On Constructing Minimal Formulae

Paul E. Dunne

Dept. of Computer Science, The University of Liverpool, Liverpool, United Kingdom

Abstract

Given a Boolean propositional formula, $\varphi(X_n)$ over the basis $\Omega = \{\wedge, \vee, \neg\}$ we consider the following decision problem: is there a subset of literals, S, for which $\varphi(X_n) \equiv \bigwedge_{y \in S} y$ or $\varphi(X_n) \equiv \bigvee_{y \in S} y$? We prove that the "obvious" Σ_2^p upper bound is sub-optimal and that the problem is decidable in $\mathrm{P}_{||}^{\mathrm{NP}}$ the class of languages decidable by polynomial time methods allowed to make non-adaptive queries to an NP oracle. We further show that the associated function problem of computing a witnessing such subset when one exists can be solved in $\mathrm{FP}_{||}^{\mathrm{NP}}$.

Key words:

1. Introduction

A long-standing problem in computational complexity theory concerns how difficult it is to determine whether a given propositional formula, $\varphi(X_n)$, can be rewritten as an equivalent formula, $\psi(X_n)$, whose size is strictly smaller. Here the *size* of a formula is interpreted as the number of occurences of literals – x and $\neg x$ – in the formula's specification. This problem was studied by Hemaspaandra and Wechsung [11, 12] who obtained the first non-trivial lower bounds on its complexity. These establish that, in its most general form, *Minimal Expression Equivalence* (MEE) is $\mathrm{P}_{||}^{\mathrm{NP}}$–hard. To date, however, no improvement to the obvious Σ_2^p upper bound has been obtained, i.e. the approach which given an instance $\langle \varphi, k \rangle$ of MEE, proceeds by guessing a formula ψ of size at most k and accepts if and only if $\varphi \equiv \psi$.

In this note, although we do not obtain improved bounds for the general MEE problem, we are able to show that a related problem, in which the *structure* of witnessing equivalent formulae is very tightly constrained, can be decided in $\mathrm{P}_{||}^{\mathrm{NP}}$. Furthermore our proof of this upper bound immediately yields an algorithm for *constructing* such a set: this algorithm places the related search problem in the function class $\mathrm{FP}_{||}^{\mathrm{NP}}$. The exact variant of MEE we consider will be called *Term Equivalence* (TE) and is defined as follows:

Term Equivalence (TE)

Instance: $\varphi(X_n)$ propositional formula over variables $\{x_1, \ldots, x_n\}$ and logical operations $\{\wedge, \vee, \neg\}$.

Question: $\exists\, S \subset \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$ such that either

$$\varphi(X_n) \;\equiv\; \left(\bigwedge_{y\in S} y\right) \text{ or } \varphi(X_n) \;\equiv\; \left(\bigvee_{y\in S} y\right) \;?$$

We denote by FTE the associated function problem, i.e.

$$\mathrm{FTE}(\varphi(X_n)) \;=\; \begin{cases} \text{undefined} & \text{if} \quad \varphi \not\in \mathrm{TE} \\ \min\; S & \text{if} \quad \varphi \equiv \wedge_{y\in S}\, y \;\text{ or } \varphi \equiv \vee_{y\in S} y \end{cases}$$

Here $\min$ is with respect to $\subseteq$ and thus covers the cases where φ is equivalent to a Boolean constant function so that $S = \emptyset$ is reported if $\varphi \equiv \top \equiv \wedge_{y\in S}\, y$ or $\varphi \equiv \bot \equiv \vee_{y\in S}\, y$.

The upper bound methods combine a translation from formulae to directed graph structures with recent complexity results on identifying subsets of vertices satisfying specific criteria from Dunne [7, 8]. The study of such sets originates from the concept of so-called "*extension semantics*" in the argument systems pioneered in work of Dung [3]. Readers interested in a general overview of these and related systems are referred to the survey of Bench-Capon and Dunne [1]; a detailed introduction to complexity-theoretic work in this model is provided in Dunne and Wooldridge [10].

In the remainder of this note we present background definitions including the graph-theoretic terminology subsequently used in Section 2. Section 3 describes the translation from formulae over the basis $\{\wedge, \vee, \neg\}$ to directed graphs that are referred to as *formula graphs*. Our main result is presented in Section 4 with conclusions given in Section 5.

2. Background Definitions

A propositional formula, $\varphi(X_n)$ where $X_n = \{x_1, \ldots, x_n\}$ is a set of n Boolean variables is any structure built according to the following rules.

a. A *literal* $y \in \{x_1, x_2, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$ is an $\wedge$-*formula* and also an $\vee$-formula.

b. If $\{\varphi_1, \ldots, \varphi_k\}$ are all $\vee$-formulae then $(\varphi_1)\wedge(\varphi_2)\cdots\wedge(\varphi_k)$ is an $\wedge$-formula.

c. If $\{\varphi_1, \ldots, \varphi_k\}$ are all $\wedge$-formulae then $(\varphi_1)\vee(\varphi_2)\cdots\vee(\varphi_k)$ is an $\vee$-formula.

d. ψ is a $\{\wedge, \vee, \neg\}$-formula (or more simply just formula) if ψ is the result of a finite number of applications of (a)–(c) above.

We say an $\wedge$-formula is an *elementary conjunction* if it has the from $\wedge_{y \in S}\ y$ for some subset S of $\{x_1, x_2, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$ such that at most one of x_i, $\neg x_i$ belongs to S ($1 \leq i \leq n$). Similarly an $\vee$-formula is an *elementary disjunction* if it is of the form $\vee_{y \in S}\ y$ (where again S does not contain both a literal and its negation).

We note a number of points concerning this definition. Firstly we do not restrict the operations $\wedge$ and an $\vee$ to be purely *binary*. Secondly we do not explicitly allow the constant symbols $\perp$ (false) or $\top$ (true) as formulae.

Suppose $\pi = \langle p_1, \ldots, p_n \rangle \in \langle \bot, \top, * \rangle^n$ is a *partial assignment* of Boolean values to the propositional variables $\langle x_1, \ldots, x_n \rangle$ defining some formula $\varphi(x_1, \ldots, x_n)$. The *value* $\varphi(\alpha) \in \{\bot, \top, *\}$ is recursively defined via:

1. If $\varphi(X_n)$ is the literal x_i then $\varphi(\pi) = p_i$.
2. If $\varphi(X_n)$ is the literal $\neg x_i$ then $\varphi(\pi) = \neg p_i$ if $p_i \neq *$ and $*$ otherwise, i.e. $\top$ if $p_i = \bot$; $\bot$ if $p_i = \top$.
3. If $\varphi(X_n)$ is an $\wedge$-formula formed from $\vee$-formulae $\{\varphi_1, \ldots, \varphi_k\}$ then:
   a. $\varphi(\pi) = \top$ if $\varphi_j(\pi) = \top$ for all $1 \leq j \leq k$.
   b. $\varphi(\pi) = \bot$ if $\varphi_j(\pi) = \bot$ for some $1 \leq j \leq k$.
   c. $\varphi(\pi) = *$ otherwise, i.e no $\varphi_j(\pi) = \bot$ and at least one $\varphi_j(\pi) = *$.
4. If $\varphi(X_n)$ is an $\vee$-formula formed from $\wedge$-formulae $\{\varphi_1, \ldots, \varphi_k\}$ then:
   a. $\varphi(\pi) = \bot$ if $\varphi_j(\pi) = \bot$ for all $1 \leq j \leq k$.
   b. $\varphi(\pi) = \top$ if $\varphi_j(\pi) = \top$ for some $1 \leq j \leq k$.
   c. $\varphi(\pi) = *$ otherwise, i.e no $\varphi_j(\pi) = \top$ and at least one $\varphi_j(\pi) = *$.

Two formulae, $\varphi(X_n)$ and $\psi(X_n)$ are *logically equivalent* – denoted $\varphi(X_n) \equiv \psi(X_n)$ – if for all assignments $\alpha \in \langle \bot, \top \rangle^n$ it holds that $\varphi(\alpha) = \psi(\alpha)$. We note that logical equivalence is well-defined in the case when the propositional variables, Y, defining ψ are a *strict subset* of those defining φ: in this case $\varphi(X) \equiv \psi(Y)$ if for every assignment α of X, $\varphi(\alpha) = \psi(\beta)$ where β is the projection of the assignment α onto the variables $Y \subset X$.

Through the *normal forms* conjunctive normal form (CNF) and disjunctive normal form (DNF) it well known that given any Boolean *function* $f : \langle \bot, \top \rangle^n \to \langle \bot, \top \rangle$ there is an $\wedge$-formula, $\psi_\wedge$ and an $\vee$-formula $\psi_\vee$ for which $\psi_\wedge(X_n) \equiv \psi_\vee(X_n) \equiv f(X_n)$, i.e. $\forall\ \alpha \in \langle \bot, \top \rangle^n\ \psi_\wedge(\alpha) = \psi_\vee(\alpha) = f(\alpha)$, see, e.g. [5, pp. 12–13].

The *size* of a formula $\varphi(X_n)$ (denoted $|\varphi(X_n)|$) is the number of occurences of literals used in defining it, i.e

$$|\varphi(X_n)| \;=\; \begin{cases} 1 & \text{if} \quad \varphi(X_n) = y \text{ for some literal } y \\ \sum_{i=1}^{k} \; |\varphi_i(X_n)| & \text{if} \quad \varphi(X_n) \text{ is an } \wedge\text{-formula formed from } \{\varphi_1, \ldots, \varphi_k\} \\ \sum_{i=1}^{k} \; |\varphi_i(X_n)| & \text{if} \quad \varphi(X_n) \text{ is an } \vee\text{-formula formed from } \{\varphi_1, \ldots, \varphi_k\} \end{cases}$$

We next introduce a number of graph-theoretic concepts which will be important in our translation of formulae to directed graphs and in the upper bound arguments of Section 4. Throughout the sequel, $H(V, F)$ is a *directed* graph with vertices $V = \{v_1, \ldots, v_n\}$ and edges $F \subseteq V \times V$. For $S \subseteq V$ the sets S^- and S^+ are

$$\begin{aligned} S^- &= \{\, w \;:\; \exists\, v \in S \text{ for which } \langle w, v\rangle \in F\} \\ S^+ &= \{\, w \;:\; \exists\, v \in S \text{ for which } \langle v, w\rangle \in F\} \end{aligned}$$

It should be noted that we permit occurences of "self-loops" in F, i.e. we allow $\langle v, v\rangle \in F$.[1]

A subset S of V is *conflict-free* if F contains no edges in $S \times S$. The subset S is *admissible* if it is both conflict-free and for every $v \in S^-$, $\{v\}^- \cap S \neq \emptyset$. An admissible set is *preferred* if it is maximal w.r.t. $\subseteq$, i.e. if S is preferred then for all $T \subseteq V$ if $S \subset T$ then T is *not* admissible. A subset S is *ideal* if S is both admissible *and* a subset of *every* preferred set.[2]

We note the following results concerning these:

Fact 1. *Given $H(V, F)$ and $S \subseteq V$,*

[1] This is solely in order to simplify some of the subsequent technical lemmata. All of our results, albeit with rather more involved constructions, can be derived when self-loops are forbidden.

[2] Our choice of terminology coincides with the treatment of these concepts in the context of argument systems: it is, of course, the case that a number of these ideas are already well-established in graph-theoretic treaments, e.g. "conflict-free" corresponds with "independent set".

a. *Deciding if S is admissible is in* P.

b. *Deciding if S is preferred is co*NP*–complete, [2].*

c. *Given $v \in V$ deciding if v is a member of* at least one *preferred set is* NP*–complete, [2].*

d. *Given $v \in V$ deciding if v is a member of* every *preferred set is* Π_2^p*–complete. [9].*

e. *Deciding if S is ideal is co*NP*–complete. [7, 8].*

f. *Every $H(V, F)$ has a unique maximal (w.r.t. $\subseteq$) ideal set. [4].*

g. *Deciding if S is the maximal ideal set is* $\mathrm{P}_{||}^{\mathrm{NP}}$*–complete, where hardness is with respect to* randomized *reductions that succeed with probability* $1 - 2^{-|V|}$*, [7, 8].*

h. *Computing the maximal ideal set is* $\mathrm{FP}_{||}^{\mathrm{NP}}$*–complete, [7, 8]. We note that the hardness proof uses a deterministic reduction.*

The techniques used to establish Fact 1(d) and (h) play a significant role in our subsequent technical development.

3. Formula Graphs and their Properties

We now present a translation from formulae, $\varphi(X_n)$ as defined in Section 2 to directed graphs, $H_\varphi(V_\varphi, F_\varphi)$ and reprise some properties of admissible and preferred sets of vertices in H_φ. Our translation while similar to the standard representation of formulae as directed graphs, e.g. as described in [5, pp. 18–23], has some important differences. The original definition of formula graph as given in Defn. 2 was presented in [9, Defn. 7, p. 193].

We start with the well-known translation from formulae $\varphi(X_n)$ to trees.

Definition 1. *Let $\varphi(X_n)$ be a propositional formula over the variables $X_n = \langle x_1, x_2, \ldots, x_n \rangle$ using the operations $\{\wedge, \vee, \neg\}$ with negation applied only to vari-*

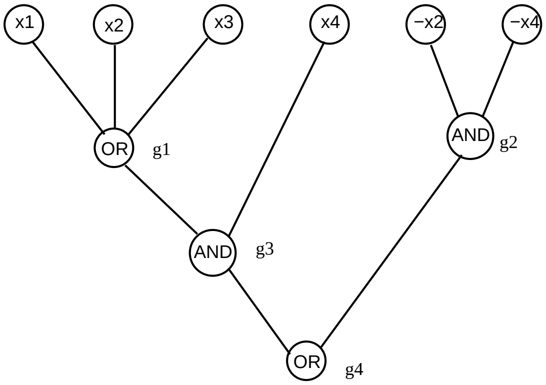

Figure 1: $T_\varphi(x_1, x_2, x_3, x_4)$ for $((x_1 \vee x_2 \vee x_3) \wedge (x_4)) \vee (\neg x_2 \wedge \neg x_4)$

ables of φ. The tree representation *of φ (denoted T_φ) is a rooted directed tree with root vertex denoted $\rho(T_\varphi)$ and inductively defined by the following rules.*

a. If $\varphi(X_n) = y$ – a single literal x or $\neg x$ – then T_φ consists of a single vertex $\rho(T_\varphi)$ labelled y.

b. If $\varphi(X_n)$ is an $\wedge$-formula $\wedge_{i=1}^{k} \psi_i(X_n)$, with $\langle \psi_1, \psi_2, \ldots, \psi_k \rangle$ $\vee$-formulae, then T_φ is formed from the k tree representations $\langle T_{\psi_i} \rangle$ by directing edges from each $\rho(T_{\psi_i})$ into a new root vertex $\rho(T_\varphi)$ labelled $\wedge$.

c. If $\varphi(X_n)$ is an $\vee$-formula $= \vee_{i=1}^{k} \psi_i(X_n)$, with $\langle \psi_1, \psi_2, \ldots, \psi_k \rangle$ $\wedge$-formulae, then T_φ is formed from the k tree representations $\langle T_{\psi_i} \rangle$ by directing edges from each $\rho(T_{\psi_i})$ into a new root vertex $\rho(T_\varphi)$ labelled $\vee$.

In what follows we use the term node *of T_φ to refer to an arbitrary tree vertex, i.e. a leaf or internal vertex.*

In the tree representation of φ, each leaf vertex is labelled with some literal w, (several leaves may be labelled with the same literal), and each internal vertex with an operation in $\{\wedge, \vee\}$. We shall subsequently refer to the internal vertices of T_φ as the *gates* of the tree. Notice that the definition of formula ensures that the successor of any $\wedge$-gate (tree vertex labelled $\wedge$) is an $\vee$-gate (tree vertex labelled $\vee$) and *vice-versa*. Let m be the number of *gate* nodes in T_φ and denote these gates by $\langle g_1, g_2, \ldots, g_m \rangle$ with g_m taken as the root $\rho(T_\varphi)$ of the tree. For any edge $\langle h, g \rangle$ in T_φ we refer to the node h as an *input* of the gate g.

The directed graph structure used in our upper bound proof is obtained from the *formula graph* of $\varphi(X_n)$ as defined below.

Definition 2. *Let $\varphi(X_n)$ be a formula with tree representation T_φ having m gates. The* Formula Graph *of φ, is the directed graph $H_\varphi(V_\varphi, F_\varphi)$ defined as follows.*

V_φ contains the following vertices

X1 $2n$ *vertices representing* literals. $\{x_i, \neg x_i : 1 \leq i \leq n\}$.

X2 For each gate g_k of T_φ, a vertex labelled $\neg g_k$ (if g_k is an $\vee$-gate) or a vertex labelled g_k (if g_k is an $\wedge$-gate). If g_m, i.e the root of T_φ, happens to be an $\vee$-gate, then an additional vertex g_{m+1} is included. We subsequently denote the set of vertices contributed by gates of T_φ as $\mathcal{G}_\varphi$.

The directed edges – F_φ – over V_φ are

F1 $\{\langle x_i, \neg x_i \rangle, \langle \neg x_i, x_i \rangle : 1 \leq i \leq n\}$

F2 $\langle \neg g_m, g_{m+1} \rangle$ if g_m is an $\vee$-gate in T_φ,

F3 If g_k is an $\wedge$-gate in T_φ with inputs $\{h_1, h_2, \ldots, h_r\}$: $\{\langle \neg h_i, g_k \rangle : 1 \leq i \leq r\}$.

It should be noted that when a literal vertex, y, is an input to an $\wedge$-gate in T_φ then this convention leads to the corresponding edge originating from the literal vertex $\neg y$ of V_φ.

F4 If g_k is an $\vee$-gate in T_φ with inputs $\{h_1, h_2, \ldots, h_r\}$: $\{\langle h_i, \neg g_k \rangle : 1 \leq i \leq r\}$.

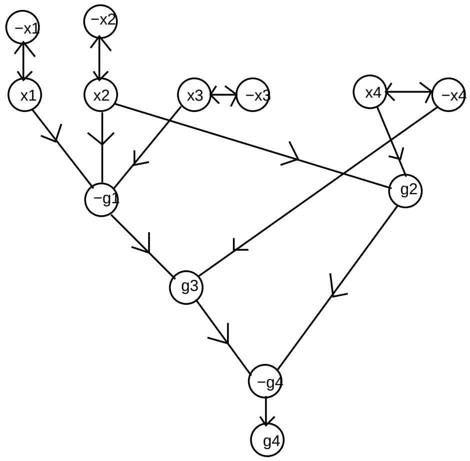

Figure 2: The Formula Graph H_φ from the tree representation of Fig. 1

Fig. 2 shows the result of this translation when it is applied to the tree representation of the formula in Fig. 1.

Suppose $\pi \ : \ X_n \rightarrow \langle \bot, \top, * \rangle^n$ is a *partial* assignment of Boolean values to X_n. Any such assignment induces a partition of V_φ which we denote

$$\tau \ : \ V_\varphi \times \langle \bot, \top, * \rangle \rightarrow \{In, \ Out, \ Open\}$$

The mapping $\tau(h, \pi)$ is given by,

$$\tau(h,\pi) \quad = \quad \begin{cases} In & \text{if} \quad h = x_i \text{ and } \pi(x_i) = \top \text{ or} \\ & \qquad h = \neg x_i \text{ and } \pi(x_i) = \bot \text{ or} \\ & \qquad h \text{ is a gate with inputs } \{h_1, \ldots, h_r\} \\ & \qquad \text{ and } \forall\, 1 \leq i \leq r \ \tau(h_i, \pi) = Out \\ Out & \text{if} \quad h = x_i \text{ and } \pi(x_i) = \bot \text{ or} \\ & \qquad h = \neg x_i \text{ and } \pi(x_i) = \top \text{ or} \\ & \qquad h \text{ is a gate with inputs } \{h_1, \ldots, h_r\} \\ & \qquad \text{ and } \exists\, 1 \leq i \leq r \ \tau(h_i, \pi) = In \\ Open & \qquad \text{otherwise} \end{cases}$$

For example, with respect to the formula graph shown in Fig. 2 which resulted from the $\vee$-formula $((x_1 \vee x_2 \vee x_3) \wedge (x_4)) \vee (\neg x_2 \wedge \neg x_4)$, if $\pi = \langle \top, *, *, \top \rangle$ then $\{x_1, x_4, g_3, g_4\}$ are all labelled In by τ; $\{\neg x_1, \neg x_4, \neg g_1, g_2, \neg g_4\}$ are labelled Out and all other nodes are labellel $Open$. For $\beta = \langle *, \bot, *, \bot \rangle$, $\{\neg x_2, \neg x_4, g_2, g_4\}$ are all labelled In and $\{x_2, x_4, g_3, \neg g_4\}$ labelled Out.

Let $In_\varphi(\pi) \ = \ \{h \in V_\varphi \ : \ \tau(h, \pi) = In\}$.

Theorem 1.

a. *For all $H(V_\varphi, F_\varphi)$ and all partial assignments π the set $In_\varphi(\pi)$ is admissible.*
b. *For $S \subseteq V_\varphi$, S is preferred if and only if $S = In_\varphi(\alpha)$ for some* total *assignment α, i.e. $\alpha(x_i) \in \{\bot, \top\}$ for every x_i.*

Proof: For (a), if $In_\varphi(\pi) = \emptyset$ then we are done since the empty set is always admissible. Otherwise $In_\varphi(\pi)$ contains a non-empty subset of V_φ. This set must be conflict-free since, by the definition of $\tau(h, \pi)$ if $\tau(h, \pi) = In$ then $\tau(p, \pi) \neq In$ for any node p which is adjacent to h in H_φ: if $\langle h, p \rangle \in F_\varphi$ then p is either a literal ($\neg h$) or a gate with h as an input; if $\langle p, h \rangle \in F_\varphi$ either h is the literal $\neg p$ or h is a gate with p as an input. In all cases $p \in In_\varphi(\pi)$ would preclude $h \in In_\varphi(\pi)$. Now consider any h in $\{In_\varphi(\pi)\}^-$. By definition, $\tau(h, \pi) = Out$. But $\tau(h, \pi) = Out$ only if h is a literal with $\tau(\neg h, \pi) = In$ or if h is a gate vertex one of whose inputs is in $In_\varphi(\pi)$: in both cases $In_\varphi(\pi) \cap \{h\}^- \neq \emptyset$ so that $In_\varphi(\pi)$ is admissible.

For (b), first note that if α is a total assignment then $\tau(h, \alpha) \in \{In, Out\}$ for all $h \in V_\varphi$. From (a) we know that $In_\varphi(\alpha)$ is admissible. To see that it is a maximal such set it suffices to note that $\tau(h, \alpha) = Out$ implies there is some $g \in In_\varphi(\alpha)$ for which $\langle g, h \rangle \in F_\varphi$, i.e. $\{h\} \cup In_\varphi(\alpha)$ would not be conflict-free and, hence, is not admissible. Conversely, if $S \subseteq V_\varphi$ is preferred then S must contain *exactly* one literal vertex from each of the n pairs $\{x_i, \neg x_i\}$. Defining the total assignment α_S by $\alpha_S(x_i) = \top$ if $x_i \in S$; $\alpha_S(x_i) = \bot$ if $\neg x_i \in S$, it is immediate that $S = In_\varphi(\alpha_S)$. □

The partition induced by a partial assignment π has a close relationship to the *value* in $\{\bot, \top, *\}$ taken by nodes, i.e. literals and gates, in the tree representation of a formula. We recall that value(h, π) for h a node in the tree representation of a formula and $\pi = \langle p_1, \ldots, p_n \rangle \in \langle \bot, \top, * \rangle^n$ a partial assignment of X_n, is defined via[3]

$$value(h,\pi) \;=\; \left\{ \begin{array}{lll} \top & \text{if} & h = x_i \text{ and } p_i = \top \text{ or} \\ & & h = \neg x_i \text{ and } p_i = \bot \\ \bot & \text{if} & h = x_i \text{ and } p_i = \bot \text{ or} \\ & & h = \neg x_i \text{ and } p_i = \top \\ & \text{if} & h = x_i \text{ and } p_i = * \text{ or} \\ & & h = \neg x_i \text{ and } p_i = * \\ \bigwedge_{i=1}^{r} \; value(h_i, \pi) & \text{if} & h \text{ is an } \wedge\text{-gate with inputs } \{h_1, \ldots, h_r\} \\ \bigvee_{i=1}^{r} \; value(h_i, \pi) & \text{if} & h \text{ is an } \vee\text{-gate with inputs } \{h_1, \ldots, h_r\} \end{array} \right.$$

Theorem 2. *Let $\varphi(X_n)$ be a formula with tree representation T_φ and formula graph $H_\varphi(V_\varphi, F_\varphi)$. For h a vertex of the formula graph – i.e. node of the tree representation – and π any partial assignment:*

[3] The value of $\wedge$ (resp. $\vee$) w.r.t partial assignments is given in the obvious way by $\wedge_{i=1}^{k} \, y_i$ is $\top$ (if $y_i = \top$ for all $1 \leq i \leq k$); $\bot$ (if at least one $y_i = \bot$); and $*$ in all other cases. Similarly $\vee_{i=1}^{k} \, y_i$ is $\bot$ (if $y_i = \bot$ for all $1 \leq i \leq k$); $\top$ (if at least one $y_i = \top$); and $*$ in all other cases.

$\tau(h,\pi) = In$ *(resp.* $\tau(h,\pi) = Out$*) if and only if*

$$\begin{cases} h \textit{ is a literal vertex} & \textit{and} \quad value(h,\pi) = \top \\ & \textit{resp.} \quad value(h,\pi) = \bot \\ h \textit{ is an } \wedge\textit{-gate} & \textit{and} \quad value(h,\pi) = \top \\ & \textit{resp.} \quad value(h,\pi) = \bot \\ h \textit{ is an } \vee\textit{-gate} & \textit{and} \quad value(h,\pi) = \bot \\ \textit{(labelled } \neg h \textit{ in } H_\varphi\textit{)} & \textit{resp.} \quad value(h,\pi) = \top \end{cases}$$

Similarly $\tau(h,\pi) = Open$ *if and only if* $value(h,\pi) = *$.

Proof: It is easily checked that $\tau(h,\pi) = Open$ if and only if $value(h,\pi) = *$. For the remaining cases, we recall that the *depth* of a node (in either representation) has $depth(h) = 0$ if h is a literal node and

$$1 + \max\ \{depth(h_i)\ :\ h \text{ is a gate node with inputs } \{h_1, \ldots, h_r\}\}$$

when h is a gate node

We proceed by induction on the depth of nodes.

For the inductive base $depth(h) = 0$, the only relevant nodes are literals and the definitions of $\tau(y,\pi)$ and $value(y,\pi)$ immediately give

$$\tau(y,\pi) = In\ \Leftrightarrow\ value(y,\pi) = \top$$
$$\tau(y,\pi) = Out\ \Leftrightarrow\ value(y,\pi) = \bot$$

Assuming the theorem holds for all nodes at depth less than $k \geq 1$ we show it holds for all nodes whose depth is equal to k. Let h be any node with $depth(h) = k \geq 1$. Then h must be a gate node with inputs $\{h_1, \ldots, h_r\}$. Furthermore, $depth(h_i) < k$ for each input h_i of h.

Suppose first that h is an $\wedge$-gate and $\tau(h,\pi) = In$. Each h_i must have $\tau(h_i,\pi) = Out$, and thus by the inductive hypothesis, if h_i is an $\vee$-gate, $value(h_i,\pi) =$

$\top$ and if h_i is the literal y then $value(y, \pi) = \bot$, i.e. $value(\neg y, \pi) = \top$. From the definition of formula graph if h is an $\wedge$-gate one of whose inputs (in T_φ) is the literal y, then the edge directed into h (in H_φ) is $\langle \neg y, h\rangle$. In other words,

$$\begin{array}{c} \langle y, h\rangle \in F_\varphi \text{ and } (h \text{ is an } \wedge\text{-gate}) \text{ and } \tau(h, \pi) = In \\ \text{implies} \\ \langle \neg y, h\rangle \in T_\varphi \text{ and } \tau(y, \pi) = Out \end{array} \qquad (1)$$

In summary from h an $\wedge$-gate with $\tau(h, \pi) = In$ we have $\tau(h_i, \pi) = Out$ for all inputs h_i, hence $value(h_i, \pi) = \top$ (from relation (1) and the inductive hypothesis) so that $value(h, \pi) = \wedge_{i=1}^{r} value(h_i, \pi) = \top$. On the other hand, if $\tau(h, \pi) = Out$ then there must be some input h_i for which $\tau(h_i, \pi) = In$. If h_i is the literal $\neg y$ we have $value(y, \pi) = \bot$ giving $value(h, \pi) = \bot$ as required. If h_i is an $\vee$-gate with $\tau(h_i, \pi) = In$ then from the inductive hypothesis, $value(h_i, \pi) = \bot$ and again $value(h, \pi) = \bot$.

Now suppose h is an $\vee$-gate with inputs $\{h_1, \ldots, h_r\}$ and $\tau(h, \pi) = In$. We again have $\tau(h_i, \pi) = Out$ so that from the inductive hypothesis (and the fact that inputs are either $\wedge$-formulae or literals), $value(h_i, \pi) = \bot$ leading to $value(h, \pi) = \vee_{i=1}^{k} value(h_i, \pi) = \bot$ as required. Similarly is $\tau(h, \pi) = Out$ then some input must have $\tau(h_i, \pi) = In$ and again (via the inductive hypothesis) $value(h_i, \pi) = \top$ leading to $value(h, \pi) = \vee_{i=1}^{r} value(h_i, \pi) = \top$.

For the converse direction, the inductive base ($depth(h) = 0$) has already been established. Thus suppose h is a gate node with $value(h, \pi) = \top$. If h is an $\wedge$-gate we need to show $\tau(h, \pi) = In$. Letting $\{h_1, \ldots, h_r\}$ be the inputs for h (which are $\vee$-gates or literals) from $value(h, \pi) = \top$ we have $value(h_i, \pi) = \top$ so that $\tau(h_i, \pi) = Out$ (if h_i is an $\vee$-gate) and $\tau(\neg y, \pi) = Out$ if h_i is the literal y providing the input to h in T_φ (so that $\neg y$ is the input to h in H_φ). It follows that since all input nodes are labelled Out, $\tau(h, \pi) = In$. If h is an $\vee$-gate, then from

$value(h,\pi) = \top$ at least one of its inputs must have $value(h_i,\pi) = \top$ giving $\tau(h_i,\pi) = In$ (by induction). But now we obtain $\tau(h,\pi) = Out$ as claimed.

Finally if $value(h,\pi) = \bot$ and h is an $\wedge$-gate, we have $value(h_i,\pi) = \bot$ for at least one input h_i: if this input is an $\vee$-gate we obtain $\tau(h_i,\pi) = In$ so that $\tau(h,\pi) = Out$. If this input is the literal y (in T_φ) then $\neg y$ (the input to h in H_φ) has $value(\neg y,\pi) = \top$ so that $\tau(\neg y,\pi) = In$ and $\tau(h,\pi) = Out$. On the other hand if h is an $\vee$-gate with $value(h,\pi) = \bot$ then all inputs h_i must have $value(h_i,\pi) = \bot$: these are either literals y (so that $value(y,\pi) = \bot$ giving $\tau(y,\pi) = Out$) or $\wedge$-gates which (by the inductive hypothesis) have $\tau(h_i,\pi) = Out$: in summary if $value(h,\pi) = \bot$ and h is an $\vee$-gate then each input has $\tau(h_i,\pi) = Out$ so that $\tau(h,\pi) = In$ as claimed. □

4. Upper bounds on Term Equivalence

The correspondence between admissible sets of vertices in the formula graph $H_\varphi(V_\varphi, F_\varphi)$ and partial assignments π, established over Thms. 1, 2, is not quite strong enough to allow the derivation of our upper bound on TE. By making a final modification to the structure of a formula graph we can, however, obtain the claimed upper bound by exploiting a correspondence between *satisfying* assignments, α (in the case of $\wedge$-formulae) and the *maximal ideal set* in the modified graph. We note that, without loss of generality, it may be assumed that that $\varphi(X_n)$ is an $\wedge$-formula: if $\varphi(X_n)$ is an $\vee$-formula simply apply De Morgan's Laws to transform the $\neg\varphi(X_n)$-formula into an $\wedge$-formula noting that $\varphi(X_n) \equiv \vee_{y \in S} S$ if and only if $\neg\varphi(X_n) \equiv \wedge_{y\in S}\,(\neg y)$.

The graph-theoretic structure considered in the upper bound is now introduced.

Definition 3. *Let $\varphi(X_n)$ be an $\wedge$-formula and $H_\varphi(V_\varphi, F_\varphi)$ its associated formula graph with g_m the $\wedge$-gate vertex corresponding to the root of the tree representation* *of φ. The* ideal graph *of φ, denoted R_{φ}, has vertex set $W_{\varphi} = V_{\varphi} \cup \{y_1, y_2, C, b\}$ (where $\{y_1, y_2, C, b\}$ are new vertices not occuring in V_{φ}) and edges $F_{\varphi} \cup E_{\varphi}$ where E_{φ} contains*

$$\begin{array}{ll} & \{\langle y_1, y_2\rangle,\ \langle y_2, y_1\rangle,\ \langle y_1, C\rangle,\ \langle y_2, C\rangle\} \\ \cup & \{\langle C, g_m\rangle,\ \langle g_m, C\rangle,\ \langle C, b\rangle, \langle b, b\rangle\} \\ \cup & \{\langle b, x_i\rangle \ :\ 1 \leq i \leq n\} \ \cup\ \{\langle b, \neg x_i\rangle \ :\ 1 \leq i \leq n\} \end{array}$$

The following characterisation of ideal sets and conditions for a vertex to be a member of the maximal ideal set are stated in [7].[4]

Fact 2.

a. *Let $H(V, F)$ be any directed graph and S a subset of V. The set S is an ideal set if and only if both of the following hold:*

   P1. S is an admissible set of $H(V, F)$.

   P2. For all $w \in S^{-}$, there is no admissible set of $H(V, F)$ containing w.

b. *Let $\mathcal{M}$ be the (unique) maximal ideal set of $H(V, F)$ and v a vertex from V. Then $v \in \mathcal{M}$ if and only if both of the following hold:*

   M1. For all $w \in \{v\}^{-}$ there is no admissible set of $H(V, F)$ that contains w.

   M2. For all $w \in \{v\}^{-}$, $\mathcal{M} \cap \{w\}^{-} \neq \emptyset$, i.e. $\mathcal{M}$ contains at least one vertex from $\{w\}^{-}$.

The key property of ideal graphs we need is

Theorem 3. *Let $\varphi(X_n)$ be an $\wedge$-formula for which $\varphi(X_n) \not\equiv \bot$ and $\varphi(X_n) \not\equiv \top$, i.e. $\varphi(X_n)$ is satisfiable and not a tautology.*[5] *Let $R_{\varphi}(W_{\varphi}, F_{\varphi} \cup E_{\varphi})$ be the ideal*

[4] The full proofs of these may be found in [8].

[5] Note that if $\varphi(X_n) = \psi_1(X_n) \wedge \psi_2(X_n)$ then $\varphi(X_n)$ will be a tautology if both ψ_1 and ψ_2 are tautologies.

graph constructed from $\varphi(X_n)$ and $\mathcal{M}_\varphi \subseteq W_\varphi$ be the maximal ideal set of R_φ.

$$\mathcal{M}_\varphi \neq \emptyset \quad \textit{if and only if } \varphi(X_n) \in \texttt{TE}$$

Proof: Suppose $\mathcal{M}_\varphi \neq \emptyset$. From Fact 2(a) we have $\mathcal{M}_\varphi \cap \{y_1, y_2, C\} = \emptyset$: the sets $\{y_1\}$ and $\{y_2\}$ are both admissible, however, $y_1 \in \{y_2\}^-$, $y_2 \in \{y_1\}^-$ and both vertices are in $\{C\}^-$. The vertex b is not in any admissible set since $\langle b, b\rangle \in F_\varphi \cup E_\varphi$. In total we deduce from $\mathcal{M}_\varphi \neq \emptyset$ that $\mathcal{M} \subseteq V_\varphi$ – the set of vertices defining the formula graph, H_φ, of φ.

We now argue that $\mathcal{M}_\varphi \neq \emptyset$ implies $g_m \in \mathcal{M}_\varphi$ (recall that g_m is the $\wedge$-gate in V_φ corresponding to the root of the tree representation, T_φ).

For suppose this were not so. From $g_m \notin \mathcal{M}_\varphi$ and Fact 2(b)

$$\mathcal{M}_\varphi \cap \{x_1, \ldots, x_n, \neg x_1, \ldots \neg x_n\} = \emptyset$$

That is, no literal vertex could belong to $\mathcal{M}_\varphi$: the vertex b is in $\{x\}^-$ hence were any literal vertex to be in $\mathcal{M}_\varphi$ this would force $g_m \in \mathcal{M}_\varphi$ since $\{b\}^- = \{b, g_m\}$. It is now easy to see that $g_m \notin \mathcal{M}_\varphi$ forces $\mathcal{M}_\varphi = \emptyset$: we have already argued that $\mathcal{M}_\varphi$ must be a subset of V_φ, thus, from the assumption that $g_m \notin \mathcal{M}_\varphi$ and the consequence of $\mathcal{M}_\varphi$ containing no literal vertices, if $\mathcal{M}_\varphi \neq \emptyset$ then it can only contain gate vertices. Consider any gate vertex h. If h has a literal, y, as an input then $h \notin \mathcal{M}_\varphi$ from Fact 2(b): the only choices from $\{y\}^-$ being $\neg y$ and b neither of which belong to $\mathcal{M}_\varphi$. If h has only gate vertices $\{h_1, \ldots, h_r\}$ as inputs then $\mathcal{M}_\varphi \cap \{h_j\}^-$ has to be non-empty: now repeating the same argument (with respect to vertices $\{h_j\}^-$) we eventually reach the position that some literal vertex must belong to $\mathcal{M}_\varphi$. We deduce that if $\mathcal{M}_\varphi \neq \emptyset$ then $g_m \in \mathcal{M}_\varphi$.

From $g_m \in \mathcal{M}_\varphi$ and our analysis above it follows that

$$\mathcal{M}_\varphi \cap \{x_1, \ldots, x_n, \neg x_1, \ldots \neg x_n\} \neq \emptyset$$

Let S be the set of literal vertices occurring in $\mathcal{M}_\varphi$. First notice that S cannot contain both a literal and its negation since $\mathcal{M}_\varphi$ must be conflict-free. Without loss of generality let $S = \{x_1, \ldots, x_k\}$. We now claim that $\varphi(X_n) = \wedge_{i=1}^{k} x_i$ so establishing that $\varphi(X_n)$ is a positive instance of TE.

Now $S \cup \{g_m\} \subseteq \mathcal{M}_\varphi$ indicates (by the definition of ideal set) that $S \cup \{g_m\}$ is a subset of *every* preferred set of R_φ. Any such preferred set consists of some subset, Q, of vertices from V_φ and exactly one of the vertices from $\{y_1, y_2\}$ so that from Thm. 1(b), $Q = In_\varphi(\alpha)$ for some satisfying[6] assignment α of $\varphi(X_n)$. It follows that *every* assignment in which $x_i \; := \; \top$ $(1 \leq i \leq k)$ will satisfy $\varphi(X_n)$ regardless of how the variables $X_n \setminus \{x_1, \ldots, x_k\}$ are assigned. In addition, however, no assignment in which $x_i \; := \; \bot$ (for $1 \leq i \leq k$) can satisfy $\varphi(X_n)$. For suppose there were a satisfying assignment, β with $x_1 = \bot$, then in this case both $In_\varphi(\beta) \cup \{y_1\}$ and $In_\varphi(\beta) \cup \{y_2\}$ would be preferred sets of R_φ: neither of these, however, contain x_1, thereby contradicting $x_1 \in \mathcal{M}$. In summary, $x_i \; := \; \top$ for each $1 \leq i \leq k$ satisfies $\varphi(X_n)$ but $x_i := \bot$ for any $1 \leq i \leq k$ fails to satisfy $\varphi(X_n)$, i.e. $\varphi(X_n) \equiv \wedge_{i=1}^{k} x_i$ as required.

For the converse direction, suppose that $\varphi(X_n) \equiv \wedge_{y \in S} \, y$ for some subset, S, of literals over X_n. Without loss of generality, assume that $S = \{x_1, \ldots, x_k\}$. We show that $\{g_m\} \cup S = \{g_m, x_1, \ldots, x_k\} \subset W_\varphi$ is a subset of $\mathcal{M}_\varphi$. From the structure of R_φ it follows that neither $\{y_1\}$ nor $\{y_2\}$ are preferred sets of R_φ (since $\{y\} \cup In_\varphi(\alpha)$ is admissible when α satisfies φ for $y \in \{y_1, y_2\}$). Thus the preferred sets of R_φ are of the form $\{y_1\} \cup In_\varphi(\alpha)$ and $\{y_2\} \cup In_\varphi(\alpha)$ for satisfying assignments α of φ. The only such assignments, however, select x_i $(1 \leq i \leq k)$ as literal vertices. It follows that every preferred set of R_φ contains $\{g_m, x_1, \ldots, x_k\}$ as a subset. Furthermore, using the partial assignment π in which $p_i = \top$ (for

[6] Satisfying since $g_m \in In_\varphi(\alpha)$ so that $value(g_m, \alpha) = \top$ from Thm. 2.

$1 \leq i \leq k$) and $p_i = *$ for all other variables we deduce from $\varphi(X_n) \equiv \wedge_{i=1}^{k} x_i$ that $In_\varphi(\pi)$ is admissible via Thm. 1(a)[7] *and* a subset of every preferred set of R_φ. In other words $In_\varphi(\pi)$ is an ideal set and $\mathcal{M}_\varphi \neq \emptyset$ as claimed. □

An immediate corollary of Thm. 3 is that Algorithm 1, below, will *construct* a witnessing subset, S, of literals such that $\varphi(X_n) \equiv \wedge_{y \in S}\, y$ if such a subset exists:

Algorithm 1 Finding S such that $\varphi(X_n) \equiv \wedge_{y \in S}\, y$

```
If $\varphi(X_n)$ is an $\vee$-formula then convert $\varphi(X_n)$ to the $\wedge$-type formula $\neg\varphi(X_n)$ applying De Morgan's Laws.
Construct the ideal graph $R_\varphi(W_\varphi, E_\varphi \cup F_\varphi)$ as described in Thm. 3.
**if** $\varphi(X_n) \equiv \top$ **then**
  Report $S = \emptyset$ and $\varphi \equiv \wedge_{y \in S} y$
**else if** $\varphi(X_n) \equiv \bot$ **then**
  Report $S = \emptyset$ and $\varphi \equiv \vee_{y \in S} y$
**else**
  Compute $\mathcal{M}_\varphi$ the maximal ideal set of $R_\varphi$
  $S \ := \ \mathcal{M}_\varphi \cap \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$
  **if** $S \neq \emptyset$ **then**
    Report $S$ and $\varphi(X_n) \equiv \wedge_{y \in S}\, y$
  **else**
    Report $\varphi(X_n) \notin$ TE.
  **end if**
**end if**
```

The correctness of Algorithm 1 is immediate from Thm. 3. All that is needed for the claimed upper bound on FTE is to show that the maximal ideal set can be

[7] Note that $C \in \{In_\varphi(\pi)\}^-$ is countered by $g_m \in \{C\}^-$.

computed in $\mathrm{FP}_{||}^{\mathrm{NP}}$. As we noted in Fact 1(h) such an upper bound has been given in Dunne [7, 8]. For completeness we outline this algorithm (as specialised to the particular instance R_φ).

Theorem 4. *Algorithm 1 can be implemented in* $\mathrm{FP}_{||}^{\mathrm{NP}}$.

Proof: Given $\varphi(X_n)$ which we assume to be an $\wedge$-formula it is clear that the ideal graph $R_\varphi(W_\varphi, F_\varphi \cup E_\varphi)$ may be constructed in polynomial time (in the size of φ). After checking if $\varphi(X_n) \equiv \top$ or $\varphi(X_n) \equiv \bot$ construct the following partition of W_φ:

$$\begin{array}{rcl} W_{\mathrm{REJ}} & = & \{v \in W_\varphi \ : \ v \text{ is not in any admissible set of } R_\varphi\} \\ W_{\mathrm{POS}} & = & \{v \in W_\varphi \ : \ \{v\}^- \cup \{v\}^+ \subseteq W_{\mathrm{REJ}}\} \setminus W_{\mathrm{REJ}} \end{array}$$

Now form the *bipartite* graph $B(W_{\mathrm{POS}}, W_{\mathrm{REJ}}, D)$ in which

$$D \ = \ F_\varphi \setminus \{\langle v, w\rangle \ : \ v \in W \setminus W_{\mathrm{POS}} \text{ and } w \in W \setminus W_{\mathrm{POS}}\}$$

It is shown in Dunne [7] that $\mathcal{M}_\varphi$ is the maximal admissible subset of W_{POS} in this bipartite graph. The algorithm below, from Dunne [6] computes this set in polynomial time.

To complete the upper bound proof it suffices to observe that constructing this bipartite graph requires only computing the set W_{REJ} which (in conjunction with testing $\varphi(X_n) \equiv \top$ or $\varphi(X_n) \equiv \bot$) can be done using $|W_\varphi| + 2$ parallel calls to an NP oracle: 2 calls to determine $\varphi(X_n) \equiv \top$ or $\varphi(X_n) \equiv \bot$; and a further $|W_\varphi|$ calls to detemrine which vertices of R_φ are members of some admissible set. □

Corollary 1. $\mathrm{TE} \in \mathrm{P}_{||}^{\mathrm{NP}}$

Proof: Immediate from the Thm. 4. □

Algorithm 2 Maximal admissible subset in Bipartite Graphs

```
$i := 0$ ; $W_0 := W_{\mathrm{POS}}$ ; $D_0 := D$
repeat
    $i := i + 1$
    $U_i := \{ y \in W_{i-1} : \exists z \in W_{\mathrm{REJ}} : \langle z, y \rangle \in D_{i-1}\, and\, \{z\}^{-} \cap W_{i-1} = \emptyset \}$
    $W_i := W_{i-1} \setminus U_i$
    $D_i := D_{i-1} \setminus \{ \langle y, z \rangle : y \in U_i \}$
until $W_i = W_{i-1}$
return $W_i$
```

5. Conclusion

In this note we have considered a variant of the Minimal Expression Equivalence problem (MEE) studied by Hemaspaandra and Wechsung [11, 12] whereby the form of witnessign smaller formulae is restricted to elementary conjunctions and disjunctions. Our main result shows this variant can be decided in $\mathrm{P}_{||}^{\mathrm{NP}}$ (with the search form belonging to the analogous function complexity class). This class has been shown to provide a lower bound for the general MEE problem. One feature of interest in our proof is the range of technical material originating from recent work on extension-based semantics in argument systems, in particular the correspondence between ideal sets and witnessing subsets of literals in the so-called ideal graph derived from a given formula. One of the original applications of argument system semantics was in modelling problems in non-classical logics (as opposed to the propositional basis of the current article). It would, therefore, be of some interest to see to what extent further interaction between argumentation semantics and algorithms for deciding properties of propositional formulae is possible.

References

[1] T. J. M. Bench-Capon and P. E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171:619–641, 2007.

[2] Y. Dimopoulos and A. Torres. Graph theoretical structures in logic programs and default theories. *Theoretical Computer Science*, 170:209–244, 1996.

[3] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and N-person games. *Artificial Intelligence*, 77:321–357, 1995.

[4] P. M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171:642–674, 2007.

[5] P. E. Dunne. *The complexity of Boolean networks* Academic Press, 1988

[6] P. E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artificial Intelligence*, 171:701–729, 2007.

[7] P. E. Dunne. The computational complexity of ideal semantics I: abstract argumentation frameworks. In *Proc. 2nd Int. Conf. on Computational Models of Argument*, volume 172 of *FAIA*, pages 147–158. IOS Press, 2008.

[8] P. E. Dunne. The computational complexity of ideal semantics. Tech. Report ULCS-08-015, Dept. of Computer Science, Univ. of Liverpool, 2008 (submitted)

[9] P. E. Dunne and T. J. M. Bench-Capon. Coherence in finite argument systems. *Artificial Intelligence*, 141:187–203, 2002.

[10] P. E. Dunne and M. Wooldridge. Complexity of abstract argumentation. In I. Rahwan and G. Simari, editors, *Argumentation in AI*. Springer-Verlag, 2009. (in press).

[11] E. Hemaspaandra and G. Wechsung The Minimization Problem for Boolean Formulas. In: *Proc. 38th Annual Symp. on Foundations of Computer Science (FOCS'97)*, page 575–584, 1997

[12] E. Hemaspaandra and G. Wechsung The Minimization Problem for Boolean Formulas. *SIAM J. Comput.*, 31(6):1948–1958, 2002