The Non-associativity of Polarized Tree-Based Grammars

Yael Cohen-Sygal[1] and Shuly Wintner[2]

[1] Department of Computer Science, University of Haifa, Israel
yaelc@cs.haifa.ac.il
[2] Department of Computer Science, University of Haifa, Israel
shuly@cs.haifa.ac.il

Abstract. Polarities are used to sanction grammar fragment combination in high level tree-based formalisms such as eXtenssible Meta-Grammar (XMG) and polarized unification grammars (PUG). We show that attaching polarities to tree nodes renders the combination operation non-associative, and in practice leads to overgeneration. We first provide some examples of non-associative combination operators in existing polarity-based formalisms. We then prove that there is no other non-trivial polarity system for which grammar combination *is* associative. This property of polarities casts doubt on the usability of polarity-based grammars for grammar engineering.

1 Introduction

Development of large scale grammars for natural languages is an active area of research in human language technology. Such grammars are developed not only for purposes of theoretical linguistic research, but also for natural language applications such as machine translation, speech generation, etc. Wide-coverage grammars are being developed for various languages in several theoretical frameworks.

In this paper we focus on tree based formalisms, e.g., Tree Adjoining Grammar (TAG, [1]). A TAG consists of a number of elementary trees, which can be combined with substitution or adjunction. Several variations and extensions of TAG exist, including lexicalized TAG ([2]) and constraint-based TAG ([3]).

A wide-coverage TAG may contain hundreds or even thousands of elementary trees, and syntactic structure can be redundantly repeated in many trees ([4,5]). Consequently, maintenance and extension of such grammars is a complex task. To address these issues, several high-level formalisms were developed ([6,7,8]). These formalisms take the *metagrammar approach*, where the basic units are tree descriptions (i.e., formulas denoting sets of trees) rather than trees. Tree descriptions are constructed by a tree logic and combined through conjunction or inheritance (depending on the formalism). The set of minimal trees that satisfy the resulting descriptions are the TAG elementary trees. In this way modular construction of grammars is supported, where a module is merely a tree description and modules are combined by means of the control tree logic.

A. Gelbukh (Ed.): CICLing 2007, LNCS 4394, pp. 208–217, 2007.

The move to tree descriptions requires a mechanism to sanction only desired combinations of descriptions. To constrain undesired combinations, each node of a tree description is associated with a name and nodes with the same name must denote the same entity and therefore must be identified ([6]). The drawback of this approach is that the only channel of interaction between two descriptions is the names of the nodes. Furthermore, the names of nodes can only be used to identify two nodes, but not to disallow such an identification. To overcome these shortcomings, [9] suggest to replace node naming by a *coloring* scheme, where nodes are colored black, white or red. When two trees are unified, a black node may be unified with 0, 1 or more white nodes and produce a black node; a white node must be unified with a black one producing a black node; and a red node cannot be unified with any other node. Furthermore, a satisfying model must be *saturated*, i.e., one in which all the nodes are either black or red. In this way some combinations can be forced and others prevented.

[10] extends this mechanism by associating each node with a set of *polarity features*. A polarity feature consists of a feature, arbitrarily determined by the grammar writer, and a polarity, which can be either positive, negative or neutral. A positive value represents an available resource and a negative value represents an expected resource. Two feature-polarity pairs can combine only if their feature is identical and their polarities are opposite (i.e., one is negative and the other is positive); the result is a feature-polarity pair consisting of the same feature and the neutral polarity. Two nodes can be identified only if their polarity features can combine. A solution is a tree whose features are all neutralized.

The concept of polarities is further elaborated by [11], who defines *Polarized Unification Grammars* (PUG). A PUG is defined over a *system of polarities* $(P, \cdot)$ where P is a set (of polarities) and '$\cdot$' is an associative and commutative product over P. A PUG generates a set of finite structures over objects which are determined for each grammar separately. The objects are associated with polarities, and structures are combined by identifying some of their objects. The combination is sanctioned by polarities: objects can only be identified if their polarities are unifiable; the resulting object has the unified polarity. A non-empty, strict subset of the set of polarities, called the set of *neutral* polarities, determines which of the resulting structures are valid: A polarized structure is *saturated* if all its polarities are neutral. The structures that are generated by the grammar are the saturated structures that result from combining different structures.

PUGs are more general than the mechanisms of polarity features and coloring, since they allow the grammar designer to decide on the system of polarities, whereas other systems pre-define it. Another difference is that while in other tree based grammars, if two nodes are identified then their predecessors must be identified as well, this is not the case in PUGs. In PUGs any two objects can be identified; the only restriction on the identification of two objects is the possibility to combine their polarities.

Combination of tree-based grammar fragments with polarities is conjectured (although not proven) to be associative ([11]). In this paper we show that attaching polarities to tree nodes results in a non-associative combination operation. Practical systems which use polarities, such as XMG ([12]), suffer from overgeneration as a result of non-associativity. In section 2 we show that existing polarity schemes induce non-associative tree combination operations. Unfortunately, this is not a result of poor choice of polarities on account of existing formalisms; in section 3 we show that *any* non-trivial polarity system induces a non-associative tree combination operation. This property of polarities casts serious doubts on the usability of polarity-based grammars for grammar engineering.

2 Existing Polarity Systems

In this section we provide a few counter-examples which demonstrate the non-associativity of grammar combination in some existing grammar formalisms. In all the examples, the relation which determines how polarities combine *is* indeed associative; it is the tree combination operation which uses polarities that is shown to be non-associative.

2.1 XMG Colors

eXtensible MetaGrammar (XMG, [13,12]) is a tool for designing large scale grammars for natural languages. Following [9], XMG uses colors to sanction tree node identification. The color combination table is presented in Figure 1. W, B and R denote white, black and red, respectively, and $\perp$ represents the impossibility to combine.

$\cdot$	W	B	R
W	W	B	$\perp$
B	B	$\perp$	$\perp$
R	$\perp$	$\perp$	$\perp$

Fig. 1. Color combination in XMG

Example 1. Consider T_1, T_2, T_3 of Figure 2. The results of combining these trees in different orders are depicted in Figure 3. While $(T_1 + T_2) + T_3$ yields possible solutions, $T_2 + T_3$ has no solution and therefore the same holds for $T_1 + (T_2 + T_3)$. Notice that the solutions of $(T_1 + T_2) + T_3$ are saturated, since all the nodes in these trees are either black or red. Clearly, the combination operation with colored trees is not associative.

Example 2. Consider T_4, T_5, T_6 of Figure 4. The results of combining these trees in different orders are depicted in Figure 5. Assume that the initial set of trees is $\{T_5, T_6\}$. Adding a new tree, T_4, is expected to result in the set of $T_4 + (T_5 + T_6)$.

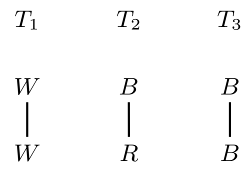

Fig. 2. Colored trees to be combined

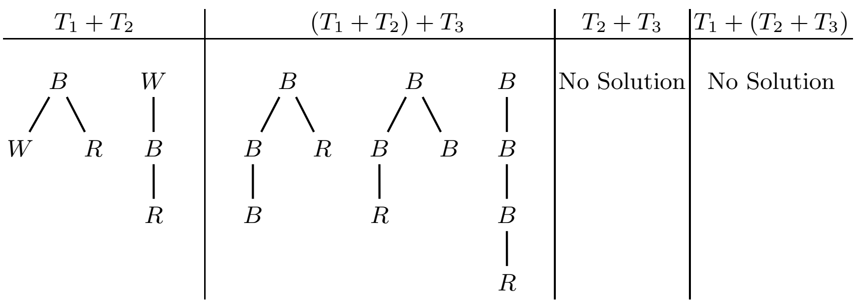

Fig. 3. The result of combining T_1, T_2, T_3

In practice, however, XMG computes all the possible combination orders, and the resulting set is $(T_4+T_5)+T_6$; observe that the resulting set overgenerates with respect to $T_4+(T_5+T_6)$. In actual grammars, where the sets of trees include hundreds of trees, the resulting solutions may include many such unexpected (and overgenerating) results. It is virtually impossible to track all the sources for such overgenerations, and therefore the maintenance of large tree-based grammars with colors is a complex, perhaps impractical task. Notice that all the intermediate and final solutions are saturated. Therefore, the saturation rule does not prevent the problem of non-associativity of colored-tree combination.

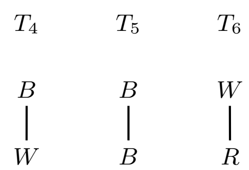

Fig. 4. Colored trees to be combined

Examples 1 and 2 sufficient for drawing the following conclusion:

Corollary 1. *Colored-tree combination is not associative.*

.

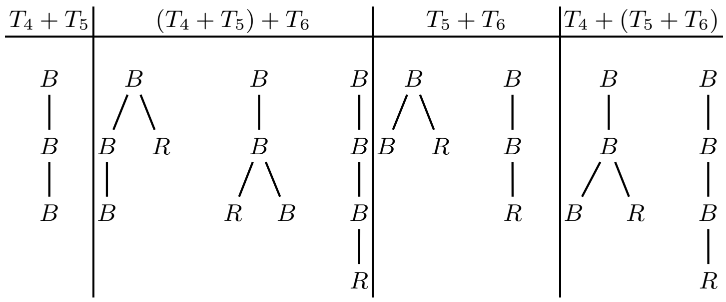

Fig. 5. The result of combining T_4, T_5, T_6

2.2 PUGs

PUGs allow arbitrary polarities to be used. However, we first consider the polarities that are used in the literature; in section 3 we consider the general case. [14] and [11] use two systems of polarities which are depicted in Figures 6 and 7, respectively. The first system includes three polarities, gray, white and black, where the neutral polarities are black and gray. A black node may be unified with 0, 1 or more gray or white nodes and produce a black node; a white node may absorb 0, 1 or more gray or white nodes but eventually must be unified with a black one producing a black node; and a gray node may be absorbed into a white or a black node. The second system extends the first by adding two more non-neutral polarities, plus and minus. The plus and minus may absorb 0, 1 or more white or gray nodes but eventually a plus node must be unified with a minus node producing a black node. The following example shows that these two operations are non-associative.

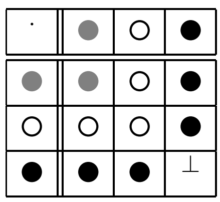

Fig. 6. A system of polarities

Example 3. Consider T_7, T_8, T_9 of Figure 8. The combination of these structures is depicted in Figure 9 (the combination is the same for both operations). Clearly $T_7 + (T_8 + T_9) \neq (T_7 + T_8) + T_9$.

Corollary 2. *PUG combination with the polarity system of either Figure 6 or 7 is not associative.*

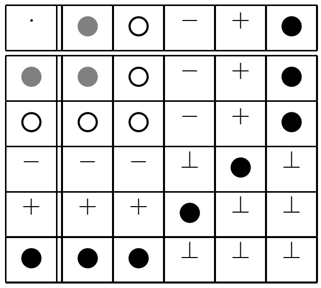

Fig. 7. A system of polarities

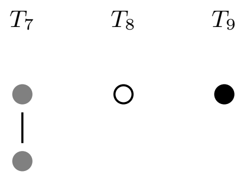

Fig. 8. Polarized trees to be combined

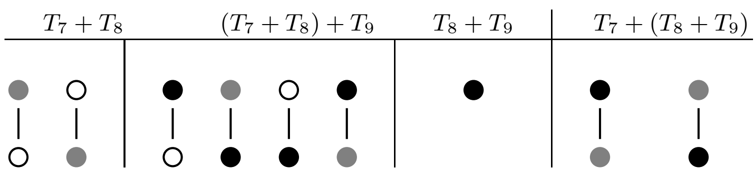

Fig. 9. The result of combining T_7, T_8, T_9

3 General Polarity Systems

In section 2 we showed that some existing polarity-based formalisms are non-associative. Unfortunately, this is not accidental; in what follows we show that the only polarity scheme that induces associative tree combination is trivial: the one in which no pair of polarities are unifiable. This scheme is useless for sanctioning tree combination since it disallows any combination.

In the sequel, if $(P, \cdot)$ is a system of polarities and $a, b \in P$, we use the shorthand notation ab instead of $a \cdot b$. $ab{\downarrow}$ means that the combination of a and b is defined and $ab{\uparrow}$ means that a and b cannot combine.

Definition 1. *A system of polarities* $(P, \cdot)$ *is trivial if for all* $a, b \in P$, $ab \uparrow$.

Proposition 1. *Let* $(P, \cdot)$ *be a system of polarities such that* $|P| > 1$. *If there exists* $a \in P$ *such that* $aa{\downarrow}$ *then the polarized tree combination based on* $(P, \cdot)$ *is not associative.*

Proof. Let $(P, \cdot)$ be a system of polarities such that $|P| > 1$ and let $a \in P$ be such that $aa\downarrow$. Assume toward a contradiction that the polarized tree combination based on $(P, \cdot)$ is associative. Let $b \in P$ be such that $a \neq b$ (such b exists since $|P| > 1$). Consider T_1, T_2, T_3 of Figure 10. Of all the trees in $(T_1 + T_2) + T_3$ and $T_1 + (T_2 + T_3)$, focus on trees of the structure depicted in Figure 11. All possible instantiations of these trees are depicted in Figure 12 (we suppress the intermediate calculations). Notice that these trees are only candidate solutions; they are actually accepted only if the polarity combinations occurring in them are defined.

As described in section 1 , PUG and XMG slightly differ in the way trees are combined. While in XMG, if two nodes are identified then their predecessors must be identified too, in PUG any two nodes can be identified. However, for the tree structure of Figure 11, the same sets of trees are accepted for both the XMG and the PUG approaches.

Since $aa\downarrow$, T_{11} is accepted as a solution of $T_1 + (T_2 + T_3)$. However, this tree is not accepted as a solution of $(T_1 + T_2) + T_3$ since $a \neq b$ and there is no tree among the possible solutions of $(T_1 + T_2) + T_3$ whose top and bottom nodes are b, a contradiction.

Proposition 2. *Let $(P, \cdot)$ be a non-trivial system of polarities such that $|P| > 1$. Then the polarized tree combination based on $(P, \cdot)$ is not associative.*

Proof. Let $(P, \cdot)$ be a non-trivial system of polarities such that $|P| > 1$. Assume toward a contradiction that the polarized tree combination based on $(P, \cdot)$ is associative. Since $(P, \cdot)$ is non-trivial, there exist $a, b \in P$ such that $ab\downarrow$. Again, consider T_1, T_2, T_3 of Figure 10 and their combinations $(T_1 + T_2) + T_3$ and $T_1 + (T_2 + T_3)$. As before, of all the trees in $(T_1 + T_2) + T_3$ and $T_1 + (T_2 + T_3)$ consider only the resulting trees having the structure of Figure 11 which are depicted in Figure 12. There are two possible cases:

1. $aa\downarrow$ or $bb\downarrow$: Then from theorem 1 it follows that the resulting tree combination operation is not associative, a contradiction.
2. $aa\uparrow$ and $bb\uparrow$: Then $(T_1 + T_2) + T_3$ has no solutions and $T_1 + (T_2 + T_3)$ has one accepted solution (T_9), a contradiction.

Proposition 3. *Let $(P, \cdot)$ be a non-trivial system of polarities such that $|P| = 1$. Then the polarized tree combination based on $(P, \cdot)$ is not associative.*

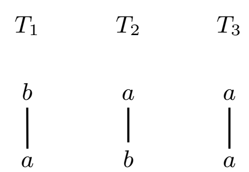

Fig. 10. Polarized trees to be combined

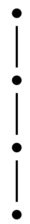

Fig. 11. A tree structure

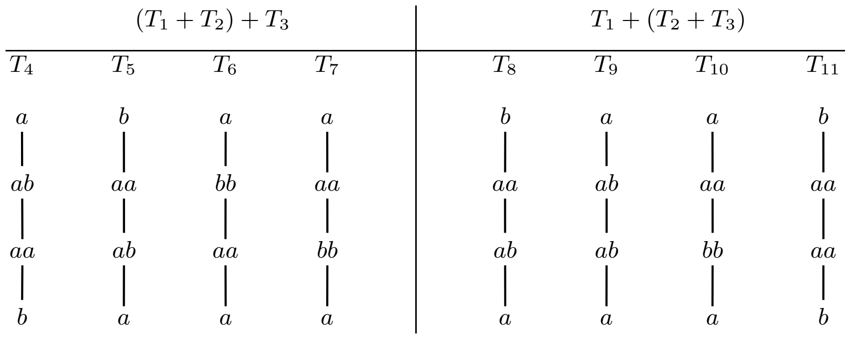

Fig. 12. Resulting trees

Proof. Let $(P, \cdot)$ be a non-trivial system of polarities such that $P = \{a\}$. Assume toward a contradiction that the polarized tree combination based on $(P, \cdot)$ is associative. Since P is non-trivial, $aa = a$. Consider T_1, T_2, T_3, T_4 of Figure 13 and the combinations $(T_1+T_2)+T_3$ and $T_1+(T_2+T_3)$. T_4 is accepted as a solution of $T_1+(T_2+T_3)$ but not as a solution of $(T_1+T_2)+T_3$ (we suppress the calculations), both in the XMG and the PUG approach. Clearly $(T_1+T_2)+T_3 \neq T_1+(T_2+T_3)$, a contradiction.

Corollary 3. *Let $(P, \cdot)$ be a non-trivial system of polarities. Then the polarized tree combination based on $(P, \cdot)$ is not associative.*

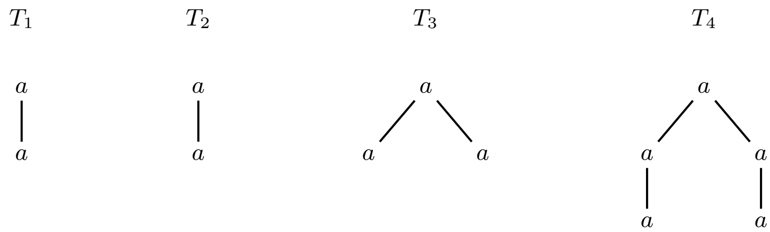

Fig. 13. Polarized trees with a single polarity

Proposition 4. *Let $(P, \cdot)$ be a trivial system of polarities. Then the polarized tree combination based on $(P, \cdot)$ is associative.*

Proof. If $(P, \cdot)$ is a trivial system of polarities then any combination of two polarized trees results in the empty set (no solutions). Evidently, polarized tree combination based on $(P, \cdot)$ is associative.

Corollary 4. *Let $(P, \cdot)$ be a system of polarities. Then polarized tree combination based on $(P, \cdot)$ is associative if and only if $(P, \cdot)$ is trivial.*

4 Conclusion

We showed that non-trivial systems of polarities induce non-associative tree combination operators. The practical implication of this non-associativity, at least in XMG, is overgeneration. This property of polarity-based systems most probably implies that they should not be used to sanction tree combination in grammar formalisms.

The non-associativity of polarized tree-based grammars is not a property of the polarities but rather of the combination operation and the way polarities are used by the tree combination operators. From proposition 3 it follows that even without polarities (where any two nodes can be identified), the combination is non-associative in the sense that different combination orders yield different structures. Furthermore, if two combination orders yield the same basic structures, their nodes are not necessarily associated with the same polarities, thus hampering combination associativity. The implication of this is that polarities cannot be used to guarantee associativity where it does not exist in the first place.

Polarities were associated with tree nodes to sanction tree combination in a more general way than the node naming mechanism introduced by [6]. As we show here, this renders grammar combination non-associative. A different mechanism, providing the generality of polarities but maintaining the associativity of tree combination, is required. For an example of such a mechanism, in the context of typed unification grammars, see [15].

Acknowledgments

We are grateful to Yannick Parmentier for his help. This research was supported by The Israel Science Foundation (grant no. 136/01).

References

1. Joshi, A.K., Levy, L.S., Takahashi, M.: Tree Adjunct Grammars. Journal of Computer and System Sciences (1975)
2. Schabes, Y., Abeillé, A., Joshi, A.K.: Parsing strategies with 'lexicalized' grammars: application to tree adjoining grammars. In: Proceedings of the 12th conference on Computational linguistics, Association for Computational Linguistics (1988) 578–583

3. Vijay-Shanker, K., Joshi, A.K.: Feature structure based tree adjoining grammar. In: Proceedings of the 12th conference on Computational linguistics, Association for Computational Linguistics (1988) 714–719
4. XTAG Research Group: A lexicalized tree adjoining grammar for English. Technical Report IRCS-01-03, IRCS, University of Pennsylvania (2001)
5. Abeillé, A., Candito, M.H., Kinyon, A.: FTAG: developing and maintaining a wide-coverage grammar for French. In Hinrichs, E., Meurers, D., Wintner, S., eds.: Proceedings of the ESSLLI-2000 Workshop on Linguistic Theory and Grammar Implementation. (2000) 21–32
6. Candito, M.H.: A principle-based hierarchical representation of LTAGs. In: Proceedings of the 16th conference on Computational linguistics, Copenhagen, Denemark, Association for Computational Linguistics (1996) 194–199
7. Duchier, D., Gardent, C.: A constraint-based treatment of descriptions. In: Third International Workshop on Computational Semantics (IWCS-3), Tilburg, Netherlands (1999)
8. Kallmeyer, L.: Local tree description grammars. Grammars **4** (2001) 85–137
9. Crabbé, B., Duchier, D.: Metagrammar redux. In: CSLP, Copenhagen, Denemark (2004)
10. Perrier, G.: Interaction grammars. In: Proceedings of the 18th conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (2000) 600–606
11. Kahane, S.: Polarized unification grammars. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia (2006) 137–144
12. Crabbé, B.: Grammatical development with XMG. In: Proceedings of the 5th International Conference on Logical Aspects of Computational Linguistics (LACL), Bordeaux, France (2005)
13. Duchier, D., Le Roux, J., Parmentier, Y.: The metagrammar compiler: An NLP application with a multi-paradigm architecture. In: Proceedings of the Second International Mozart/Oz Conference (MOZ 2004), Charleroi, Belgium (2004)
14. Kahane, S., Lareau, F.: Meaning-text unification grammar: modularity and polarization. In: Proceedings of the 2nd International Conference on Meaning-Text Theory, Moscow (2005) 197–206
15. Cohen-Sygal, Y., Wintner, S.: Partially specified signatures: A vehicle for grammar modularity. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia (2006) 145–152