Appendix:
Integrating 'Inside Out' with broader theories of local context computation

As announced in Section **Error! Reference source not found.**, we provide as a 'proof of concept' an algorithm that integrates 'inside out' order of evaluation for modifiers with different orders for other constructions. We will specify a procedure that takes as input a sentence with a distinguished position (written as _), and returns the set of alternatives that must obey the equivalence in **Error! Reference source not found.**, thus yielding the value of the local context of _ .

Due to the complexity of the problem, it is important to make things as simple as possible by investigating a tiny fragment (as is done in all studies that seek to reconstruct local contexts in a predictive fashion). We thus start from the fragment in (1), which generates sentences of the form *(p_1 and p_2)*, *(_ and p_2)* (when we wish to assess the local context of _), but also quantified statements with a main predicate *F*, of the form *(Q F)*. Here *Q* stands for a quantified noun phrase of the form *someone, some journalist, every student,* etc. (proper names can also be thought of as special cases of generalized quantifiers)[1]. We do not seek to decompose *Q* further because we will be solely interested in NPs that appear in predicative (i.e. verbal) position. We will also consider modified predicates of the form *(P_1 P_2)*, where P_2 corresponds to a nominal predicate and P_1 to its modifier. It is important that nominal modifiers are preposed, as in English and Chinese, because our goal is to determine how 'inside out' (for NPs) and linear-looking orders of evaluation (for conjunction) can be combined when they 'pull' in different directions. For this reason, we will also consider quantified statements of the form *(Q (P_1 P_2))*, where *(P_1 P_2)* is a modified predicate – e.g. *Someone is a female engineer*. We will also consider formulas for the form *(Q (_ P_1))*, e.g. when we wish to assess the local context of an intersective modifier added to P_1.

(1) a. *Vocabulary:* we consider a highly simplified fragment with (i) propositional and predicative constants p_i, P_i (ii) unanalyzed quantificational Noun Phrases Q_i, (iii) predicative and propositional conjunctions written as *and*. To specify the position whose local context is computed, the symbol _ is added to the set of propositional and to the set of predicative expressions.
b. *Syntax:*
If f and g are propositional expressions, so is *(f and g)*.
If *F* and *G* are predicative expressions, so are *(F and G)* and *(F G)*
If *F* is a predicative expression, *(Q F)* is a propositional expression.

We then consider a formula with _ in a certain position, e.g. *(_ and p_1)*, and we gradually replace expressions with variables of the same type when the information these expressions provide is not accessible in the computation of the local context. For notational simplicity, we use primed expressions – e.g. p'_1, P'_3 as variables added to our official language to obtain this result, with the same types as the unprimed expressions. In our example, then, the goal is to transform *(_ and p_1)* into *(_ and p'_1)*, thus indicating that the information about the second conjunct is not accessible in the computation of the local context of the first conjunct.

But how can this transformation be achieved in a systematic fashion? We will transform an entire formula step by step. Re-write will be done constituent by constituent, marking an expression with $_m$ after it has been treated in the appropriate fashion. In the initial stage, we always have $_m$ on _, the position whose local context we are computing. The goal is then to move $_m$ to the outermost position in the sentence by way of an iteration of the rewrite rules in (2). Importantly, these rules are just the compact specification of a useful algorithm; they have nothing to do with rules used in syntax to indicate how constituents are generated.

(2) Permissible transformations (we use lowercase letters for propositional expressions and uppercase letters for predicative expressions, and expressions like *g'*, *G'* for 'fresh' propositional or predicative variables, e.g. p_1', P_1')
1. (f_m and g) → (f and g')$_m$
2. (f and g_m) → (f and g)$_m$

---

[1] In generalized quantifier theory, *Sam* can be given a quantifier type <et, t>. If s is the individual denoted by *Sam*, the generalized quantifier value of the proper name can be defined as: $\mathbf{Sam}^w = \lambda f_{<e,t>}.\ f(s) = 1$.

3. $\mathbf{(F_m \text{ and } G) \to (F \text{ and } G')_m}$
4. $(F \text{ and } G_m) \to (F \text{ and } G)_m$
5. $\mathbf{(F_m\ G) \to (F\ G)_m}$
6. $\mathbf{(F\ G_m) \to (F'\ G)_m}$
7. $(Q\ G_m) \to (Q\ G)_m$

The boldfaced rules in (2)3,5,6 require a comment. First, despite the similarities in their semantics, predicate conjunction and predicate modification don't have the same rewrite effects: (2)3 specifies that the first conjunct doesn't have access to information in the second, while (2)5 specifies that a preposed predicate modifier *does* have access to the predicate it modifies. Second, (2)6 specifies that a predicate *G* modified by a preposed predicate *F* does *not* have access to the value of *F*, which is essential to account for the acceptability of *a pregnant woman*, for instance.

The procedure then works as follows. We start from a sentence with a distinguished position $\__m$ to indicate which local context we are computing. We then gradually replace the necessary expressions with variables, moving the $_m$ around in accordance with the rules in (2). To facilitate cross-reference to relevant rules in derivations, we superscript $\to$ with a number corresponding to the rule from (2) which is invoked (e.g. $\to^3$ indicates that the rewrite is permitted by rule (2)3). When the $_m$ is in outermost position, we use the formula (without $_m$) to tell us which alternatives must satisfy the equivalence in **Error! Reference source not found.**.[2]

Crucial cases are derived in (3). To illustrate, it might help to say in words about what (3)a does. It seeks to compute the local context of the first conjunction, marked as _, in a formula $(\_ \text{ and } p_1)$. It does so by starting with a version of the formula with the position _ marked with $_m$, hence: $(\__m \text{ and } p_1)$. Then it pushes $_m$ to the outermost position by rewriting this formula as $(\_ \text{ and } p_1')_m$ thanks to rule (2)1. Replacing the distinguished position _ with *d'* or with *(c' and d')*, we finally require that the local context **c'** should be the strongest value x which, for all appropriate *d'* and p_1', satisfies relative to C the equivalence: $((c' \text{ and } d') \text{ and } p_1') \Leftrightarrow (d' \text{ and } p_1')$.

(3) a. $(\__m \text{ and } p_1) \to^1 (\_ \text{ and } p_1')_m$ and thus we require that for all d', p_1',
$C \vDash^{c'\to x} ((c' \text{ and } d') \text{ and } p_1') \Leftrightarrow (d' \text{ and } p_1')$.

b. $(Q\ (\__m\ P_2)) \to^5 (Q\ (\_\ P_2)_m) \to^7 (Q\ (\_P_2))_m$, hence to compute the local context of a modifier of P_2 we will require that for all d',
$C \vDash^{c'\to x} (Q\ ((c' \text{ and } d')\ P_2)) \Leftrightarrow (Q\ (d'\ P_2))$

c. The same result is obtained if P_2 is replaced with a modified Noun Phrase $P_3(P_2)$: the final condition will ensure that the local context of a further modifier, starting from $(Q\ (\__m\ P_3(P_2)))$, is computed in a way that ensures access to both innermost predicates, thanks to the equivalence:
$C \vDash^{c'\to x} (Q\ ((c' \text{ and } d')(P_3P_2))) \Leftrightarrow (Q\ (d'\ (P_3P_2)))$

d. By contrast, the local context of a modified Noun Phrase P_2 in P_1P_2 does not have access to the value of the modifier:
$(Q\ (P_1\ \__m)) \to^6 (Q\ (P_1'\ \_)_m) \to^7 (Q\ (P_1'\ \_))_m$, and we will require that for all d', P_1',
$C \vDash^{c'\to x} (Q\ (P_1'\ (c' \text{ and } d'))) \Leftrightarrow (Q\ (P_1'\ d'))$

e. When we compute the local context of a modifier in a first conjunct, we take into account part of the information that linearly follows, namely that pertaining to the modifie Noun Phrase, but we don't take into account the second conjunct:
$((Q\ (\__m\ P_2)) \text{ and } p_1) \to^5 ((Q\ (\_\ P_2)_m) \text{ and } p_1) \to^7 ((Q\ (\_P_2))_m \text{ and } p_1) \to^3 ((Q\ (\_P_2)) \text{ and } p_1')_m$, hence a requirement that for all d', p_1',
$C \vDash^{c'\to x} ((Q\ ((c' \text{ and } d')\ P_2)) \text{ and } p_1') \Leftrightarrow ((Q\ (d'\ P_2)) \text{ and } p_1')$

---

[2] More formally: If *(… d …)* is a sentence of the fragment in (1), and if $(\ldots \__m \ldots)$ can be rewritten into another formula $(\bullet\bullet\bullet\ \_\ \bullet\bullet\bullet)_m$ in accordance with (2), with (primed, lowercase or uppercase) variables $v'_1 \ldots v'_n$, the local context of *d* in *(… d …)* relative to a context C is the strongest proposition or property x which guarantees that for all *d'* of the same type as *d*, for all expressions $v'_1 \ldots v'_n$ (of the appropriate types):

$C \vDash^{c'\to x} (\bullet\bullet\bullet(c' \text{ and } d')\ \bullet\bullet\bullet) \Leftrightarrow (\bullet\bullet\bullet\ d'\ \bullet\bullet\bullet)$

f. When a conjunction *(P_1 and P_2)* modifies a Noun Phrase P_3 in the structure *((P_1 and P_2) P_3)*, the local context of P_1 has access to P_3 but not to P_2:
$(Q\ ((\_{}_m \text{ and } P_2)\ P_3)) \rightarrow^{1} (Q\ ((\_ \text{ and } P_2')_m\ P_3)) \rightarrow^{5} (Q\ ((\_\text{and } P_2')\ P_3)_m) \rightarrow^{7} (Q\ ((\_\text{and } P_2')\ P_3))_m$, hence a requirement that for all d', P_2',
$C \vDash^{\rightarrow_x} (Q\ (((c' \text{ and } d') \text{ and } P_2')\ P_3)) \Leftrightarrow (Q\ ((d' \text{ and } P_2')\ P_3))$

g. On the other hand, in the same structure *((P_1 and P_2) P_3)*, the local context of P_2 has access to both P_3 and P_1:
$(Q\ ((P_1 \text{and } \_{}_m)\ P_3)) \rightarrow^{2} (Q\ ((P_1 \text{and } \_)_m\ P_3)) \rightarrow^{5} (Q\ ((P_1 \text{and } \_)\ P_3)_m) \rightarrow^{7} (Q\ ((P_1 \text{and } \_)\ P_3))_m$, hence a requirement that for all d',
$C \vDash^{\rightarrow_x} (Q\ ((P_1 \text{and } (c' \text{ and } d'))\ P_3)) \Leftrightarrow (Q\ ((P_1 \text{and } d')\ P_3))$

Examples (3)a-d are unsurprising since they just encode old or new generalizations: the second conjunct isn't accessed in the computation of the local context of the first conjunct (= (3)a); the local context of a higher preposed modifier has access a lower noun or noun phrase (= (3)b,c); and the local context of a noun doesn't have access to preposed modifiers (= (3)d).

(3)e highlights the interplay between different notions of order: in computing the local context of the modifier embedded in the first conjunct, we have access to the modified Noun Phrase but not to the second conjunct, although both follow the modifier. (3)f,g present new benefits of the analysis: the system correctly predicts that in a modified Noun Phrase of the form *((P_1 and P_2) P_3),* the local context of P_2 has access to P_1 and P_3, whereas the local context of P_1 has access to P_3 but not to P_2, despite the fact that both appear to its right. This was exactly the challenge we announced in Section **Error! Reference source not found.** in relation to **Error! Reference source not found.** and **Error! Reference source not found.** (= **Error! Reference source not found.**a,b and **Error! Reference source not found.**a,b respectively).[3]

In sum, we have provided a 'proof of concept' for a procedure that integrates the 'insider out' order of evaluation for NP modifiers with linear-looking orders for conjunctions. We had to provide this integration because as things stand no more general notion of order appears to work for both construction types – a situation that ought to be remedied in the future.

---

[3] We also predict that reversing the order of the conjunctions in **Error! Reference source not found.** should yield related contrasts. One consultant helpfully re-rated **Error! Reference source not found.** and provided comparative judgments with (i). Contrasts go in the same direction: **Error! Reference source not found.**a,b,c,d are rated as 6, 2, 7, 7, while with the order of the conjuncts reversed as in (i), ratings are 5, 1, 7, 7.

(i) Under Nazi occupation, this heroic family hid…
a. ^{5} a dark-skinned and Hasidic French Jew.
b. ^{1} a dark-skinned and Jewish French Hasid.
c. ^{7} a dark-skinned and Hasidic French child.
d. ^{7} a dark-skinned and Jewish French child.