Algorithm to Count the Number of Signed Paths in an Electrical Network via Boolean Formulas.

Guillermo De Ita Luna*, Helena Gómez* and Belem Merino*

ABSTRACT

This article presents a practical method to count the different signed paths which maintain an electric charge on each one of the lines of an electrical network. We assume that there is just one charge (positive or negative) on each network node. We model the problem of counting the signed paths via the #2SAT problem. The #2SAT problem consists on counting models of Boolean formulas in two conjunctive forms. Our method is based on the topology of the graph representing the electrical network and from which we get its Boolean formula in two conjunctive form. A set of recurrence equations are applied, starting from the terminal nodes up to the root node of the network. Such recurrence equations allow us to compute #2SAT for the formula associated to the electrical network. The computed value (#2SAT) represents the different ways to keep charge on all line of the electrical network.

RESUMEN

Este artículo presenta un método práctico para contar los diferentes caminos signados los cuales mantienen una carga eléctrica sobre cada una de las líneas de una red eléctrica. Consideramos que hay sólo una carga (positiva o negativa) en cada nodo de la red. Nosotros modelamos el problema de contar los caminos signados vía el problema #2SAT. El problema #2SAT consiste en contar modelos de fórmulas booleanas en dos forma conjuntiva.Nuestro método esta basado en la topología del grafo que representa la red eléctrica y de la cual se obtiene su fórmula booleana en dos forma conjuntiva. Un conjunto de ecuaciones de recurrencia son aplicadas, partiendo de los nodos terminales hacia el nodo raíz de la red. Tales ecuaciones de recurrencia nos permiten calcular el valor #2SAT para la fórmula asociada a la red eléctrica. El valor calculado (#2SAT) representa las diferentes formas de mantener carga sobre todas las líneas de la red eléctrica.

Recibido: 7 de Enero del 2012
Aceptado: 14 de Febrero del 2012

INTRODUCTION

The problem of propositional *satisfiability* (SAT) is a classic NP-complete problem, but there is a wide range of research on how to identify limited instances for which the SAT problem can be solved efficiently.

The problem of counting models of a boolean formula, problem denoted as ($\#SAT$), has been relevant in the Artificial Intelligence area (AI). For example, it has been known that #SAT has been used for modeling problems as: for estimating the degree of reliability in a communication network, for computing the degree of belief in propositional theories, for the generation of explanations to propositional queries, in Bayesian inference, in a truth maintenance systems and for repairing inconsistent databases [3, 4, 5, 9, 10]. Those previous problems come from several AI applications such as planning, expert systems, reasoning, etc.

$\#SAT$ is as difficult as the SAT problem, but even when SAT can be solved in polynomial time, is not known an efficient computational method for $\#SAT$. For example, the 2-SAT problem (SAT limited to consider two conjunctive forms) can be solved in linear time. However, the corresponding counting problem #2-SAT is a #P-complete problem. The maximum polynomial class recognized for #2SAT is the class $(\leq 2, 2\mu)$-CF (conjunction of binary or unary clauses where each variable appears twice at most) [5, 9].

Palabras clave:
Problema SAT; problema #SAT, modelos de conteo, grafo con signo, redes eléctricas.
Keywords:
SAT problem, #SAT problem; counting models; signed graph; electrical networks.

* Computer Sciences Department, Benemérita Universidad Autónoma de Puebla Edif.104, 14 Sur con San Claudio, Sn. Manuel, C.P. 72570, Puebla, Pue., México Email: deita@cs.buap.mx, helena.adorno@gmail.com, poxbelem@hotmail.com

Here, we extend such class for considering the topological structure of the undirected graph induced by the restrictions (clauses) of the formula.

We start considering *signed graphs* (graphs whose edges are designated positive or negative). That class of graphs have been very useful for modeling different type of problems in the social sciences area [6, 7, 8]. For example, the signed graphs has been used for modeling the interaction among a group of persons and the type of relationship between certain pair of individuals of the group. Special focus has been given on social inequalities, for example, what is it about such characteristics as sex, race, occupation, education, and so on that leads to inequalities in social interaction? [8].

We consider here an extension of the signed graphs where all edge in the graph has associated a pair of signs. In this way, any binary clause can be represented by such signed edges. And we model an electric network by those extended signed graphs.

We are interested in computing the different assignments (charges) associated to the variables (nodes) in the network which allow that all edge in the network keep a charge transmitted by at least one of its two endpoints (nodes).

The aim is to develop a method to count the different signed paths in which an electric network held energy on each of its electric lines, by counting the number of satisfy assignments (models) of the Boolean formula associated to the electrical network.

METHODOLOGY

Below we summarize a basic set of concepts through which we try to improve a more complete understanding of terms that will be used in this work.

Topology: Systematic study of the components of an electrical network and its reciprocal links [2].

Element: Individual constituent of a network with two or more terminals or bounds that are used for interconnection with other elements [2].

Nodes and branches: The junction points of the elements in a network are called nodes. Several elements can join together and to be considered as a single unit or a whole. When several elements are joined together so that the flow is the same in all of them, we say that the alements are connected in serie. The junctions between two components are called internal nodes. The set of elements in series between the nodes themselves are called branches [2].

Graph: It is the geometric, topological, representation of a network, forming a backbone of the physical layout of the network [2]. A graph G is represented by a pair (V, E) where V is the set of nodes and E the set of edges.

A signed graph Γ (also called sigraph) is an ordered pair $\Gamma = (G, \sigma)$ where $G = (V, E)$ is a graph called the underlying graph of Γ and $\sigma : E \rightarrow \{+, -\}$ is a function called a *signature* or *signing*. $E^{+}(\Gamma)$ denotes the set of edges from E that are mapped by σ to '+', and $E^{-}(\Gamma)$ denotes the set of edges from E that are mapped by σ to '-'.

The elements of $E^{+}(\Gamma)$ are called positive edges and those of $E^{-}(\Gamma)$ are called negative edges of Γ. A signed graph is *all-positive* (respectively, *all-negative*) if all of its edges are positive (negative); further, it is said that Γ is *homogeneous* if Γ is either all-positive or all-negative, otherwise Γ is *heterogeneous* .

Signed graphs have been very useful for modeling interactions among a group of persons and for representing the type of relationship between certain pair of individuals of the group. Special focus is on the social inequalities [8].

Given a set of de n Boolean variables, $X = \{x_1, x_2, ..., x_n\}$. It is called a *literal* to any variable x or the negation $\overline{x}$ of it. We use $v(l)$ to indicate the variable involved by the literal l.

The disjunction of different literals is called a *clause*. For $k \in \mathbb{N}$, a $k-clause$ is a *clause* consisting of exactly k *literals*. A variable $x \in X$ appears in a clause c if x or $\overline{x}$ is an element of c. Let $v(c) = \{x \in X : x \text{ appears in } c\}$. A conjunctive form (CF) is a conjunction of *clauses*. A $k-CF$ is a CF containing only $k-clauses$ and, $(\leq k)-CF$ denotes a CF containing *clauses* with at most k literals.

Let Σ be a 2-CF, an assignment s for Σ is a function $s : v(\Sigma) \rightarrow \{0, 1\}$. An assignment can also be considered as a set of non-complementary pairs of literals. If $l \in s$, being s an assignment, then s makes l *true* and makes $\bar{l}$ *false*. A *clause* c *is satisfied* if and only if $c \cap s \neq \emptyset$, and if for all $l \in c, \bar{l} \in s$ then s falsifies c.

A CF Σ is satisfied by an assignment s if each *clause* in Σ is satisfied by s and Σ is contradicted if it is not satisfied. s is a model of Σ if s is a satisfied assignment of Σ.

Let $SAT(\Sigma)$ be the set of models that Σ has over its variables: $v(\Sigma)$. Σ is a contradiction or unsatisfiable if $SAT(\Sigma) = \emptyset$. Let $\mu_{v(\Sigma)}(\Sigma) = |SAT(\Sigma)|$, be the cardinality of $SAT(\Sigma)$. Given Σ a CF, the SAT problem consists in determining if Σ has a model. The $\#SAT$ consists of counting the number of models of F defined over $v(\Sigma)$. We will also denote $\mu_{v(\Sigma)}(\Sigma)$ by $\#SAT(\Sigma)$ [1].

Given a signed graph G_Σ, we obtain the associated Boolean formula Σ. Σ can be expressed as a two Conjunctive Form (2-CF) in the following way. Let $G_\Sigma = ((V, E), \sigma)$ be the signed graph, then $v(\Sigma) = V$ and for all positive edge $x \overset{+}{-} y$ in G_Σ the clause (x, y) is part of Σ, while for a negative edge $x \overset{-}{-} y$ in G_Σ the clause $(\overline{v(x)}, \overline{v(y)})$ is part of Σ. This means that the vertices of G_Σ are the variables of Σ, and each signed edge in E represents a clause in Σ.

BUILDING RECURRENCE EQUATIONS FOR COUNTING SIGNED PATHS IN AN ELECTRICAL NETWORK

The signed graphs are very useful to represent some kind of relationships among individuals. In this case, each person is a node of the graph and there is an edge between $\{x, y\}$ if x is in some relation to y. Many of the relationships of interest have natural opposites, for example: likes/dislikes, associates with/avoids, and so on [8]. For this, one of the signs: $\{+, -\}$ is used as label of each edge of the graph.

For example, if we consider that x likes y, then a positive edge is denoted between x and y. But this representation does not show what happens about the relation from y to x. y might dislike x or maybe like him. In order to be more precise in the kind of relationships between two individuals is better to consider two signs in each edge, one sign associated with each end-point of the edge.

Furthermore, in order to consider general Boolean formulas in 2-CF, we extend the concept of signed graphs. Instead to consider just one sign as label of an edge of the graph $G = ((V, E), \sigma)$, we consider here that all edge in E has associated a pair of signs. Then the signed function σ is of type $\sigma : E \to \{(+,+), (+,-), (-,+), (-,-)\}$ which gives a pair of signs to each edge of G.

In this way, all binary clause $\{x, y\}$ can be represented by a signed edge in the graph in a natural way without importance of the signs associated to the variables in the clause. For example the clause $\{\overline{x}, y\}$ will be represented as the signed edge: $x \overset{-+}{—} y$ which is equivalent to $y \overset{+-}{—} x$, and in this case, '-' is called the adjacent sign of x and '+' will be the adjacent sign of y.

Moreover, we can consider those extended signed graphs as an electric network. $\Sigma = (G, \sigma)$ where $G = (V, E)$ is the underlying graph of Σ and $\sigma : E \to \{(+,+), (+,-), (-,+), (-,-)\}$ is the signature function. That means that all edge in the network is energized in its both end-points.

Here, we are interested in count the different assignments associated to the nodes of the network in such a way that all edge in the network will be satisfied by at least one of its two end-points. That means, that we want that all line in the network keep the charge from at least one of the two possible sources (its end-points). For this, we start analyzing the way to build satisfied assignment in the network considering first, the most simple topologies associated with the underlying graph of the electrical network.

If the Electrical Network is a Linear Path

Let us consider that the electrical network: $G_\Sigma = (V, E)$ *is a linear path*. Let us write down its associated formula Σ, without a loss of generality (ordering the clauses and its literals, if it were necessary), as: $\Sigma = \{c_1, ..., c_m\} = \left\{\{x_1^{\epsilon_1}, x_2^{\delta_1}\}, \{x_2^{\epsilon_2}, x_3^{\delta_2}\}, \ldots, \{x_m^{\epsilon_m}, x_{m+1}^{\delta_m}\}\right\}$, where $|\upsilon(c_i) \cap \upsilon(c_{i+1})| = 1$, $i \in [\![m-1]\!]$, and $\delta_i, \epsilon_i \in \{0, 1\}$, $i = 1, ..., m$.

In order to compute the number of signed paths in such electrical network, we start with a pair of values (α_i, β_i) associated with each node i of the network. We call the charge of the node i to the pair (α_i, β_i). The value (α_i) indicates the number of times that node i takes the positive value and (β_i) indicate the number of times that node i takes the negative value into the set of satisfied assigments of Σ.

Let f_i be a family of clauses of Σ built as follows: $f_0 = \emptyset, f_i = \{c_j\}_{j \le i}$, $i \in [\![m]\!]$. Note that $f_i \subset f_{i+1}$, $i \in [\![m-1]\!]$. Let $SAT(f_i) = \{s : s \ satisfies \ f_i\}$, $A_i = \{s \in SAT(f_i) : x_i \in s\}$, $B_i = \{s \in SAT(f_i) : \overline{x}_i \in s\}$. Let $\alpha_i = |A_i|$; $\beta_i = |B_i|$ and $\mu_i = |SAT(f_i)| = \alpha_i + \beta_i$. From the total number of models in $\mu_i, i \in [\![m]\!]$, there are α_i of which x_i takes the logical value 'true' and β_i models where x_i takes the logical value 'false'.

For example, $c_1 = (x_1^{\epsilon_1}, x_2^{\delta_1})$, $f_1 = \{c_1\}$, and $(\alpha_1, \beta_1) = (1, 1)$ since x_1 can take one logical value 'true' and one logical value 'false' and with whichever of those values satisfies the subformula f_0 while $SAT(f_1) = \{x_1^{\epsilon_1} x_2^{\delta_1}, x_1^{1-\epsilon_1} x_2^{\delta_1}, x_1^{\epsilon_1} x_2^{1-\delta_1}\}$, and then $(\alpha_2, \beta_2) = (2, 1)$ if δ_1 were 1 or rather $(\alpha_2, \beta_2) = (1, 2)$ if δ_1 were 0.

In general, we can compute the values for (α_i, β_i)

associated to each node x_i, $i = 2, .., m$, according to the signs (ϵ_i, δ_i) of the literals in the clause c_i, by the next recurrence equation:

$$(\alpha_i, \beta_i) = \begin{cases} (\beta_{i-1}, \alpha_{i-1} + \beta_{i-1}) \ if (\epsilon_i, \delta_i) = (-,-) \\ (\alpha_{i-1} + \beta_{i-1}, \beta_{i-1}) \ if (\epsilon_i, \delta_i) = (-,+) \\ (\alpha_{i-1}, \alpha_{i-1} + \beta_{i-1}) \ if (\epsilon_i, \delta_i) = (+,-) \\ (\alpha_{i-1} + \beta_{i-1}, \alpha_{i-1}) \ if (\epsilon_i, \delta_i) = (+,+) \end{cases} \quad (1)$$

In these recurrences, (ϵ_i, δ_i) represent the signs associated with the literals of the clause c_i. We denote with $' \rightarrow '$ the application of one of the four rules of the recurrence (1). Thus, the expression $(2, 3) \rightarrow (5, 2)$ denotes the application of one of the rules (in this case, the rule 4), over the pair $(\alpha_{i-1}, \beta_{i-1}) = (2, 3)$ in order to obtain $(\alpha_i, \beta_i) = (\alpha_{i-1} + \beta_{i-1}, \alpha_{i-1}) = (5, 3)$.

These recurrence equations allow to carry the count of the different ways to keep all line of the electrical network with charge. In fact, the sum $\alpha_m + \beta_m$ obtained in the last node of the network, indicates the total number of models that the Boolean formula associated with the network has. The value $\alpha_m + \beta_m$ also represents the number of different ways to keep charge on all electric line in the electrical network.

We give the following examples for illustrating how the recurrence is applied for specific cases, although equation (1) considers all the possible combinations of signs on the edges.

Example 1 *Let* $\Sigma = \{(x_1, x_2), (x_2, \overline{x}_3), (\overline{x_3}, x_4), (x_4, x_5)\}$ *be a 2-CF which conforms a linear path, the series* $(\alpha_i, \beta_i), i \in [\![5]\!]$, *is computed as:* $(\alpha_1, \beta_1) = (1, 1) \rightarrow (\alpha_2, \beta_2) = (2, 1)$ *since* $(\epsilon_1, \delta_1) = (1, 1)$, *and the rule 4 has to be applied. In general, applying the corresponding rule of the recurrence (1) according to the signs expressed by* $(\epsilon_i, \delta_i), i = 1, ..., 5$, *we have* $(2, 1) \rightarrow (\alpha_3, \beta_3) = (2, 3) \rightarrow (\alpha_4, \beta_4) = (5, 3) \rightarrow (\alpha_5, \beta_5) = (8, 5)$, *and then,* $\#SAT(\Sigma) = \mu(\Sigma) = \mu_5 = \alpha_5 + \beta_5 = 8 + 5 = 13$ *models.*

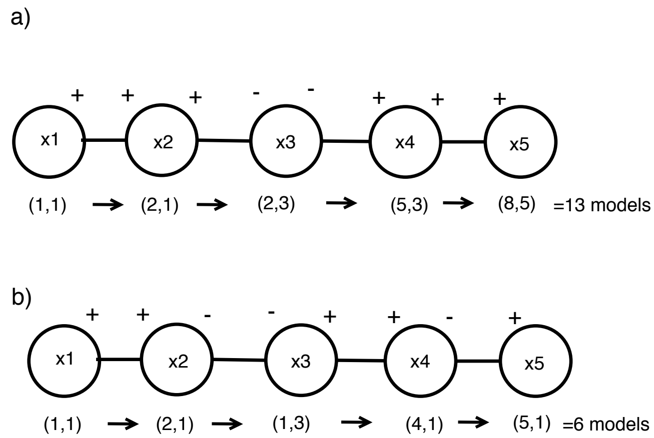

Figure 1 . a) Counting models over a monotone electrical path network b) Counting models over a general electrical path network

Example 2 *Let* $\Sigma = \{(x_1, x_2), (x_2, \overline{x}_3), (\overline{x}_3, x_4), \ (x_4, x_5)\}$ *be a path, the series* $(\alpha_i, \beta_i), i \in [\![5]\!]$, *is computed according to the signs of each clause, as it is illustrated in figure (1a). A similar path with 5 nodes but with different signs in the edges is shown in figure (1b).*

If Σ is a path, we apply (1) in order to compute $\mu(\Sigma)$. The procedure has a linear time complexity over the number of variables of Σ, since (1) is applied on each node while we are traversing by the path, from the initial node x_1 to the final node x_n.

If the Electrical Network is a Tree

Given a constrained graph G_Σ of a $2 - CF\Sigma$, if we apply a depth-first search on G_Σ and none backedges are found, then G_Σ is a tree and we can consider any node of G_Σ as the root node of the tree.

The charge of the nodes (α_i, β_i) for $i = 1, ...n$, are calculated while the nodes are visited in a post-order traversal. The first pair of values (α_i, β_i) for each terminal node in the network (tree leaves) is initialized with the value $(1, 1)$. So, traversing by the tree in post-order; we start by the most left child-node first, followed by the right child-node, and so on, until visit all child-node and after, we visit the father node.

At each visit of a child node $i-1$ to a father node i, the new values (α_i, β_i) associated with the father node are computed in the following way.

Algorithm Count_Models (A_Σ)

Input: A_Σ the tree defined by the electrical network G_Σ

Output: The number of signed paths in the electrical network, where a signed path means that all line in the network keep one of the charge of its two end-points.

Procedure: Traversing A_Σ in post-order, and when a node $v \in A_\Sigma$ is left (all of its edges have been processed) assign:

1. $(\alpha_v, \beta_v) = (1, 1)$, If v is a leaf node in A_Σ

2. If v is a father node with a list of child nodes associated, i.e., $u_1, u_1, ...u_k$, are the child nodes of v, then the charge of the father-node (α_v, β_v) is computed in the following way. As we have already visited all the child nodes of v, then each pair $(\alpha_{u_j}, \beta_{u_j})$ $j = 1, ..., k$ has been determined. Then, $(\alpha_{v_i}, \beta_{v_i})$ is obtained by apply the recurrences in (1) over $(\alpha_{i-1}, \beta_{i-1}) = (\alpha_{u_j}, \beta_{u_j})$. This step is iterated until computes all the values $(\alpha_{v_j}, \beta_{v_j})$, $j = 1, \ldots, k$. And finally, let $\alpha_v = \prod_{j=1}^{k} \alpha_{v_j}$ y $\beta_v = \prod_{j=1}^{k} \beta_{v_j}$

3. If v is the root node of A_Σ then returns$(\alpha_v + \beta_v)$

This procedure returns the number of signed paths for an electrical network in time $O(n+m)$ which is the necessary time for traversing G_Σ in post-order.

Example 3 *Let* $\Sigma = \{(\overline{x}_1, \overline{x}_2), (x_2, \overline{x}_3), (\overline{x}_5, x_3), (x_3, x_4), (x_8, x_7), (\overline{x}_9, \overline{x}_7), (\overline{x}_{10}, \overline{x}_7), (x_6, x_7), (x_{11}, x_{12}), (\overline{x}_{13}, \overline{x}_{12}), (x_{12}, x_6), (x_6, \overline{x}_4)\}$ *be a* $2-CF$ *obtained from a connectivity graph of an electrical network. The tree generated by the formula as well as the number of signed paths in each level of the tree is shown in figure 2. The procedure* $Count\_Models$ *returns for* $\alpha_{x_1} = 168$, *and for* $\beta_{x_1} = 84$. *Thus, the total number of signed paths is* $\#SAT(\Sigma) = 168+84 = 252$.

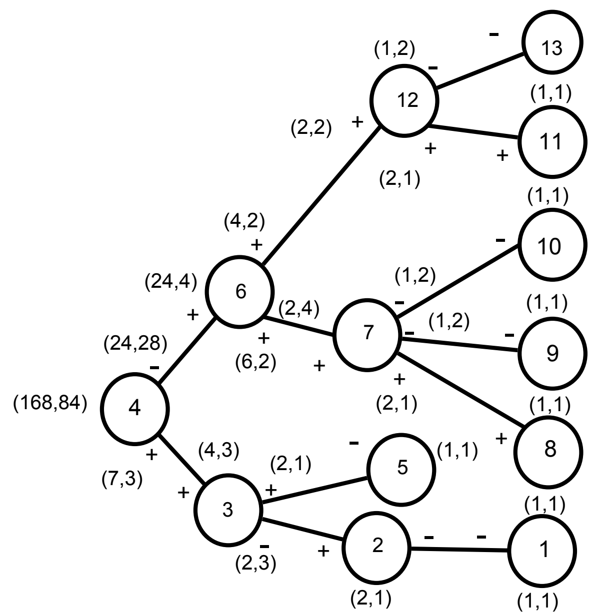

Figure 2 . The tree graph for the formula of the example 3

The order of processing for the nodes of the tree in figure 2, is: $1-2, 8-7, 9-7, 10-7, 11-12, 13-12, 2-3, 5-3, 7-6, 12-6, 3-4, 6-4$. Notice that there are as equal number of application of recurrence (1) as edges in the tree. And the step 2 in the procedure $Count\_Models$ is applied as much as the number of internal nodes are in the tree. That is, because the traverse of a tree in post-order is in $O(n+m)$, where m is the number of edges and n the number of nodes in the tree.

CONCLUSION

We present an extension of the signed graphs which are relevant for modeling new class of problems. For example, we model problems associated to electrical networks using the new extended signed graphs.

We show that the problem of counting the number of models of a Boolean formula can be applied to count the number of signed paths in an electrical network. We have designed an exact algorithm which in linear time computes #2SAT for formulas where its constrained graph is acyclic.

We have presented different topologies: paths and trees which represent the backbone of an electrical network and where to compute the number of energized paths of the network is done for computing the value #2SAT of the formula associated with that graph.

REFERENCES

[1] De Ita G., Bello P. y Contreras M. (2007). New Polynomial Classes for #2SAT Established Via Graph-Topological Structure. *Engeneering Letters.*

[2] Lópina M., Rodríguez C. (1994). *Ecuaciones de redes Topología* , Facultad de Ingenieria - Universidad de Buenos Aires.

[3] Angelsmark O., Jonsson P. (2003). Improved Algorithms for Counting Solutions in Constraint Satisfaction Problems. *In ICCP: Int. Conf. on Constraint Programming.*

[4] Darwiche, A. (2001). On the Tractability of Counting Theory Models and its Application to Belief Revision and Truth Maintenance. *Jour. of Applied Non-classical Logics,* 11(1-2):11-34.

[5] De Ita G. (2004). Polynomial Classes of Boolean Formulas for Computing the Degree of Belief. *Advances in Artificial Intelligence LNAI,* 3315:430-440.

[6] Germina K.A. Hameed S.K. (2010). On signed paths, signed cycles and their energies. *Applied Mathematical Sciences,* 4(70):3455-3466.

[7] Kota P. S., Subramanya M.S. (2010). Note on path signed graphs. *Notes on Number Theory and Discrete Mathematics* 15(4):1-6.

[8] Fred S. Roberts. (1978). *Graph Theory and Its Applications to Problems of Society.* Edit. SIAM Collection.

[9] Roth, D. (1996). On the hardness of approximate reasoning. *Artificial Intelligence,* 82:273-302.

[10] Vadhan, S. P. (2001). *The complexity of Counting in Sparse, Regular, and Planar Graphs* , SIAM Journal on Computing, 31(2):398-427.