Exh and only don't really compete – they just answer different questions[1]

Adèle Hénot-Mortier (MIT)

May 20, 2025

35th meeting of Semantics and Linguistic Theory

[1]**Many thanks to Amir Anvari, Athulya Aravind, Gennaro Chierchia, Danny Fox, Martin Hackl, Nina Haslinger, Manfred Krifka, Viola Schmitt and Raven Zhang for their advising, feedback, or input. Thanks also to the audiences of the BerlinBrnoVienna Workshop 2024, SuB29, and AC2024 for relevant feedback on related projects. All mistakes are my own.**

Two odd constructions

- Disjunctions featuring **entailing disjuncts** are typically odd.

(1) Hurford Disjunction[1] (**Paris** $\models$ **France**)
Jo grew up in **France or Paris**.

- And so are sequences of conditionals with **entailing antecedents** and **incompatible consequents**.

(2) Sobel Sequence[2] (**friendly** $\wedge$ **rude** $\models \bot$)
If Jo grew up in **France** she is **friendly**,
but **if** she grew up in **Paris** she is **rude**.

[1] Hurford, 1974, i.a.

[2] Sobel, 1970; Lewis, 1973; von Fintel, 2001, i.a.

Two odd constructions

- Disjunctions featuring **entailing disjuncts** are typically odd.

(1) Hurford Disjunction[1] (**Paris** $\models$ **France**)
Jo grew up in **France** **or** **Paris**.

- And so are sequences of conditionals with **entailing antecedents** and **incompatible consequents**.

(2) Sobel Sequence[2] (**friendly** $\wedge$ **rude** $\models \bot$)
If Jo grew up in **France** she is **friendly**,
but **if** she grew up in **Paris** she is **rude**.

[1] Hurford, 1974, i.a.

[2] Sobel, 1970; Lewis, 1973; von Fintel, 2001, i.a.

Assymetries in scalar Hurford Disjunctions

- Hurford Disjunctions with scalar disjuncts are fine if **the weaker disjunct precedes the stronger one** (3a).

(3) "Scalar" Hurford Disjunctions[3]

a. Jo did **some** **or** **all** of the problems.

b. ?? Jo did **all** **or** **some** of the problems.

[3] Gazdar, 1979; Chierchia et al., 2009; Hénot-Mortier, 2023, i.a.

Assymetries in scalar Sobel Sequences

- Likewise, Sobel Sequences are fine if **the weaker antecedent precedes the stronger one**.

(4) "Scalar" Sobel Sequences[4]

a. **If** Jo solved **some** of the problems she'll **fail**, but **if** she solved **all** she'll **pass**.

b. ?[5] **If** Jo solved **all** of the problems she'll **pass**, but **if** she solved **some** she'll **fail**.

[4] Sobel, 1970; Lewis, 1973; von Fintel, 2001; Singh, 2008a; Ippolito, 2019, i.a.

[5] Contrast gets crisper with **or** vs. **and**.

Challenge 1: capturing the asymmetry

- In both Hurford and Sobel cases, infelicity seems to stem from the **entailment between contrasted stronger and weaker items**, which:
  - creates "redundancy" in Hurford cases;
  - creates a contradiction in Sobel cases.
- Dominant view: these issues can be fixed by **exh**,[6] a "covert **only**" which locally strengthens the weaker item to contradict the stronger one. E.g., **exh(some)** = **some but not all**.

How to prevent exh from rescuing the "bad" cases?

[6] Singh, 2008a; Chierchia et al., 2009, i.a.

Challenge 1: capturing the asymmetry

- In both Hurford and Sobel cases, infelicity seems to stem from the **entailment between contrasted stronger and weaker items**, which:
  - creates "redundancy" in Hurford cases;
  - creates a contradiction in Sobel cases.
- Dominant view: these issues can be fixed by **exh**,[6] a "covert **only**" which locally strengthens the weaker item to contradict the stronger one. E.g., **exh(some)** = **some but not all**.

How to prevent exh from rescuing the "bad" cases?

[6] Singh, 2008a; Chierchia et al., 2009, i.a.

Challenge 1: capturing the asymmetry

- In both Hurford and Sobel cases, infelicity seems to stem from the **entailment between contrasted stronger and weaker items**, which:
  - creates "redundancy" in Hurford cases;
  - creates a contradiction in Sobel cases.
- Dominant view: these issues can be fixed by **exh**,[6] a "covert **only**" which locally strengthens the weaker item to contradict the stronger one. E.g., **exh(some)** = **some but not all**.

How to prevent exh from rescuing the "bad" cases?

[6] Singh, 2008a; Chierchia et al., 2009, i.a.

Challenge 1: capturing the asymmetry

- In both Hurford and Sobel cases, infelicity seems to stem from the **entailment between contrasted stronger and weaker items**, which:
  - creates "redundancy" in Hurford cases;
  - creates a contradiction in Sobel cases.
- Dominant view: these issues can be fixed by **exh**,[6] a "covert **only**" which locally strengthens the **weaker** item to contradict the **stronger** one. E.g., **exh(some)** = **some but not all**.

How to prevent exh from rescuing the "bad" cases?

[6] Singh, 2008a; Chierchia et al., 2009, i.a.

Challenge 1: capturing the asymmetry

- In both Hurford and Sobel cases, infelicity seems to stem from the **entailment between contrasted stronger and weaker items**, which:
  - creates "redundancy" in Hurford cases;
  - creates a contradiction in Sobel cases.
- Dominant view: these issues can be fixed by **exh**,[6] a "covert **only**" which locally strengthens the **weaker** item to contradict the **stronger** one. E.g., **exh(some)** = **some but not all**.

How to prevent exh from rescuing the "bad" cases?

[6] Singh, 2008a; Chierchia et al., 2009, i.a.

Sketching a solution to Challenge 1

How to prevent exh from rescuing the "bad" cases?

- The distribution of **exh** should be somehow **constrained by preceding material**.[7]
- One view is that **exh** is not licensed if incrementally vacuous, which happens in (5b).
- This makes **exh** an "**asymmetric rescuer**".

(5) a. Jo did ✓**exh(some) or all** of the problems.

b. ?? Jo did **all or** ✗**exh(some)** of the problems.

- Now what about overt **only**?

[7] Singh, 2008b; Fox and Spector, 2018; Ippolito, 2019; Tomioka, 2021; Hénot-Mortier, 2023, i.a.

Only is a "symmetric rescuer"!

- Hurford Disjunctions with **only** are fine **regardless of the ordering of the disjuncts**.[8]

(6) a. ?[9] Jo did **only** **some** or **all** of the problems.

b. Jo did **all** or **only** **some** of the problems.

- Same holds for Sobel Sequences.[10]

(7) a. If Jo solved **only** **some** of the problems she'll **fail**, but if she solved **all** of the problems she'll **pass**.

b. If Jo solved **all** of the problems she'll **pass**, but if she solved **only** **some** of the problems she'll **fail**.

[8] Singh, 2008a; Fox and Spector, 2018; Ippolito, 2019; Tomioka, 2021; Krifka, 2024.

[9] ? can be attributed to the felicity of (3a), which has same meaning as (6a) collapsing presupposition and assertion, but is strictly simpler to produce.

[10] Singh, 2008a; Ippolito, 2019.

Only is a "symmetric rescuer"!

- Hurford Disjunctions with **only** are fine **regardless of the ordering of the disjuncts**.[8]

(6) a. ?[9] Jo did **only** **some** or **all** of the problems.

b. Jo did **all** or **only** **some** of the problems.

- Same holds for Sobel Sequences.[10]

(7) a. If Jo solved **only** **some** of the problems she'll **fail**,
but if she solved **all** of the problems she'll **pass**.

b. If Jo solved **all** of the problems she'll **pass**,
but if she solved **only** **some** of the problems she'll **fail**.

---

[8] Singh, 2008a; Fox and Spector, 2018; Ippolito, 2019; Tomioka, 2021; Krifka, 2024.

[9] ? can be attributed to the felicity of (3a), which has same meaning as (6a) collapsing presupposition and assertion, but is strictly simpler to produce.

[10] Singh, 2008a; Ippolito, 2019.

Challenge 2: capturing a contrast between exh and only

- Let's assume we solved Challenge 1, i.e. we have a constraint on **exh** explaining why **exh(some)** = **some but not all** is not allowed in the 2nd disjunct/antecedent of (3b)/(4b).

Why would overt only escape this constraint, and rescue both orderings?

Challenge 2: capturing a contrast between exh and only

- Let's assume we solved Challenge 1, i.e. we have a constraint on **exh** explaining why **exh(some)** = **some but not all** is not allowed in the 2nd disjunct/antecedent of (3b)/(4b).

Why would overt only escape this constraint, and rescue both orderings?

Goal for Today

- We will take for granted that (c)overt exhaustification rescues scalar Hurford and Sobel cases.
- We will focus on explaining the contrast between **exh** and **only** in the 2nd disjunct of scalar Hurford Disjunctions (**SHD**s):

(3b) Jo did **all** **or** ✗**exh**/✓**only** **some** of the problems.

- We will explain it using three core ingredients:
  1. covert and overt exhaustification are compatible with **distinct Questions under Discussion** (**QuD**[11]).
  2. QuDs are raised **incrementally and implicitely**,[12]...
  3. and must be **felicitously addressed by following material**.[13]

[11] Roberts, 1996.

[12] Ippolito, 2019; Zhang, 2022; Hénot-Mortier, to appear.

[13] Heim, 2015; Aravind et al., 2022; Doron and Wehbe, 2024.

Goal for Today

- We will take for granted that (c)overt exhaustification rescues scalar Hurford and Sobel cases.
- We will focus on explaining the contrast between **exh** and **only** in the 2nd disjunct of scalar Hurford Disjunctions (**SHD**s):

(3b) Jo did **all** **or** ✗**exh**/✓**only** **some** of the problems.

- We will explain it using three core ingredients:
  1. covert and overt exhaustification are compatible with **distinct Questions under Discussion** (**QuD**[11]).
  2. QuDs are raised **incrementally and implicitely**,[12]...
  3. and must be **felicitously addressed by following material**.[13]

[11] Roberts, 1996.

[12] Ippolito, 2019; Zhang, 2022; Hénot-Mortier, to appear.

[13] Heim, 2015; Aravind et al., 2022; Doron and Wehbe, 2024.

Goal for Today

- We will take for granted that (c)overt exhaustification rescues scalar Hurford and Sobel cases.
- We will focus on explaining the contrast between **exh** and **only** in the 2nd disjunct of scalar Hurford Disjunctions (**SHD**s):

(3b) Jo did **all** **or** ✗**exh**/✓**only** **some** of the problems.

- We will explain it using three core ingredients:
  1. covert and overt exhaustification are compatible with **distinct Questions under Discussion** (**QuD**[11]).
  2. QuDs are raised **incrementally and implicitely**,[12]...
  3. and must be **felicitously addressed by following material**.[13]

[11] Roberts, 1996.

[12] Ippolito, 2019; Zhang, 2022; Hénot-Mortier, to appear.

[13] Heim, 2015; Aravind et al., 2022; Doron and Wehbe, 2024.

Goal for Today

- We will take for granted that (c)overt exhaustification rescues scalar Hurford and Sobel cases.
- We will focus on explaining the contrast between **exh** and **only** in the 2nd disjunct of scalar Hurford Disjunctions (**SHD**s):

(3b) Jo did **all** **or** ✗**exh**/✓**only** **some** of the problems.

- We will explain it using three core ingredients:
  1. covert and overt exhaustification are compatible with **distinct Questions under Discussion** (**QuD**[11]).
  2. QuDs are raised **incrementally and implicitely**,[12]...
  3. and must be **felicitously addressed by following material**.[13]

[11] Roberts, 1996.

[12] Ippolito, 2019; Zhang, 2022; Hénot-Mortier, to appear.

[13] Heim, 2015; Aravind et al., 2022; Doron and Wehbe, 2024.

Goal for Today

- We will take for granted that (c)overt exhaustification rescues scalar Hurford and Sobel cases.
- We will focus on explaining the contrast between **exh** and **only** in the 2nd disjunct of scalar Hurford Disjunctions (**SHD**s):

(3b) Jo did **all** **or** ✗**exh**/✓**only** **some** of the problems.

- We will explain it using three core ingredients:
  1. covert and overt exhaustification are compatible with **distinct Questions under Discussion** (**QuD**[11]).
  2. QuDs are raised **incrementally and implicitely**,[12]...
  3. and must be **felicitously addressed by following material**.[13]

[11] Roberts, 1996.

[12] Ippolito, 2019; Zhang, 2022; Hénot-Mortier, to appear.

[13] Heim, 2015; Aravind et al., 2022; Doron and Wehbe, 2024.

Goal for Today

- We will take for granted that (c)overt exhaustification rescues scalar Hurford and Sobel cases.
- We will focus on explaining the contrast between **exh** and **only** in the 2nd disjunct of scalar Hurford Disjunctions (**SHD**s):

(3b) Jo did **all** **or** ✗**exh**/✓**only** **some** of the problems.

- We will explain it using three core ingredients:
  1. covert and overt exhaustification are compatible with **distinct Questions under Discussion** (**QuD**[11]).
  2. QuDs are raised **incrementally and implicitely**,[12]...
  3. and must be **felicitously addressed by following material**.[13]

[11] Roberts, 1996.

[12] Ippolito, 2019; Zhang, 2022; Hénot-Mortier, to appear.

[13] Heim, 2015; Aravind et al., 2022; Doron and Wehbe, 2024.

Previous accounts

TL;DR: capturing the data at stake under standard assumptions and limited stipulations is challenging.

Singh (2008a): assumptions

- Both **exh** and **only** assert their prejacent and the negation of non-weaker relevant alternatives, i.e. have **same at-issue contributions**.
- **exh** is presuppositionless; while **only** presupposes that an alternative different from its prejacent is made noteworthy; i.e. **only has a strictly stronger presupposition**.

(8) $\text{exh}(\textbf{some}) = \begin{bmatrix} P & : & \\ A & : & \textbf{some but not all} \end{bmatrix}$

(9) $\text{only}(\textbf{some}) = \begin{bmatrix} P & : & \exists X \in ALT(\textbf{some}) \text{ salient and } \neq \textbf{some} \\ A & : & \textbf{some but not all} \end{bmatrix}$

Singh (2008a): assumptions

- Both **exh** and **only** assert their prejacent and the negation of non-weaker relevant alternatives, i.e. have **same at-issue contributions**.
- **exh** is presuppositionless; while **only** presupposes that an alternative different from its prejacent is made noteworthy; i.e. **only has a strictly stronger presupposition**.

(8) $\text{exh}(\textbf{some}) = \begin{bmatrix} P & : & \\ A & : & \textbf{some but not all} \end{bmatrix}$

(9) $\text{only}(\textbf{some}) = \begin{bmatrix} P & : & \exists X \in ALT(\textbf{some}) \text{ salient and } \neq \textbf{some} \\ A & : & \textbf{some but not all} \end{bmatrix}$

Singh (2008a): assumptions

- Both **exh** and **only** assert their prejacent and the negation of non-weaker relevant alternatives, i.e. have **same at-issue contributions**.
- **exh** is presuppositionless; while **only** presupposes that an alternative different from its prejacent is made noteworthy; i.e. **only has a strictly stronger presupposition**.

(8) $\text{exh}(\textbf{some}) = \begin{bmatrix} P & : & \\ A & : & \textbf{some but not all} \end{bmatrix}$

(9) $\text{only}(\textbf{some}) = \begin{bmatrix} P & : & \exists X \in ALT(\textbf{some}) \text{ salient and } \neq \textbf{some} \\ A & : & \textbf{some but not all} \end{bmatrix}$

Singh (2008a): proposal

- **Exh** and **only** compete under Local Maximize Presupposition (**LMP!**).
- **Only**'s presupposition is verified whenever its prejacent is preceded by one of its alternatives.
- In such cases, **only** should be preferred over **exh** as per LMP!

(10) Jo did **all** **or** $\begin{bmatrix} {}^{\#}\textbf{some} \\ {}^{LMP!\text{✗}}\textit{exh}(\textbf{some}) \\ {}^{LMP!\text{✓}}\textit{only}(\textbf{some}) \end{bmatrix}$ of the problems.

⚠ Shortcomings: **only**'s entry is non-standard, and this LMP! story cannot fully replace another constraint on **exh** needed for other cases.[14]

[14] Fox and Spector, 2018.

Singh (2008a): proposal

- **Exh** and **only** compete under Local Maximize Presupposition (**LMP!**).
- **Only**'s presupposition is verified whenever its prejacent is preceded by one of its alternatives.
- In such cases, **only** should be preferred over **exh** as per LMP!

(10) Jo did **all** **or** $\begin{bmatrix} {}^{\#}\textbf{some} \\ {}^{LMP!\text{✗}}\textit{exh}(\textbf{some}) \\ {}^{LMP!\text{✓}}\textit{only}(\textbf{some}) \end{bmatrix}$ of the problems.

⚠ Shortcomings: **only**'s entry is non-standard, and this LMP! story cannot fully replace another constraint on **exh** needed for other cases.[14]

[14] Fox and Spector, 2018.

Singh (2008a): proposal

- **Exh** and **only** compete under Local Maximize Presupposition (**LMP!**).
- **Only**'s presupposition is verified whenever its prejacent is preceded by one of its alternatives.
- In such cases, **only** should be preferred over **exh** as per LMP!

(10) Jo did **all** **or** $\begin{bmatrix} {}^{\#}\textbf{some} \\ {}^{LMP!\text{✗}}\mathit{exh}(\textbf{some}) \\ {}^{LMP!\text{✓}}\mathit{only}(\textbf{some}) \end{bmatrix}$ of the problems.

⚠ Shortcomings: **only**'s entry is non-standard, and this LMP! story cannot fully replace another constraint on **exh** needed for other cases.[14]

[14] Fox and Spector, 2018.

Singh (2008a): proposal

- **Exh** and **only** compete under Local Maximize Presupposition (**LMP!**).
- **Only**'s presupposition is verified whenever its prejacent is preceded by one of its alternatives.
- In such cases, **only** should be preferred over **exh** as per LMP!

(10) Jo did **all** **or** $\begin{bmatrix} {}^{\#}\textbf{some} \\ {}^{LMP!\text{✗}}\textit{exh}(\textbf{some}) \\ {}^{LMP!\text{✓}}\textit{only}(\textbf{some}) \end{bmatrix}$ of the problems.

⚠ Shortcomings: **only**'s entry is non-standard, and this LMP! story cannot fully replace another constraint on **exh** needed for other cases.[14]

[14] Fox and Spector, 2018.

Ippolito (2019): assumptions

- F-marked sentences evoke Structured Sets of Alternatives (**SSA**s), trees whose nodes are alternatives to the sentence and whose branches are induced by $\models$.
- Salient alternatives are defined as mothers and siblings of the asserted alternative.

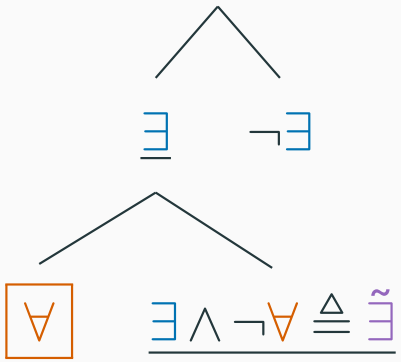


Figure 1: SSA evoked by **all**.

Ippolito (2019): assumptions

- F-marked sentences evoke Structured Sets of Alternatives (**SSA**s), trees whose nodes are alternatives to the sentence and whose branches are induced by $\models$.
- Salient alternatives are defined as mothers and siblings of the asserted alternative.

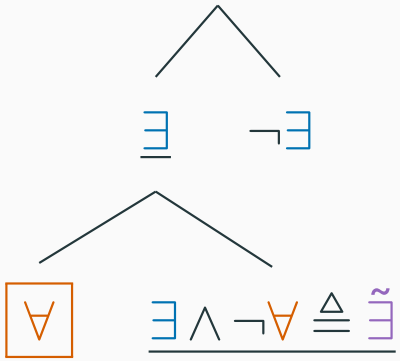


Figure 1: SSA evoked by **all**.

Ippolito (2019): assumptions

- F-marked sentences evoke Structured Sets of Alternatives (**SSA**s), trees whose nodes are alternatives to the sentence and whose branches are induced by $\models$.
- Salient alternatives are defined as mothers and siblings of the asserted alternative.

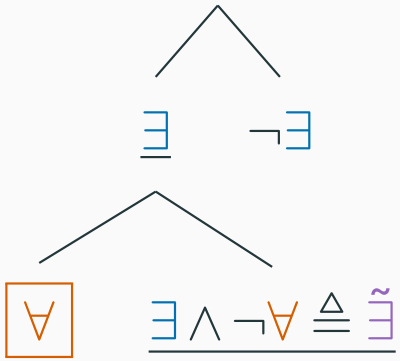


Figure 1: SSA evoked by **all**.

Ippolito (2019): proposal

- **Economy constraint on covert operations**: **exh** cannot be inserted if it results in a meaning equivalent to an alternative made salient by a preceding SSA.
  - In SHDs, **all** in the 1st disjunct makes **some but not all** salient.
  - Because **exh**(**some**) = **some but not all**, **exh** cannot be inserted in the 2nd disjunct of a SHD, as per Economy.
  - **Only** can, because it's overt.

⚠ Shortcomings:

  - the structural definition of salience is stipulated;
  - why should the above Economy condition care about covertness? **some but not all**/**only some** appear most costly than **exh**(**some**)!

Ippolito (2019): proposal

- **Economy constraint on covert operations**: **exh** cannot be inserted if it results in a meaning equivalent to an alternative made salient by a preceding SSA.
  - In SHDs, **all** in the 1st disjunct makes **some but not all** salient.
  - Because **exh**(**some**) = **some but not all**, **exh** cannot be inserted in the 2nd disjunct of a SHD, as per Economy.
  - **Only** can, because it's overt.

⚠ Shortcomings:

  - the structural definition of salience is stipulated;
  - why should the above Economy condition care about covertness? **some but not all**/**only some** appear most costly than **exh**(**some**)!

Ippolito (2019): proposal

- **Economy constraint on covert operations**: **exh** cannot be inserted if it results in a meaning equivalent to an alternative made salient by a preceding SSA.
  - In SHDs, **all** in the 1st disjunct makes **some but not all** salient.
  - Because **exh**(**some**) = **some but not all**, **exh** cannot be inserted in the 2nd disjunct of a SHD, as per Economy.
  - **Only** can, because it's overt.

⚠ Shortcomings:

- the structural definition of salience is stipulated;
- why should the above Economy condition care about covertness? **some but not all**/**only some** appear most costly than **exh**(**some**)!

Ippolito (2019): proposal

- **Economy constraint on covert operations**: **exh** cannot be inserted if it results in a meaning equivalent to an alternative made salient by a preceding SSA.
  - In SHDs, **all** in the 1st disjunct makes **some but not all** salient.
  - Because **exh**(**some**) = **some but not all**, **exh** cannot be inserted in the 2nd disjunct of a SHD, as per Economy.
  - **Only** can, because it's overt.

⚠ Shortcomings:

- the structural definition of salience is stipulated;
- why should the above Economy condition care about covertness? **some but not all**/**only some** appear most costly than **exh**(**some**)!

Ippolito (2019): proposal

- **Economy constraint on covert operations**: **exh** cannot be inserted if it results in a meaning equivalent to an alternative made salient by a preceding SSA.
  - In SHDs, **all** in the 1st disjunct makes **some but not all** salient.
  - Because **exh**(**some**) = **some but not all**, **exh** cannot be inserted in the 2nd disjunct of a SHD, as per Economy.
  - **Only** can, because it's overt.

⚠ Shortcomings:

- the structural definition of salience is stipulated;
- why should the above Economy condition care about covertness? **some but not all**/**only some** appear most costly than **exh**(**some**)!

Ippolito (2019): proposal

- **Economy constraint on covert operations**: **exh** cannot be inserted if it results in a meaning equivalent to an alternative made salient by a preceding SSA.
  - In SHDs, **all** in the 1st disjunct makes **some but not all** salient.
  - Because **exh**(**some**) = **some but not all**, **exh** cannot be inserted in the 2nd disjunct of a SHD, as per Economy.
  - **Only** can, because it's overt.

⚠ Shortcomings:

  - the structural definition of salience is stipulated;
  - why should the above Economy condition care about covertness? **some but not all**/**only some** appear most costly than **exh(some)**!

Desiderata and next steps

Building on Ippolito's SSAs, and on Singh's intuition that **exh** and **only** semantically differ, **we propose a story requiring less initial assumptions.**

- We ground the difference between **exh** and **only** in the independently motivated *semantics* of these operators. **(C)overtness will not play a role.**
- In place of LMP!/Economy, we recycle a constraint on presuppositions and the QuD, *deriving* distinct licensing conditions for **exh** and **only**.
- Under this view, **exh** and **only** don't compete – they are just used to answer different implicit QuDs!

Desiderata and next steps

Building on Ippolito's SSAs, and on Singh's intuition that **exh** and **only** semantically differ, **we propose a story requiring less initial assumptions.**

- We ground the difference between **exh** and **only** in the independently motivated *semantics* of these operators. **(C)overtness will not play a role.**
- In place of LMP!/Economy, we recycle a constraint on presuppositions and the QuD, *deriving* distinct licensing conditions for **exh** and **only**.
- Under this view, **exh** and **only** don't compete – they are just used to answer different implicit QuDs!

Desiderata and next steps

Building on Ippolito's SSAs, and on Singh's intuition that **exh** and **only** semantically differ, **we propose a story requiring less initial assumptions.**

- We ground the difference between **exh** and **only** in the independently motivated *semantics* of these operators. **(C)overtness will not play a role.**
- In place of LMP!/Economy, we recycle a constraint on presuppositions and the QuD, *deriving* distinct licensing conditions for **exh** and **only**.
- Under this view, **exh** and **only** don't compete – they are just used to answer different implicit QuDs!

Desiderata and next steps

Building on Ippolito's SSAs, and on Singh's intuition that **exh** and **only** semantically differ, **we propose a story requiring less initial assumptions.**

- We ground the difference between **exh** and **only** in the independently motivated *semantics* of these operators. **(C)overtness will not play a role.**
- In place of LMP!/Economy, we recycle a constraint on presuppositions and the QuD, *deriving* distinct licensing conditions for **exh** and **only**.
- Under this view, **exh** and **only** don't compete – they are just used to answer different implicit QuDs!

Exhaustification and the QuD

TL;DR: exh and only are used to answer different questions, which can be linked to how they divide presupposition and assertion.

Assumptions about (c)overt exhaustifiers

- We assume a covert, optional, *presuppositional* exhaustification operator **pex**.[15]
- We use a standard entry for **only**.

(11) $\text{pex}(\textbf{some}) = \begin{bmatrix} P & : & \textbf{not all} \\ A & : & \textbf{some} \end{bmatrix}$

(12) $\text{only}(\textbf{some}) = \begin{bmatrix} P & : & \textbf{some} \\ A & : & \textbf{not all} \end{bmatrix}$

- Let's see how these operators interact with overt QuDs.

[15] Bassi et al., 2021.

Assumptions about (c)overt exhaustifiers

- We assume a covert, optional, *presuppositional* exhaustification operator **pex**.[15]
- We use a standard entry for **only**.

(11) $\text{pex}(\textbf{some}) = \begin{bmatrix} P & : & \textbf{not all} \\ A & : & \textbf{some} \end{bmatrix}$

(12) $\text{only}(\textbf{some}) = \begin{bmatrix} P & : & \textbf{some} \\ A & : & \textbf{not all} \end{bmatrix}$

- Let's see how these operators interact with overt QuDs.

[15] Bassi et al., 2021.

Assumptions about (c)overt exhaustifiers

- We assume a covert, optional, *presuppositional* exhaustification operator **pex**.[15]
- We use a standard entry for **only**.

(11) $\text{pex}(\textbf{some}) = \begin{bmatrix} P & : & \textbf{not all} \\ A & : & \textbf{some} \end{bmatrix}$

(12) $\text{only}(\textbf{some}) = \begin{bmatrix} P & : & \textbf{some} \\ A & : & \textbf{not all} \end{bmatrix}$

- Let's see how these operators interact with overt QuDs.

[15] Bassi et al., 2021.

Assumptions about (c)overt exhaustifiers

- We assume a covert, optional, *presuppositional* exhaustification operator **pex**.[15]
- We use a standard entry for **only**.

(11) $\text{pex}(\textbf{some}) = \begin{bmatrix} P & : & \textbf{not all} \\ A & : & \textbf{some} \end{bmatrix}$

(12) $\text{only}(\textbf{some}) = \begin{bmatrix} P & : & \textbf{some} \\ A & : & \textbf{not all} \end{bmatrix}$

- Let's see how these operators interact with overt QuDs.

[15] Bassi et al., 2021.

Whether-some questions

- If the QuD is **?some**, answering with a bare **some** is fine, **some but not all** is okay; see (13a-b). Both assert **some**.
- **Only some**, which presupposes **some**, is out; see (13c).

(13) Did Jo solve **some** of the problems, or **none** of them? $\{\exists, \neg\exists\}$

a. – Jo solved **some** of the problems.

b. ? – Jo solved **some but not all** of the problems.

c. # – Jo solved **only some** of the problems.

- In line with the idea that **sentences should not settle overt QuDs via their presupposition**.[16]

[16] Heim, 2015; Aravind et al., 2022; Doron and Wehbe, 2024.

Whether-some questions

- If the QuD is **?some**, answering with a bare **some** is fine, **some but not all** is okay; see (13a-b). Both assert **some**.
- **Only some**, which presupposes **some**, is out; see (13c).

(13) Did Jo solve **some** of the problems, or **none** of them? $\{\exists, \neg\exists\}$

a. – Jo solved **some** of the problems.

b. ? – Jo solved **some but not all** of the problems.

c. # – Jo solved **only some** of the problems.

- In line with the idea that **sentences should not settle overt QuDs via their presupposition**.[16]

[16] Heim, 2015; Aravind et al., 2022; Doron and Wehbe, 2024.

Whether-some questions

- If the QuD is **?some**, answering with a bare **some** is fine, **some but not all** is okay; see (13a-b). Both assert **some**.
- **Only some**, which presupposes **some**, is out; see (13c).

(13) Did Jo solve **some** of the problems, or **none** of them? $\{\exists, \neg\exists\}$

a. – Jo solved **some** of the problems.

b. ? – Jo solved **some but not all** of the problems.

c. # – Jo solved **only some** of the problems.

- In line with the idea that **sentences should not settle overt QuDs via their presupposition**.[16]

[16] Heim, 2015; Aravind et al., 2022; Doron and Wehbe, 2024.

Whether-**all** questions

- If the QuD is **?all**, answering with **all** or **only some** is fine, and **some but not all** is okay; see (14a-c). All options assert either **all**, or **not all**.
- F-marked **SOME**, which can be understood as forcing **pex** and thus presupposes **not all**, is out; see (14d).

(14) Did Jo solve **all** of the problems, or **not all** of them? $\{\forall, \neg\forall\}$

a. – Jo solved **all** of the problems.

b. – Jo solved **only some** of the problems.

c. ? – Jo solved **some but not all** of the problems.

d. ?? – Jo solved **SOME** of the problems.

- Again in line with the idea that sentences should not settle overt QuDs *via* their presupposition!

Whether-**all** questions

- If the QuD is **?all**, answering with **all** or **only some** is fine, and **some but not all** is okay; see (14a-c). All options assert either **all**, or **not all**.
- F-marked **SOME**, which can be understood as forcing **pex** and thus presupposes **not all**, is out; see (14d).

(14) Did Jo solve **all** of the problems, or **not all** of them? $\{\forall, \neg\forall\}$

a. – Jo solved **all** of the problems.

b. – Jo solved **only some** of the problems.

c. ? – Jo solved **some but not all** of the problems.

d. ?? – Jo solved **SOME** of the problems.

- Again in line with the idea that sentences should not settle overt QuDs *via* their presupposition!

Whether-**all** questions

- If the QuD is **?all**, answering with **all** or **only some** is fine, and **some but not all** is okay; see (14a-c). All options assert either **all**, or **not all**.
- F-marked **SOME**, which can be understood as forcing **pex** and thus presupposes **not all**, is out; see (14d).

(14) Did Jo solve **all** of the problems, or **not all** of them? $\{\forall, \neg\forall\}$

a. – Jo solved **all** of the problems.

b. – Jo solved **only some** of the problems.

c. ? – Jo solved **some but not all** of the problems.

d. ?? – Jo solved **SOME** of the problems.

- Again in line with the idea that sentences should not settle overt QuDs *via* their presupposition!

Summary

Answer → / QuD ↓	some	only(some)	pex(some)[17]	some but not all =exh(some)[18]
?some	✓	✗	✓	meh
?all	✗	✓	✗	meh

- The compatibility between **pex**ed/**only**ed assertions and QuDs is constrained by the idea that QuDs should not be presuppositionally settled.

We adapt this to cover SHDs like (3b), arguing that their 1st disjunct evokes a QuD which must be felicitously addressed by their 2nd disjunct.

[18] We extrapolated a bit for **pex** here, because it remains optional in simplex sentences. But it will be *needed* to rescue SHDs.

[18] **pex**(some), unlike **exh**(some) is predicted to be out under a **?all** QuD. This will turn out crucial for our account to work.

Summary

Answer → / QuD ↓	some	only(some)	pex(some)[17]	some but not all =exh(some)[18]
?some	✓	✗	✓	meh
?all	✗	✓	✗	meh

- The compatibility between **pex**ed/**only**ed assertions and QuDs is constrained by the idea that QuDs should not be presuppositionally settled.

We adapt this to cover SHDs like (3b), arguing that their 1st disjunct evokes a QuD which must be felicitously addressed by their 2nd disjunct.

[18] We extrapolated a bit for **pex** here, because it remains optional in simplex sentences. But it will be *needed* to rescue SHDs.

[18] **pex**(some), unlike **exh**(some) is predicted to be out under a **?all** QuD. This will turn out crucial for our account to work.

Summary

Answer → / QuD ↓	some	only(some)	pex(some)[17]	some but not all =exh(some)[18]
?some	✓	✗	✓	meh
?all	✗	✓	✗	meh

- The compatibility between **pex**ed/**only**ed assertions and QuDs is constrained by the idea that QuDs should not be presuppositionally settled.

We adapt this to cover SHDs like (3b), arguing that their 1st disjunct evokes a QuD which must be felicitously addressed by their 2nd disjunct.

[18] We extrapolated a bit for **pex** here, because it remains optional in simplex sentences. But it will be *needed* to rescue SHDs.

[18] **pex**(some), unlike **exh**(some) is predicted to be out under a ?all QuD. This will turn out crucial for our account to work.

Incremental implicit QuDs

TL;DR: individual disjuncts evoke implicit QuDs "on-the-fly", which provides strong cues about what the global QuD should look like.

Relating disjuncts to question-answer pairs

- We want to relate SHDs to pairs made of:
  - an **overt QuD** like *Did Jo solve* **all** *of the problems?*
  - and a **follow-up assertion** like *Jo solved* **pex/only some** *of the problems.*
- We need to devise **a model of the QuD** evoked by **all**...
- and to flesh out the **interaction between this QuD and the 2nd disjunct** (**only**/**pex**) **some**, in particular when it comes to presuppositions.

Relating disjuncts to question-answer pairs

- We want to relate SHDs to pairs made of:
  - an **overt QuD** like *Did Jo solve* **all** *of the problems?*
  - and a **follow-up assertion** like *Jo solved* **pex/only some** *of the problems.*
- We need to devise **a model of the QuD** evoked by **all**...
- and to flesh out the **interaction between this QuD and the 2nd disjunct** (**only/pex**) **some**, in particular when it comes to presuppositions.

Relating disjuncts to question-answer pairs

- We want to relate SHDs to pairs made of:
  - an **overt QuD** like *Did Jo solve* **all** *of the problems?*
  - and a **follow-up assertion** like *Jo solved* **pex/only some** *of the problems.*
- We need to devise **a model of the QuD** evoked by **all**...
- and to flesh out the **interaction between this QuD and the 2nd disjunct** (**only**/**pex**) **some**, in particular when it comes to presuppositions.

Relating disjuncts to question-answer pairs

- We want to relate SHDs to pairs made of:
  - an **overt QuD** like *Did Jo solve* **all** *of the problems?*
  - and a **follow-up assertion** like *Jo solved* **pex/only some** *of the problems.*
- We need to devise **a model of the QuD** evoked by **all**...
- and to flesh out the **interaction between this QuD and the 2nd disjunct** (**only**/**pex**) **some**, in particular when it comes to presuppositions.

Relating disjuncts to question-answer pairs

- We want to relate SHDs to pairs made of:
  - an **overt QuD** like *Did Jo solve* **all** *of the problems?*
  - and a **follow-up assertion** like *Jo solved* **pex/only some** *of the problems.*
- We need to devise **a model of the QuD** evoked by **all**...
- and to flesh out the **interaction between this QuD and the 2nd disjunct** (**only**/**pex**) **some**, in particular when it comes to presuppositions.

Implicit QuDs

- Simplex sentences (e.g. disjuncts) evoke the implicit QuDs they could felicitously answer.[19]
- These QuDs take the form of **nested partitions of the Context Set (CS)**, graphically represented as trees.[20]

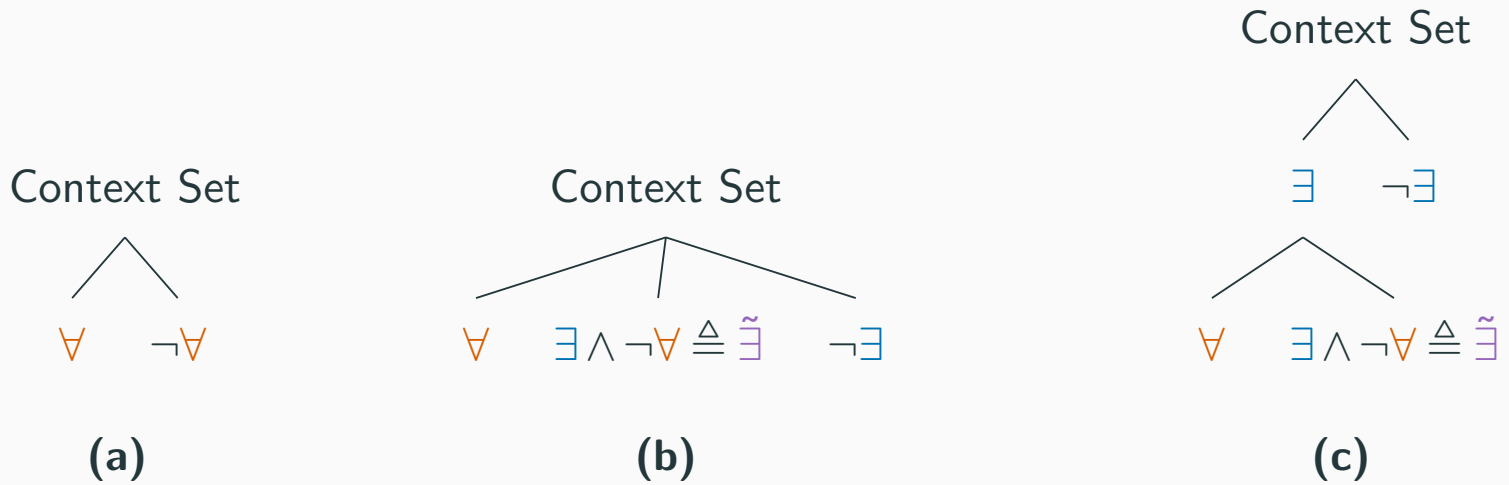


Figure 2: Possible QuDs evoked by *Jo solved* **all** *of the problems*.

[19] Zhang, 2022.

[20] Close in spirit to Ippolito's SSAs and Zhang's QuD-trees. This complicates the story for SHDs, but (i) explains the subtlety of the contrast in (3), and (ii) accounts for Sobel cases, plus cases involving Distant Entailing Alternatives and Subdomain Alternatives. See Appendix.

Implicit QuDs

- Simplex sentences (e.g. disjuncts) evoke the implicit QuDs they could felicitously answer.[19]
- These QuDs take the form of **nested partitions of the Context Set** (**CS**), graphically represented as trees.[20]

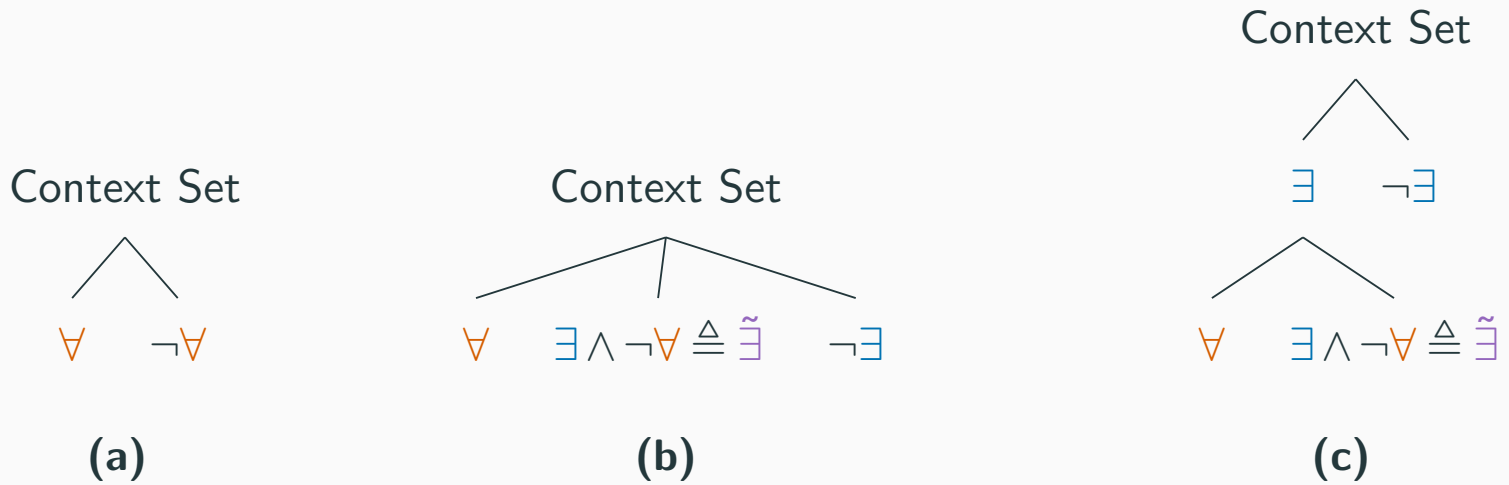


Figure 2: Possible QuDs evoked by *Jo solved* **all** *of the problems*.

[19] Zhang, 2022.

[20] Close in spirit to Ippolito's SSAs and Zhang's QuD-trees. This complicates the story for SHDs, but (i) explains the subtlety of the contrast in (3), and (ii) accounts for Sobel cases, plus cases involving Distant Entailing Alternatives and Subdomain Alternatives. See Appendix.

Incremental processing of disjunctive QuDs

- Disjuncts usually have to answer the same QuD.[21]
- Our interpretation: the QuD of a disjunction results from the **merger of QuDs evoked by its constitutive disjuncts**.
- After computing some QuD Q evoked by the 1st disjunct, we know that the global QuD should be a **(non-strict) "supertree" of Q!**

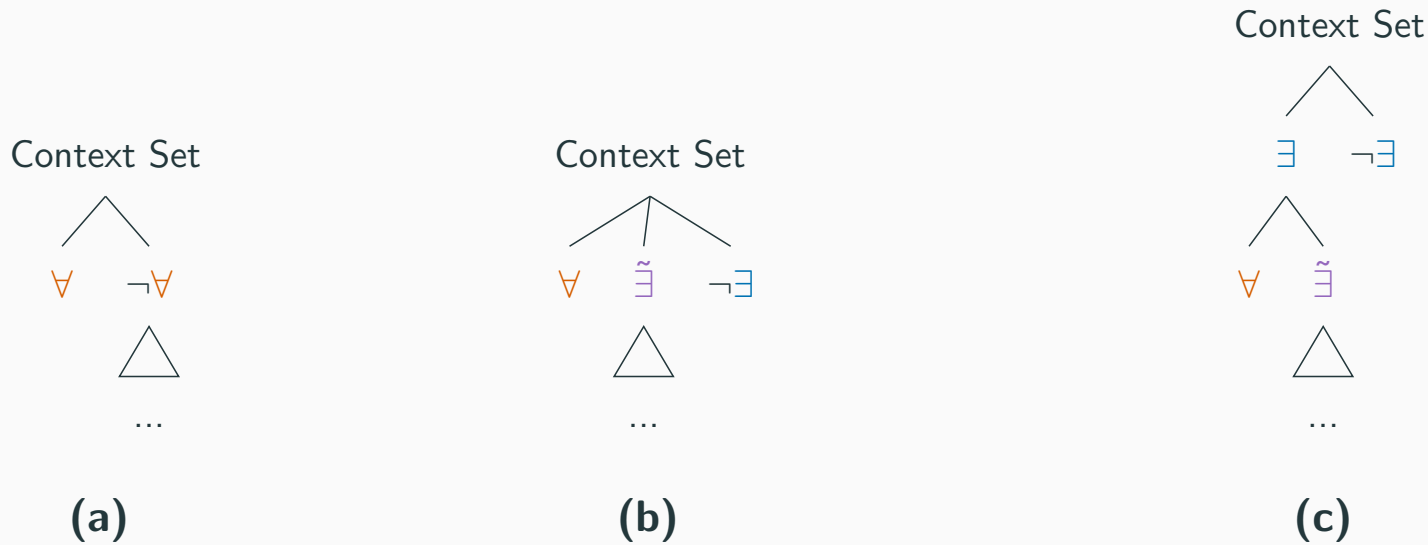


Figure 3: "Supertrees" of QuD trees evoked by *Jo solved* **all** *of the problems*.

[21] Simons, 2001; Westera, 2020; Zhang, 2022.

Incremental processing of disjunctive QuDs

- Disjuncts usually have to answer the same QuD.[21]
- Our interpretation: the QuD of a disjunction results from the **merger of QuDs evoked by its constitutive disjuncts**.
- After computing some QuD Q evoked by the 1st disjunct, we know that the global QuD should be a **(non-strict) "supertree" of** Q**!**

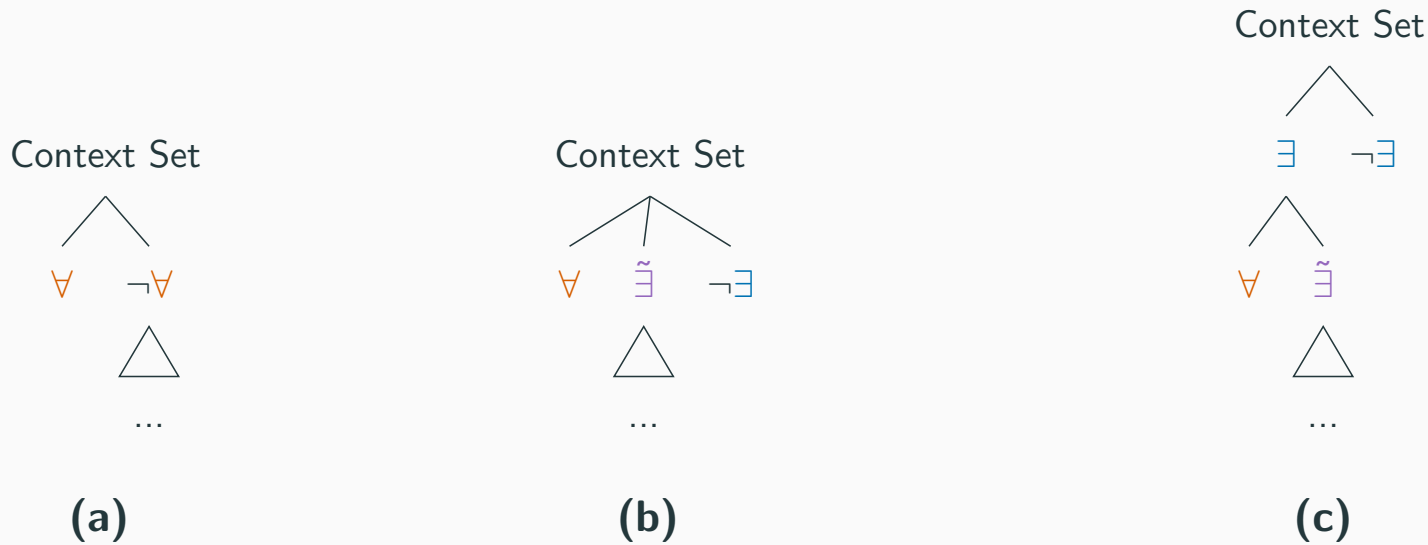


Figure 3: "Supertrees" of QuD trees evoked by *Jo solved* **all** *of the problems*.

[21] Simons, 2001; Westera, 2020; Zhang, 2022.

Incremental processing of disjunctive QuDs

- Disjuncts usually have to answer the same QuD.[21]
- Our interpretation: the QuD of a disjunction results from the **merger of QuDs evoked by its constitutive disjuncts**.
- After computing some QuD Q evoked by the 1st disjunct, we know that the global QuD should be a **(non-strict) "supertree" of Q!**

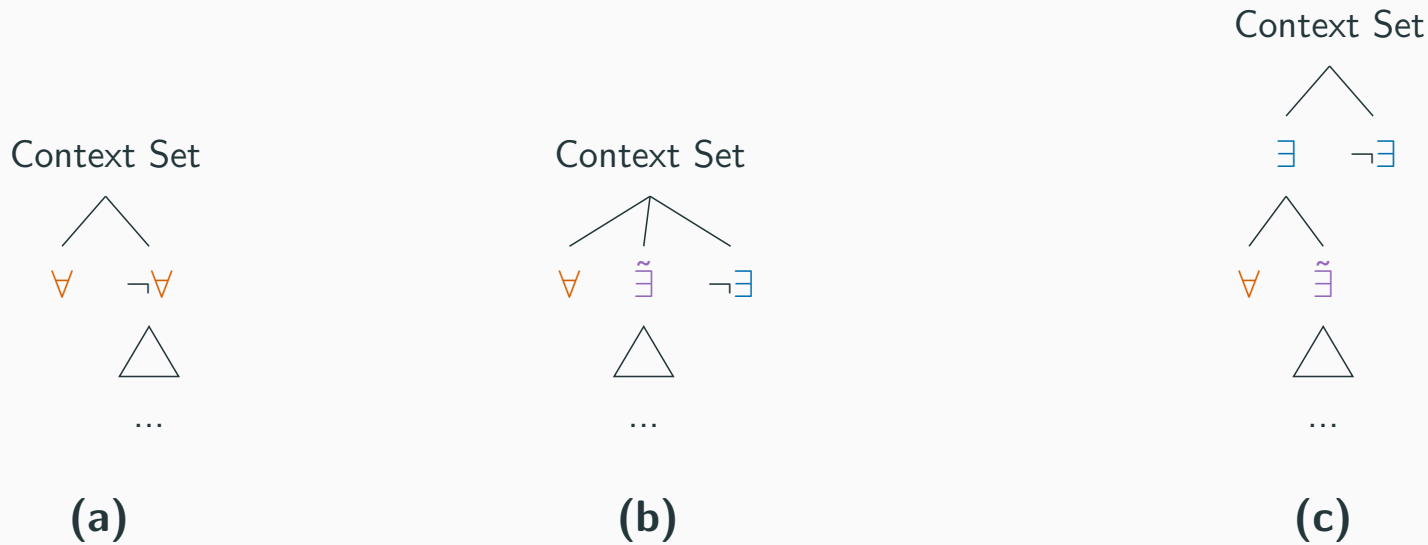


Figure 3: "Supertrees" of QuD trees evoked by *Jo solved* **all** *of the problems*.

[21] Simons, 2001; Westera, 2020; Zhang, 2022.

Effect of accommodation on QuD trees

- We now have a model of QuDs evoked by the 1st disjunct, and how they shape the global QuD.
- How do presuppositions introduced by the 2nd disjunct interact with such implicit, incremental QuDs?
  - Accommodating a presupposition p typically amounts to **intersecting the CS with p.**
  - Given that QuDs are nested partitions of the CS (=trees), accommodating p on a QuD tree T amounts to **intersecting each node of T with p**, removing empty nodes, dangling or unary branches. We call the result $T \cap p$.
- We now detail how **not all** (presupposition of **pex**(**some**)) and **some** (presupposition of **only**(**some**)) get accomodated on the QuDs from Figure 3.

Effect of accommodation on QuD trees

- We now have a model of QuDs evoked by the 1st disjunct, and how they shape the global QuD.
- How do presuppositions introduced by the 2nd disjunct interact with such implicit, incremental QuDs?
  - Accommodating a presupposition p typically amounts to **intersecting the CS with p.**
  - Given that QuDs are nested partitions of the CS (=trees), accommodating p on a QuD tree T amounts to **intersecting each node of T with p**, removing empty nodes, dangling or unary branches. We call the result $T \cap p$.
- We now detail how **not all** (presupposition of **pex**(**some**)) and **some** (presupposition of **only**(**some**)) get accomodated on the QuDs from Figure 3.

Effect of accommodation on QuD trees

- We now have a model of QuDs evoked by the 1st disjunct, and how they shape the global QuD.
- How do presuppositions introduced by the 2nd disjunct interact with such implicit, incremental QuDs?
  - Accommodating a presupposition p typically amounts to **intersecting the CS with p.**
  - Given that QuDs are nested partitions of the CS (=trees), accommodating p on a QuD tree T amounts to **intersecting each node of T with p**, removing empty nodes, dangling or unary branches. We call the result $T \cap p$.
- We now detail how **not all** (presupposition of **pex**(**some**)) and **some** (presupposition of **only**(**some**)) get accomodated on the QuDs from Figure 3.

Effect of accommodation on QuD trees

- We now have a model of QuDs evoked by the 1st disjunct, and how they shape the global QuD.
- How do presuppositions introduced by the 2nd disjunct interact with such implicit, incremental QuDs?
  - Accommodating a presupposition p typically amounts to **intersecting the CS with p.**
  - Given that QuDs are nested partitions of the CS (=trees), accommodating p on a QuD tree T amounts to **intersecting each node of T with p**, removing empty nodes, dangling or unary branches. We call the result $T \cap p$.
- We now detail how **not all** (presupposition of **pex**(**some**)) and **some** (presupposition of **only**(**some**)) get accomodated on the QuDs from Figure 3.

Effect of accommodation on QuD trees

- We now have a model of QuDs evoked by the 1st disjunct, and how they shape the global QuD.
- How do presuppositions introduced by the 2nd disjunct interact with such implicit, incremental QuDs?
  - Accommodating a presupposition p typically amounts to **intersecting the CS with p.**
  - Given that QuDs are nested partitions of the CS (=trees), accommodating p on a QuD tree T amounts to **intersecting each node of T with p**, removing empty nodes, dangling or unary branches. We call the result $T \cap p$.
- We now detail how **not all** (presupposition of **pex**(**some**)) and **some** (presupposition of **only**(**some**)) get accomodated on the QuDs from Figure 3.

Accommodating not all on our incremental ?all QuD

(3b) ?? Jo did $\underbrace{\textbf{all}}_{\text{QuD(all)}}$ **or** $\underbrace{\textbf{pex(some)}}_{P\ :\ \textbf{not all}}$ of the problems.

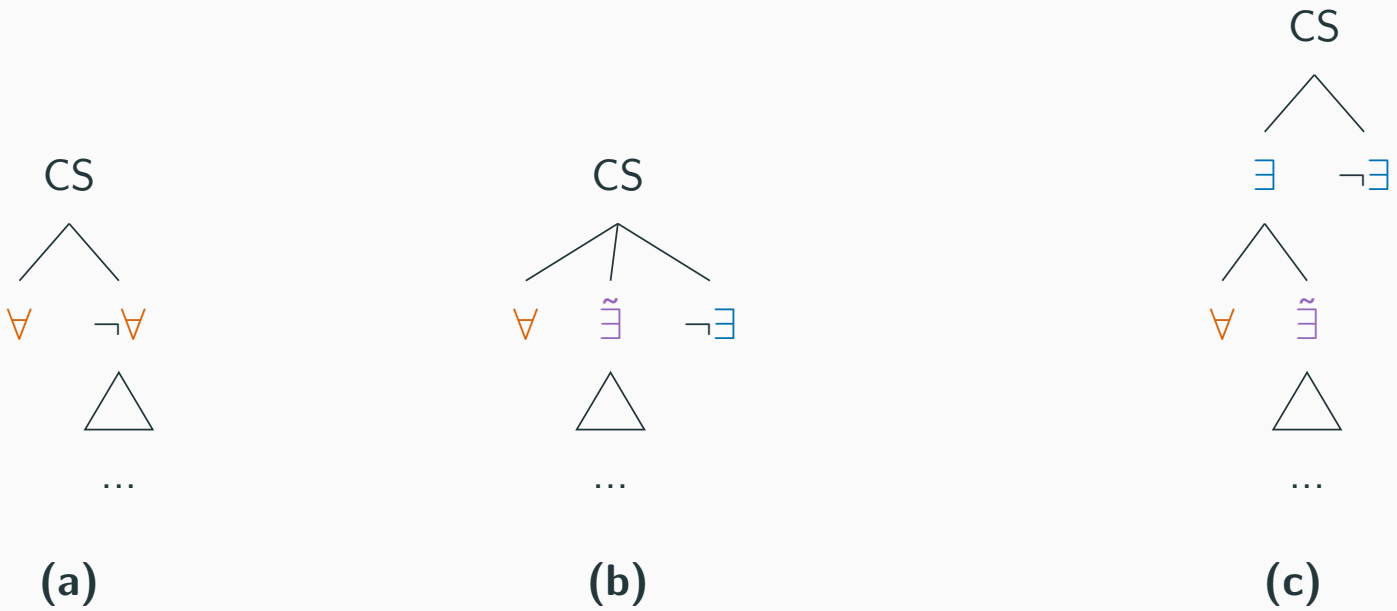
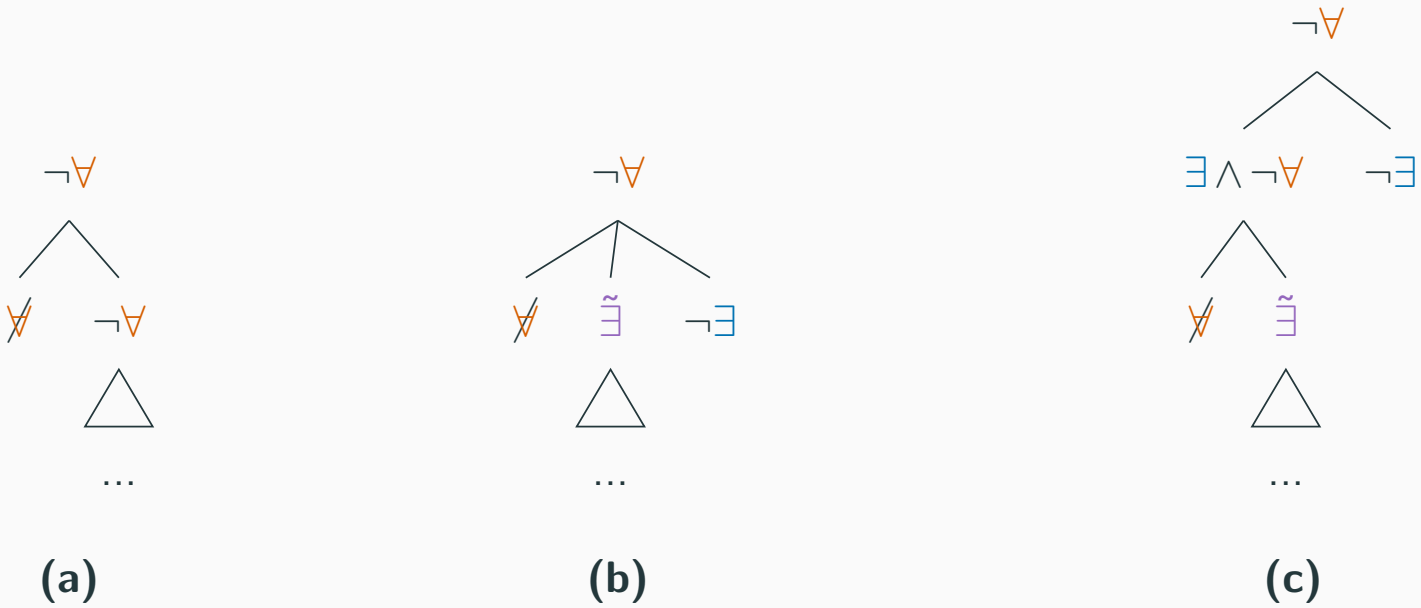


Figure 4: "Supertrees" of QuD trees evoked by *Jo solved* **all** *of the problems*. **Prior to accommodating not all.**

- Intuition: the **not all** presupposition carried by **pex(some)** addresses the **?all** QuDs trees pretty well, i.e. **removes a lot of their initial structure**!

Accommodating not all on our incremental ?all QuD

(3b) ?? Jo did $\underbrace{\textbf{all}}_{\text{QuD(all)}}$ **or** $\underbrace{\textbf{pex(some)}}_{P\ :\ \textbf{not all}}$ of the problems.



(a) **(b)** **(c)**

Figure 4: Accommodating **not all** on the "supertrees" of *Jo solved* **all** *of the problems*. **Each node gets intersected with not all.**

- Intuition: the **not all** presupposition carried by **pex(some)** addresses the **?all** QuDs trees pretty well, i.e. **removes a lot of their initial structure**!

Accommodating not all on our incremental ?all QuD

(3b) ?? Jo did $\underbrace{\textbf{all}}_{\text{QuD(all)}}$ **or** $\underbrace{\textbf{pex(some)}}_{P \,:\, \textbf{not all}}$ of the problems.

Figure 4: Accommodating **not all** on the "supertrees" of *Jo solved* **all** *of the problems*. **After removing empty nodes, dangling and unary branches.**

- Intuition: the **not all** presupposition carried by **pex(some)** addresses the **?all** QuDs trees pretty well, i.e. **removes a lot of their initial structure**!

Accommodating some on our incremental ?all QuD

(6b) Jo did $\underbrace{\textbf{all}}_{\text{QuD(all)}}$ **or** $\underbrace{\textbf{only(some)}}_{P\ :\ \textbf{some}}$ of the problems.

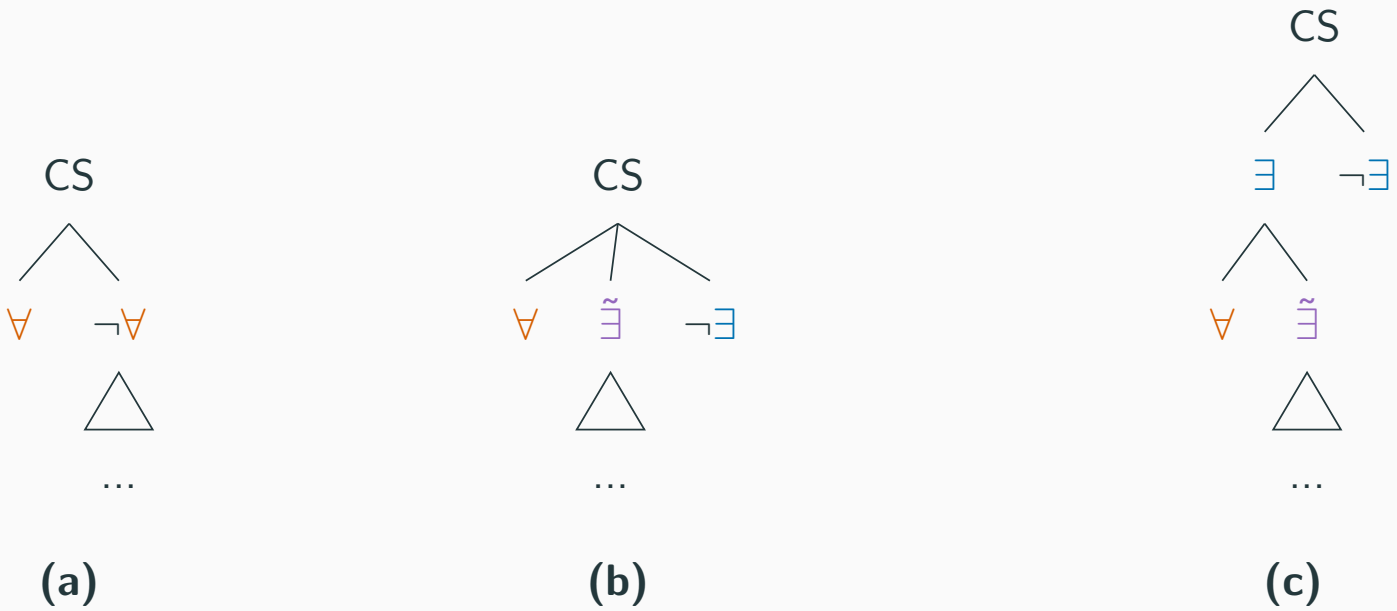


Figure 5: "Supertrees" of QuD trees evoked by *Jo solved* **all** *of the problems*. **Prior to accommodating some.**

- Intuition: the **some** presupposition carried by **only(some)** addresses the **?all** QuD trees differently than **not all**. The resulting tree, is s.t. some structure is left to be addressed by **only(some)**'s assertion!

Accommodating some on our incremental ?all QuD

(6b) Jo did **all** **or** **only(some)** of the problems.

QuD(**all**)

P : **some**

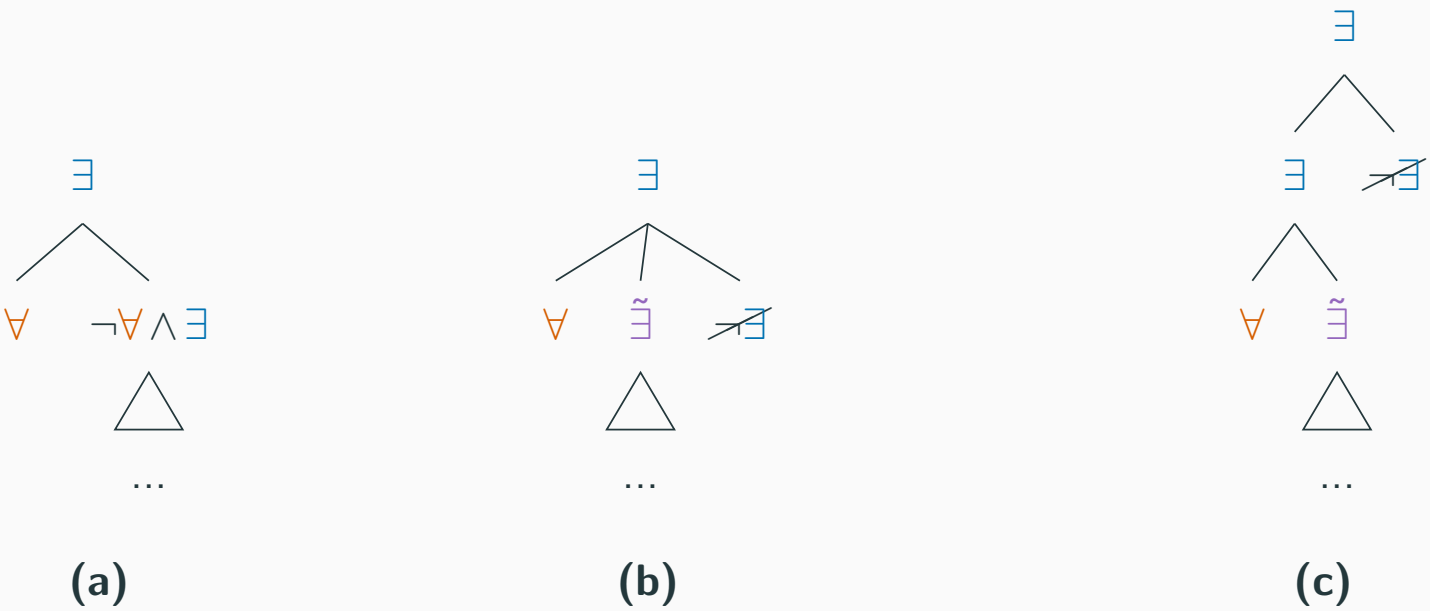
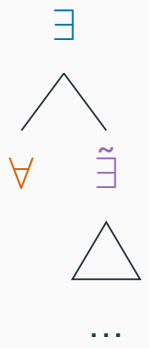


Figure 5: Accommodating **some** on the "supertrees" of *Jo solved* **all** *of the problems*. **Each node gets intersected with some.**

- Intuition: the **some** presupposition carried by **only(some)** addresses the **?all** QuD trees differently than **not all**. The resulting tree, is s.t. some structure is left to be addressed by **only(some)**'s assertion!

Accommodating some on our incremental ?all QuD

(6b) Jo did $\underbrace{\textbf{all}}_{\text{QuD}(\textbf{all})}$ **or** $\underbrace{\textbf{only(some)}}_{P\ :\ \textbf{some}}$ of the problems.



(a) / (b) / (c)

Figure 5: Accommodating **some** on the "supertrees" of *Jo solved* **all** *of the problems*. **After removing empty nodes, dangling and unary branches.**

- Intuition: the **some** presupposition carried by **only(some)** addresses the **?all** QuD trees differently than **not all**. The resulting tree, is s.t. some structure is left to be addressed by **only(some)**'s assertion!

Felicitously addressing (implicit) QuDs

TL;DR: The distribution of pex and only in the 2nd disjunct of SHDs can be captured assuming their presuppositions should not trivialize the incremental QuD evoked by the 1st disjunct.

Felicitously addressing overt QuDs

- The idea that QuDs should not be settled by presuppositions is implemented by the Post Accommodation Informativity (**PAI**) condition:[22]
  - If S presupposes p and intends to answer a question Q (partition of the CS), S **has to be informative w.r.t.** Q **after the CS gets updated with** p.
  - A sentence S is informative w.r.t. Q if it allows to rule out at least one cell in Q.[23]

[22] Doron and Wehbe, 2024.

[23] Roberts, 2012.

Felicitously addressing overt QuDs

- The idea that QuDs should not be settled by presuppositions is implemented by the Post Accommodation Informativity (**PAI**) condition:[22]
  - If S presupposes p and intends to answer a question Q (partition of the CS), S **has to be informative w.r.t.** Q **after the CS gets updated with** p.
  - A sentence S is informative w.r.t. Q if it allows to rule out at least one cell in Q.[23]

[22] Doron and Wehbe, 2024.

[23] Roberts, 2012.

Felicitously addressing overt QuDs

- The idea that QuDs should not be settled by presuppositions is implemented by the Post Accommodation Informativity (**PAI**) condition:[22]
  - If S presupposes p and intends to answer a question Q (partition of the CS), S **has to be informative w.r.t.** Q **after the CS gets updated with** p.
  - A sentence S is informative w.r.t. Q if it allows to rule out at least one cell in Q.[23]

[22] Doron and Wehbe, 2024.

[23] Roberts, 2012.

Felicitously addressing implicit, incremental QuDs

- We adapt PAI to incremental implicit QuDs.
- Given a partial LF C evoking a set of possible QuD trees $\mathbb{T}_C$, and a continuation S of C presupposing p, **for any QuD tree $T \in \mathbb{T}_C$, S should rule-out a node in $T \cap p$ ($=T$ updated with p).**
- In our target case (3b), C is **all or ...**, $\mathbb{T}_C$ is Figure 3 and p is either **not all** (if S is **pex(some)**) or **some** (if S is **only(some)**).
- We have already done most of the heavy lifting in the previous Section, when we accommodated **not all**/**some** on the **?all** incremental QuD trees.
- Let's check Incremental PAI on the resulting structures.

Felicitously addressing implicit, incremental QuDs

- We adapt PAI to incremental implicit QuDs.
- Given a partial LF C evoking a set of possible QuD trees $\mathbb{T}_C$, and a continuation S of C presupposing p, **for any QuD tree $T \in \mathbb{T}_C$, S should rule-out a node in $T \cap p$ ($=T$ updated with p).**
- In our target case (3b), C is **all or ...**, $\mathbb{T}_C$ is Figure 3 and p is either **not all** (if S is **pex(some)**) or **some** (if S is **only(some)**).
- We have already done most of the heavy lifting in the previous Section, when we accommodated **not all**/**some** on the **?all** incremental QuD trees.
- Let's check Incremental PAI on the resulting structures.

Felicitously addressing implicit, incremental QuDs

- We adapt PAI to incremental implicit QuDs.
- Given a partial LF C evoking a set of possible QuD trees $\mathbb{T}_C$, and a continuation S of C presupposing p, **for any QuD tree $T \in \mathbb{T}_C$, S should rule-out a node in $T \cap p$ ($=T$ updated with p).**
- In our target case (3b), C is **all or ...**, $\mathbb{T}_C$ is Figure 3 and p is either **not all** (if S is **pex(some)**) or **some** (if S is **only(some)**).
- We have already done most of the heavy lifting in the previous Section, when we accommodated **not all**/**some** on the **?all** incremental QuD trees.
- Let's check Incremental PAI on the resulting structures.

Felicitously addressing implicit, incremental QuDs

- We adapt PAI to incremental implicit QuDs.
- Given a partial LF C evoking a set of possible QuD trees $\mathbb{T}_C$, and a continuation S of C presupposing p, **for any QuD tree $T \in \mathbb{T}_C$, S should rule-out a node in $T \cap p$ ($=T$ updated with p).**
- In our target case (3b), C is **all or ...**, $\mathbb{T}_C$ is Figure 3 and p is either **not all** (if S is **pex(some)**) or **some** (if S is **only(some)**).
- We have already done most of the heavy lifting in the previous Section, when we accommodated **not all**/**some** on the **?all** incremental QuD trees.
- Let's check Incremental PAI on the resulting structures.

Felicitously addressing implicit, incremental QuDs

- We adapt PAI to incremental implicit QuDs.
- Given a partial LF C evoking a set of possible QuD trees $\mathbb{T}_C$, and a continuation S of C presupposing p, **for any QuD tree $T \in \mathbb{T}_C$, S should rule-out a node in $T \cap p$ ($=T$ updated with p).**
- In our target case (3b), C is **all or ...**, $\mathbb{T}_C$ is Figure 3 and p is either **not all** (if S is **pex(some)**) or **some** (if S is **only(some)**).
- We have already done most of the heavy lifting in the previous Section, when we accommodated **not all**/**some** on the **?all** incremental QuD trees.
- Let's check Incremental PAI on the resulting structures.

Deriving all or only(some)

(6b) Jo did **all** **or** **only(some)** of the problems.

QuD(**all**)

P : **some**
A : **not all**

Figure 6: Checking Incremental PAI after accommodating **some** on the "supertrees" of *Jo solved* **all** *of the problems*. **Nodes compatible with the assertion not all are underlined**.

- In the only possible Tree 6, the assertion of **only some**, **not all**, rules out the ∀-node.
- Incremental PAI is satisfied! And (6b) is ruled-in.

Deriving all or only(some)

(6b) Jo did $\underbrace{\textbf{all}}_{\text{QuD}(\textbf{all})}$ **or** $\underbrace{\textbf{only(some)}}_{\begin{array}{lll} P & : & \textbf{some} \\ A & : & \textbf{not all} \end{array}}$ of the problems.

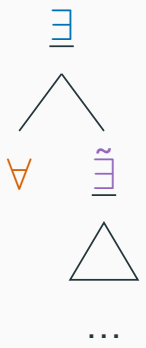


Figure 6: Checking Incremental PAI after accommodating **some** on the "supertrees" of *Jo solved* **all** *of the problems*. **Nodes compatible with the assertion not all are underlined**.

- In the only possible Tree 6, the assertion of **only some**, **not all**, rules out the ∀-node.
- Incremental PAI is satisfied! And (6b) is ruled-in.

Deriving all or only(some)

(6b) Jo did **all** **or** **only(some)** of the problems.

QuD(**all**)

P : **some**
A : **not all**

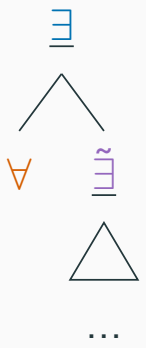


Figure 6: Checking Incremental PAI after accommodating **some** on the "supertrees" of *Jo solved* **all** *of the problems*. **Nodes compatible with the assertion not all are underlined**.

- In the only possible Tree 6, the assertion of **only some**, **not all**, rules out the ∀-node.
- Incremental PAI is satisfied! And (6b) is ruled-in.

Deriving all or only(some)

(6b) Jo did $\underbrace{\textbf{all}}_{\text{QuD}(\textbf{all})}$ **or** $\underbrace{\textbf{only(some)}}_{\begin{array}{lll} P & : & \textbf{some} \\ A & : & \textbf{not all} \end{array}}$ of the problems.

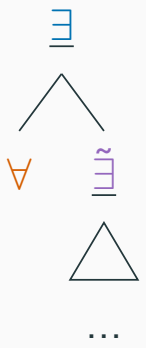


Figure 6: Checking Incremental PAI after accommodating **some** on the "supertrees" of *Jo solved* **all** *of the problems*. **Nodes compatible with the assertion not all are underlined**.

- In the only possible Tree 6, the assertion of **only some**, **not all**, rules out the ∀-node.
- Incremental PAI is satisfied! And (6b) is ruled-in.

Deriving #all or pex(some)

(3b) ?? Jo did $\underbrace{\textbf{all}}_{\text{QuD}(\textbf{all})}$ **or** $\underbrace{\textbf{pex(some)}}_{\begin{array}{lll} P & : & \textbf{not all} \\ A & : & \textbf{some} \end{array}}$ of the problems.

(a) (b)

Figure 7: Checking Incremental PAI after accommodating **not all** on the "supertrees" of *Jo solved* **all** *of the problems*. **Nodes compatible with the assertion some are underlined**.

- In Tree 7a, the assertion of **pex(some)**, **some**, **does not rule out any node**, i.e. is not informative!
- Incremental PAI, which must hold for *all* implicit QuD trees, is thus violated, and (3b) is ruled-out!
- (We don't care that in Tree 7b, **some** rules out the ¬∃-node.)

Deriving #all or pex(some)

(3b) ?? Jo did $\underbrace{\textbf{all}}_{\text{QuD}(\textbf{all})}$ **or** $\underbrace{\textbf{pex(some)}}_{\begin{array}{lll} P & : & \textbf{not all} \\ A & : & \textbf{some} \end{array}}$ of the problems.

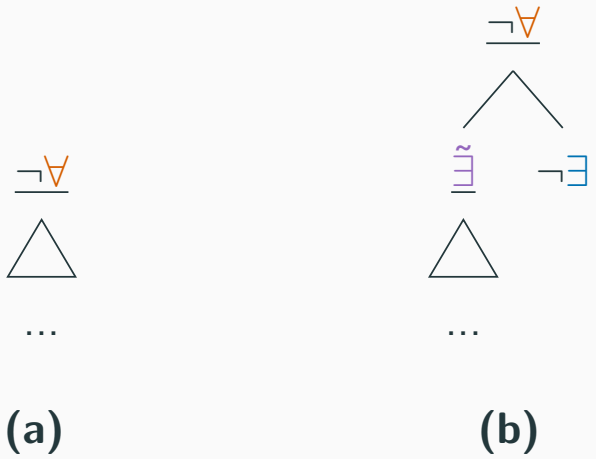
(a) **(b)**

Figure 7: Checking Incremental PAI after accommodating **not all** on the "supertrees" of *Jo solved* **all** *of the problems*. **Nodes compatible with the assertion some are underlined**.

- In Tree 7a, the assertion of **pex(some)**, **some**, **does not rule out any node**, i.e. is not informative!
- Incremental PAI, which must hold for *all* implicit QuD trees, is thus violated, and (3b) is ruled-out!
- (We don't care that in Tree 7b, **some** rules out the ¬∃-node.)

Deriving #all or pex(some)

(3b) ?? Jo did $\underbrace{\textbf{all}}_{\text{QuD(all)}}$ **or** $\underbrace{\textbf{pex(some)}}_{\begin{array}{lll} P & : & \textbf{not all} \\ A & : & \textbf{some} \end{array}}$ of the problems.



(a) **(b)**

Figure 7: Checking Incremental PAI after accommodating **not all** on the "supertrees" of *Jo solved* **all** *of the problems*. **Nodes compatible with the assertion some are underlined**.

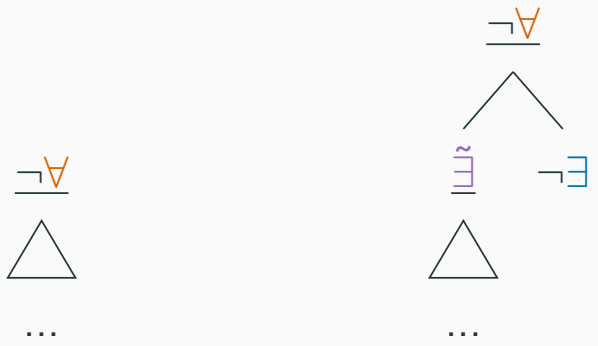
- In Tree 7a, the assertion of **pex(some)**, **some**, **does not rule out any node**, i.e. is not informative!
- Incremental PAI, which must hold for *all* implicit QuD trees, is thus violated, and (3b) is ruled-out!
- (We don't care that in Tree 7b, **some** rules out the ¬∃-node.)

Deriving #all or pex(some)

(3b) ?? Jo did **all** **or** **pex(some)** of the problems.

QuD(**all**)

P : **not all**
A : **some**



(a) **(b)**

Figure 7: Checking Incremental PAI after accommodating **not all** on the "supertrees" of *Jo solved* **all** *of the problems*. **Nodes compatible with the assertion some are underlined**.

- In Tree 7a, the assertion of **pex(some)**, **some**, **does not rule out any node**, i.e. is not informative!
- Incremental PAI, which must hold for *all* implicit QuD trees, is thus violated, and (3b) is ruled-out!
- (We don't care that in Tree 7b, **some** rules out the ¬∃-node.)

Deriving #all or pex(some)

(3b) ?? Jo did $\underbrace{\textbf{all}}_{\text{QuD}(\textbf{all})}$ **or** $\underbrace{\textbf{pex(some)}}_{\begin{array}{lll} P & : & \textbf{not all} \\ A & : & \textbf{some} \end{array}}$ of the problems.

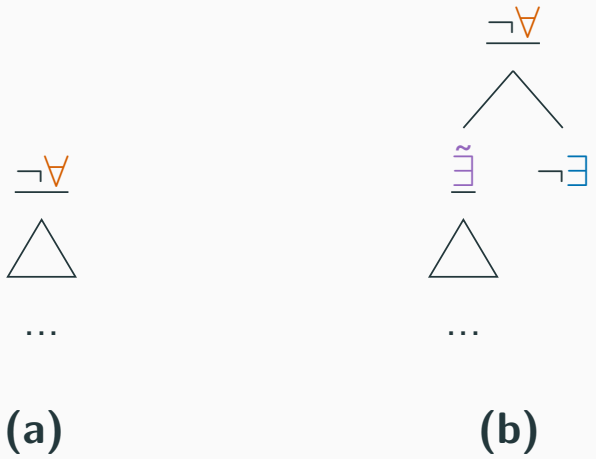
(a) **(b)**

Figure 7: Checking Incremental PAI after accommodating **not all** on the "supertrees" of *Jo solved* **all** *of the problems*. **Nodes compatible with the assertion some are underlined**.

- In Tree 7a, the assertion of **pex(some)**, **some**, **does not rule out any node**, i.e. is not informative!
- Incremental PAI, which must hold for *all* implicit QuD trees, is thus violated, and (3b) is ruled-out!
- (We don't care that in Tree 7b, **some** rules out the ¬∃-node.)

Deriving ??all or pex(some) – alternative route

(3b) ?? Jo did $\underbrace{\textbf{all}}_{\text{QuD(all)}}$ **or** $\underbrace{\textbf{pex(some)}}_{\begin{array}{ll} P & : \textbf{not all} \\ A & : \textbf{some} \end{array}}$ of the problems.



(a) **(b)**

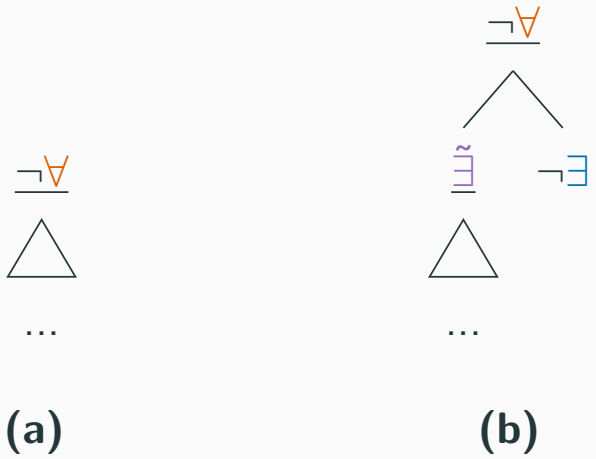
Figure 8: Checking Incremental PAI after accommodating **not all** on the "supertrees" of *Jo solved* **all** *of the problems*. **Nodes compatible with the assertion some are underlined**.

- Alternative view: maybe Incremental PAI has **existential force**, and the "bad" Tree 8a simply corresponds to the **most salient implicit QuD**.
- (3b)'s **relative infelicity** would then result from a QuD "garden-path".[1]

[1] Thanks to Jad and Nina for suggesting this.

Deriving ??all or pex(some) – alternative route

(3b) ?? Jo did $\underbrace{\textbf{all}}_{\text{QuD(all)}}$ **or** $\underbrace{\textbf{pex(some)}}_{\begin{array}{ll} P & : \textbf{not all} \\ A & : \textbf{some} \end{array}}$ of the problems.



(a) **(b)**

Figure 8: Checking Incremental PAI after accommodating **not all** on the "supertrees" of *Jo solved* **all** *of the problems*. **Nodes compatible with the assertion some are underlined**.

- Alternative view: maybe Incremental PAI has **existential force**, and the "bad" Tree 8a simply corresponds to the **most salient implicit QuD**.
- (3b)'s **relative infelicity** would then result from a QuD "garden-path".[1]

[1] Thanks to Jad and Nina for suggesting this.

Getting the flipped cases for free

- So far we have overlooked the flipped case (3a), whereby both **pex** and **only** are licensed.

(3a) Jo did ✓**pex**/✓**only** **some** **or** **all** of the problems.

- When processing (3a)'s 1st disjunct (where the presupposition-bearing operators get inserted), the context C is empty, so $\mathbb{T}_C$ is empty and **Incremental PAI is trivially verified**.[24]
- Therefore, nothing prevents **pex**/**only** to be inserted in the 1st disjunct of (3a).

[24] At least under its universal flavor.

Getting the flipped cases for free

- So far we have overlooked the flipped case (3a), whereby both **pex** and **only** are licensed.

(3a) Jo did ✓**pex**/✓**only** **some** **or** **all** of the problems.

- When processing (3a)'s 1st disjunct (where the presupposition-bearing operators get inserted), the context C is empty, so $\mathbb{T}_C$ is empty and **Incremental PAI is trivially verified**.[24]
- Therefore, nothing prevents **pex**/**only** to be inserted in the 1st disjunct of (3a).

[24] At least under its universal flavor.

Getting the flipped cases for free

- So far we have overlooked the flipped case (3a), whereby both **pex** and **only** are licensed.

(3a) Jo did ✓**pex**/✓**only** **some** **or** **all** of the problems.

- When processing (3a)'s 1st disjunct (where the presupposition-bearing operators get inserted), the context C is empty, so $\mathbb{T}_C$ is empty and **Incremental PAI is trivially verified**.[24]
- Therefore, nothing prevents **pex**/**only** to be inserted in the 1st disjunct of (3a).

[24] At least under its universal flavor.

Extension: Sobel

- This result extends to Sobel Sequences, thanks to the nested character of implicit QuD trees, and modulo two assumptions:
  - Conditionals "plug" the consequent QuD into leaves of the antecedent QuD verifying the antecedent.[25]
  - The *but* linking Sobel conditionals behaves like an *or* at the QuD level.

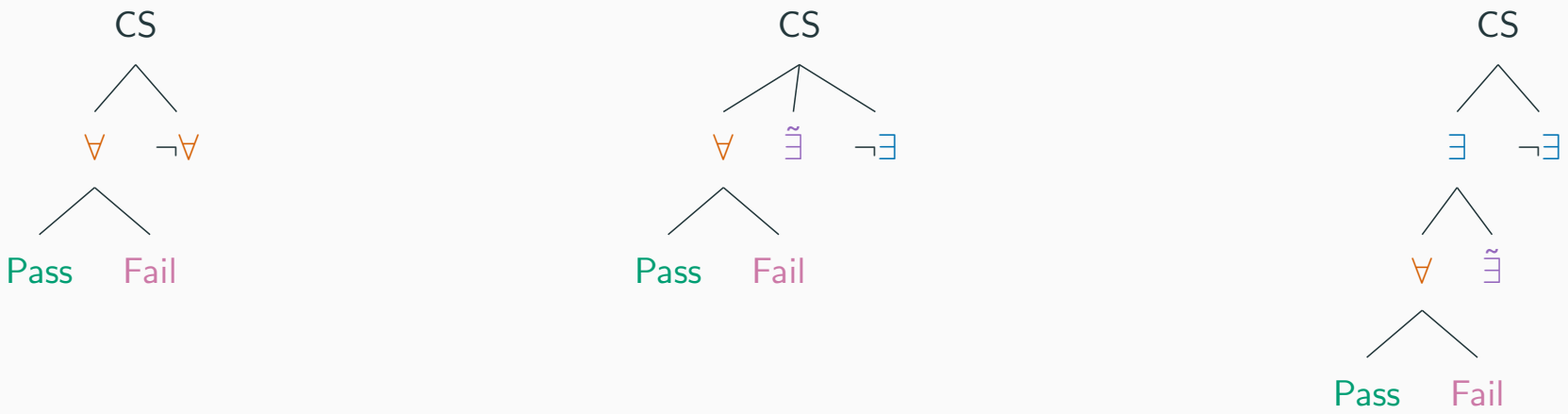


Figure 9: QuDs evoked by **If all then pass**: similar to those evoked by **all** except there is one extra {Pass, Fail} subpartition.

[25] An idea already entertained by Enguehard (2021) for independent reasons.

Conclusion: REUSE, REDUCE, RECYCLE

- We accounted for the contrast between (presuppositional) **exh** (asymmetric rescuer in SHDs), and **only** (symmetric rescuer), based on **standard entries** for these operators, and the incremental adaptation of PAI, an **independently motivated constraint on presupposition accommodation**.
- Under that view, the two disjuncts of a SHD behave pretty much like an overt QuD (∼1st disjunct), and its answer (2nd disjunct), with the constraint that the answer should not presuppositionally settle the QuD.
- Lastly, we explored a way to capture the **subtleness** of the asymmetry introduced by **pex**, anaylzed as a "QuD garden-path".
- Further extensions: scales involving "distant-entailing"/partially ordered alternatives.

Conclusion: REUSE, REDUCE, RECYCLE

- We accounted for the contrast between (presuppositional) **exh** (asymmetric rescuer in SHDs), and **only** (symmetric rescuer), based on **standard entries** for these operators, and the incremental adaptation of PAI, an **independently motivated constraint on presupposition accommodation**.
- Under that view, the two disjuncts of a SHD behave pretty much like an overt QuD (∼1st disjunct), and its answer (2nd disjunct), with the constraint that the answer should not presuppositionally settle the QuD.
- Lastly, we explored a way to capture the **subtleness** of the asymmetry introduced by **pex**, anaylzed as a "QuD garden-path".
- Further extensions: scales involving "distant-entailing"/partially ordered alternatives.

Conclusion: REUSE, REDUCE, RECYCLE

- We accounted for the contrast between (presuppositional) **exh** (asymmetric rescuer in SHDs), and **only** (symmetric rescuer), based on **standard entries** for these operators, and the incremental adaptation of PAI, an **independently motivated constraint on presupposition accommodation**.
- Under that view, the two disjuncts of a SHD behave pretty much like an overt QuD (∼1st disjunct), and its answer (2nd disjunct), with the constraint that the answer should not presuppositionally settle the QuD.
- Lastly, we explored a way to capture the **subtleness** of the asymmetry introduced by **pex**, anaylzed as a "QuD garden-path".
- Further extensions: scales involving "distant-entailing"/partially ordered alternatives.

Conclusion: REUSE, REDUCE, RECYCLE

- We accounted for the contrast between (presuppositional) **exh** (asymmetric rescuer in SHDs), and **only** (symmetric rescuer), based on **standard entries** for these operators, and the incremental adaptation of PAI, an **independently motivated constraint on presupposition accommodation**.
- Under that view, the two disjuncts of a SHD behave pretty much like an overt QuD ($\sim$1st disjunct), and its answer (2nd disjunct), with the constraint that the answer should not presuppositionally settle the QuD.
- Lastly, we explored a way to capture the **subtleness** of the asymmetry introduced by **pex**, anaylzed as a "QuD garden-path".
- Further extensions: scales involving "distant-entailing"/partially ordered alternatives.

Thank you!

Selected references i

Sobel, J. H. (1970).**Utilitarianisms: Simple and general.** *Inquiry: An Interdisciplinary Journal of Philosophy*, *13*(1-4), 394–449. https://doi.org/10.1080/00201747008601599

Lewis, D. K. (1973). **Counterfactuals.** Blackwell.

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

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

Roberts, C. (1996).**Information Structure in Discourse: Towards an Integrated Formal Theory of Pragmatics.** *Semantics and Pragmatics*, *5*, 1–69.

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

von Fintel, K. (2001, April). **Counterfactuals in a dynamic context.** In *Ken hale* (pp. 123–152). The MIT Press. https://doi.org/10.7551/mitpress/4056.003.0006

Singh, R. (2008a). **Modularity and locality in interpretation [Doctoral dissertation, MIT].**

Selected references ii

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

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

Roberts, C. (2012).**Information structure in discourse: Towards an integrated formal theory of pragmatics.** *Semantics and Pragmatics*, *5*. https://doi.org/10.3765/sp.5.6

Heim, I. (2015). **Lecture notes.**

Fox, D., & Spector, B. (2018).**Economy and embedded exhaustification.** *Natural Language Semantics*, *26*(1), 1–50. https://doi.org/10.1007/s11050-017-9139-6

Ippolito, M. (2019).**Varieties of sobel sequences.** *Linguistics and Philosophy*, *43*(6), 633–671. https://doi.org/10.1007/s10988-019-09281-8

Westera, M. (2020).**Hurford disjunctions: An in-depth comparison of the grammatical and the pragmatic approach.** *Under review*.

Selected references iii

Bassi, I., Pinal, G. D., & Sauerland, U. (2021).**Presuppositional exhaustification.** *Semantics and Pragmatics*, *14*, 1–42.

Enguehard, É. (2021).**Explaining presupposition projection in (coordinations of) polar questions.** *Natural Language Semantics*, *29*(4), 527–578. https://doi.org/10.1007/s11050-021-09182-2

Tomioka, S. (2021).**Scalar Implicature, Hurford's Constraint, Contrastiveness and How They All Come Together.** *Frontiers in Communication*, *5*. https://doi.org/10.3389/fcomm.2020.461553

Aravind, A., Fox, D., & Hackl, M. (2022).**Principles of presupposition in development.** *Linguistics and Philosophy*, *46*(2), 291–332. https://doi.org/10.1007/s10988-022-09364-z

Zhang, Y. (2022). **New perspectives on inquisitive semantics [Doctoral dissertation, University of Maryland].**

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. https://doi.org/10.18148/sub/2023.v27.1071

Doron, O., & Wehbe, J. (2024).**On the pragmatic status of locally accommodated presuppositions.**

Selected references iv

Krifka, M. (2024).**Hurford's constraint and disjunctions over speech acts.** *Proceedings of the 2024 Amsterdam Colloquium*.

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

Appendix

Extension: Distant Entailing Alternatives

- Unlike Singh's LMP! story, our account (modulo one extra assumption about QuD tree "monotonicity") can explain why **pex** becomes a *symmetric* rescuer when **most** is made salient, beside **some** and **all**.