THE EDITING DISTANCE IN SHARED FOREST

M. Vilares **D. Cabrero** **F.J. Ribadas**

Computer Science Department

University of A Coruña

Campus de Elviña s/n, 15071 A Coruña, Spain

{vilares, cabrero, ribadas}@dc.fi.udc.es

Abstract

In an information system indexing can be accomplished by creating a citation based on context-free parses, and matching becomes a natural mechanism to extract patterns. However, the language intended to represent the document can often only be approximately defined, and indices can become shared forests. Queries could also vary from indices and an approximate matching strategy becomes also necessary. We present a proposal intended to prove the applicability of tabulation techniques in this context.

1 A dynamic frame for parsing

We use ICE [2] as parsing frame. The formalism is an extended,LALR(1), push-down transducer (PDT). It proceeds by building *items*, compact representations of the PDT stacks, which are produced by applying transitions to existing ones, until no new application is possible. These items provide an optimal sharing of computations for non-deterministic inputs. A merit ordering guarantees fairness and completeness, and redundant items are ignored by using a simple subsumption relation.

We represent a parse as the chain of the context-free rules used in a leftmost reduction of the input sentence, whose non-terminals are items. The output grammar is then represented in finite shared form by an AND-OR graph that in our case is precisely the shared-forest. The time complexity (resp. space complexity) for this bottom-up parser is $\mathcal{O}(n^3)$ (resp. $\mathcal{O}(n^2)$), for inputs of length n. This complexity is lineal for deterministic inputs, which favourices the performance in practice.

2 A dynamic frame for approximate pattern matching

Given P, a target tree, and D, a data tree, we define an *edit operation* as a pair $a \to b$ with an associated cost, $\gamma(a \to b)$, that we extend to a sequence S of edit operations $s_1,\ s_2,\ \ldots,\ s_n$ in the form $\gamma(S) = \sum_{i=1}^{|S|}(\gamma(s_i))$. The distance between P and D is defined by the metric δ, in the form $\delta(P, D) = \min\{\gamma(S),\ S$ editing sequence taking P *to* $D\}$. Given an inverse postorder traversal to name each node i of a tree T by $T[i]$, we focus on mappings [1], a particular kind of sequences. Given $\mathcal{D}$ and $\mathcal{I}$, respectively, the nodes in P and D not touched by edit operations; the cost of a mapping M is computed by $\gamma(M) = \sum_{(i,j)\in M} \gamma(P[i] \to D[j]) + \sum_{i\in\mathcal{D}} \gamma(P[i] \to \varepsilon) + \sum_{j\in\mathcal{I}} \gamma(\varepsilon \to D[j])$. It can be proved that $\delta(P, D) = \min\{\gamma(M),\ M$ mapping from P to $D\}$.

In relation to Zhang *et al.* in [3], we can deal with data shared forest. The original approach would require a less efficient top-down parsing to avoid redundant computations, due to postorder tree traversal applied and computing the distance by left-recursion on this search. The time complexity is $\mathcal{O}(\mid P \mid\mid D \mid \min(\text{depth}(P), \text{leaves}(P)) \min(\text{depth}(D), \text{leaves}(D)))$, where $\mid P \mid$ (resp. $\mid D \mid$) is the number of nodes in P (resp. in the tree of D with a maximum number of nodes), leaves(P) (resp. leaves(D)) is the number of leaves in P (resp. in the tree of D with a maximum number of leaves), and depth(P) (resp. depth(D)) is the depth of P (resp. in the tree of D with maximal depth).

3 Experimental results

We consider the language, $\mathcal{L}$, of arithmetic expressions to compare our proposal with Tai [1], and Zhang *et al.* [3], using two deterministic grammars, $\mathcal{G}_L$ and $\mathcal{G}_R$, for the left and right associative versions of $\mathcal{L}$; and one non-deterministic $\mathcal{G}_N$. Parses are built using ICE, and tests have been applied on data inputs $a_1 + a_2 + \ldots + a_i + a_{i+1}$, with i even. As pattern, parse trees from inputs $a_1 + b_1 + a_3 + b_3 + \ldots + b_{i-1} + a_{i-1} + b_{i+1} + a_{i+1}$, where $b_j \neq a_{j-1}$, for all $j \in \{1, 3, \ldots i-1, i+1\}$.

Patterns are built from the left-associative (resp. right-associative) interpretation for $\mathcal{G}_L$ (resp. $\mathcal{G}_R$), as in the non-deterministic case, to evaluate the impact of traversal orientation. So, Fig. 1 proves the adaptation of our proposal (resp. Zhang *et al.*) to left-recursive (resp. right-recursive) derivations, and the gain in efficiency due to sharing of computations in a dynamic frame for $\mathcal{G}_N$.

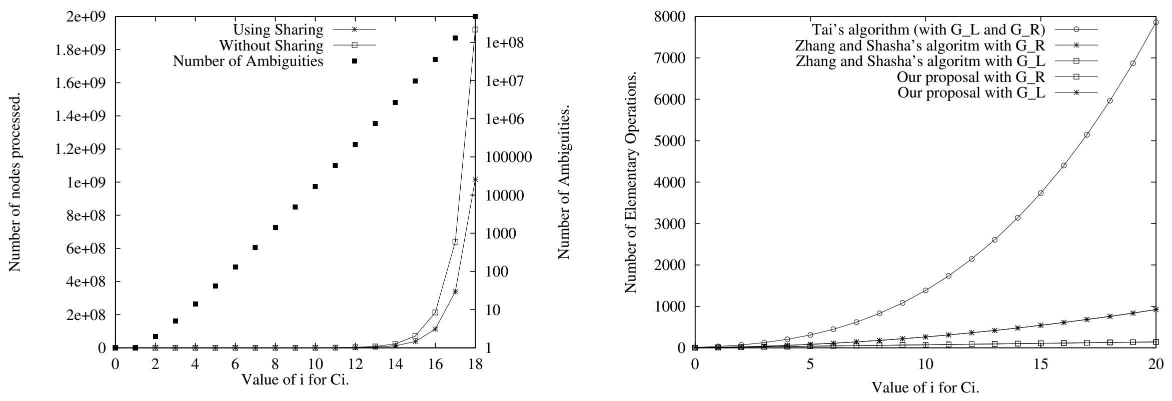

Figure 1: Results on approximate tree matching

Acknowledgements

Work partially supported by projects XUGA 20402B97 of the Autonomous Government of Galicia, project PGIDT99XI10502B of Spanish Government and project 1FD97-0047-C04-02 by the EU.

References

[1] Kuo-Chung Tai. 1978. Syntactic error correction in programming languages. IEEE Transactions on Software Engineering, SE-4(5):414-425.

[2] M. Vilares and B. A. Dion. 1994. Efficient incremental parsing for context-free languages. Proc. of the 5^{th} IEEE Int. Conf. on Computer Languages, pages 241-252, Toulouse, France.

[3] K. Zhang and D. Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. SIAM Journal on Computing, 18:1245-1262.