Constraints on System Entrance in a Systemic Functional Generator

Víctor M. Castel

Consejo Nacional de Investigaciones Científicas y Técnicas
InCiHuSA. Avda. Ruiz Leal s/n, 5500 Mendoza, Argentina
vcastel@mendoza-conicet.gob.ar

Abstract. The current system network of the Cardiff Grammar generator GeneSys is defined by implications of the form ‘p $\rightarrow$ q’, where ‘p’ is a (disjunction or conjunction of) feature(s), and ‘q’ is an instruction to incorporate a (conjunction of) feature(s) into the selection expression under construction. The graphs generated by GeneSys reveal that there is not a strict separation between the argument that is intended to be more delicately specified by the features in ‘q’, and the conditions under which such a specification is to be carried out. This paper remedies this problem by eliminating system network conjunctive conditions altogether in favour of single entry conditions through a relocation in preference rules on relevant features of the constraining conjuncts, typically the conditioning feature(s) of the old conjunctive conditions.

Keywords: Text Generation, Cardiff Grammar Generator, Systemic Functional Grammar, System Networks, Preference Rules, Conjunctive Entry Conditions.

1 Introduction

GeneSys [7, 8, 9] is an implementation of the Cardiff Grammar (CG) in its generation oriented dimension [3, 6, 7]. CG is organized into a semantic component, defining the meaning potential of a language, and a form component, defining its associated linguistic expression potential. The semantic component is a set of System Network Rules (SNRs), and Same Pass Preference Resetting Rules (PRs), and the form component is a set of Realization Rules (RRs). SNRs and PRs jointly construct selection expression graphs that feed RRs. The task of RRs is to define form representations, i.e. graphs which account for morpho-syntactic, lexical, and punctuational (or intonational) properties of linguistic units realizing a given selection expression graph. This basic organization of CG is captured in figure 1.

This paper addresses an aspect of the organization of the semantic component of CG as defined for the micro- and mini-grammars of English [4-5]. It focuses on the effects on graph construction of a specific subtype of SNR, namely: conjunctive entry condition rules. The purpose of the paper is to show that this type of SNR does not capture transparently the relationship between a feature and the more delicate specification offered by the system of which it is a candidate argument. In section 2, this problematic relationship is illustrated with examples of conjunctive condition SNRs extracted from the micro- and mini-grammars of English, and a solution is suggested which is simpler and does not require an extension of the existing theoretical apparatus. In section 3, the general properties of the solution proposed in section 2 are described. Finally, in section 4 a general conclusion is reached and future work is oulined.

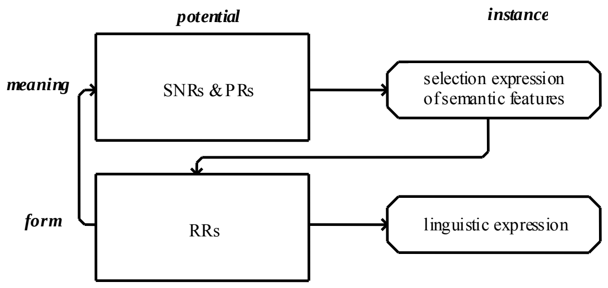

Fig. 1. Main components and output representations of CG. Figure extracted from [4] with minor adjustments. See note 1.

2 Constraints on system entrance

There are two subtypes of SNR: (i) single feature condition rules, as illustrated by figure 2, and (ii) complex entry condition rules, as illustrated by figures 3-5, extracted from [4]:

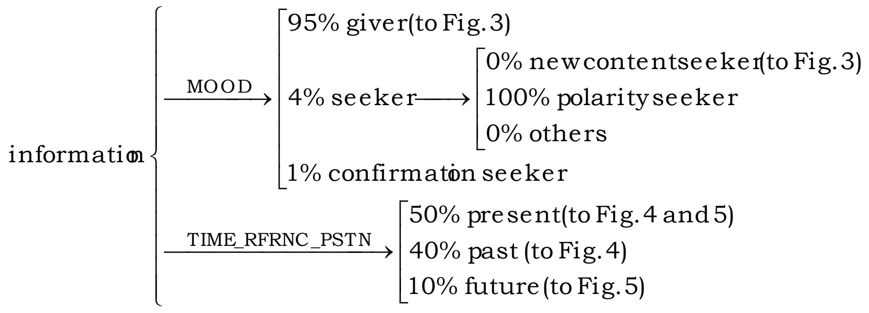

Fig. 2.

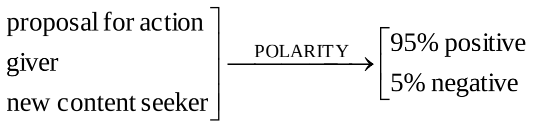

Fig. 3.

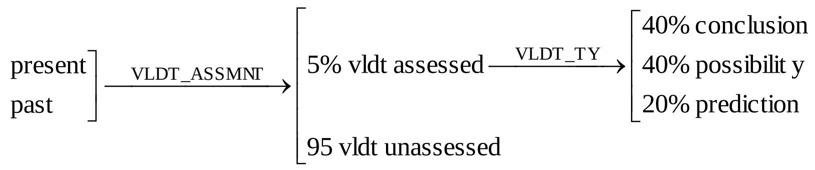

Fig. 4.

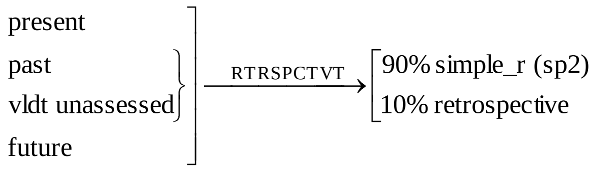

Fig. 5.

Key: TIME_RFRNC_PSTN = TIME_REFERENCE_POSITION; VLDT_ASSMNT = VALIDITY_ASSESSMENT; vldt = validity; VLDT_TY = VALIDITY_TYPE; RTRSPCTVT = RETROSPECTIVITY.

In the micro-grammar of English [4], as implemented in GeneSys [7, 8, 9], the graphic definitions given in figures 2-5 are formulated in a linear fashion as in (1-5), (6-7), (8-11), and (12-13), respectively:

$$\text{information} \rightarrow \text{MOOD} \wedge \text{TIME\_RFRNC\_PSTN}\ . \tag{1}$$

$$\text{MOOD} \rightarrow 95\%\ \text{giver} \vee 4\%\ \text{seeker} \vee 1\%\ \text{confirmation seeker}\ . \tag{2}$$

$$\text{seeker} \rightarrow \text{SKR\_TYPE}\ . \tag{3}$$

$$\text{SKR\_TYPE} \rightarrow 0\%\ \text{new content seeker} \vee 100\%\ \text{polarity seeker}\ . \tag{4}$$

$$\text{TIME\_RFRNC\_PSTN} \rightarrow 50\%\ \text{present} \vee 40\%\ \text{past} \vee 10\%\ \text{future}\ . \tag{5}$$

$$\text{proposal for action} \vee \text{giver} \vee \text{new content seeker} \rightarrow \text{POLARITY}\ . \tag{6}$$

$$\text{POLARITY} \rightarrow 95\%\ \text{positive} \vee 5\%\ \text{negative}\ . \tag{7}$$

$$\text{present} \vee \text{past} \rightarrow \text{VLDT\_ASSMNT}\ . \tag{8}$$

$$\text{VLDT\_ASSMN} \rightarrow 5\%\ \text{vldt assessed} \vee 95\%\ \text{vldt unassessed}\ . \tag{9}$$

$$\text{vldt assessed} \rightarrow \text{VLDT\_TY}\ . \tag{10}$$

$$\text{VLDT\_TY} \rightarrow 40\%\ \text{conclusion} \vee 40\%\ \text{possibility} \vee 20\%\ \text{prediction}\ . \tag{11}$$

$$\text{present} \vee (\text{past} \wedge \text{vldt unassessed}) \vee \text{future} \rightarrow \text{RTRSPCTVT}\ . \tag{12}$$

$$\text{RTRSPCTVT} \rightarrow 90\%\ \text{simple\_r (sp2)} \vee 10\%\ \text{retrospective}\ . \tag{13}$$

$$\text{proposal for action} \rightarrow \text{POLARITY}\ . \tag{14}$$

$$\text{giver} \rightarrow \text{POLARITY}\ . \tag{15}$$

$$\text{new content seeker} \rightarrow \text{POLARITY}\ . \tag{16}$$

$$\text{present} \rightarrow \text{VLDT\_ASSMN} \wedge \text{RTRSPCTVT}\ . \tag{17}$$

$$\text{past} \rightarrow \text{VLDT\_ASSMN}\ . \tag{18}$$

$$\text{past} \wedge \text{vldt unassessed} \rightarrow \text{RTRSPCTVT}\ . \tag{19}$$

$$\text{future} \rightarrow \text{RTRSPCTVT}\ . \tag{20}$$

Both the left hand side of the system names in figures 3-5, and the left hand side of the equivalent formulas in (6), (8) and (12) are abbreviations of the more simple definitions given in (14-16), (17-18), and (19-20), respectively:

2.1 System entrance constraints as complex condition SNRs

RTRSPCTVT applies to 'present' and 'future' in absolute terms, i.e. independently of the value chosen from VLDT_ASSMNT, as in the case of selection expressions containing 'present' (cf. (1-8) below, or even if the system VLDT_ASSMNT does not apply at all, as in the case of selection expressions containing 'future' (cf. (9-10) below). In the (partial) selection expressions associated with each of the following sentences (cf. the features between square brackets), the options defined by RTRSPCTVT are underlined.[1]

1. Ike is kicking Victoria.
   [present, vldt unassessed, simple_r]
2. Ike has kicked Victoria.
   [present, vldt unassessed, retrospective]
3. Ike must be kicking Victoria.
   [present, vldt assessed, conclusion, simple_r]
4. Ike may be kicking Victoria.
   [present, vldt assessed, possibility, simple_r]
5. Ike will be kicking Victoria.
   [present, vldt assessed, prediction, simple_r]
6. Ike must have kicked Victoria.
   [present, vldt assessed, conclusion, retrospective]
7. Ike may have kicked Victoria.
   [present, vldt assessed, possibility, retrospective]
8. Ike will have kicked Victoria.
   [present, vldt assessed, prediction, retrospective]
9. Ike will kick Victoria.
   [future, simple_r]
10. Ike will have kicked Victoria.
    [future, retrospective]
11. Ike kicked Victoria.
    [past, vldt unassessed, simple_r]
12. Ike had kicked Victoria.
    [past, vldt unassessed, retrospective]
13. Ike must have kicked Victoria.
    [past, vldt assessed, conclusion]
14. Ike may have kicked Victoria.
    [past, vldt assessed, possibility]
15. Ike will have kicked Victoria.
    [past, vldt assessed, prediction]

Note, however, that RTRSPCTVT applies to 'past' only if the feature 'vldt unassessed' has been previously chosen from VLDT_ASSMNT; i.e. the feature 'past' is more delicately specified by RTRSPCTVT only if the selection expression under construction contains also the feature 'vldt unassessed' (cf. sentences (11) and (12)).

[1] Space limitations refrain us from providing complete selection expression graphs from which RRs derive the sentences in (1-15). In this respect, both the bracketed selection expressions and the associated sentences are simplifications of complex graphs. The sentences in (1-15) result from a stripping process whose function is to eliminate semantic and form structure so that only a final string of characters is shown in the output (cf. figure 1).

If the feature 'vldt assessed' were chosen, the system RTRSPCTVT would not subcategorize 'past', i.e. the system could not be accessed and therefore the options 'simple_r' and 'restrospectivity', would not be available. In such a case, the sentences in (13-15) would be generated.

The purpose of a conjunctive condition SNR like (19) is to restrict 'past' from entering the system RTRSPCTVT when it co-occurs with 'vldt assessed'. But a problem arises when one draws graphs that capture the relationship between 'past' and 'vldt unassessed', on one hand, and the possible values 'simple_r' and 'retrospective', on the other: Which of the two edges is RTRSPCTVT supposed to label in figures 6 and 7?[2]

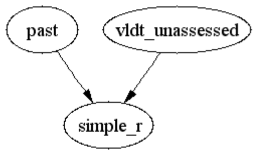

Fig. 6.

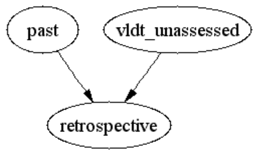

Fig. 7.

Figures 6 and 7 are graphic representations of the two instances defined by a rule like (19) along with (13). Contrast these graphs with the instances in figures 8-11 obtained when 'present' is involved:

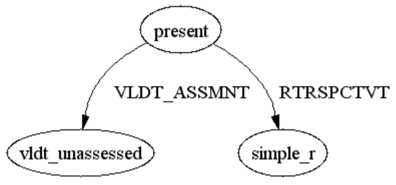

Fig. 8.

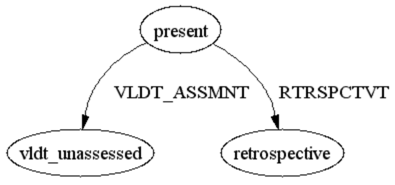

Fig. 9.

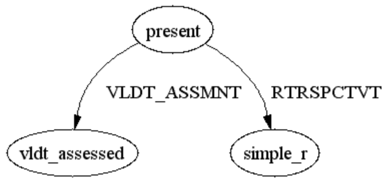

Fig. 10.

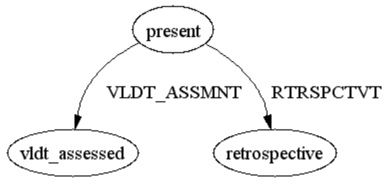

Fig. 11.

Figures 8-11 perspicuously show that RTRSPCTVT applies to 'present' and assigns to it either 'simple_r' or 'retrospective' as its values, independently of whether 'vldt unassessed' or 'vldt assessed' is part of the selection expression being constructed.

The graphs in figures 6 and 7 serve the purpose of revealing that one could not assign the label RTRSPCTVT to both edges, for this decision would not capture the

[2] GeneSys [9] uses the type library WinGraphviz 1.01.7 to implement the algorithm involved in graph generation with the Dot Language [10].

fact that RTRSPCTVT applies to ‘past’ under the condition that the feature ‘vldt unassessed’ be also part of the current selection expression (i.e. the selection expression under construction). Furthermore, if one assigned RTRSPCTVT to the edge joining ‘past’ to its values, what label should the other edge be assigned? A very simple solution is presented in what follows.

2.2 System entrance constraints as feature preferences

The same effects envisioned by Fawcett [4] can be attained without a conjunctive condition SNR like (19). Instead, we propose a PR like sp41 on ‘vldt assessed’ (see (21) below), plus the feature ‘nil’ (= undefined) as one option of the system RTRSPCTVT, which is assigned 100% probability whenever ‘vldt assessed’ is chosen. This is equivalent to preventing ‘past’ from being subcategorized as ‘simple_r’ or ‘restrospective’.

The goal is to eliminate the conjunctive condition on the left hand side of rule (19), redefine the systems in figures 4 and 5 as the systems in figures 12 and 13, and incorporate a PR on the feature ‘vldt assessed’ like (21):

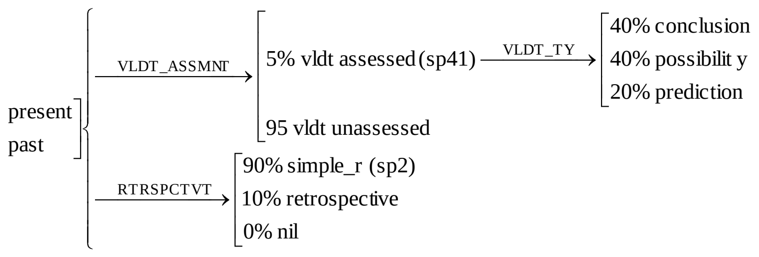

Fig. 12.

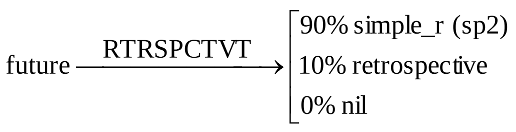

Fig. 13.

$$\text{sp41} \rightarrow \text{for same pass prefer RTRSPCTVT} < 100\% \text{ nil} > . \qquad (21)$$

The role of this PR is to state that the system RTRSPCTVT is entered but the value assigned to ‘past’, namely ‘nil’, amounts to specifying it as “undefined”, i.e. that neither ‘simple_r’ nor ‘restrospective’ are possible values for it.

The linear versions of figures 12 and 13 are defined as (22-28). Note the following changes: (i) instead of (18) and (19) we how have (23), and (ii) instead of (13) we now have (28).

$$\text{present} \rightarrow \text{VLDT\_ASSMN} \wedge \text{RTRSPCTVT} \ . \tag{22}$$

$$\text{past} \rightarrow \text{VLDT\_ASSMN} \wedge \text{RTRSPCTVT} \ . \tag{23}$$

$$\text{future} \rightarrow \text{RTRSPCTVT} \ . \tag{24}$$

$$\text{VLDT\_ASSMN} \rightarrow 5\%\ \text{vldt assessed(sp41)} \vee 95\%\ \text{vldt unassessed} \ . \tag{25}$$

$$\text{vldt assessed} \rightarrow \text{VLDT\_TY} \ . \tag{26}$$

$$\text{VLDT\_TY} \rightarrow 40\%\ \text{conclusion} \vee 40\%\ \text{possibilit y} \vee 20\%\ \text{prediction} \ . \tag{27}$$

$$\text{RTRSPCTVT} \rightarrow 90\%\ \text{simple\_r (sp2)} \vee 10\%\ \text{retrospective} \vee 0\%\ \text{nil} \ . \tag{28}$$

These rules allow for the instances illustrated by the graphs in figures 14-16, which contrast significantly with the graphs in figures 6 and 7. The instantiations in figures 14-16 show clearly that RTRSPCTVT is a function that applies to the argument 'past' and delivers the values 'simple_r', 'retrospective' or 'nil' depending on whether 'vldt unassessed' or 'vldt assessed' are also part of the current selection expression, respectively.

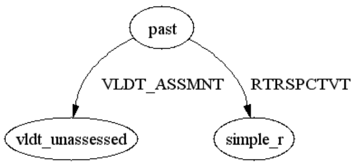

Fig. 14.

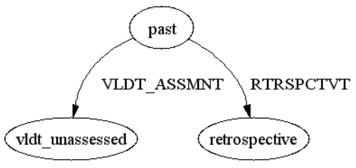

Fig. 15.

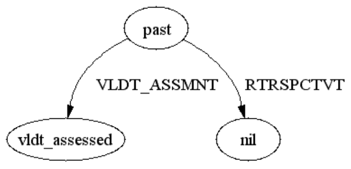

Fig. 16.

Crucially, it is a PR like (21), triggered by the feature 'vldt assessed' (cf. figure 12/rule (25)), which guarantees the set of instances in figures 14-16. In fact, PR (21) enforces the constraint that RTRSPCTVT proper, i.e. 'simple_r' and 'retrospective', applies only if the selection expression has previously incorporated the feature 'vldt unassessed'. This reformulation imposes a strict separation between arguments that need to be further specified by more delicate options, and the conditions under which such a subcategorization can take place. As in the case of 'present', now, RTRSPCTVT also applies to 'past' independently of whether 'vldt unassessed' or 'vldt assessed' are part of the current selection expression. The value 'nil' assigned to 'past' by RTRSPCTVT (figure 16) enforces Fawcett's original formulation [4] in the sense that neither 'simple_r' nor 'retrospective' are possible if 'vldt assessed' has been chosen.

Why would anyone make this move of eliminating conjunctive conditions in rules such as (19)? For one thing, RTRSPCTVT can now be viewed as a functor that takes ‘present’ or ‘past’ or ‘future’ as its argument, and delivers for it ‘simple_r’, ‘retrospective’, or ‘nil’ as possible values. The enforcement of value ‘nil’ for the function ‘RTRSPCTVT(past)’ is a consequence of PR (21) on ‘vldt assessed’. Instead of avoiding entrance to RTRSPCTVT in the case of selection expressions containing both ‘past’ and ‘vldt assessed’, our reformulation lets this configuration of features to enter the system but ‘past’ is assigned the value ‘nil’ by the relevant function. The net effect of this treatment of conjunctive condition SNRs is that now all system names can be viewed as functors taking simple conditions, i.e. single features, as arguments, and delivering feature options as the possible values.

2.3 Entering alternate sets of features

Fawcett’s mini-grammar [5] contains conjunctive condition SNRs that involve pairs of systems like the ones in figures 17-18 and 19-20:

$$f1 \xrightarrow{\text{SYS1}} F$$

Fig. 17.

$$f1 \xrightarrow{\text{SYS1}} F'$$

Fig. 18.

$$f2 \xrightarrow{\text{SYS2}} G$$

Fig. 19.

$$f2 \xrightarrow{\text{SYS2}} G'$$

Fig. 20.

where ‘f1’, ‘f2’ are conjunctions of features, SYS1, SYS2 are system names, F, F’, G, G’ are disjunctions of non-‘nil’ features with probabilities assigned, F’ is identical to F except for the probability assignments, and some of the disjuncts in G’ differ from G not only in the probability assignments but also in the features themselves.

Only one of the conjunctive features in ‘f1’, ‘f2’ is the natural candidate for further specification. The other conjunct(s) is/are the condition(s) under which such a subcategorization must take place. Thus, a reformulation along the lines of the preceding section appears to be also appropriate for the mini-grammar [5]: the candidate argument remains as the single feature entry condition to SYS1 and SYS2, and the conditioning features are eliminated from ‘f1’, ‘f2’ so that when chosen in a system traversal they trigger the PRs responsible for resetting the relevant probabilities.

3 Preference rules as constraints on feature relationships

The two cases discussed in the preceding section illustrate the problem posed by conjunctive conditions on SNRs in CG: they mix arguments proper that need to be further specified by other SNRs with the conditions under which such SNRs (i) should not apply in a general fashion (cf. 2.1) or (ii) should be sensitive to alternative feature specifications and/or feature probability reconfigurations (cf. 2.3).

This problem is apparent when one draws the graphs of the relevant selections expressions, for they reveal, as illustrated by the instances in figures 6 and 7, that the features that further specify ‘past’ are also related to other features of the selection expression that serve the sole, though significant, role of stipulating the conditions under which the subcategorization is to be carried out.

Given the approach of section 2, the cases addressed in 2.1 and 2.3 can be both seen as actually referring to conditions whose role is to reset probabilities of the same or alternative features, including the "undefined" value 'nil', which represents "vacuous" functional application or, in terms of Fawcett's [4] original formulation for restricting the scope of RTRSPCTVT, the exclusion of the values 'simple_r' and 'retrospective' for the argument 'past' when this feature co-occurs with 'vldt assessed'.

In both the CG micro- [4] and mini-grammar [5], all SNRs requiring conjunctive conditions mix an argument proper with the condition(s) under which the rule can be applied. Put differently, in all SNRs having a conjunctive condition, one of the conjuncts is the argument proper while the other conjuncts, whether simple or complex, define the conditions under which the argument is to be further specified by other features.

This state of affairs can be transformed into an empirically equivalent, and formally more appropriate, formulation that (i) reduces the original conjunctive condition to a simple feature condition, where this feature is the argument proper of the system in question (i.e. the functor), (ii) redefines the system options (i.e. the possible values) in accordance with the original formulation, whether by adding the feature 'nil' or by incorporating separate feature options, and (iii) incorporates a PR on the relevant conditional feature so that it assigns the necessary probabilities to 'nil' or other features.

Our reformulation separates arguments from the condition(s) under which they are to be assigned a set of possible values. The theoretical advantage of this treatment is that it allows for SNRs to be defined entirely by a set of one-place functions, for system entrance constraints are now controlled by PRs. This reformulation of system entrance constraints is possible because they can be defined by an existing, independently motivated rule type of CG, namely PRs.

4 Conclusions and future work

The semantic component of CG is defined by a set of material implications of the form '$p \rightarrow q$', where the condition 'p' is either a single feature or a conjunction of features, and 'q' is an instruction to incorporate a single feature or a conjunction of features into the selection expression under construction. This paper has shown that rules where 'p' is a conjunction of features generate graphs revealing that there is not a strict separation between the argument that is intended to be more delicately specified by the features in 'q', and the conditions under which such a specification is to be carried out. Typically, one of the conjuncts of the conjunctive condition 'p' is the argument of the function(s) responsible for assigning the values defined in 'q'. The other conjuncts in 'p', however, are rather the conditions under which functional assignment takes place. This is an unwanted conceptualization, for it does not represent transparently the relationships between features and thus the resulting graphs do not capture the facts adequately, namely: the functional argument-value structure of the system network, on one hand, and the conditioning feature configurations constraining it, on the other.

This paper remedies the problem by eliminating conjunctive condition SNRs altogether in favour of single feature condition SNRs, i.e. one-place functions, through a relocation of the constraining conjuncts in PRs on relevant features, typically the constraining features of the old conjunctive conditions. Now the strict separation between the argument in need of further specification and the conditions under which this more delicate subcategorization is to be carried out allows for a division of labour between two of the existing rule types of CG: SNRs and PRs. The former, all redefined now as single feature condition rules, are used to construct richly labeled, structured selection expressions, and the latter, to define the conditions under which certain systems reset feature probabilities so that the demands of empirical adequacy are met.

Two hypothesis currently being explored within GeneSys are: (i) being definable as sets of one-place functions, system networks can be viewed as resources that can be specified entirely in terms of the RDF technology, and (ii) system networks can be conceptualized and formalized as algorithms for the construction of logical form graphs [1, 2].

References

1. Castel, V.M. Acerca de la representación de relaciones entre semántica y forma en el signo lingüístico de la Gramática de Cardiff. En XXII Jornadas de investigación de la Universidad Nacional de Cuyo, Mendoza, Argentina (2010)
2. Castel, V.M. GeneSys_IDE: Entorno integral de desarrollo de gramáticas orientadas a la generación de textos. *Software* para la especificación de gramáticas sistémico-funcionales en el marco del modelo de Cardiff. Las lenguas española e inglesa como casos testigos. PIP CONICET (2010-2012) y Proyecto SeCyT, UNCuyo (2010-2011).
3. Fawcett, R.P.: A theory of syntax for systemic functional linguistics. John Benjamins, Amsterdam (2000)
4. Fawcett, R.P.: Systemic Functional Grammar as a formal model of language: a micro-grammar for some central elements of the English clause. Computational Linguistics Unit, Cardiff UK (2003)
5. Fawcett, R.P.: The mini-grammar of GeneSys version 5. Basic Meanings realized in the clause other than transitivity: register, mood, time, aspect, vldt, polarity, co-ordination, information focus and subject theme. Computational Linguistics Unit, Cardiff UK (2004)
6. Fawcett, R.P.: Invitation to Systemic Functional Linguistics through the Cardiff Grammar: An extension and simplification of Halliday's Systemic Functional Grammar (third edition). Equinox, London (2008)
7. Fawcett, R.P., Tucker, G.H., Lin, Y.Q.: How a systemic functional grammar works: the role of realization in realization. In: Horacek, H., Zock, M. (eds.) New Concepts in Natural Language Generation, pp. 114-86. Pinter, London (1993)
8. Fawcett, R.P., Tucker, G.H., Lin, Y.Q.: The GENESYS Lexicogrammar Version 5.3. Computational Linguistics Unit, Cardiff University (1996)
9. Fawcett, R.P. and V.M. Castel: Software for GENESYS: The Cardiff Grammar Text-Sentence Generator. Prototype 3. Computational Linguistics Unit, Cardiff University (2006)
10. Gansner, E.R., Koutsofios, E., North, S.: Drawing graphs with dot. Available at http://www.graphviz.org/pdf/dotguide.pdf (2009)