

## Dynamic representations of stack- and queue-like syntactic structures

André Grüning

Dept. of Psychology, Univ. of Warwick, UK

e-mail: a.gruening@warwick.ac.uk

Stack-like syntactic structures have been found to be far more frequent in natural languages than queue-like ones. Typical stack-like structures are palindromes *abccba* which occur naturally when one marks grammatical dependencies in e. g. nested relative clauses by matching parentheses:

(<sub>a</sub> The mouse (<sub>b</sub> the cat (<sub>c</sub> the dog teased )<sub>c</sub> caught )<sub>b</sub> escaped. )<sub>a</sub>

The most prominent example of similar queue-like structures *abcabc*, a duplication, is the ordering of infinite verbs in Dutch, abusing brackets in marking matching constituents:

[<sub>a</sub>Aad heft [<sub>b</sub> Jantje [<sub>c</sub> de lerare de knikkers laten ]<sub>a</sub> helpen ]<sub>b</sub> opruimen ]<sub>c</sub>  
*Aad has Jantje the teacher the marbles let help collect*  
'Aad let Jantje help the teacher to collect the marbles.'

The traditional explanation of the different frequencies of stack-like and queue-like syntactic structures relies on a symbolic computing view of human language processing and uses the Chomsky hierarchy: stack-like structures belong to the class of context-free languages while queue-like ones need context-sensitive grammars. Thus processing queue-like structures is computationally more difficult.

Yet, within the framework of connectionism human cognition has been described using subsymbolic processes as in neural networks and dynamical systems. Since the "hardware" in the connectionist framework is rather an analogue computer than a symbolic one it is not obvious whether the argumentation with a language's rank in the Chomsky hierarchy applies here.

Therefore we set up simulations with simple recurrent networks (SRN) trained on binary palindrome and duplication languages, respectively: the copy language was more difficult to acquire mirroring the lower frequency of queue-like structures in natural languages.

In order to find an explanation of this fact, we analysed the networks' emerging dynamics and found that successful networks represented their language as simple dynamical systems in two dynamical variables: while these variables essentially evolve independently for the palindrome language, their dynamics were coupled for the duplication language, rendering the dynamical system more complex and thus harder to implement in a SRN.

In conclusion, we have analysed how SRN represent and process stack-like and queue-like structures and found an explanation why queue-like structures are harder to process even in an analogue computing or connectionist setting.