Last Resort is a Global Economy Condition

Author: Daniel Gallagher

Date: February 24, 2025

1 A Global or Local Economy Condition?

In Collins 2001, a number of economy conditions are outlined which aim to formalise how syntactic "operations, derivations and representations" are subject to constraints which determine their optimality. The underlying assumption here is that syntactic derivations should be as efficient and economical as they can be, therefore aiming to reduce unnecessary complexity. One such economy condition is Last Resort, defined in (1).

(1) "An operation OP may apply only if the derivation would otherwise result in an ungrammatical representation (at PF or LF)." (Collins 2001: p. 46)

In Section 5, Collins points out that there is often not a clear distinction between what constitutes a *local* and *global* optmality condition, positing the question *"Is the general form of Last Resort in (1) a global principle?"*. He complements this question with the issue of whether the spell-out of the trace of resumptive verbs in a predicate cleft construction makes reference to global information or not (e.g "Do, Kofi will **do** it"). In these cases, it is claimed that the grammaticality of the entire structure must be checked in order to determine whether the spell-out of the resumptive verb ought to take place.

The local versus global distinction is commonly defined in terms of the search space that is made available to the optimiser. A local condition will deal with a more limited area of the given structure while a global condition may require access to the search space as a whole. Local conditions are therefore more desirable due to their need for less computational power. Take the *compiler* for instance from the field of computer science, which is the component responsible for "understanding" computer code and converting it into a lower-level representation. In the case of local optimisation we may have *peephole* optimisation. This involves choosing a subset of the operations carried out inside the compiler and optimising using only this subset, without the need to see the program as a whole. This can be used to find small redundancies in the code such as an operation being applied such that $A \to B \to A$ as in a Duke of York derivation. On the other hand, these compilers also use the global method of *dead code elimination*. In this case, the entire program is available to the optimiser which searches for code that can be proven to be impossible to reach or never has any effect on the output of the program. In this case it is necessary to have a wider lens over the source code in order to identify code which may never be reached due to the design of the overall program structure. In the local case, there is a limited search space in which the optimiser can move in order to satisfy the economy conditions, while in the latter case the final program is available as a whole to the optimiser and an unknown number of steps may be run before it is known whether an economy condition is satisfied. Drawing lessons from how these optimisation types are defined in other fields, we may therefore define them for syntactic optimisation as in (2) and (3).

(2) local: An economy condition is local if, at any given step of the derivation where the syntactic structure is a representation R_i after *i* operations have been applied, only information provided by the current representation R_i is required to determine whether an operation should apply or not to satisfy that economy condition.

(3) global: An economy condition is global if, at any given step of the derivation where the syntactic structure is a representation R_i after *i* operations have been applied, a potentially unknown number of further operations *j* may be applied such that representation R_{i+j} is required in order to determine whether an operation should apply to R_i to satisfy that economy condition.

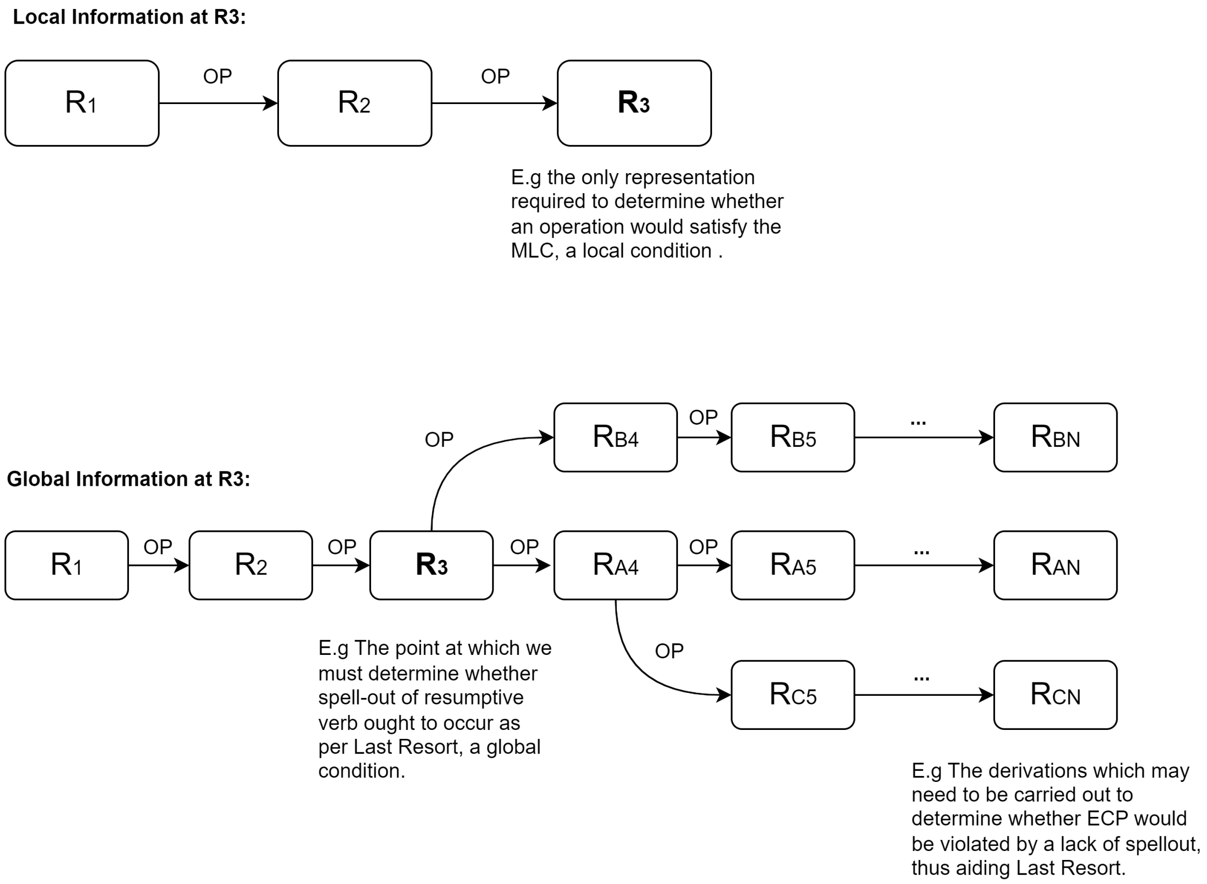
Local Information at R3:
R1
OP
R2
OP
R3
E.g the only representation required to determine whether an operation would satisfy the MLC, a local condition .
Global Information at R3:
R1
OP
R2
OP
R3
OP
RA4
OP
RA5
...
RAN
OP
RB4
OP
RB5
...
RBN
OP
RC5
...
RCN
E.g The point at which we must determine whether spell-out of resumptive verb ought to occur as per Last Resort, a global condition.
E.g The derivations which may need to be carried out to determine whether ECP would be violated by a lack of spellout, thus aiding Last Resort.

Figure 1: Representations Required for Local vs Global Conditions

This draws a clearer distinction between those economy conditions which work only with the current representation R_i, and those which must apply further operations to compute further representations $\{R_{i+1}, R_{i+2}, R_{i+3}, ...\}$. In Figure 1 an example is shown of what information local versus global optimality conditions have access to at a given point in the derivation, in this case where a number of operations have been carried out and the current syntactic structure is represented by R_3. Checking the grammaticality of a potential final syntactic structure means that an economy condition is global, as this requires a potential final representation R_N derived through the application of N operations. Referring back to Last Resort in (1), we note that a final representation R_N must be derived in order to evaluate whether the lack of an operation OP would produce a grammatical error in R_N. In the case mentioned above for the spell-out of resumptive verbs in a predicate cleft construction, it can therefore be determined that the evaluation of whether the ECP condition would be violated by the lack of trace spell-out constitutes global optimisation. This is because the potential final representation R_N would need to be derived in order to determine whether the syntactic structure without spell-out of the resumptive verb would be grammatical or not, and this is then used to determine whether the spell-out of this trace should take place. LAST RESORT can hence be slotted in with SHORTEST DERIVATION and PROCRASTINATE as global economy conditions. It also predicts that ASAP and MINIMALITY conditions ought to be defined as local. These results are summarised in Table 1.

Economy Condition	**Scope**	**Reason**
ASAP	Local	Requires only current derivation to know whether requirement can be fulfilled immediately e.g case checking.
Shortest Derivation	Global	Requires alternative derivations to be calculated.
Minimality	Local	Distance calculations for movement need only current representation to be calculated.
Procrastinate	Global	Avoiding movement until you have to can be carried out one representation at a time.
Last Resort	Global	Requires the potential final representation to be calculated.

Table 1: Scope of Various Economy Conditions

References

Collins, Chris. 2001. Economy Conditions in Syntax. In 45–61. https://doi.org/10.1002/9780470756416.ch2.