the maximal element (i.e. the weaker proposition) between the root of T_X and the root of T_Y. This will typically be the entire CS. If there is no such maximum, then the output cannot be a Qtree. [17]
- defining $T_\vee$'s verifying nodes as the union of T_X's and T_Y's verifying nodes: $\mathcal{N}^+(T_\vee) = \mathcal{N}^+(T_X) \cup \mathcal{N}^+(T_Y)$.

[16] I thank Amir who helped me see this.

[17] Indeed, suppose R_X and R_Y are the roots of respectively T_X and T_Y, and that R_X and R_Y are not in any kind of inclusion relation. We show by contradiction that $T_X \cup T_Y$ cannot be a Qtree. If $T_X \cup T_Y$ were a Qtree, then, R_X and R_Y would not be in an ancestry relation, meaning, R_X would not be an ancestor of R_Y, and R_Y would not be an ancestor of R_X. So, neither R_X nor R_Y could be the root of $T_X \cup T_Y$, because the root is an ancestor of all the other nodes. Let's call R this root. R is a common ancestor of both R_X and R_Y in $T_X \cup T_Y$. So R must be a strict superset of R_X and R_Y. Also, because $T_X \cup T_Y$ is obtained *via* node- and edge-union, we must have, in the input Qtrees: $R_X \overset{T_X}{\rightsquigarrow} R$ and $R_Y \overset{T_Y}{\rightsquigarrow} R$. In other words, R_X is an ancestor of R in T_X, and R_Y is an ancestor of R in T_Y. Because T_X and T_Y are Qtrees, this implies that R is a strict subset of R_X, and also strict subset of R_Y. Contradiction.

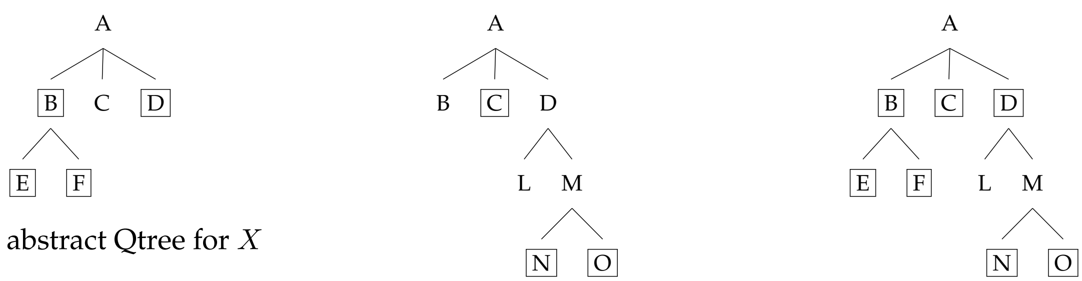
- returning the output only if it is a Qtree.

In other words, $Qtrees(X \vee Y) = \{T_X \cup T_Y \mid T_X \cup T_Y$ verifies (52) $\wedge (T_X, T_Y) \in Qtrees(X) \times Qtrees(Y)\}$

(107) *Graph union.* Let $G = (\mathcal{N}, \mathcal{E})$ and $G' = (\mathcal{N}', \mathcal{E}')$ be two graphs. The union of G and G', noted $G \cup G'$, is the graph $G'' = (\mathcal{N}'', \mathcal{E}'')$ s.t.:

- $\mathcal{N}'' = \mathcal{N} \cup \mathcal{N}'$
- $\mathcal{E}'' = \mathcal{E} \cup \mathcal{E}'$

Figure 2.20 below exemplifies Qtree union applied to two abstract Qtrees, represented in Figures 2.20a and 2.20b. In these Qtrees, nodes with different labels are assumed to correspond to a different propositions. By definition, $\{B, C, D\}$ partitions A; $\{E, F\}$ partitions B, $\{L, M\}$ partition D, and $\{N, O\}$ partition M. The disjunction of Figures 2.20a and 2.20b is shown in Figure 2.20c. Nodes, edges, and verifying nodes, are unioned, and the output is a Qtree, that contains the two input Qtrees. So, whatever answered either Qtree in Figures 2.20a and 2.20b, also answers their disjunction in Figure 2.20c.



(a) An abstract Qtree for X

(b) An abstract Qtree for Y.

(c) An abstract Qtree for $X = Y \vee Z$, derived from Figures 2.20a and 2.20b.

Figure 2.20: Successful attempt at deriving a Qtree from the union of two Qtrees.

It can be shown that if a disjunctive Qtree $T_\vee$ is well-formed and results from the union of two Qtrees T and T' sharing the same root, $T_\vee$ will always constitute a refinement of both T and T'.

What about cases in which the union of two Qtrees, is not a well-formed Qtree? A prediction of (106) is that two Qtrees sharing the same root can be properly disjoined iff they do not involve a common node that gets partitioned in two different ways in the two different input Qtrees.[18] We call this problematic configuration a partition "clash" (or simply a clash). It is formally defined in (108), and related to disjoinability in (109).

[18]We show that if T and T' exhibit such a clash, their disjunction is not a Q-tree. Let's call C and C' the

(108) *Partition clash.* Let $T = (\mathcal{N}, \mathcal{E}, R)$ and $T' = (\mathcal{N}', \mathcal{E}', R')$ be two Qtrees. T and T' feature a partition clash iff there is $N \in \mathcal{N}$ and $N' \in \mathcal{N}'$ s.t. $N = N'$ but the sets of children of N and N' differ.

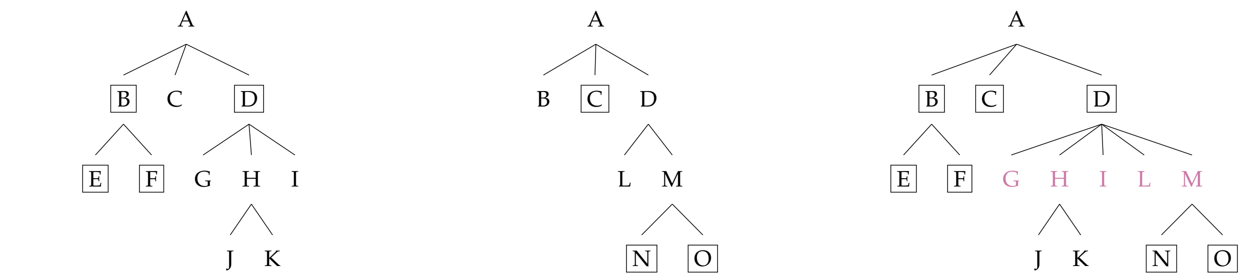
(109) *Partition clashes and Qtree disjoinability.* Let $T = (\mathcal{N}, \mathcal{E}, R)$ and $T' = (\mathcal{N}', \mathcal{E}', R)$ be two Qtrees. T and T' are disjoinable (i.e., their union is a well-formed Qtre) iff T and T' do not exhibit any partition clash.

So, under a recursive interpretation of nodes, two Qtrees with the same root can be disjoined iff, for each node N present in both Qtrees, N's recursive interpretation is the same across Qtrees, or one interpretation constitutes a refinement of the other. This means that, to be disjoinable Qtrees should not introduce different subquestions at the local level.

Figure 2.21 illustrates a degenerate case of Qtree union, arising from a partition clash between two abstract input Qtrees. The two input Qtrees, represented in Figures 2.21a and 2.21b, minimally differ from those in Figures 2.21a and 2.21b: Figures 2.20b and 2.21b are the same, but, in Figure 2.21a, $\{G, H, I\}$ are extra nodes that partition D, and $\{J, K\}$ partitions H. The "clash" between the Qtrees in Figures 2.21a and 2.21b comes from the $\{G, H, I\}$ nodes in Figure 2.20a and the $\{L, M\}$ nodes in Figure 2.20b: these two sets partitions node D in different ways. As a result, the union of these two sets of nodes *cannot* partition D. Figure 2.21c, which represents the disjunction of Figures 2.21a and 2.21b, thus features nodes $\{G, H, I, L, M\}$ as children of node D, and this configuration violates the partition property of Qtrees. This prevents the tree in Figure 2.21c from being a well-formed Qtree.

---

sets of nodes of resp. T and T' that induce a bracketing clash; by assumption, 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 $C \neq C'$, $C \cup C' \supset C, C'$. Given that both C and C' are partitions of N, $C \cup C'$ cannot be a partition of N. Conversely, if two Q-trees T and T' sharing the same CS as root are s.t. their union $T \cup T'$ is not a Qtree, it must be because T and T' had a bracketing clash. Indeed, under those assumptions, $T \cup T'$ not being a Qtree means one node N in $T \cup T'$ is not partitioned by its children. Given N is in $T \cup T'$, N is also in T, T', or both. If N was only in, say, T, then it means N's children are also only in T, but then, T itself would have had a node not partitionned by its children, contrary to the assumption T is a Qtree. The same holds *mutatis mutandis* for T', so, N must come from *both* T and T'. Let us call C and C' the partitioning introduced by N in resp. T and T'. The fact C, C', but not $C \cup C'$ partition N entails $C \neq C'$, i.e. T and T' feature a bracketing clash.



(a) An abstract Qtree for X (b) An abstract Qtree for Y. (c) An abstract Qtree for $X = Y \vee Z$, derived from Figures 2.21a and 2.21b.

Figure 2.21: Unsuccessful attempt at deriving a Qtree from the union of two Qtrees exhibiting a bracketing clash.

The badness of this kind of configuration, captures the intuition that two disjoined Qtrees should not raise orthogonal issues locally. We call two issues (partitions) orthogonal if they involve two nodes/cells that strictly overlap; see (110). This definition can be shown to be equivalent to that of a partition clash,[19] It is interesting, because it can be more directly related to some concept of Relevance discussed in Chapter 1; see (111).

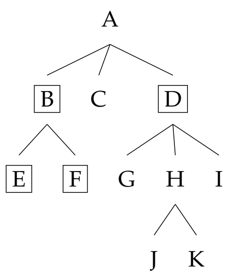
(110) *Orthogonal partitions.* Let $T = (\mathcal{N}, \mathcal{E}, R)$ and $T' = (\mathcal{N}', \mathcal{E}, R)$ be two depth-1 Qtrees sharing the same root R (equivalently, two partitions of the same CS). T and T' are orthogonal iff they involves two nodes that are strictly overlapping, i.e. $\exists (N, N') \in \mathcal{N} \times \mathcal{N}'.\ N \cap N' \neq \emptyset \wedge N \neq N'$. T and T' are orthogonal iff T and T' exhibit a partition clash.

(111) *Orthogonal partitions and relevance.* Let $T = (\mathcal{N}, \mathcal{E}, R)$ and $T' = (\mathcal{N}', \mathcal{E}, R)$ be two depth-1 Qtrees sharing the same root R. T and T' are orthogonal iff some maximal answer (leaf) of T is not Lewis-Relevant to T'.

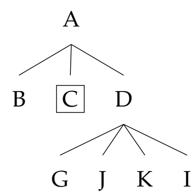
Figure 2.22 illustrates yet another degenerate case, that may seem more subtle when looking at the two input Qtrees, but with more drastic consequences when looking at the output structure, that is not even a tree. In this example, the two input Qtrees, represented

[19] Let us show that if two partitions are different (i.e. involve different cells), then, there is one cell from the former partition and one cell from the latter partition that strictly overlap. Let us assume two partitions P_1 and P_2 are distinct. We show that there is a cell in P_1 and a cell in P_2 that strictly overlap. We consider P_1' and P_2' the partitions obtained from P_1 and P_2 by removing the cells P_1 and P_2 have in common. P_1' and P_2' are not empty, because otherwise P_1 and P_2 would be identical. Moreover, there must be 2 cells c_1 and c_2 in P_1' and P_2' that overlap, because P_1' and P_2' are partitions and as such must be fully covered by their cells. Moreover, c_1 and c_2 cannot be the same, otherwise, they would not be in P_1' and P_2' by construction. So c_1 and c_2 strictly overlap. The other direction of the proof is trivial: if two partitions of the same space P_1 and P_2 involve two strictly overlapping cells, then these two cells must be distinct, and so P_1 and P_2 must be different sets.

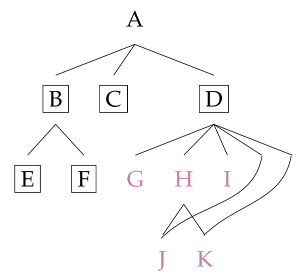
in Figures 2.22a and 2.22b clash again at the level of the D node: both $\{G, H, I\}$ $\{G, J, K, I\}$ partition D, but in different ways, since the latter partition is finer grained ($\{J, K\}$ partitions H). This kind of clash, though subtle, generates a disjunctive Qtree that is not even a tree: in Figure 2.22c, J/K is connected to D *via* two distinct paths: directly, and *via* H. So Figure 2.22c is not acyclic. Zooming out, this degenerate configuration stems from the fact that Qtree union "collapsed" the J and K nodes from the two input Qtrees, and that these nodes, being located at different levels in the two Qtree, were connected differently to the other nodes. This example outlines the idea that, in order to be disjoinable, two Qtrees must match in terms of their layering, i.e. in terms of their degrees of granularity.



(a) An abstract Qtree for X



(b) An abstract Qtree for Y.



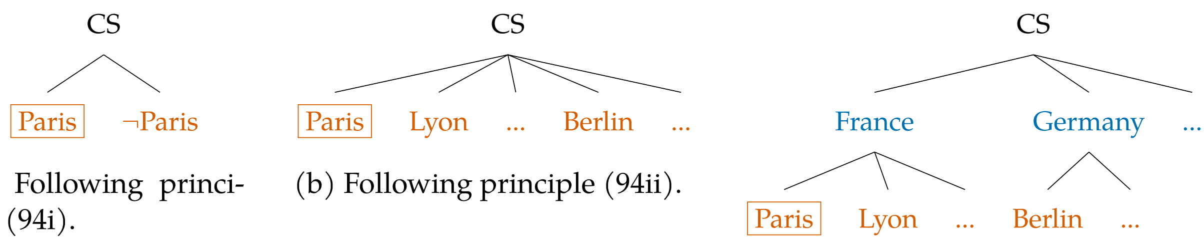
(c) An abstract Qtree for $X = Y \vee Z$, derived from Figures 2.22a and 2.22b.

Figure 2.22: Yet another unsuccessful attempt at deriving a Qtree from the union of two Qtrees exhibiting a bracketing clash.

Now that we have defined how disjunctive Qtrees are formed and what the well-formedness conditions for such trees are, we come back to our more concrete disjunctive example (104), repeated below.

(104) Jo grew up in Paris or Lyon.

To derive the Qtrees evoked by this disjunctive LF, one must first derive the Qtrees evoked by its two disjuncts, abbreviated *Paris* and *Lyon*. This has been done already in Figure 2.15 (repeated in Figure 2.23) for *Paris*. Additionally, *Paris* and *Lyon* have same granularity, and therefore, give rise to the same tiered set of propositional alternatives. This in turn ensures that both *Paris* and *Lyon* give rise to similar Qtrees, that mostly differ in terms of their verifying nodes: *Paris* will flag *Paris*-nodes, and *Lyon*, *Lyon*-nodes. The Qtrees evoked by *Lyon* can be found in Figure 2.24.

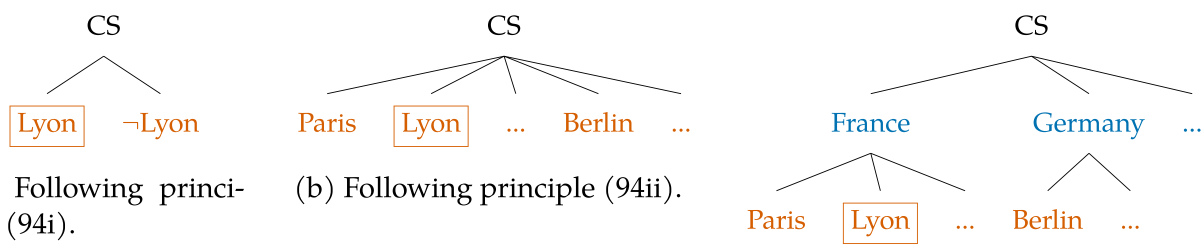


(a) Following principle (94i).

(b) Following principle (94ii).

(c) Following principle (94iii).

Figure 2.23: Possible Qtrees evoked by the assertion (73a)=*Jo grew up in Paris*.



(a) Following principle (94i).

(b) Following principle (94ii).

(c) Following principle (94iii).

Figure 2.24: Possible Qtrees evoked by the assertion *Jo grew up in Lyon*.

We could now compute all possible unions of the Qtrees in Figures 2.23 and 2.24, and retain those that are Qtrees. This would effectively yield the Qtree evoked by (104). But instead of computing all these unions, let us use the notion of a partition clash to retain the input Qtrees that will in fact give rise to well-formed disjunctive Qtrees. We now evaluate all pairs of Qtrees from Figures 2.23 and 2.24 for partition clashes, and compute unions only if no clash is detected.

Starting with the two "polar" Qtrees 2.23a and 2.24a, we notice an obvious clash between the 2-cells partitions $\{Paris, \neg Paris\}$ and $\{Lyon, \neg Lyon\}$. So we can ignore the union of these two Qtrees. Qtrees 2.23a and 2.24b also clash, because $\{Paris, \neg Paris\}$ and $\{Paris, Lyon...\}$ are different partitions. Again, we ignore this combination. Same holds for Qtrees 2.23a and 2.24c, because $\{Paris, \neg Paris\}$ and $\{France, Germany...\}$ are different. We thus once again ignore this combination. From this, we conclude that the "polar" Qtree for *Paris* 2.23a is not disjoinable with any Qtree *Lyon* evokes. Reciprocally, the "polar" Qtree for *Lyon* 2.24a is not disjoinable with any Qtree *Paris* evokes.

Moving on to the "*wh*" Qtree for *Paris* 2.23b, it is structurally identical to the "*wh*" Qtree for *Lyon* 2.24b. Therefore, these two Qtrees do not clash, and can de disjoined. Their union is given in Figure 2.25a. Because the two input Qtrees are structurally identical, the structure of the disjunctive output Qtree is also similar. The only difference between inputs and output, is that the nodes flagged as verifying by the output Qtree, are both the *Paris* and the *Lyon* node. Considering now the "*wh*" Qtree for *Paris* 2.23b, and the "*wh*-articulated" Qtree for *Lyon* 2.24c, we notice yet another partition clash: $\{Paris, Lyon, ...\}$ is different from $\{France, Germany, ...\}$. So these two Qtrees cannot be disjoined. Reciprocally, the "*wh*" Qtree for *Lyon* 2.24b, and the "*wh*-articulated" Qtree for *Paris* 2.23c, will not be disjoinable.

This leaves us with one last pair to evaluate, namely the pair made by the two "*wh*-articulated" Qtrees in Figures 2.23c and 2.24c. These two Qtrees are structurally identical, and so can be disjoined. The result of their union is given in Figure 2.25b. Because the two input Qtrees are structurally identical, the output is also similar. The only difference between inputs and output, is that the nodes flagged as verifying by the output disjunctive Qtree, are both the *Paris* and the *Lyon* node.

(a) Qtree 2.23b ∨ Qtree 2.24b

(b) Qtree 2.23c ∨ Qtree 2.24c

Figure 2.25: Possible Qtrees evoked by the assertion (104)=*Jo grew up in Paris or Lyon*.

Figure 2.25 capture the idea that a disjunction like (104), evokes the same Qtrees as the *wh*-question *In which city did Jo grow up?*: either a simple "*wh*" Qtree partitioning the CS according to cities, or a more complex "*wh*-articulated" Qtree corresponding to the question of which city Jo grew up in, but such that this question is decomposed into two subquestions: first, which country Jo grew up in; then, knowing the country, which city Jo grew up in, in that country.[20] We will see more examples of Qtree disjunctions in the next Chapters, including pathological cases in which the disjuncts may not share the same degree of specificity. We now proceed to define Qtree evoked by conditionals. Crucially, the way such Qtrees will be defined, will not be a function of the "recipes" we just devised for negated and disjunctive LFs. In other words, conditional Qtrees will not be "material".

[20] One might wonder at this point why a condition on Qtree disjoinability should not involve structural equality between inputs. After all, the two Qtrees we just derived, depicted in Figure 2.25, were associated with structurally identical inputs. Chapter **??** will discuss why this identity condition might be too strong, on top of being stipulative.

2.4.3 Questions evoked by conditional LFs

Material implication, defined in (112) is perhaps the simplest way to analyze natural language conditionals.

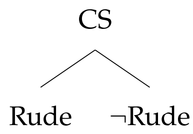
(112) *Material Implication.* Let X and Y be two LFs denoting p and q respectively. $[\![$ *If* X *then* $Y]\!]$ is true iff $\neg p \vee q$ is true. $\rightarrow$ is used as a shorthand for $\lambda p.\, \lambda q.\, \lambda w.\, \neg p \vee q$.

It may be tempting to adapt this definition to the domain of Qtrees evoked by assertions. This tentative translation is given in (113).

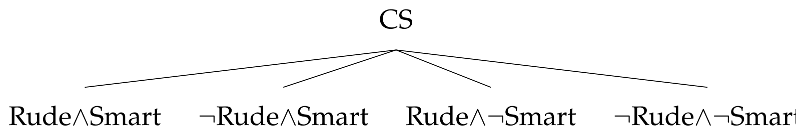
(113) *"Material" Conditional Qtrees.* Let X be an LF of the form *If* Y *then* Z. A Qtree for X is a Qtree for $\neg Y \vee Z$.

Because we defined Qtrees for negated and disjunctive LFs in the previous Sections, we already have the tools to understand what (113) would predict for Qtrees evoked by natural language conditionals. In particular, we noted that negation preserves Qtree structure, and that disjunction forces the two disjuncts to evoke structurally similar Qtrees. These properties combined, predict that, under (113) the antecedent and consequent of a conditional, should evoke similar Qtrees, devoid of any partition clash. In other words, two Qtrees evoked by X and Y should be "conditionalizable" (in the material sense) iff they are disjoinable. This does not seem to match intuitions about conditionals. (114a) for instance, sounds fine, even if the antecedent *Jo is rude* and the consequent *Jo grew up in Paris*, appear to evoke Qtrees with very different structures. The previous Section already detailed what the latter Qtrees for *Jo grew up in Paris* should looks like, and Figure 2.26 sketches how Qtrees for *Jo is rude should look like*. Clearly, partitions of the CS induces by personality traits, are unlikely to match partitions induced by countries, so under the material analysis, a sentence like (114a) should behave exactly like (114b) at the inquisitive level. Therefore, it should not give rise to any Qtree and should be deemed odd.

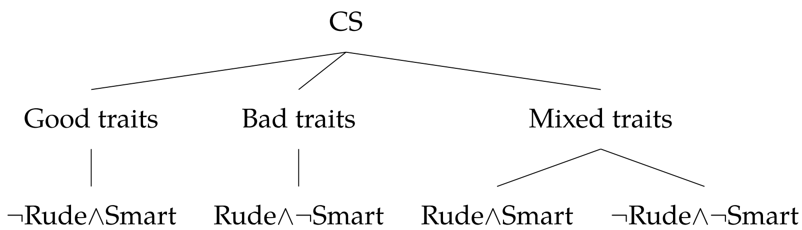
(114) a. If Jo is rude, she grew up in Paris.
b. # Jo is not rude, or she grew up in Paris.



(a) Following principle (94i).



(b) Following principle (94ii).



(c) Following principle (94iii).

Figure 2.26: Possible Qtrees evoked by the assertion *Jo is rude*.

This empirical difference between conditionals and disjunctions regarding the questions they evoke, motivates a non-material model of conditionals at the inquisitive level. Intuitively, what a conditional statement like (114a) seems to convey, is that figuring out Jo's rudeness may help narrow down where Jo grew up. So, (114a) seems to primarily answer a question about where Jo grew up, taking for granted that she is a rude person. This introduces an asymmetry between antecedent and consequent; it seems that the question evoked by the consequent gets *restricted* to the CS updated with the antecedent. A Qtree for (114a) would then look like the one in Figure 2.27. In this tree, the Qtree corresponding to the consequent, is "plugged" into the node corresponding to the *Jo is rude* worlds.

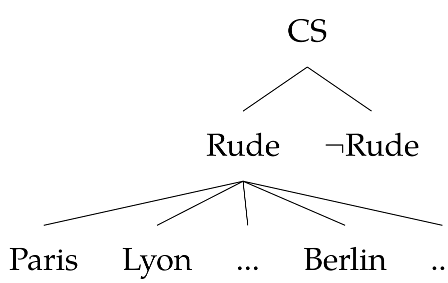


Figure 2.27: An intuitive Qtree for (114a)= *If Jo is rude, she grew up in Paris*.

We already have the tools to cash out this intuition in a compositional way. Recall that in Section 2.2.3, we discussed how a subtree rooted in N in a given Qtree, could be interpreted as the nodewise intersection between the entire Qtree and N, following (66). The definition of this operation is repeated below.

(66) *Nodewise intersection.* Let $T = (\mathcal{N}, \mathcal{E}, R)$ be a Qtree. Let p be a proposition. The nodewise intersection between T and p, noted $T \cap p$, is defined iff $R \cap p \neq \emptyset$ and, if so, is the Qtree $T' = (\mathcal{N}', \mathcal{E}', R')$ s.t.:

- $\mathcal{N}' = \{N \cap p \mid N \in \mathcal{N} \wedge N \cap p \neq \emptyset\}$
- $\mathcal{E}' = \{\{N_1 \cap p, N_2 \cap p\} \mid \{N_1, N_2\} \in \mathcal{E} \wedge (N_1 \cap p) \neq (N_2 \cap p) \wedge N_1 \cap p \neq \emptyset \wedge N_2 \cap p \neq \emptyset\}$
- $R' = R \cap p$

Nodewise intersection, seen as a form of contextual restriction, in fact allows to "plug" specific Qtrees into the node(s) of another Qtree–producing an output that is still a Qtree. Let us see the mechanics of this operation through an example.

(115) and (116) define conditional Qtrees as Qtrees evoked by the antecedent of the conditional, but whose verifying nodes get replaced by their intersection with a Qtree evoked by the consequent (*modulo* reduction). This process is assumed to filter out the outputs that do not qualify at Qtrees. 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.

(115) *Qtrees for conditional LFs.* A Qtree T for $X \to Y$ is obtained from a Qtree T_X for X and a Qtree T_Y for Y by:

- replacing each node N of T_X that is in $\mathcal{N}^+(T_X)$ with $N \cap T_Y$ (see (116));
- returning the result only if it is a Qtree.

In other words, $Qtrees(X \to Y) = \{T_X \cup \bigcup_{N \in \mathcal{N}^+(T_X)}(N \cap T_Y) | (T_X, T_Y) \in Qtrees(X) \times Qtrees(Y) \wedge T_X \cup \bigcup_{N \in \mathcal{N}^+(T_X)}(N \cap T_Y)$verifies (52)$\}$, and $\mathcal{N}^+(T_X \to T_Y) = \{N \cap N' | (N, N') \in \mathcal{N}^+(T_X) \times \mathcal{N}^+(T_Y) \wedge N \cap N' \neq \emptyset\}$.

(116) *Node-Qtree intersection.* If N is a node (set of worlds) and T a Qtree, $N \cap T_Y$ is defined as T_Y, where each node gets intersected with N and empty nodes as well as trivial ("only child") links get removed (in line with (**??**)); and where T_Y's verifying nodes are preserved.

Influential work in psychology (Wason, 1968), showed that, when asked to verify the truth of a conditional statement, participants tend to massively overlook the eventualities falsifying the antecedent. Building on this finding, and insights from the recent linguistic literature (Aloni, 2022), we assume conditional LFs preferentially evoke questions pertaining to their consequent, *in the domain(s) of the CS where the antecedent holds*. The antecedent of a conditional LF therefore plays the role of a question "restrictor", rather than a question "generator".

The Node-Qtree intersection operation is schematized in (2.28).

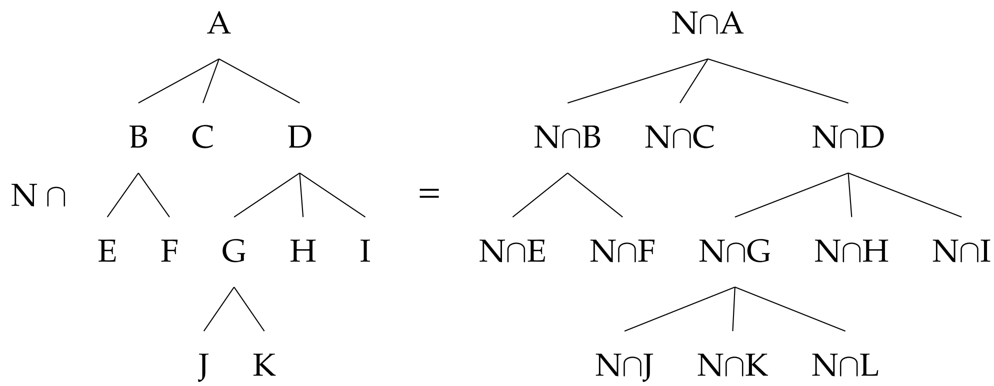


Figure 2.28: Node-Qtree intersection.

There are two additional things to note about this operation. First, the verifying nodes of a conditional Qtree are inherited from its input *consequent* Qtree; meaning, verifying nodes contributed by the antecedent Qtree are *disregarded*. This is in line with the idea that, when checking the truth of natural language conditionals, speaker tend to overlook the possible the falsity of the antecedent. The whole operation is schematized in Figure 2.29.

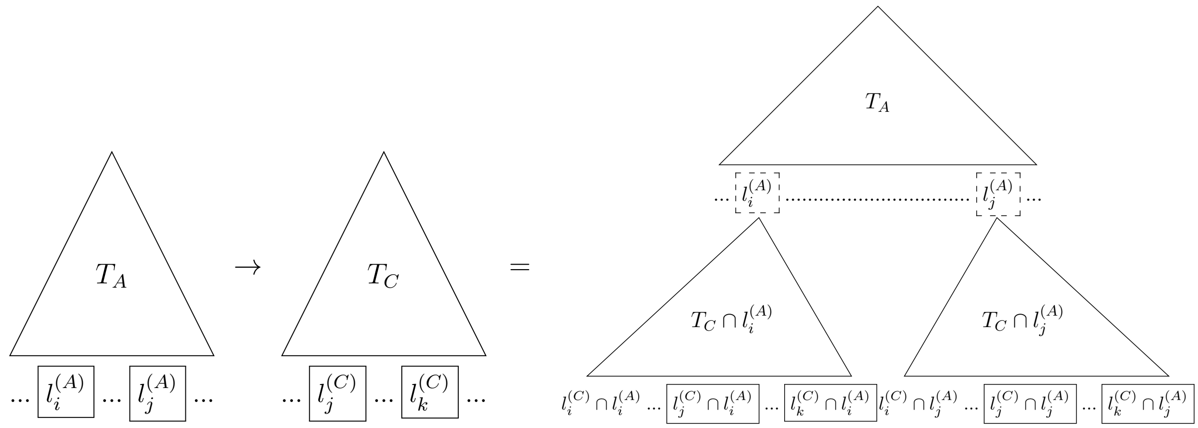


Figure 2.29: Schema of the derivation of a conditional Qtree. Nodes in dashed boxes refer to the nodes that were verifying in the input antecedent Qtree, but are no longer verifying in the output conditional Qtree. Nodes in $\boxed{\text{solid boxes}}$ refer to the nodes that were verifying in the input consequent Qtree, and are thus still verifying in the output conditional Qtree.

Second, the Node-Qtree intersection operation ($N \cap T$), which is part of the conditional Qtree formation process, is "vacuous" iff N entails a specific leaf in T. We call the operation $N \cap T$ vacuous if it outputs N; the status of N as verifying still depends on T's verifying nodes. This is exemplified in Figure (2.30) assuming the node N intersecting the Qtree entails (i.e. is a subset of) the leaf labeled L in T. What happens is the following. The definition of a Qtree (see (52)) states that each intermediate node is partitioned by the set of its children. A corollary of this definition, is that, given a leaf L, all the nodes present on the path from L to the root will be supersets of L, while all the other nodes will have no overlap with L. So, if $N \subseteq L$, N will be a subset of all the nodes in L's path to the root, and have no overlap with the other nodes, as well. Performing $N \cap T$ will thus initially yield a tree with same structure as the input Qtree T, but with nodes equal to N along the path between the root and L's original position, and empty nodes in all other positions. The Qtree reduction process devised in (**??**) then removes all these empty nodes, and collapses the path made of N-nodes into one single node, namely, N. The whole operation therefore returns the input node N. Because the status of being a verifying node percolates when reduction takes place, as per (116), the output N will be verifying iff L was in T.

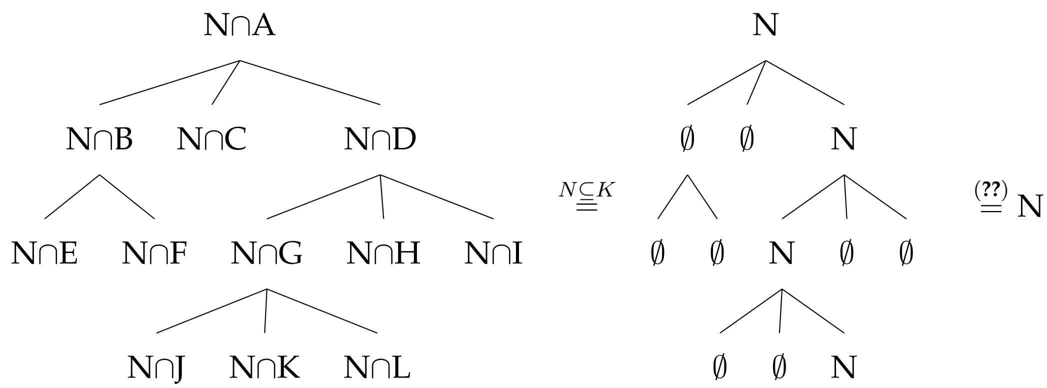
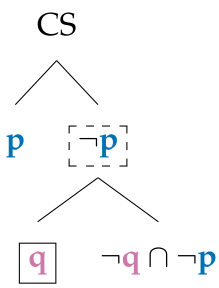


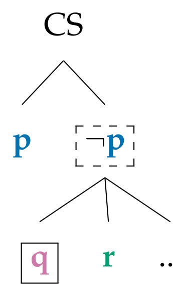
Figure 2.30: Vacuous Node-Qtree intersection if N entails a leaf of T, e.g. K.

The whole conditional Qtree formation process will then be vacuous if each verifying leaf in the antecedent Qtree entails a specific leaf of the consequent Qtree. Moreover, if each verifying leaf in the antecedent Qtree entails a specific *non-verifying* leaf of the consequent Qtree, the output Qtree will be structurally identical to the antecedent Qtree but, will be left with *no* verifying node. Such a tree will be deemed ill-formed as per principle (75).

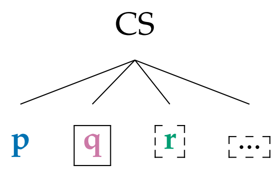
We are now equipped to build conditional Qtrees from the sentences S_p=*Ido is at SuB*, $\neg S_p$=*Ido is not at SuB*, S_q=*Ido is in Cambridge*, and $\neg S_q$=*Ido is not in Cambridge*, whose Qtrees where computed in the previous Sections. This is done for $\neg S_p \rightarrow S_q$ in Figure 2.31, using Qtrees for $\neg S_p$ from Figure **??** and Qtrees for S_q from Figure **??**. Figure 2.32, does the same for $\neg S_q \rightarrow S_p$, just swapping the roles of p and q. It is worth noting that the Qtrees in Figure (2.31c) and (2.32c) are structurally identical to the antecedent Qtree used to build them. Such Qtrees are thus examples of a vacuous application of the Node-Qtree intersection operation. Their verifying nodes are however different from those of their antecedent Qtree, since, by definition, they are inherited from their consequent Qtree.



(a) Figure (**??**) → Figure (**??**)

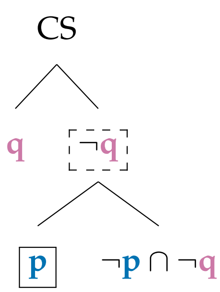


(b) Figure (**??**) → Figure (**??**)

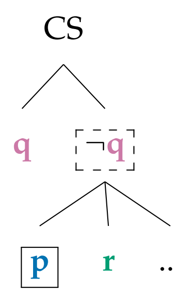


(c) Figure (**??**) → Figure (**??**)/(**??**)[21]

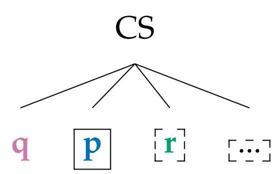
Figure 2.31: Qtrees for $\neg S_{\mathbf{p}} \rightarrow S_{\mathbf{q}}$ = *If Ido is not at SuB then he is in Cambridge*. Nodes in dashed boxes refer to the nodes that were verifying in the input antecedent Qtree, but are no longer verifying in the output conditional Qtree. Nodes in $\boxed{\text{solid boxes}}$ refer to the nodes that were verifying in the input consequent Qtree, and are thus still verifying in the output conditional Qtree.



(a) Figure (**??**) → Figure (**??**)



(b) Figure (**??**) → Figure (**??**)



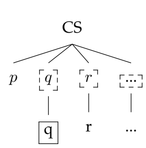
(c) Figure (**??**) → Figure (**??**)/(**??**)

Figure 2.32: Qtrees for $\neg S_{\mathbf{q}} \rightarrow S_{\mathbf{p}}$ = *If Ido is not in Cambridge then he is at SuB*; obtained *mutatis mutandis* from Figure 2.31

At that point, we can already observe that Qtrees built from $\neg S_p \rightarrow S_q$, do not flag the p-node as verifying, since this corresponds to falsifying the antecedent of the conditional, a strategy that is typically overlooked. This feature of the model will be crucial in deriving the felicity of (**??**): because p is not treated as verifying in the Qtrees in Figure 2.31, it will be possible to disjoin them with a Qtree for S_p, without producing redundant Qtrees as output. To clarify this intuition, we proceed to defining disjunction over Qtrees.

[21]This Qtree is derived *via* intersection and reduction as defined in (116). The Qtree derived *before* reduction is given in (i). Reduction on this Qtree 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)



2.5 Conclusion

Unlike inquisitive semantics (Mascarenhas, 2008; Ciardelli, 2009; Groenendijk and Roelofsen, 2009; Ciardelli et al., 2018), which proposes an *unified* view of questions and assertions at the semantic level, what we propose here is a form of inquisitive *pragmatics*: sentences are still assigned "standard" extensional/intensional meaning, but also have an inquisitive contribution at the pragmatic level. In fact, the current machinery may be closer in spirit to Dynamic Semantics (Heim, 1983; Hei, 1983), where different operators give rise to different incremental updates of the Context Set. Under our view, different operators will give rise to different *parses* of the Context Set, at the inquisitive level. This will eventually allow to capture a contrast between (46) and (45).

Bibliography

(1983). On the projection problem for presuppositions. In Barlow, M., Flickinger, D. P., and Wescoat, M. T., editors, *Proceedings of the Second West Coast Conference on Formal Linguistics*, pages 114–126, Stanford, CA. Stanford University Department of Lingustics.

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

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

Chierchia, G., Fox, D., and Spector, B. (2009). Hurford's constraint and the theory of scalar implicatures: Evidence for embedded implicatures. In Egré, P., , and Magri, G., editors, *Presuppositions and implicatures*, pages 47–62, Cambridge, MA. MIT Press.

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

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

Cosmides, L. (1989). The logic of social exchange: Has natural selection shaped how humans reason? studies with the wason selection task. *Cognition*, 31(3):187–276.

Fox, D. (2007). *Free Choice and the Theory of Scalar Implicatures*, page 71–120. Palgrave Macmillan UK.

Fox, D. (2018). Partition by exhaustification: comments on dayal 1996. *ZAS Papers in Linguistics*, 60:403–434.

Fox, D. and Katzir, R. (2011). On the characterization of alternatives. *Natural Language Semantics*, 19(1):87–107.

Gazdar, G. (1979). Pragmatics, Implicature, Presuposition and Logical Form. *Critica*, 12(35):113–122.

Ginzburg, J. (1996). Dynamics and the semantics of dialogue. In Seligman, J. and Westerstahl, D., editors, *Language, Logic, and Computation*, volume 1, Stanford, CA: CSLI. University of Chicago Press.

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

Grice, H. P. (1989). *Studies in the Way of Words*. Harvard University Press, Cambridge, MA.

Groenendijk, J. (1999). The logic of interrogation: Classical version. *Semantics and Linguistic Theory*, 9:109.

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

Groenendijk, J. and Stokhof, M. (1984). *Studies in the semantics of questions and the pragmatics of answers*. PhD thesis, University of Amsterdam. (Unpublished doctoral dissertation).

Hamblin, C. L. (1958). Questions. *Australasian Journal of Philosophy*, 36(3):159–168.

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

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

Heim, I. (1982). *The Semantics of Definite and Indefinite Noun Phrases*. PhD thesis, UMass Amherst.

Heim, I. (1983). *File Change Semantics and the Familiarity Theory of Definiteness*, page 164–189. DE GRUYTER.

Hulstijn, J. (1997). Structured information states: Raising and resolving issues.

Hurford, J. R. (1974). Exclusive or Inclusive Disjunction. *Foundations of Language*, 11(3):409–411.

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.

Hénot-Mortier, A. (2024a). "One tool to rule them all"? An integrated model of the QuD for Hurford sentences. In *Proceedings of the 29th Sinn und Bedeutung*. Draft (`https://adelemortier.github.io/files/SuB_2024_TMT_paper.pdf`), to appear.

Hénot-Mortier, A. (2024b). Redundancy under Discussion. In *Proceedings of the 29th Sinn un Bedeutung*. Draft (`https://adelemortier.github.io/files/SuB_2024_main_session_paper.pdf`), to appear.

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

Jäger, G. (1996). Only updates: On the dynamics of the focus particle only. In *Proceedings of the 10th Amsterdam Colloquium*.

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

Karttunen, L. (1977). Syntax and semantics of questions. *Linguistics and Philosophy*, 1(1):3–44.

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

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

Katzir, R. and Singh, R. (2015). Economy of structure and information: Oddness, questions, and answers. *Proceedings of Sinn und Bedeutung*, 19:322–339.

Kaufmann, M. (2016). Free Choice is a Form of Dependence. *Natural Language Semantics*, 24(3):247–290.

Kroch, A. (1972). Lexical and inferred meanings for some time adverbs. *Quarterly Progress Report of the Research Laboratory of Electronics 104*.

Lewis, D. (1988). Relevant Implication. *Theoria*, 54(3):161–174.

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

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

Mascarenhas, S. (2008). Inquisitive semantics and logic. Master's thesis, University of Amsterdam. Manuscript, University of Amsterdam.

Mayr, C. and Romoli, J. (2016). 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*. PhD thesis, Massachusetts Institute of Technology.

Onea, E. (2016). *Potential Questions at the Semantics-Pragmatics Interface*. BRILL.

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

Roberts, C. (1996). Information structure in discourse: Towards an integrated formal theory of pragmatics. In Yoon, J. H. and Kathol, A., editors, *Ohio State University Working Papers in Linguistics*, volume 49. Ohio State University.

Roberts, C. (2012). Information structure in discourse: Towards an integrated formal theory of pragmatics. *Semantics & Pragmatics*, 5:1–69.

Rooth, M. (1992). A theory of focus interpretation. *Natural Language Semantics*, 1(1):75–116.

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

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

Singh, R. (2008a). *Modularity and locality in interpretation*. PhD thesis, MIT.

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

Spector, B., Fox, D., and Chierchia, G. (2009). Hurford's Constraint and the Theory of Scalar Implicatures: Evidence for embedded implicatures. In Égré, P. and Magri, G., editors, *Presuppositions and Implicatures: Proceedings of the MIT-Paris Workshop*.

Stalnaker, R. (1974). Pragmatic Presuppositions. In Munitz, M. K. and Unger, P. K., editors, *Semantics and Philosophy*, page 197–213. New York University Press, New York.

Stalnaker, R. (1978). Assertion. *Syntax and Semantics*, 9:315–332.

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

Tomioka, S. (2021). Scalar Implicature, Hurford's Constraint, Contrastiveness and How They All Come Together. *Frontiers in Communication*, 5.

van Kuppevelt, J. (1995a). Discourse structure, topicality and questioning. *Journal of Linguistics*, 31(1):109–147.

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

von Fintel, K. and Heim, I. (2023). *Intensional Semantics*.

Wason, P. C. (1968). Reasoning about a rule. *Quarterly Journal of Experimental Psychology*, 20(3):273–281.

Zhang, Y. (2022). *New Perspectives on Inquisitive Semantics*. PhD thesis, University of Maryland.

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