Conventional implicatures in Dependent Type Semantics

Daiki Matsuoka (PhD student at Department of Computer Science, The University of Tokyo)

A guest lecture at the ESSLLI2025 course: *Composing Meaning via Dependent Types* Jul. 31, 2025

Starting point: appositive relative clause

This talk mainly focuses on a construction called an **appositive relative clauses (ARC)**.

(1) Kim, who is away, will return tomorrow.

The content of an ARC (e.g., "Kim is away" in (1)) is **projective**.

- (2) a. Negation
 It is not true that [Kim, who is away, will return tomorrow].
 - b. Conditional antecedent

 If Kim, who is away, returns tomorrow, we ...

Is it a presupposition?

However, the content of an ARC is NOT a presupposition.

To see why, recall that presupposition can be **filtered**.

(3) If Kim is away, Alex will have realized that she is away.

In contrast, the content of an ARC can NOT be filtered. It must be **new information** to the (local) context (Potts, 2005).

(4) # If Kim is away, Alex will tell us when Kim, who is away, will return.

Challenge

- This type of projective content is called a **conventional implicature (CI)**.
 - Other triggers of CIs: expressives (Potts, 2005; McCready, 2010), evidentials (Murray, 2014)
- CIs pose a challenge to DTS.
 - o The @-type cannot be directly used to represent CIs, since it would allow filtering.
 - Although there was a study on CI with an early version of DTS (Bekki & McCready, 2015), the analysis is difficult to reformulate in the current setting. In addition, it cannot handle some properties of CIs.

Q. How can we formally analyze CIs in the framework of DTS?

Some properties of CIs

Not-at-issueness

- A CI is **not at-issue**, in the sense that it does not constitute the "central content" of an utterance (Simons et al., 2010; Koev, 2018).
- A diagnostics for (not-)at-issueness: **Direct Reply Test** (Tonhauser, 2012)
 - o Only the at-issue content can be the target of the direct assent/dissent.
 - (5) a. A: Kim, who is away, will return tomorrow.
 - b. B: No, she won't. She is planned to be away for a week.
 - c. B': #No, she isn't. She has just come back.

CIs update the discourse in a way that the hearer cannot directly respond to the update (Murray, 2014; AnderBois et al., 2015).

Cross-dimensional anaphora

Although the content of an ARC seems to be independent from the at-issue content, there can be an **anaphoric dependency** between the two (Amaral et al., 2007; Nouwen, 2007).

(6) Kim, who has $a^i \log$, takes it_i for a walk.

Presupposition filtering shows the same pattern (AnderBois et al., 2015).

(7) Kim, who used to smoke, has stopped smoking.

The at-issue content can be anaphorically dependent on CIs.

Interaction with quantifier scope

When a CI involves a pronoun bound by a quantifier, it may lead to a **universally quantified** projective content.

- (8) Everyⁱ cyclist met Lance, who gave him $_i$ a Tour de France souvenir.
 - \implies For each cyclist x, Lance gave x a TDF souvenir.

(Martin, 2016, (13b))

- (9) Noⁱ candidate suspects that his $_i$ wife, who is after all his $_i$ biggest supporter, will vote against him $_i$.
 - \implies For each candidate x, x's wife is x's biggest supporter.

(Schlenker, 2020, (40))

Denoting a proposition φ with a CI ψ as φ_{ψ} ...

$$Qx \in A.(\varphi_{Px}) \implies \forall x \in A.Px$$

[+] Interaction with quantifier scope: projection

Doesn't the CI take scope below the quantifier in cases like (8)?

- —No. The implications is indeed projective.
 - (10) If [everyⁱ student has bid farewell to Nate, who had given them $_i$ some great advice during their $_i$ individual meetings], then we can officially close off the session.

 \implies For each student x, Nate gave x some great advice. (Zhao, 2023, (11a))

Note that the conditional antecedent is a **scope island**: the scope of "every student" cannot go out of it.

CI type

Recap

As we have seen, DTS assumes the following steps to obtain semantic representations.

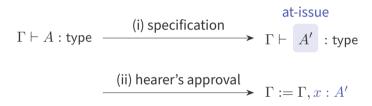
- **Type checking** derives the felicity condition $\Gamma \vdash A$: type
- @-elimination substitutes x @ A with the term found in proof search.

Crucially, these steps have a role of **specifying the at-issue content** (= the central message of the speaker's utterance).

Intuition: the hearer cannot accept/reject an utterance with pronouns without specifying their referents.

The pipeline

We can formulate the process of discourse update as follows.

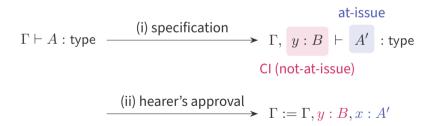


Q. Then, how can we incorporate CIs into this picture?

Sketch of the account

Recall: a CI is not subject to the hearer's direct reply.

This indicates that the update with a CI occurs while the hearer is specifying the at-issue content.



CI type

To formalize this intuition, we introduce a new type constructor (Matsuoka et al., 2024).

Definition (CI type)

We add $(x \lhd \Lambda) \times \Lambda$ to the recursive definition of UDTT preterms.

We represent a CI A with a type of the form $(x \lhd A) \times \cdots$.

(1) Kim, who is away, will return tomorrow.

$$ightsquigar \left[egin{align*} u & \lhd \mathtt{away}(\mathtt{k}) \\ \mathtt{rtn}(\mathtt{k}) \end{array}
ight]$$
 (we will use the box notation for the CI type, too.)

[+] Semantic composition

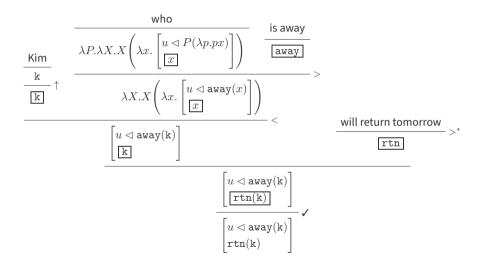
An appositive relative pronoun lexically introduces a CI type.

$$(11) \quad \llbracket \mathsf{who} \vdash (NP^* \backslash NP^*) / (S \backslash NP)^* \rrbracket = \lambda P.\lambda X. X \left(\lambda x. \begin{bmatrix} u \lhd P(\lambda p.px) \\ \hline x \end{bmatrix} \right)$$

Syntactic derivation:

$$\frac{\frac{\mathsf{Kim}}{NP}}{NP^{\star}} \uparrow \frac{\frac{\mathsf{who}}{(NP^{\star} \backslash NP^{\star})/(S \backslash NP)^{\star}} \frac{\mathsf{is} \, \mathsf{away}}{(S \backslash NP)^{\star}}}{\frac{NP^{\star} \backslash NP^{\star}}{NP^{\star}}} < \frac{\mathsf{will} \, \mathsf{return} \, \mathsf{tomorrow}}{(S \backslash NP)^{\star}} > \frac{S^{\star}}{S} \checkmark$$

[+] Semantic composition (contd.)



Formal preliminaries

The next step is to define the behavior of the CI type by prescribing its typing rules.

Since we will be concerened about context updates, we adopt a slightly different notation, where the **typing contexts are explicitly mentioned**.

Example:
$$(\Sigma E_1)$$

$$\frac{M: (x:A) \times B}{\pi_1 M: A} (\Sigma E_1) \quad \mapsto \quad \frac{\Gamma \vdash M: (x:A) \times B}{\Gamma \vdash \pi_1 M: A} (\Sigma E_1)$$

Formal preliminaries (contd.)

The next step is to define the behavior of the CI type by prescribing its inference rules.

Since we will be concerened about context updates, we adopt a slightly different notation, where the **typing contexts are explicitly mentioned**.

CI type: Type checking

First, the formation rule of the CI type is the same as (ΠF) and (ΣF) .

Formation rule of the CI type

$$\frac{\Gamma \vdash A : \mathsf{type} \qquad \Gamma, x : A \vdash B : \mathsf{type}}{\Gamma \vdash (x \lhd A) \times B : \mathsf{type}} \, (\lhd F)$$

Note: the "official" version of $(\lhd F)$, where the @-elimination for A is appropriately considered, will be given later.

CI type: @-elimination

The trick lies in the definition of @-elimination. (or <-elimination, so to speak ...)

@-elimination for the CI type

$$\left[\begin{array}{ccc} \mathcal{D}_A & \mathcal{D}_B \\ \hline \Gamma \vdash A : \mathsf{type} & \Gamma, x : A \vdash B : \mathsf{type} \\ \hline \Gamma \vdash (x \lhd A) \times B : \mathsf{type} \end{array} \right] = \left[\begin{array}{c} \llbracket \mathcal{D}_B \rrbracket \\ \hline \Gamma, x : A \vdash B : \mathsf{type} \end{array} \right]$$

This rule has the effect of "rewriting" the context from Γ to $\Gamma, x:A$.

 \rightarrow Consequently, the CI type **updates the context during** @-elimination (as desired!).

[+] Type checking: some formal details

Since @-elimination [-] may change the context of the given typing judgment, we need to care about this effect in the recursive definitions of the UDTT rules.

$$\frac{\Gamma \vdash A : \mathsf{type} \quad \Gamma, \Delta, x : A' \vdash B : \mathsf{type}}{\Gamma, \Delta \vdash (x \lhd A) \times B : \mathsf{type}} \, (\lhd F) \quad \left(\mathsf{where} \, \left[\begin{array}{c} \mathcal{D}_A \\ \Gamma \vdash A : \mathsf{type} \end{array} \right] = \frac{\llbracket \mathcal{D}_A \rrbracket}{\Gamma, \Delta \vdash A' : \mathsf{type}} \right) \\ & \qquad \qquad \qquad \mathsf{context} \, \mathsf{extension} \, \mathsf{due} \, \mathsf{to} \, A \\ \\ \left[\underbrace{ \begin{array}{c} \mathcal{D}_A & \mathcal{D}_B \\ \Gamma \vdash A : \mathsf{type} & \Gamma, \Delta, x : A' \vdash B : \mathsf{type} \\ \hline \Gamma, \Delta \vdash (x \lhd A) \times B : \mathsf{type} \end{array}}_{\Gamma, \Delta, x : A, \; \Theta} \right] = \underbrace{ \begin{bmatrix} \mathcal{D}_B \rrbracket}{\Gamma, \Delta, x : A, \; \Theta} \vdash B' : \mathsf{type} \\ \mathsf{context} \, \mathsf{extension} \, \mathsf{due} \, \mathsf{to} \, B \\ \end{aligned}}_{\Gamma, \Delta, x : A, \; \Theta}$$

Interim summary

We have introduced the CI type $(x \lhd A) \times B$, which extends the context with A during the specification of the semantic representation.

$$\Gamma \vdash \begin{bmatrix} x \lhd A \\ B \end{bmatrix} : \mathsf{type} \quad \xrightarrow{\mathsf{specification}} \quad \Gamma, \quad x : A \vdash B : \mathsf{type}$$

Account

Preview

So far, we have seen that the CI type predicts ...

• The not-at-issueness of CIs

What remains to see:

- Projection behavior
- · Cross-dimensional anaphora
- Interaction with quantifier scope

Account: projection

(2a) It is not true that [Kim, who is away, will return tomorrow].

The update with this utterance will proceed as follows.

$$\Gamma \vdash \left(v : \begin{bmatrix} u \lhd \mathsf{away}(\mathtt{k}) \\ \mathsf{rtn}(\mathtt{k}) \end{bmatrix}\right) \to \bot : \mathsf{type} \xrightarrow{\mathsf{spec.}} \Gamma, \underbrace{u : \mathsf{away}(\mathtt{k})}_{} \vdash (v : \mathsf{rtn}(\mathtt{k})) \to \bot : \mathsf{type}$$

$$\checkmark \mathsf{projected}$$

$$\frac{\mathsf{app.}}{} \Gamma := \Gamma, \underline{u : \mathsf{away}(\mathtt{k})}, \underline{w : (v : \mathsf{rtn}(\mathtt{k}))}_{} \to \bot$$

The resulting context indeed entails the content of the ARC ("Kim is away").

Cross-dimensional anaphora

(6) Kim, who has $a^i \operatorname{dog}$, takes it_i for a walk.

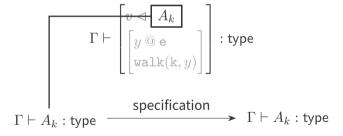
Felicity condition:

$$\Gamma \vdash \begin{bmatrix} v \lhd A_k \\ y @ e \\ \texttt{walk}(\texttt{k}, y) \end{bmatrix} \end{bmatrix} : \mathsf{type} \qquad \left(A_k \equiv \begin{bmatrix} u : \begin{bmatrix} x : e \\ \texttt{dog}(x) \end{bmatrix} \end{bmatrix} \right)$$

Cross-dimensional anaphora (contd.)

(6) Kim, who has $a^i \operatorname{dog}$, takes it_i for a walk.

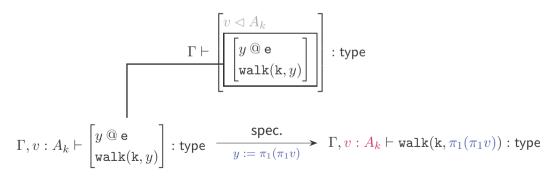
Specification (step 1):



Cross-dimensional anaphora (contd.)

(6) Kim, who has $a^i \log$, takes it, for a walk.

Specification (step 2):



[+] Cross-dimensional anaphora: Order-sensitivity

Cross-dimensional anaphora is sensitive to the order (AnderBois et al., 2015; Elliott & Sudo, 2021).

(12) * Its $_i$ reviewers praised Kim, who had submitted a^i paper on CI.

$$\Gamma \vdash \begin{bmatrix} x @ e \\ v \lhd \begin{bmatrix} u : \begin{bmatrix} y : e \\ paper(y) \end{bmatrix} \\ \dots \\ praise(reviewer(x), k) \end{bmatrix} \end{bmatrix} : \mathsf{type}$$

In this case, the @-type must be eliminated **before the CI is added to the context**, correctly blocking the anaphoric dependency.

Interaction with quantifier scope

(13) Everyⁱ student bid farewell to Nate, who had given them_i some great advice.

Felicity condition:

$$\Gamma \vdash \left(u : \begin{bmatrix} x : \mathbf{e} \\ \mathtt{std}(x) \end{bmatrix}\right) \rightarrow \begin{bmatrix} v \lhd \begin{bmatrix} y @ \mathbf{e} \\ \mathtt{adv}(\mathbf{n}, y) \end{bmatrix} \end{bmatrix} : \mathsf{type}$$

By resolving the @-type (inside the CI type) with $\pi_1 u$, we have ...

$$\left(u: \begin{bmatrix} x : \mathbf{e} \\ \mathbf{std}(x) \end{bmatrix}\right) \to \begin{bmatrix} v \lhd \mathbf{adv}(\mathbf{n}, \boxed{\pi_1 u}) \\ \mathbf{fwl}(\pi_1 u, \mathbf{n}) \end{bmatrix}$$

Π -closure

Issue: A CI type containing local varibleleads leads to a failure of type checking.

$$\Gamma \vdash (u:A) \rightarrow \begin{bmatrix} v \lhd B(u) \\ C \end{bmatrix} : \mathsf{type} \xrightarrow{\quad \mathsf{spec.} \quad} \Gamma, v:B(\underbrace{u}) \vdash (u:A) \rightarrow C : \mathsf{type}$$

Solution: we allow the variables to be bound by the Π -type.

$$\Gamma, f: (u:A) \to B(u) \vdash (u:A) \to C[fu/v]: \mathsf{type}$$

[+] Π -closure: some formal details

We revise the typing rule as follows (the @-elimination remains the same).

$$(\lhd)$$
 (a revised version)

$$\frac{\Gamma \vdash A : \mathsf{type} \quad \Gamma, \Delta, x : A^c \vdash B[xx_0 \cdots x_n/x] : \mathsf{type}}{\Gamma, \Delta \vdash (x \lhd A) \times B : \mathsf{type}} \, (\lhd F)$$

$$\left(\text{where }\left[\left[\begin{array}{c}\mathcal{D}_A\\\Gamma\vdash A:\text{type}\end{array}\right]\right]=\left[\left[\begin{array}{c}\mathbb{D}_A\right]\right]\\\Gamma,\Delta\vdash A':\text{type}\end{array}, \begin{array}{c}A^c=(x_0:A_0)\to\cdots\to(x_n:A_n)\to A'\\(x_i:A_i\in\Gamma\text{ for each }0\le i\le n)\end{array}\right)$$

Note: since this rule may potentially "overgenarate" many readings (with irrelevant $(x_i : A_i) \to \text{ being introduced}$), we need some pragmatic mechanisms to rule them out.

Interaction with quantifier scope (contd.)

(13) Everyⁱ student bid farewell to Nate, who had given them $_i$ some great advice.

"Nate gave every student some advise."

Summary

Q. How can we formally analyze CIs in the framework of DTS?

- We have introduced **the CI type**, which extends the context when eliminated.
 - not-at-isue update = context extension during the specification process
- The CI type can capture some important properties of CIs.
 - o **Projection**: CIs are directly added to the context, escaping the scope of operators.
 - Cross-dimensional anaphora: after added to the context, CIs behave in the same way as the at-issue content.
 - o Interaction with quantifier scope: it can be handled by assuming Π -closure.

Thank you for listening!

daiki.matsuoka@is.s.u-tokyo.ac.jp

For any follow-up questions, please contact me at:

contact file at.

Appendix

Manipulating the context

We have a slight technical issue when a CI type appears in the scope of some operators.

Example: $(x:A) \rightarrow ((y \triangleleft B) \times C)$

$$\frac{\Gamma \vdash A : \mathsf{type} \qquad \Gamma, x : A \vdash (y \lhd B) \times C : \mathsf{type}}{\Gamma \vdash (x : A) \to ((y \lhd B) \times C) : \mathsf{type}} \, (\Pi F)$$

After eliminating \lhd , we cannot put back the local variable x:A since it is not at the right edge of the context.

Manipulating the context (contd.)

To avoid this problem, we need to use weakening and permutation.

$$\frac{\Gamma \vdash A : \mathsf{type} \qquad \Gamma \vdash B : \mathsf{type}}{\Gamma, y : B \vdash A : \mathsf{type}} \xrightarrow{(\mathsf{wk})} \frac{\Gamma, x : A, y : B \vdash C : \mathsf{type}}{\Gamma, y : B, x : A \vdash C : \mathsf{type}} \xrightarrow{(\mathsf{Trred})} \frac{\Gamma, y : B \vdash (x : A) \to C : \mathsf{type}}{(\Pi F)}$$

Details of the derivation: projection

(2a) It is not true that [Kim, who is away, will return tomorrow].

Felicity condition:

$$\Gamma dash \left(v: egin{bmatrix} u \lhd \mathtt{away} \ \mathtt{k} \\ \mathtt{rtn} \ \mathtt{k} \end{bmatrix}
ight)
ightarrow oldsymbol{oldsymbol{oldsymbol{\mathsf{k}}}} = \mathtt{type}$$

(2a) It is not true that [Kim, who is away, will return tomorrow].

Type checking:

$$rac{\Gamma dash \left[egin{array}{c} u \lhd \mathtt{away} \ \mathtt{k} \\ \mathtt{rtn} \ \mathtt{k} \end{array}
ight] : \mathsf{type} }{\Gamma dash \left(v : \left[egin{array}{c} u \lhd \mathtt{away} \ \mathtt{k} \\ \mathtt{rtn} \ \mathtt{k} \end{array}
ight]
ight)
ightarrow ot : \mathsf{type} }$$

(2a) It is not true that [Kim, who is away, will return tomorrow].

Type checking:

$$\mathcal{D}_{away}$$
 \mathcal{D}_{return} $\Gamma dash ext{away } ext{k} : ext{type}$ $\Gamma, u : ext{away } ext{k} dash ext{rtn } ext{k} : ext{type}$ $\Gamma dash \left[egin{array}{c} u \lhd ext{away } ext{k} \\ ext{rtn } ext{k} \end{array}
ight] : ext{type}$ $\Gamma dash \left[v : egin{array}{c} u \lhd ext{away } ext{k} \\ ext{rtn } ext{k} \end{array}
ight]
ight)
ightarrow ota : ext{type}$

(2a) It is not true that [Kim, who is away, will return tomorrow].

Type checking:

$$\frac{\mathcal{D}_{A}}{\Gamma \vdash \begin{bmatrix} u \lhd \mathsf{away} \, \mathsf{k} \\ \mathsf{rtn} \, \mathsf{k} \end{bmatrix} : \mathsf{type} \qquad \Gamma, u : \mathsf{away} \, \mathsf{k}, v : \mathsf{rtn} \, \mathsf{k} \vdash \bot : \mathsf{type}}{\Gamma \vdash \left(v : \begin{bmatrix} u \lhd \mathsf{away} \, \mathsf{k} \\ \mathsf{rtn} \, \mathsf{k} \end{bmatrix} \right) \to \bot : \mathsf{type}}$$

$$\left(\because \quad \llbracket \mathcal{D}_{A} \rrbracket = \quad \mathcal{D}_{return} \\ \Gamma, u : \mathsf{away} \, \mathsf{k} \vdash \mathsf{rtn} \, \mathsf{k} : \mathsf{type} \right)$$

(2a) It is not true that [Kim, who is away, will return tomorrow].

Applying @-elimination to the whole derivation ...

$$\begin{split} & \llbracket \mathcal{D}_A \rrbracket & \llbracket \mathcal{D}_\bot \rrbracket \\ & \frac{\Gamma, u : \mathsf{away} \, \mathsf{k} \vdash \mathsf{rtn} \, \mathsf{k} : \mathsf{type} \qquad \Gamma, u : \mathsf{away} \, \mathsf{k}, v : \mathsf{rtn} \, \mathsf{k} \vdash \bot : \mathsf{type} }{\Gamma, u : \mathsf{away} \, \mathsf{k} \vdash (v : \mathsf{rtn} \, \mathsf{k}) \to \bot : \mathsf{type}} \end{split}$$

References i

- Amaral, P., Roberts, C., & Smith, E. A. (2007). Review of the logic of conventional implicatures by chris potts. Linguistics and Philosophy, 30(6), 707–749. https://doi.org/10.1007/s10988-008-9025-2
- AnderBois, S., Brasoveanu, A., & Henderson, R. (2015). At-issue Proposals and Appositive Impositions in Discourse.

 <u>Journal of Semantics</u>, 32(1), 93–138. https://doi.org/10.1093/jos/fft014
- Bekki, D., & McCready, E. (2015). CI via DTS. In T. Murata, K. Mineshima, & D. Bekki (Eds.),

 New frontiers in artificial intelligence (pp. 23–36). Springer. https://doi.org/10.1007/978-3-662-48119-6_3
- Elliott, P. D., & Sudo, Y. (2021). Generalised Crossover. Semantics and Linguistic Theory, 30, 396–408. https://doi.org/10.3765/salt.v30i0.4841
- Koev, T. (2018). Notions of at-issueness. <u>Language and Linguistics Compass</u>, <u>12</u>(12), e12306. https://doi.org/10.1111/lnc3.12306
- $\textbf{Martin, S. (2016). Supplemental update.} \underline{\textbf{Semantics and Pragmatics}}, \underline{9} (5), 1-61. \ \text{https://doi.org/10.3765/sp.9.5}$
- Matsuoka, D., Bekki, D., & Yanaka, H. (2024). Appositive projection as implicit context extension in Dependent Type Semantics. In D. Bekki, K. Mineshima, & E. McCready (Eds.),

 Logic and engineering of natural language semantics (pp. 224–243). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-60878-0_13
- McCready, E. (2010). Varieties of conventional implicature. $\underline{Semantics\ and\ Pragmatics}, \underline{3}(8), 1-57. \\ https://doi.org/10.3765/sp.3.8$

References ii

- Murray, S. E. (2014). Varieties of update. Semantics and Pragmatics, 7(2), 1–53. https://doi.org/10.3765/sp.7.2
- Nouwen, R. (2007). On appositives and dynamic binding. Research on Language and Computation, $\underline{5}(1)$, 87–102. https://doi.org/10.1007/s11168-006-9019-6
- Potts, C. (2005). The logic of conventional implicatures. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199273829.001.0001
- Schlenker, P. (2020). The semantics and pragmatics of appositives. In <u>The wiley blackwell companion to semantics</u> (pp. 1–33). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118788516.sem110
- Simons, M., Tonhauser, J., Beaver, D., & Roberts, C. (2010). What projects and why. Semantics and linguistic theory, 309–327. https://doi.org/10.3765/salt.v20i0.2584
- Tonhauser, J. (2012). Diagnosing (not-)at-issue content.

 Proceedings of Semantics of Under-represented Languages of the Americas 6.
- Zhao, Z. (2023). The scope of supplements. Proceedings of Sinn und Bedeutung 27. https://doi.org/10.18148/sub/2023.v27.1099