Contextual Tree Adjoining Grammars

Martin Kappes

Fachbereich Informatik, Johann Wolfgang Goethe-Universität, D - 60054 Frankfurt am Main, Germany, E-Mail: kappes@psc.informatik.uni-frankfurt.de

Abstract

In this paper, we introduce a formalism called contextual tree adjoining grammar (CTAG). CTAGs are a generalization of multi bracketed contextual rewriting grammars (MBICR) which combine tree adjoining grammars (TAGs) and contextual grammars. The generalization is to add a mechanism similar to obligatory adjoining in TAGs. Here, we present the definition of the model and some results concerning the generative capacity and closure properties of the classes of languages generated by CTAGs.

Introduction

Contextual grammars are a formalization of the linguistic idea that more complex, well formed strings are obtained by inserting contexts into already well formed strings. They were first introduced by Marcus in 1969; all models presented here are based on so-called internal contextual grammars which were introduced by Păun and Nguyen. References and further details about contextual grammars can be found in the monograph (Păun, 1997); a survey is given in (Ehrenfeucht *et al.*, 1997).

Tree adjoining grammars (TAGs) and contextual grammars are linguistically well motivated and have been considered as a good model for the description of natural languages (c.f. (Marcus, 1997)). Although contextual grammars and tree adjoining grammars seem very different at first sight, a closer look reveals many similarities between both formalisms. Therefore, it seems natural to combine those formalisms in order to obtain a generalized class of grammars for the description of natural languages, which combines the mechanisms of various classes. A first step were so-called multi-bracketed contextual grammars (MBIC) and multi-bracketed contextual rewriting grammars (MBICR), c.f. (Kappes, 1999). These grammars operate on a tree structure induced by the grammar (the first approach aiming in this direction was introduced in (Martin-Vide & Păun, 1998)).

However, the families of languages generated by MBIC and MBICR-grammars are either strictly included in or incomparable to the family of languages generated by TAGs. This is the case since, in MBIC and MBICR-grammars, each yield of a derived tree is immediately a word in the language generated by the grammar. In other words, there is no mechanism to distinguish between "finished" and "unfinished" trees like obligatory adjoining allows in TAGs. Here, by adding obligatory adjoining to MBICR-grammars, we obtain a generalized class which is also a proper extension of TAGs.

Definition and Example

Let Σ^* denote the set of all words over the finite alphabet Σ and $\Sigma^+ = \Sigma^* - \{\lambda\}$, where λ denotes the empty word. We denote the length of a string x by $|x|$. In this paper, we use the term derived tree for a tree where the internal nodes are labelled by symbols from a nonterminal alphabet Δ and the leaves are labelled by symbols from a terminal alphabet Σ. We use

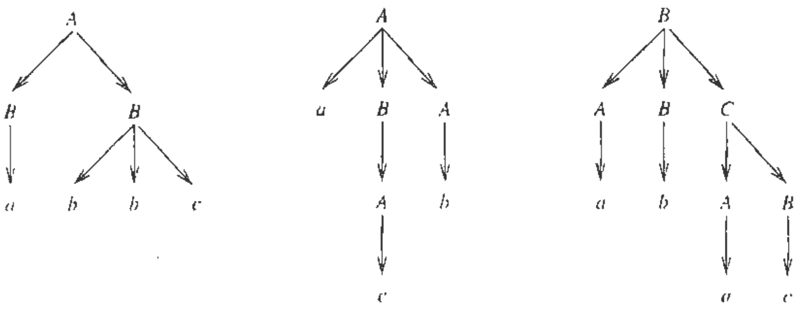

Figure 1: Derived trees corresponding to the Dyck-covered words (from left to right) $[_A[_Ba]_B[_Bbbc]_B]_A$, $[_Aa[_B[_Ac]_A]_B[_Ab]_A]_A$ and $[_B[_Aa]_A[_Bb]_B[_C[_Aa]_A[_Bc]_B]_C]_B$.

a linear representation of derived trees called Dyck-covered words. A Dyck-covered word is a string consisting of terminal symbols and opening and closing brackets indexed with nonterminal symbols. Formally, for the nonterminal alphabet Δ we define the bracket alphabet $B_\Delta = \{[_A,]_A \mid A \in \Delta\}$. Throughout the paper we always assume $\Sigma \cap B_\Delta = \emptyset$. The set of all Dyck-covered words $\mathrm{DC}_\Delta(\Sigma)$ over Σ with respect to the index alphabet Δ is inductively defined by

- For all $w \in \Sigma^+$ and $A \in \Delta$, $[_Aw]_A$ is in $\mathrm{DC}_\Delta(\Sigma)$.
- Let $n \geq 1$ be a positive integer. If $A \in \Delta$ and $\alpha_1, \alpha_2, \ldots, \alpha_n$ are in $\mathrm{DC}_\Delta(\Sigma) \cup \Sigma$, then $[_A\alpha_1\alpha_2\ldots\alpha_n]_A$ is in $\mathrm{DC}_\Delta(\Sigma)$.

It is not difficult to see that each $\alpha \in \mathrm{DC}_\Delta(\Sigma)$ can be interpreted as unique encoding for a derived tree, where Δ is the label alphabet for the internal nodes and Σ is the label alphabet for the leaf nodes in the following way: A string $[_A\alpha]_A \in \mathrm{DC}_\Delta(\Sigma)$ is identified with a tree where the root is labelled by A, and the subtrees of the root are determined by the unique decomposition of $\alpha = \alpha_1\alpha_2\ldots\alpha_n$ such that $\alpha_i \in \mathrm{DC}_\Delta(\Sigma) \cup \Sigma$, $1 \leq i \leq n$. For examples see Figure 1. By $\mathrm{DC}_\Delta^A(\Sigma)$ we denote the set of all elements in $\mathrm{DC}_\Delta(\Sigma)$ where the root node is labelled by A.

A contextual tree adjoining grammar (CTAG) is a tuple $G = (\Sigma, \Delta, \Upsilon, \Omega, P)$, where Σ is a finite set of terminals, Δ is a finite set of indices, $\Upsilon \subseteq \Delta$ is a set of permitted indices, $\Omega \subseteq \mathrm{DC}_\Delta(\Sigma) \cup \{\lambda\}$ is a finite set of axioms and P is a finite set of productions. Each production is of the form (S, C, K, H), where $S \subseteq \Sigma^+$ is the selector language, $K, H \subseteq \Delta$ are sets of nonterminals and C is a finite subset of contexts where each context is of the form (μ, ν) such that $\mu\nu \in \mathrm{DC}_\Delta(\Sigma)$.

The derivation process in a CTAG is illustrated in Figure 2: A context (μ, ν) may be adjoined to an $\alpha = \alpha_1[_B\alpha_2]_B\alpha_3$ yielding a tree $\alpha_1\mu[_E\alpha_2]_E\nu\alpha_3$ if and only if there is an $(S, C, K, H) \in P$ such that the yield of α_2 is in S, $(\mu, \nu) \in C$, $[_B\alpha_2]_B \in \mathrm{DC}_\Delta^B(\Sigma)$, $B \in K$ and $E \in H$. The string $[_A\alpha_2]_A$ is called selector. In the above figure, we have $\alpha \in \mathrm{DC}_\Delta^A(\Sigma)$, $\mu\nu \in \mathrm{DC}_\Delta^D(\Sigma)$ and the yield of α_1, α_2, α_3, μ, ν is w_1, w_2, w_3, u, v respectively. The set of all sentential forms of G, $S(G)$, consists of all trees which can be derived in the above way starting from an axiom in Ω. The set of all trees derived by a CTAG G, $T(G)$, consists of all trees in $S(G)$ where the internal nodes are only labelled by nonterminals in Υ. The weak generative capacity $L(G)$ is the yield of all trees in $T(G)$. Hence, internal nodes labelled by symbols from $\Delta - \Upsilon$ have to be relabelled during the derivation process in order to obtain a tree in $T(G)$.

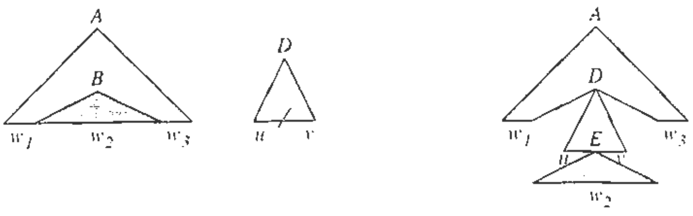

Figure 2: The derivation process in a CTAG

Up to some technical modifications necessary to keep our formalism consistent to the usual model of contextual grammars, we only added selector languages to the productions of a TAG. These selector languages are used to control the derivation process as they do in contextual grammars, the adjunction of an auxilliary tree is only possible if the yield of the node where the adjunction takes place is in the selector language.

We can classify CTAGs by their selector languages: A CTAG $G = (\Sigma, \Delta, \Upsilon, \Omega, P)$ is called with F-choice for a family of languages F, if $S \in F$ for all $(S, C, K, H) \in P$.

Consider for example the CTAG with Σ^+-selection

$$\begin{aligned} G &= (\{a, b, c, d, e\}, \{A, B\}, \{A\}, \{[_A a[_B bc]_B d]_A\}, \{\pi_1, \pi_2\}) \text{ where} \\ \pi_1 &= (\Sigma^+, \{([_A a[_B b, c]_B d]_A)\}, \{B\}, \{A\}) \text{ and} \\ \pi_2 &= (\Sigma^+, \{([_A c, e]_A)\}, \{B\}, \{A\}). \end{aligned}$$

It is not difficult to see that using π_1 i times yields a derivation

$$[_A a[_B bc]_B d]_A \overset{i}{\Longrightarrow}_G ([_A a)^{i+1}[_B b([_A b)^i (c]_A)^i c]_B (d]_A)^{i+1}.$$

In order to obtain a string in $T(G)$ we have to use production π_2 exactly once to remove the pair of brackets indexed by B from the sentential form. After applying π_2 once, no further derivation steps are possible, hence $L(G) = \{a^n cb^n c^n cd^n \mid n \geq 1\}$.

Generative Capacity

CTAGs are a generalization of MBICR-grammars. For $\Delta = \Upsilon$ these models are equivalent (CTAGs could thus also be called multi-bracketed contextual grammars with obligatory rewriting (MBICRO)). The obligatory adjoining feature increases the generative capacity. For instance, the language in the above example cannot be generated by any MBICR-grammar. This is due to the fact that each language L generated by an MBICR-grammar fulfills the so-called internal bounded step property (c.f. (Păun, 1997)): There is a constant p such that for each string $x \in L$, $|x| > p$ there is a $y \in L$ such that $x = x_1ux_2vx_3$, $y = x_1x_2x_3$ and $0 < |uv| \leq p$.

CTAGs using only the selector language Σ^+, i.e., in effect ignoring the selector language mechanism, and TAGs are, up to some details, descriptions of the same model. It is possible to construct a TAG equivalent to a given CTAG with Σ^+-choice and vice versa. The technical detail is that all elementary trees of a TAG must be elements of $\mathrm{DC}_\Delta(\Sigma)$ if the foot nodes of the auxilliary trees are not taken into account. Formally, the equivalence holds if the initial trees in a TAG are elements of $\mathrm{DC}_\Delta(\Sigma)$ and each auxilliary tree i of G is of the form $\alpha_i = \mu_i[_{A_i}]_{A_i}\nu_i$ such that $\mu_i\nu_i \in \mathrm{DC}_\Delta^{A_i}(\Sigma)$. Notice that the pair $[_{A_i}]_{A_i}$ represents the foot node of α_i. The construction of an equivalent TAG for a given CTAG with Σ^+-choice is a straightforward generalization of a similar construction for MBICR-grammars which can be found in (Kappes, 1999).

For the other direction, consider a TAG G of the above form. Let X denote the selective (or $\overline{X}$ in case of an obligatory) adjoining constraint of an internal node in an elementary tree. X (or $\overline{X}$) thus dereferences the subset of auxilliary trees which may be adjoined at this node. We can construct an equivalent CTAG $G' = (\Sigma, \Delta', \Upsilon', \Omega', P')$ with Σ^+-choice as follows: The set of indices Δ' and the set of permitted indices Υ' of G'' is given by

$$\begin{aligned} \Delta' &= \{(A, \tilde{X}) \mid A \in \Delta \text{ and } \tilde{X} \text{ is a (selective or obligatory) adjoining constraint}\} \\ \Upsilon' &= \{(A, X) \mid A \in \Delta \text{ and } X \text{ is a selective adjoining constraint}\}. \end{aligned}$$

For each initial tree α of G we insert a tree α' into Ω', where each node labelled by $A \in \Delta$ with (selective or obligatory) adjoining constraint $\tilde{X}$ is replaced by the index $(A, \tilde{X})$. We thus consider the adjoining constraint of a node as part of its index. For each auxilliary tree $i : \alpha_i = \mu_i[_{A_i}]_{A_i}\nu_i$ we insert a production $\pi_i = (\Sigma^+, \{(\mu_i', \nu_i')\}, \{(A_i, \tilde{X}) \mid i \in \tilde{X}\}, \{(A_i, \tilde{Z})\})$ into P' where $\mu_i'\nu_i'$ is obtained from $\mu_i\nu_i$ by the same procedure as above and $\tilde{Z}$ is the (selective or obligatory) adjoining constaint of the foot node of α_i. It is possible to prove that both grammars are equivalent.

It can be shown that each CTAG with finite selection generates a context-free language. This is the case since the length of each string which may be used as selector in a derivation step can be bounded by some constant. Due to the bracket structure it is impossible to shift information through the sentential form of a CTAG if the length of the selectors is finite. Therefore it is possible to construct a context-free grammar generating the same language. Also, for each context-free language there is a CTAG with finite selection generating that language. So, CTAGs with finite selectors generate exactly the context-free languages.

CTAGs with regular selectors can generate languages which cannot be generated by TAGs even if we do not take advantage of the obligatory adjoining feature. The language $L(G) = \{a^n b^n c^m d^n e^n \mid m \geq n \geq 1\}$ can be generated by an MBICR-grammar and hence by a CTAG with regular selector languages (c.f. (Kappes, 1999)) but not by any TAG because of the pumping-lemma for TAGs (cf. (Vijay-Shanker, 1988)).

With context-sensitve selector languages, CTAGs generate exactly the context-sensitive languages: Let $L \subseteq \Sigma^+$ be a context-sensitive language. We construct the CTAG

$$\begin{aligned} G &= (\Sigma, \{A, B\}, \{A\}, \Omega, \{\pi\} \cup \{\pi_\sigma \mid \sigma \in \Sigma\}), \text{ where} \\ \Omega &= \{[_A x]_A \mid x \in L, |x| = 1\} \cup \{[_B \sigma]_B \mid \sigma \in \Sigma\} \\ \pi &= (\Sigma^+, \{([_B \sigma,]_B) \mid \sigma \in \Sigma\}, \{B\}, \{A\}) \text{ and} \\ \pi_\sigma &= (\{x \in \Sigma^+ \mid \sigma x \in L\}, \{([_A \sigma,]_A)\}, \{B\}, \{A\}). \end{aligned}$$

Since the family of context-sensitive language is closed under quotient with singleton sets, all selector languages are context-sensitive, and it is not difficult to prove $L(G) = L$.

This result shows that the combined use of selector languages and obligatory adjoining leads to a very powerful formalism. Whereas there are context-sensitive languages (such as $L = \{a^n c b^n c^n c d^n \mid n \geq 1\}$) which cannot be generated by any MBICR-grammar regardlessly of the used selector languages, the above construction shows that for each family of languages F closed under quotient with singleton sets and containing all finite languages each $L \in F$ can also be generated by a CTAG with F-choice.

Closure Properties

The class of languages generated by CTAGs with F-choice is closed under union, concatenation and Kleene-star for all families of languages F with $\Sigma^+ \in F$. Let $G_1 = (\Sigma_1, \Delta_1, \Upsilon_1, \Omega_1, P_1)$

and $G_2 = (\Sigma_2, \Delta_2, \Upsilon_2, \Omega_2, P_2)$ be two CTAGs with F-choice for a family of languages F with $\Sigma^* \in F$. Without loss of generality we may assume that $\Delta_1 \cap \Delta_2 = \emptyset$. Therefore it is easy to see that for $G = (\Sigma_1 \cup \Sigma_2, \Delta_1 \cup \Delta_2, \Upsilon_1 \cup \Upsilon_2, \Omega_1 \cup \Omega_2, P_1 \cup P_2)$ we have $L(G) = L(G_1) \cup L(G_2)$. For concatenation we take a new index $S \notin \Delta_1 \cup \Delta_2$ and construct $G' = (\Sigma_1 \cup \Sigma_2, \Delta_1 \cup \Delta_2 \cup \{S\}, \Upsilon_1 \cup \Upsilon_2 \cup \{S\}, \{[_S\alpha\beta]_S \,|\, \alpha \in \Omega_1 - \{\lambda\}, \beta \in \Omega_2 - \{\lambda\}\} \cup \{\alpha \in \Omega_1 \,|\, \lambda \in \Omega_2\} \cup \{\alpha \in \Omega_2 \,|\, \lambda \in \Omega_1\}, P_1 \cup P_2)$. Clearly $L(G') = L(G_1) \cdot L(G_2)$. For Kleene-star we construct $G'' = (\Sigma_1, \Delta_1 \cup \{S\}, \Upsilon_1 \cup \{S\}, \{[_S\alpha]_S \,|\, \alpha \in \Omega_1 - \{\lambda\}\} \cup \{\lambda\}, P \cup \{\pi\})$, where $\pi = (\Sigma^*, \{([_S\alpha]_S) \,|\, \alpha \in \Omega_1 - \{\lambda\}\}, \{S\}, \{S\})$ It is a technical exercise to prove $L(G'') = L(G_1)^*$.

For each CTAG G and regular language R we can construct a CTAG G' such that $L(G') = L(G) \cap R$. Furthermore, G' uses the same selector languages as G. Hence, this construction directly proves that the class of languages generated by CTAGs with F-choice is closed under intersection with regular languages for any family of languages F. For the relevance of closure under intersection with regular sets we refer the reader to (Lang, 1994).

In the following, we will present a sketch of the proof. Let $G = (\Sigma, \Delta, \Upsilon, \Omega, P)$ be an arbitrary CTAG and R a regular language. Without loss of generality we assume that G is in a normal form such that each internal node either has exactly one leaf or only internal nodes as immediate successors: formally for each $\alpha_1\alpha_2\alpha_3 \in T(G)$ such that $\alpha_2 \in DC_\Delta(\Sigma)$ we either have $\alpha_2 = [_A a]_A$ for some $a \in \Sigma$ and $A \in \Delta$ or $\alpha_2 = [_A\beta_1 \ldots \beta_n]_A$ for an $A \in \Delta$ and $\beta_i \in DC_\Delta(\Sigma)$, $1 \leq i \leq n$. Since R is regular, there exists a deterministic finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ with $L(M) = R$ (c.f. (Hopcroft & Ullman, 1979) for notational details). We construct a grammar G' where the label of each internal node additionally carries two pairs of states of M, formally the set of indices of G' is given by $\Phi = \{(A, [p, q], [r, s]) \,|\, A \in \Delta, p, q, r, s \in Q\}$.

Intuitively, in the tree interpretation, if an internal node is labelled by $(A, [p, q], [r, s])$ then $[p, q]$ is a value propagated from the immediate predecessor of the node stating that this node is supposed to generate a yield w such that $\delta(p, w) = q$. The pair $[r, s]$ denotes that the immediate successors of the node are supposed to generate a yield w such that $\delta(r, w) = s$.

G' generates as sentential forms exactly the sentential forms of G where a label of an internal node A in $S(G)$ is replaced by all labels $(A, [p, q], [r, s])$, $p, q, r, s \in Q$, in $S(G')$ such that for the resulting strings $\alpha \in S(G')$ the following properties hold:

(1) For each partition $\alpha = \alpha_1\alpha_2\alpha_3$ such that $\alpha_2 \in DC_\Delta(\Sigma)$ and $\alpha_2 = [_X\gamma_1 \ldots \gamma_n]_X$ we have $X = (A, [p, q], [p_0, p_n])$, $\gamma_i \in DC_\Delta(\Sigma)$ and $\gamma_i = [_{Y_i}\gamma_i']_{Y_i}$, $Y_i = (B_i, [p_{i-1}, p_i], [r_i, s_i])$, $1 \leq i \leq n$. In other words, for each internal node with other internal nodes as immediate successors, the second pair of states of the node is consistent with the first pairs of states of its immediate descendants (in the sense of the usual triple costruction). See Figure 3 for an illustration.

(2) For each partition $\alpha = \alpha_1\alpha_2\alpha_3$ such that $\alpha_2 \in DC_\Delta(\Sigma)$ and $\alpha_2 = [_X\sigma]_X$ where $X = (A, [p, q], [r, s])$ and $\sigma \in \Sigma$ we have $\delta(r, \sigma) = s$. In other words, for all internal nodes having a leaf labelled by σ as immediate successor we have $\delta(r, \sigma) = s$ for the second pair of states $[r, s]$.

(3) For each $\alpha = [_X\alpha']_X$ we have $X = (A, [q_0, f], [r, s])$ where q_0 is the initial state of M and f is a final state of M, $f \in F$. In other words, the first pair of states of the root node of each tree consists of M's initial state and a final state of M.

The details of converting the axioms and contexts of G into axioms and contexts of G' are omitted due to the limited space. The conversion leaves the selector languages untouched, so

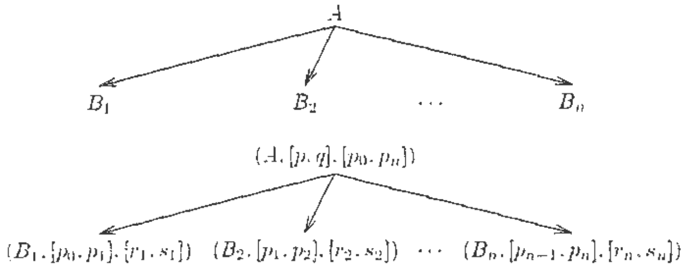
A
B_1
B_2
$\cdots$
B_n
$(A, [p, q], [p_0, p_n])$
$(B_1, [p_0, p_1], [r_1, s_1])$
$(B_2, [p_1, p_2], [r_2, s_2])$
$\cdots$
$(B_n, [p_{n-1}, p_n], [r_n, s_n])$

Figure 3: Example for a part of a tree in $S(G')$ corresponding to a part of a tree in $S(G)$. The above part of a tree with root labelled by A and immediate nonterminal successors $B_1, \ldots, B_n$ is converted into all parts of the above form for arbitrary $p, q, p_i, r_i, s_i \in Q$, $0 \leq i \leq n$ (not considering further restrictions due to the immediate predecessor or the immediate descendants of this part of the tree).

G' uses the same selector languages as G. If we define the set of permitted indices of G' by $\Phi' = \{(A, [p, q], [p, q]) \mid A \in \Upsilon, p, q \in Q\}$ we obtain $L(G') = L(G) \cap R$.

The same construction can also be used to show the closure of TAL under intersection with regular sets without involving a corresponding automata model like EPDAs.

Conclusion and Further Work

In this paper, we introduced CTAGs and discussed their generative capacity and some closure properties. CTAGs seem a significant progress compared to MBICR-grammars. As allowing both obligatory adjoining and selector languages leads to a very powerful model, our future work will focus on CTAGs with "weak" selector languages. Open problems which we would like to tackle in the future are whether the classes of languages generated by such grammars are closed under homomorphism and inverse homomorphism or not and the relationship to other formalisms such as range concatenation grammars and recursive matrix systems.

References

EHRENFEUCHT A., PĂUN G. & ROZENBERG G. (1997). Contextual grammars and formal languages. In G. ROZENBERG & A. SALOMAA, Eds., *Handbook of Formal Languages Vol.2: Linear Modeling*, p. 237–294. Berlin: Springer.

HOPCROFT J. E. & ULLMAN J. D. (1979). *Introduction to Automata Theory, Languages and Computation*. Reading, MA, USA: Addison-Wesley.

KAPPES M. (1999). Combining contextual grammars and tree adjoining grammars. In *Proceedings of the 6. Meeting on Mathematics of Language*, p. 337–345, Orlando.

LANG B. (1994). Recognition can be harder than parsing. *Computational Intelligence*.

MARCUS S. (1997). Contextual grammars and natural languages. In G. ROZENBERG & A. SALOMAA, Eds., *Handbook of Formal Languages Vol.2: Linear Modeling*, p. 215–235. Berlin: Springer.

MARTIN-VIDE C. & PĂUN G. (1998). Structured contextual grammars. *Grammars*, 1 (1 p.), 33–55.

PĂUN G. (1997). *Marcus Contextual Grammars*. Dordrecht, Boston, London: Kluwer Academic Publishers.

VIJAY-SHANKER K. (1988). *A Study of Tree Adjoining Grammars*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA.