THE COMPLEXITY OF SATISFIABILITY PROBLEMS

Thomas J. Schaefer*
Department of Mathematics
University of California, Berkeley, California 94720

ABSTRACT

The problem of deciding whether a given propositional formula in conjunctive normal form is satisfiable has been widely studied. It is known that, when restricted to formulas having only two literals per clause, this problem has an efficient (polynomial-time) solution. But the same problem on formulas having three literals per clause is NP-complete, and hence probably does not have any efficient solution.

In this paper, we consider an infinite class of satisfiability problems which contains these two particular problems as special cases, and show that every member of this class is either polynomial-time decidable or NP-complete. The infinite collection of new NP-complete problems so obtained may prove very useful in finding other new NP-complete problems. The classification of the polynomial-time decidable cases yields new problems that are complete in polynomial time and in nondeterministic log space.

We also consider an analogous class of problems, involving quantified formulas, which has the property that every member is either polynomial-time decidable or complete in polynomial space.

1. INTRODUCTION -- A GENERALIZED SATISFIABILITY PROBLEM

We start with an introductory example. Let $R(x,y,z)$ be a 3-place logical relation whose truth-table is $\{(1,0,0),(0,1,0),(0,0,1)\}$ -- that is, $R(x,y,z)$ is true iff exactly one of its three arguments is true. Consider the problem of deciding whether an arbitrary conjunction of clauses of the form $R(x,y,z)$ is satisfiable. We call this the ONE-IN-THREE SATISFIABILITY problem. For example, the formula $R(x,y,z) \wedge R(x,y,u) \wedge R(u,u,y)$ is satisfiable, because it is made true by assigning the values 0,1,0,0 to the variables x,y,z,u respectively. As will be seen, the ONE-IN-THREE SATISFIABILITY problem is NP-complete.

The similarity between this problem and the standard satisfiability problem for propositional formulas in conjunctive normal form leads to the generalization which is the subject of this paper. Consider the problem of deciding whether a given CNF formula with 3 literals in each clause is satisfiable -- a well-known NP-complete problem. Since a clause may contain any number of negated variables from 0 to 3, there are four distinct relations among variables which occur as conjuncts in the formulas of this problem -- namely, the relations R_0,R_1,R_2,R_3 defined by $R_0(x,y,z) \equiv x \vee y \vee z$, $R_1(x,y,z) \equiv \neg x \vee y \vee z$, $R_2(x,y,z) \equiv \neg x \vee \neg y \vee z$ and $R_3(x,y,z) \equiv \neg x \vee \neg y \vee \neg z$. An input to this satisfiability problem is just a conjunction of clauses of the form $R_i(\xi,\xi',\xi'')$ for various variables ξ,ξ',ξ'' and various $i \in \{0,1,2,3\}$.

This sets the stage for the following generalization. Let $S = \{R_1,\dots,R_m\}$ be any finite set of logical relations. (A logical relation is defined to be any subset of $\{0,1\}^k$ for some integer $k \geq 1$. The integer k is called the rank of the relation.) Define an S-formula to be any conjunction of clauses, each of the form $\underset{\sim}{R}_i(\xi_1,\xi_2,\dots)$, where $\xi_1,\xi_2,\dots$ are variables whose number matches the rank of R_i, $i \in \{1,\dots,m\}$, and $\underset{\sim}{R}_i$ is a relation symbol representing the relation R_i. The S-satisfiability problem is the problem of deciding whether a given S-formula is satisfiable. We denote by SAT(S) the set of all satisfiable S-formulas.

The main result of this paper characterizes the complexity of SAT(S) for every finite set S of logical relations. The most striking feature of this characterization is that for any such S, SAT(S) is either polynomial-time decidable or NP-complete. This dichotomy is somewhat surprising, since one might expect that any such large and diverse class of problems, that includes both polynomial-time decidable and NP-complete members, would also contain some representatives of the many intermediate degrees of complexity which presumably lie between these two extremes.

Furthermore, we give an interesting classification of the polynomial-time decidable cases. We show that (assuming $P \neq NP$) SAT(S) is polynomial-time decidable only if at least one of the following conditions holds:

(a) Every relation in S is satisfied when all variables are 0.

(b) Every relation in S is satisfied when all variables are 1.

(c) Every relation in S is definable by a CNF formula in which each conjunct has at most one negated variable.

(d) Every relation in S is definable by a CNF formula in which each conjunct has at most one unnegated variable.

---

*Author's current address: CALMA, 527 Lakeside Drive, Sunnyvale, CA 94086

(e) Every relation in S is definable by a CNF formula having at most 2 literals in each conjunct.

(f) Every relation in S is the set of solutions of a system of linear equation over the two-element field {0,1}.

Sections 2-4 are devoted to the statement and proof of this Dichotomy Theorem. (Although we use the word "dichotomy" to describe this result, it should be borne in mind that the dichotomy holds only if P≠NP; if P=NP, the dichotomy would collapse.)

A variation of the problem consists of allow- the constants 0 and 1 to occur in input formulas (e.g. a clause R(x,0,y) is allowed). We denote this "satisfiability-with-constants" problem by $SAT_C(S)$. Our results for $SAT_C(S)$ are sharper than for SAT(S): we obtain a complete characterization up to log-space equivalence. For any finite set S of logical relations, $SAT_C(S)$ lies in one of seven log-space equivalence classes, described as follows:

1. $SAT_C(S)$ is decidable deterministically in log space.
2. The complement of $SAT_C(S)$ is log-equivalent to the graph reachability problem (given a graph G and nodes s,t of G, do s and t lie in the same connected component of G?).
3. The complement of $SAT_C(S)$ is log-equivalent to the digraph reachability problem (given a directed graph G and nodes s,t of G, is there a directed path from s to t?). In this case, $SAT_C(S)$ is log-complete in co-NSPACE(log n).
4. $SAT_C(S)$ is log-equivalent to the problem of deciding whether a graph is bipartite.
5. $SAT_C(S)$ is log-equivalent to the problem of whether an arbitrary system of linear equations over the field {0,1} is consistent.
6. $SAT_C(S)$ is log-complete in P.
7. $SAT_C(S)$ is log-complete in NP.

This result is presented in Section 5. For "most" sets S, $SAT_C(S)$ is essentially identical to, and has the same complexity as, SAT(S). (See Lemma 4.2.) Of course, it is not known that the above seven classes are distinct.

In Section 6, we present a polynomial-space analogue of the Dichotomy Theorem, involving quantified formulas. We define $QF_C(S)$ to be the analog of $SAT_C(S)$ in which formulas contain universal and existential quantifiers quantifying the propositional variables. The main theorem of this section states that for any finite set S of logical relations, $QF_C(S)$ is either polynomial-time decidable or log-complete in polynomial space. For both $QF_C(S)$ and $SAT_C(S)$ the polynomial-time decidable cases are just the cases (c)-(f) listed above; cases (a) and (b) are excluded.

We mention here a few particular completeness results which follow from these general theorems. Problems NP1,NP2 and NP3 are NP-complete.

NP1. ONE-IN-THREE SATISFIABILITY

Given sets $S_1,\ldots,S_m$ each having at most 3 members, is there a subset T of the members such that for each i, $|T \cap S_i| = 1$?

NP2. NOT-ALL-EQUAL SATISFIABILITY

Given sets $S_1,\ldots,S_m$ each having at most 3 members, can the members be colored with two colors so that no set is all one color?

NP3. TWO-COLORABLE PERFECT MATCHING

Given a graph G, can the nodes of G be colored with two colors so that each node has exactly one neighbor the same color as itself? (G may be restricted to be planar and cubic.) (Theorem 7.1)

Problems P1 and P2 are log-complete in P.

P1. SAT3W (Weakly Positive Satisfiability)

Given a CNF formula having at most 3 literals in each clause, and having at most one negated variable in each clause, is it satisfiable? (Corollary 5.2)

P2. NOT-EXACTLY-ONE SATISFIABILITY

Given sets $S_1,\ldots,S_m$ each having at most 3 members, and a distinguished member s, can one choose a subset of the members, containing s, so that no set has exactly one member chosen? (Corollary 5.2)

This paper contains a full proof of the Dichotomy Theorem. The other results are, for the most part, stated without proof.

Technical Note. The definition of "logical relation" given above is deficient in that it fails to differentiate between empty relations of differing ranks. Therefore, we formally define a logical relation to be a pair (k,R) with $R \subseteq \{0,1\}^k$; but informally we shall continue to regard R itself as being the relation.

2. THE DICHOTOMY THEOREM

This section states and discusses the main result of this paper, the following theorem.

Theorem 2.1. (Dichotomy Theorem for Satisfiability). Let S be a finite set of logical relations. If S satisfies one of the conditions (a)-(f) below, then SAT(S) is polynomial-time decidable. Otherwise, SAT(S) is log-complete in NP. (See below for definitions).

(a) Every relation in S is 0-valid.

(b) Every relation in S is 1-valid.

(c) Every relation in S is weakly positive.

(d) Every relation in S is weakly negative.

(e) Every relation in S is affine.

(f) Every relation in S is bijunctive.

Definitions.
(The following definitions were all invented for this paper and should not be assumed to agree with terminology used elsewhere.)

The logical relation R is 0-valid if $(0,\ldots,0) \in R$. The logical relation R is 1-valid if $(1,\ldots,1) \in R$.

The logical relation R is weakly positive (resp. weakly negative) if $\underset{\sim}{R}(x_1,\ldots)$ is logically equivalent to some CNF formula having at most one negated (resp. unnegated) variable in each conjunct.

The logical relation R is bijunctive if $\underset{\sim}{R}(x_1,\ldots)$ is logically equivalent to some CNF formula having at most 2 literals in any conjunct.

The logical relation R is affine if $\underset{\sim}{R}(x_1,\ldots)$ is logically equivalent to some system of linear equations over the two-element field {0,1}; that is, if $\underset{\sim}{R}(x_1,\ldots)$ is logically equivalent to a conjunction of formulas of the forms $\xi_1 \oplus \ldots \oplus \xi_n = 0$ and $\xi_1 \oplus \ldots \oplus \xi_n = 1$, where $\oplus$ denotes addition modulo 2.

Complexity-theoretic notions, such as P, NP, log-space reducibility, etc. are defined briefly in the Appendix.

Examples

The relation $R_1 = \{(1,0,0,0),(0,1,1,0),(0,1,0,1),(1,0,1,1)\}$ is affine, since $\underset{\sim}{R}_1(u,x,y,z)$ is equivalent to $(u \oplus x = 1) \wedge (x \oplus y \oplus z = 0)$.

The relation $R_2 = \{(0,0,0),(0,0,1),(0,1,0),(1,1,0)\}$ is bijunctive and weakly negative, since $\underset{\sim}{R}_2(x,y,z)$ is equivalent to $(\neg x \vee y) \wedge (\neg y \vee \neg z)$. It is also, obviously, 0-valid.

The relation $R_3 = \{(0,1),(1,0)\}$ is defined by the formula $(x \vee y) \wedge (\neg x \vee \neg y)$, or equivalently, $x \oplus y = 1$. Hence this relation is bijunctive and affine. It is not, however, weakly positive or weakly negative -- this can be shown using Lemma 3.1W.

The relation $R_4 = \{(0,0,0),(1,1,1)\}$ is defined by the formula $(x \equiv y) \wedge (y \equiv z)$, or equivalently, $(x \vee \neg y) \wedge (y \vee \neg x) \wedge (y \vee \neg z) \wedge (z \vee \neg y)$. Hence, it is 0-valid, 1-valid, weakly positive, weakly negative, affine and bijunctive.

The relation $R_5 = \{(0,0,1),(0,1,0),(0,1,1),(1,0,0),(1,0,1),(1,1,0)\}$ is the complement of R_4. It does not have any of the six properties listed for R_4 -- this can be proved using Lemmas 3.1A, 3.1B, and 3.1W. Thus, this example shows that none of these properties is preserved under complement.

The relation $R_6 = \{(0,0,1),(0,1,0),(1,0,0)\}$ is the relation "exactly one of three" mentioned in the Introduction. It can be shown, using Lemmas 3.1A, 3.1B and 3.1W, that it is not weakly positive, not weakly negative, not affine and not bijunctive.

By applying Theorem 2.1 with $S=\{R_5\}$ and $S=\{R_6\}$ respectively, it can be deduced that the NOT-ALL-EQUAL and ONE-IN-THREE satisfiability problems, defined in Section 1, are log-complete in NP. We omit the proofs.

Method of Proof

The key question on which the proof of the Dichotomy Theorem centers is: For a given S, what relations are definable by existentially quantified S-formulas? For example, is S={R}, where R is the relation "exactly one of x,y,z," then the existentially quantified S-formula $(\exists u_1,u_2,u_3)(\underset{\sim}{R}(x,u_1,u_3) \wedge \underset{\sim}{R}(y,u_2,u_3) \wedge \underset{\sim}{R}(u_1,u_2,z))$ defines the relation $\{(1,1,1),(1,0,0),(0,1,0),(0,0,1)\}$, which in the notation of Section 3 could be written $[x \oplus y \oplus z = 1]$. Moreover, for this particular S, it turns out that every logical relation is definable by some existentially quantified S-formula. This fact readily implies the NP-completeness of SAT(S).

Another way to state this fact is as a closure property: The smallest set of relations which contains S and is closed under certain operations (conjunction and existential quantification) is the set of all logical relations. From this point of view, the general problem can be phrased as follows: What sets of logical relations are closed under these operations? If we can obtain a reasonably succinct classification of the sets of relations that are closed in this way, then this may serve as a basis for classifying the complexity of SAT(S) for various S.

We do in fact obtain a classification theorem along these lines. Section 3 is devoted to its statement and proof, and a refinement of it is given in Section 5. This theorem classifies the sets of logical relations that are closed under composition, substitution of constants for variables, and existential quantification. Although the classification is not so thorough as to give a complete enumeration of the sets having this closure property, it does permit the complexity of the corresponding satisfiability problem to be determined up to log-space equivalence in all cases.

The closure of the set S under these three operations is denoted Rep(S). It is interesting to note that the corresponding satisfiability-with-constants problem, $SAT_C(S)$, is NP-complete just when Rep(S) is the set of all logical relations. Thus, NP-completeness is closely tied to a kind of functional completeness. (In Section 3 of [Sch] we observed and exploited a similar "logical completeness property" which is probably exhibited in some form by all known NP-complete problems.)

Relation to Earlier Work

The work presented here is similar in spirit to the classification by Post [P] of the sets of logical functions that are closed under functional composition. In both cases, it is shown that "functional completeness" holds provided that the generating set is not included in one of a finite number of restricted classes of functions or relations. But the generating operations are quite different, and to the best of our knowledge, none of the particulars of Post's proof carry over to this work.

Our generalized satisfiability problem embraces, as particular cases, a number of previously studied problems. Of the NP-complete cases, so far as we know, only the standard CNF satisfiability problem with 3 literals per clause has appeared in the literature [C]. Of the polynomial-time decidable cases, all are either trivial or previously known. The satisfiability problem for weakly negative formulas is essentially identical to the problem called UNIT which is shown to be complete in P in [JL]. A restricted form of weakly positive satisfiability is equivalent (under complement) to the digraph reachability problem, a complete problem in nondeterministic log space [Sav]. Our work makes use of all these earlier completeness results.

3. CLASSIFICATION OF LOGICAL RELATIONS

This section presents the classification theorem (Theorem 3.0) which is the essential part of the proof of the Dichotomy Theorem. This theorem classifies the sets of logical relations that are closed under certain operations (conjunction, substitution of constants for variables, and existential quantification), showing that any such set consists exclusively of relations which are in one of the four classes weakly positive, weakly negative, affine or bijunctive, or else is the set of all logical relations.

A key part of the proof, which is also of independent interest, is a series of lemmas (Lemmas 3.1A, 3.1B and 3.1W) which characterize these four classes of relations in semantic terms, that is, in terms of what elements are in the relation, rather than in terms of defining formulas as in the definitions.

The results of this section deal purely with logical relations; no complexity-theoretic notions are involved.

Definitions

The definition of S-formula was given in Section 1. We use the term formula in a larger sense, to mean any well-formed formula, formed from variables, constants, logical connectives, parentheses, logical relation symbols and existential and universal quantifiers -- the intent here is to include whatever notation is handy for expressing a relation among propositional variables.

To clarify these terms: (a) A variable, for purposes of this paper, is an element of the set $\{x,x_0,x_1,\ldots,y,y_0,y_1,\ldots,z,z_0,\ldots,u,u_0,\ldots,v,v_0,v_1,\ldots\}$. Variables, like formulas, are strings of symbols; and we construe, e.g., the variable x_{18} to be a string of length 3. (b) A constant is one of the symbols 0,1 (1=true, 0=false). (c) A logical connective is one of the symbols $\neg, \wedge, \vee, \rightarrow, \equiv, \not\equiv$ which have their usual meanings of "not", "and", "or", "implies", "equals", "does not equal." (d) Each logical relation R has associated with it a logical relation symbol, denoted $\underset{\sim}{R}$, as in Section 1. (e) The quantifiers $(\exists x)$ and $(\forall x)$ are interpreted to mean "for some $x \in \{0,1\}$" and "for all $x \in \{0,1\}$".

A literal is a variable or a negated variable, i.e., ξ or $\neg\xi$ for some variable ξ.

The notation $\underset{\sim}{R}(x_1,\ldots)$ is shorthand for $\underset{\sim}{R}(x_1,\ldots,x_k)$ where k is the rank of R.

If A is a formula, then Var(A) denotes the set of free (i.e., unquantified) variables occurring in A. An assignment for A is a function $s:\mathrm{Var}(A)\rightarrow\{0,1\}$. We say the assignment s satisfies A if s makes A true under the usual rules of interpretation.

We define Sat(A) to be the set of all assignments $s:\mathrm{Var}(A)\rightarrow\{0,1\}$ which satisfy A. Two formulas A and B are logically equivalent if Var(A) = Var(B) and Sat(A) = Sat(B).

Let A be a formula, $V \subseteq \mathrm{Var}(A)$, and $i \in \{0,1\}$. Then $K^A_{i,V}$ denotes the assignment $s:\mathrm{Var}(A)\rightarrow\{0,1\}$ defined by $s(\xi)=i$ iff $\xi \in V$. Usually we write just $K_{i,V}$ and let the domain be inferred from context. K_i denotes the assignment which has the constant value i; again the domain is inferred from context.

If A is a formula, ξ is a variable, and w is a literal, then $A[^{\xi}_{w}]$ denotes the formula formed from A by replacing each occurrence of ξ by w. If V is a set of variables, then $A[^{V}_{w}]$ denotes the result of substituting w for every occurrence of every variable in V. Multiple substitutions are denoted by expressions such as $A[^{V}_{w}, ^{V'}_{w'}, ^{V''}_{w''}]$ with obvious meaning.

The set of existentially quantified S-formulas with constants is denoted Gen(S). Specifically, Gen(S) is the smallest set of formulas such that (a) for all $R \in S$, $\underset{\sim}{R}(x_1,\ldots) \in \mathrm{Gen}(S)$, and (b) for all $A,B \in \mathrm{Gen}(S)$ and all variables ξ,η, the following are all in Gen(S): $A \wedge B$, $A[^{\xi}_{\eta}]$, $A[^{\xi}_{0}]$, $A[^{\xi}_{1}]$ and $(\exists\xi)A$.

$\mathrm{Gen}^+(S)$ denotes the set of all formulas which are logically equivalent to some formula in Gen(S).

If A is a formula, then we denote by [A] the logical relation defined by A, when the variables are taken in lexicographic order. For example, $[z \not\equiv (x \vee y)]$ is the 3-place relation $\{(0,0,1),(0,1,0),(1,0,0),(1,1,0)\}$.

Finally we define $\mathrm{Rep}(S) := \{[A] : A \in \mathrm{Gen}(S)\}$. Rep(S) is the set of relations that are "representable" by quantified S-formulas with constants. Observe that if $S \subseteq S'$, then $\mathrm{Rep}(S) \subseteq \mathrm{Rep}(S')$.

Classification Theorem for Rep(S)

Theorem 3.0. Let S be any set of logical relations. If S satisfies one of the conditions (a)-(d) below, then Rep(S) satisfies the same condition. Otherwise, Rep(S) is the set of all logical relations.

(a) Every relation in S is weakly positive.
(b) Every relation in S is weakly negative.
(c) Every relation in S is affine.
(d) Every relation in S is bijunctive.

The remainder of this section is devoted to the proof of Theorem 3.0.

Lemma 3.1A. Let R be a logical relation and let $A := \underset{\sim}{R}(x_1,\ldots)$. Then R is affine if and only if for all $s_1,s_2,s_3 \in \mathrm{Sat}(A)$, $s_1 \oplus s_2 \oplus s_3 \in \mathrm{Sat}(A)$.

Proof. We use the following fact, which can be proved using elementary linear algebra. If K is a field, then a subset $D \subseteq K^n$ is the solution set of a system of linear equations over K iff for all $b_1,b_2,b_3 \in D$ and all $c_1,c_2,c_3 \in K$ with $c_1+c_2+c_3=1$, $c_1b_1+c_2b_2+c_3b_3 \in D$. In case K is the field $\{0,1\}$, this condition is equivalent to "the sum of any three elements of D is in D." Since R is affine iff A is equivalent to a system of linear equations over $\{0,1\}$, the lemma follows from this fact.[]

Remark. The cardinality of an affine relation is always a power of 2. (This follows from standard results in linear algebra.) This fact is often of use for showing that a relation is not affine.

We now define some terminology for the next lemma.

If ξ is a variable, we use the notation $\langle\xi,i\rangle$ to denote the literal ξ if i=1 and $\neg\xi$ if i=0. As is customary, $\bar{\alpha}$ denotes the complementary literal of α, that is, the literal $\langle\xi,1-i\rangle$ where $\alpha = \langle\xi,i\rangle$. We say the literal $\alpha = \langle\xi,i\rangle$ is consistent with a formula A if $s(\xi)=i$ for some $s \in \mathrm{Sat}(A)$. We say the assignment s agrees with the literal α if $\alpha = \langle\xi, s(\xi)\rangle$ for some variable ξ. A set of literals is consistent if it does not contain α and $\overline{\alpha}$ for any literal α.

If s is an assignment and Q is a consistent set of literals, we denote by s#Q the assignment which differs from s just on the set $\{\xi : \langle\xi, 1-s(\xi)\rangle \in Q\}$.

Let α be a literal and A a formula. We define $Imp_A(\alpha)$ to be the set of all literals β such that every $s \in Sat(A)$ which agrees with α also agrees with β. Thus, $Imp_A(\alpha)$ is the set of literals which are "implied" by the literal α. For example, if $A = x \vee y$, then $\langle y,1\rangle \in Imp_A(\langle x,0\rangle)$.

Let A be a formula and $s \in Sat(A)$. A change set for (A,s) is any set $V \subseteq Var(A)$ such that $s \oplus K_{1,V} \in Sat(A)$. That is, V is a change set for (A,s) iff the assignment which differs from s just on V satisfies A.

Lemma 3.1B. Let R be a logical relation and let $A := \underset{\sim}{R}(x_1,\ldots)$. Then the following are equivalent:

(a) R is bijunctive.

(b) For every $s \in Sat(A)$, if V_1 and V_2 are change sets for (A,s) then so is $V_1 \cap V_2$.

(c) For every $s \in Sat(A)$ and every literal α which is consistent with A, $s\#Imp_A(\alpha) \in Sat(A)$. (See Note on last page of this section.)

Proof. (a)$\Rightarrow$(b): Assume R is bijunctive. Thus, A is logically equivalent to some formula B which is a conjunction of clauses of the form $(\alpha \rightarrow \beta)$, where α,β are literals. Let $s \in Sat(A)$ be given, and let V_1, V_2 be change sets for (A,s). Let Q be the smallest set of literals such that (a) $\{\langle\eta, 1-s(\eta)\rangle : \eta \in V_1 \cap V_2\} \in Q$, and (b) whenever $\alpha \in Q$ and $(\alpha \rightarrow \beta)$ or $(\beta \rightarrow \overline{\alpha})$ is a conjunct of B for some literal β, then $\beta \in Q$. Clearly, any assignment $t \in Sat(B)$ which differs from s on all of $V_1 \cap V_2$ must agree with every literal in Q. Since $s \oplus K_{1,V_1}$ is such an assignment, Q is consistent and Q cannot contain any literal $\langle\eta, 1-s(\eta)\rangle$ with $\eta \notin V_1$. Similarly, Q cannot contain any literal $\langle\eta, 1-s(\eta)\rangle$ with $\eta \notin V_2$. Hence, $s\#Q = s \oplus K_{1,V_1 \cap V_2}$. It is straightforward to show that s#Q satisfies every conjunct of B; hence $s \oplus K_{1,V_1 \cap V_2} \in Sat(B) = Sat(A)$. Hence $V_1 \cap V_2$ is a change set for (A,s).

(b)$\Rightarrow$(c): Assume that (b) holds. Let $s \in Sat(A)$, and let α be a literal consistent with A. We want to show that $s\#Imp_A(\alpha) \in Sat(A)$. Assume† that s disagrees with α; that is, $\alpha = \langle\xi, 1-s(\xi)\rangle$ for some variable ξ. Let $W := \{\eta : \langle\eta, 1-s(\eta)\rangle \in Imp_A(\alpha)\}$; that is, W is the set of variables on which $Imp_A(\alpha)$ clashes with s. We claim that $W = \bigcap\{V : V$ is a change set for (A,s) & $\xi \in V\}$. To prove this claim, first note that any change set containing ξ must also contain all of W, since W consists of all variables which are forced to change as a result of changing ξ. On the other hand, if some variable η is not contained in W, then there is some assignment t such that $t(\xi) \neq s(\xi)$ but $t(\eta) = s(\eta)$, so that η is not contained in the change set $\{\zeta : t(\zeta) \neq s(\zeta)\}$. This proves the claim.

Now by multiple application of hypothesis (b), W is itself a change set for (A,s). Thus, $s \oplus K_{1,W} = s\#Imp_A(\alpha) \in Sat(A)$, as was to be shown.

(c)$\Rightarrow$(a): Assume that for every $s \in Sat(A)$ and every literal α which is consistent with A, $s\#Imp_A(\alpha) \in Sat(A)$. Define B to be the conjunction of $\{(\alpha \rightarrow \beta) : \alpha,\beta$ are literals & $\beta \in Imp_A(\alpha)\}$. Note that Var(B)=Var(A), since B has the conjunct $(\xi \rightarrow \xi)$ for each $\xi \in Var(A)$. We claim that B is logically equivalent to A, and hence R is bijunctive.

We must show that Sat(B)=Sat(A). Clearly, $Sat(A) \subseteq Sat(B)$, since any assignment satisfying A must satisfy each conjunct of B.

It remains to show $Sat(B) \subseteq Sat(A)$. Suppose, for sake of contradiction, that $s_1 \in Sat(B) - Sat(A)$. Choose $s_2 \in Sat(A)$ such that $|W|$ is maximum, where $W = \{\eta : s_1(\eta) = s_2(\eta)$. Choose $\xi \in Var(A) - W$, and let $\alpha := \langle\xi, s_1(\xi)\rangle$. The literal α is consistent with A, because if α were inconsistent with A, then B would have a conjunct asserting this fact (that is, if $\alpha = \langle\xi,0\rangle$ is inconsistent with A, then B has a conjunct $(\neg\xi \rightarrow \xi)$, and if $\alpha = \langle\xi,1\rangle$ is inconsistent with A, then B has a conjunct $(\xi \rightarrow \neg\xi)$), and this would force $s_2(\xi) = s_1(\xi)$. Let $s_3 := s_2\#Imp_A(\alpha)$. By hypothesis s_3 satisfies A.

We claim that for all $\eta \in W$, $s_3(\eta) = s_2(\eta)$. To see this, suppose $\eta \in W$ with $s_3(\eta) \neq s_2(\eta)$. Since now $\langle\eta, 1-s_2(\eta)\rangle \in Imp_A(\alpha)$, B has a conjunct $(\alpha \rightarrow \langle\eta, 1-s_2(\eta)\rangle)$, or equivalently $(\langle\xi, s_1(\xi)\rangle \rightarrow \langle\eta, 1-s_1(\eta)\rangle)$. This conjunct is not satisfied by s_1, contradicting the assumption that s_1 satisfies B. This proves the claim.

Thus, s_3 agrees with s_1 on all of $W \cup \{\xi\}$. This contradicts the fact that s_2 was chosen to maximize $|W|$. The contradiction completes the proof. []

Let A be a formula, let $i \in \{0,1\}$, and let $V \subseteq Var(A)$. Define the i-closure of V with respect to A to be the set $Cl_{i,A}(V) := \{\xi \in Var(A) :$ for all $s \in Sat(A)$ such that $s|V \equiv i$, $s(\xi) = i\}$. In other words, $Cl_{0,A}(V)$ (resp. $Cl_{1,A}(V)$) is the set of variables which are forced to be false (resp. true) by all variables of V being false (resp. true).

It is easy to see that $V \subseteq Cl_{i,A}(V)$, and that $V \subseteq V'$ implies $Cl_{i,A}(V) \subseteq Cl_{i,A}(V')$, for all $V,V' \subseteq Var(A)$, $i \in \{0,1\}$. Call the set $V \subseteq Var(A)$ i-closed for A if $V = Cl_{i,A}(V)$. Also, call V i-consistent for A if there is some $s \in Sat(A)$ such that $s|V \equiv i$. We say V is i-nonclosed (resp. i-inconsistent) for A if V is not i-closed (resp. i-consistent) for A.

Lemma 3.1W. Let R be a logical relation and let $A := \underset{\sim}{R}(x_1,\ldots)$. Then (a) R is weakly positive if and only if whenever $V \subseteq Var(A)$ is 0-consistent and 0-closed for A, $K_{0,V} \in Sat(A)$; and (b) R is weakly negative if and only if whenever $V \subseteq Var(A)$ is 1-consistent and 1-closed for A, $K_{1,V} \in Sat(A)$.

Proof. We just prove part (a). The proof of (b) is similar. If R is empty, the lemma holds trivially, so assume R is nonempty.

($\Rightarrow$): Assume that R is weakly positive. Thus A is logically equivalent to some CNF formula A' having at most one negated variable per conjunct. It suffices to show that if $V \subseteq Var(A')$ is 0-consistent and 0-closed for A', then $K_{0,V} \in Sat(A')$. Let V be such a set and suppose to the contrary that $K_{0,V} \notin Sat(A')$. Let C be a conjunct of A' on which $K_{0,V}$ fails. Let U be the set of unnegated variables of C. Since $K_{0,V}$ fails on C, $U \subseteq V$. If C has no

---

†If s agrees with α, the conclusion is immediate, since then $s\#Imp_A(\alpha) = s$.

negated variable, then this contradicts the fact that V is 0-consistent for A'. Otherwise, let η be the unique negated variable of C. It can be seen that $\eta \in Cl_{0,A'}(U)$. Also, since $K_{0,V}$ fails on C, $\eta \notin V$. This contradicts the fact that V is 0-closed for A'. This proves that in fact $K_{0,V} \in Sat(A')$.

($\Leftarrow$). Assume that $K_{0,V} \in Sat(A)$ for all 0-closed, 0-consistent sets $V \subseteq Var(A)$. Let A' be the conjunction of all the clauses $\{(\xi_1 \vee \ldots \vee \xi_n) : \{\xi_1,\ldots,\xi_n\}$ is 0-inconsistent for A$\} \cup \{(\neg\eta \vee \xi_1 \vee \ldots \vee \xi_n) : \eta \in Cl_{0,A}(\{\xi_1,\ldots,\xi_n\})\}$. Since every variable is contained in its own 0-closure, A' has a conjunct $(\neg\xi \vee \xi)$ for each $\xi \in Var(A)$; hence, Var(A')=Var(A). We claim that A' is logically equivalent to A.

To show $Sat(A) \subseteq Sat(A')$, suppose that $s \notin Sat(A')$. Let C be a conjunct of A' on which s fails. If C is of the form $(\xi_1 \vee \ldots \vee \xi_n)$, then $s(\xi_1)=\ldots=s(\xi_n)=0$ and, by the definition of A', $\{\xi_1,\ldots,\xi_n\}$ is 0-inconsistent for A; hence, $s \notin Sat(A)$. Otherwise C is of the form $(\neg\eta \vee \xi_1 \vee \ldots \vee \xi_n)$, and so $s(\eta)=1$, $s(\xi_1)=\ldots=s(\xi_n)=0$. Then by the definition of A', $\eta \in Cl_{0,A}(\{\xi_1,\ldots,\xi_n\})$, hence $s \notin Sat(A)$. This proves that $Sat(A) \subseteq Sat(A')$.

Next we show that $Sat(A') \subseteq Sat(A)$. Suppose $s \notin Sat(A)$. Let $V := \{\xi : s(\xi)=0\}$. By the property assumed for A, V is either 0-inconsistent of 0-non-closed for A. If it is 0-inconsistent, then $(\xi_1 \vee \ldots \vee \xi_n)$ is a conjunct of A', where $V = \{\xi_1, \ldots,\xi_n\}$, and hence $s \notin Sat(A')$. If V is 0-non-closed, let $\eta \in Cl_{0,A}(V) - V$. Then $(\neg\eta \vee \xi_1 \vee \ldots \vee \xi_n)$ is a conjunct of A', and hence $s \notin Sat(A')$. This proves that $Sat(A') \subseteq Sat(A)$.

Thus Sat(A)=Sat(A') and so A' is logically equivalent to A. Hence R is weakly positive. []

Lemma 3.2. Let R be a logical relation. If R is not weakly negative, then $Rep(\{R\}) \cap \{[x \not\equiv y],[x \vee y]\} \neq \phi$. If R is not weakly positive, then $Rep(\{R\}) \cap \{[x \not\equiv y],[\neg x \vee \neg y]\} \neq \phi$.

Corollary 3.2.1. If S contains some relation which is not weakly positive and some relation which is not weakly negative, then $[x \not\equiv y] \in Rep(S)$.

Proof of Corollary. Assume $R,R' \in S$ with R not weakly positive and R' not weakly negative. Suppose, for sake of contradiction, that $[x \not\equiv y] \notin Rep(S)$. Then, by Lemma 3.2, $[x \vee y]$ and $[\neg x \vee \neg y]$ are in Rep(S). Hence, Rep(S) contains $[(x \vee y) \wedge (\neg x \vee \neg y)]$, which is just $[x \not\equiv y]$, contrary to assumption. []

Proof of Lemma 3.2. Let R be a logical relation which is not weakly negative, and let $A := \underline{R}(x_1,\ldots)$. By Lemma 3.1W, there is a set $V \subseteq Var(A)$ which is 1-consistent and 1-closed such that $K_{1,V} \notin Sat(A)$. Let $\overline{V} := Var(A) - V$. Choose $W \subseteq \overline{V}$ of maximum cardinality such that $K_{0,W} \in Sat(A)$. It can be seen from the definitions that $1 \leq |W| < |\overline{V}|$. Choose $\xi \in \overline{V} - W$, and let s be an assignment satisfying A such that $s|V \equiv 1$ and $s(\xi)=0$. Such an s exists because V is 1-closed and 1-consistent.

For j=0,1 define $W_j := \{\eta \in W : s(\eta)=j\}$, $\overline{W}_j := \{\eta \in Var(A) - W : s(\eta)=j\}$, and let

$$B := A[{}^{W_0}_{0}, {}^{W_1}_{x}, {}^{\overline{W}_0}_{y}, {}^{\overline{W}_1}_{1}].$$

W_1 is nonempty by maximality of $|W|$. $\overline{W}_0$ is non-empty since it contains ξ. Thus x and y both actually occur in B.

Clearly $[B] \in Rep(\{R\})$. Also, [B] contains (0,1) and (1,0), because A is satisfied by $K_{0,W}$ and s respectively. And [B] does not contain (0,0), by maximality of $|W|$. Thus, depending on whether (1,1) is in [B], [B] is either $[x \not\equiv y]$ or $[x \vee y]$. This proves the lemma for the case where R is weakly negative.

The proof of the weakly positive case is similar. []

The following lemma is of frequent use in what follows.

Negated Substitution Lemma. Assume $[x \not\equiv y] \in Rep(S)$. Then Gen+(S) is closed under negated substitution; that is, if $A \in Gen^+(S)$ and ξ,η are variables, $A[{}^{\xi}_{\neg\eta}] \in Gen^+(S)$.

Proof. By hypothesis $Gen^+(S)$ contains the formula $x \not\equiv y$. Observe that $A[{}^{\xi}_{\neg\eta}]$ is logically equivalent to $(\exists u)(A[{}^{\xi}_{u}] \wedge \eta \not\equiv u)$, where u is a variable not occurring in A. Hence, $A[{}^{\xi}_{\neg\eta}] \in Gen^+(S)$. []

By a "3-element binary logical relation" we mean a 2-place logical relation having exactly 3 elements. It is easy to verify that there are exactly four such relations, and that these are $[x \vee y]$, $[\neg x \vee y]$, $[x \vee \neg y]$ and $[\neg x \vee \neg y]$.

Lemma 3.3. Let R be a relation which is not affine. Then $Rep(\{R ,[x \not\equiv y]\})$ contains all 3-element binary logical relations.

Proof. It suffices to show that $Rep(\{R,[x \not\equiv y]\})$ contains some 3-element binary relation, since the others can then be obtained by use of the Negated Substitution Lemma.

Let $A := \underline{R}(x_1,\ldots)$. Using Lemma 3.1A, let s_0,s_1,s_2 be assignments satisfying A such that $s_0 \oplus s_1 \oplus s_2$ does not satisfy A. Form A' from A by negating all occurrences of variables in the set $\{\eta : s_0(\eta)=1\}$. By the Negated Substitution Lemma, $A' \in Gen^+(\{R,[x \not\equiv y]\})$. Define $s_i' := s_i \oplus s_0$, for i=1,2. Observe that an assignment t satisfies A' iff $t \oplus s_0$ satisfies A. Thus, K_0 (the all-zero assignment), s_1', s_2' all satisfy A', but $s_1' \oplus s_2'$ does not.

For i,j = 0,1, let $V_{i,j} := \{\xi \in Var(A') : s_1'(\xi)=i \ \& \ s_2'(\xi)=j\}$, and let

$$B := A[{}^{V_{0,0}}_{0}, {}^{V_{0,1}}_{x}, {}^{V_{1,0}}_{y}, {}^{V_{1,1}}_{z}].$$

Clearly, $B \in Gen^+(\{R,[x \not\equiv y]\})$. Assume without loss of generality that x,y,z all actually occur in B. (For example, if x does not occur, one can add a conjunct $(\exists w)(w \not\equiv x)$ just to make it occur.)

By the statement just made about satisfaction of A', [B] contains (0,0,0),(0,1,1) and (1,0,1), but not (1,1,0).

Assume, for sake of contradiction, that $Rep(\{R,[x \not\equiv y]\})$ does not contain any 3-element binary relation. Then [B] must contain (0,1,0), or else $[(\exists x)B]$ is {(0,0),(1,1),(0,1)}. Also, [B] must contain (1,0,0), or else $[(\exists y)B]$ is{(0,0),(0,1),(1,1)}. But then $[B[{}^{z}_{0}]]$ is {(0,0),(0,1),(1,0)}, and this contradiction completes the proof. []

Lemma 3.4. Let R be a logical relation which is not bijunctive. Then Rep({R,[$x \neq y$],[$x \vee y$]}) contains the relation "exactly one of x,y,z."

Proof. Let $A := \underline{R}(x_1,\ldots)$. By part (b) of Lemma 3.1B, there exist $s_0 \in \mathrm{Sat}(A)$ and $U,V \subseteq \mathrm{Var}(A)$ such that U and V are change sets for (A,s), but $U \cap V$ is not a change set for (A,s).

Form A' from A by negating all occurrences of each variable in the set $\{\eta : s_0(\eta)=1\}$. By the Negated Substitution Lemma, $A' \in \mathrm{Gen}^+(\{R,[x \neq y]\})$. Observe that U and V are change sets for (A',K_0), where K_0 is the all-zero assignment, but $U \cap V$ is not. (Also, K_0 satisfies A'.) Define

$$B := A'\left[\begin{smallmatrix}\mathrm{Var}(A')-(U \cup V) \\ 0\end{smallmatrix}, \begin{smallmatrix}U \cap V \\ x\end{smallmatrix}, \begin{smallmatrix}U-V \\ y\end{smallmatrix}, \begin{smallmatrix}V-U \\ z\end{smallmatrix}\right]$$

By the above remarks about change sets for (A',K_0), [B] contains (0,0,0), (1,1,0) and (1,0,1), but not (1,0,0). Now define

$$B' := B\left[\begin{smallmatrix}x \\ \neg x\end{smallmatrix}\right] \wedge (\neg x \vee \neg y) \wedge (\neg y \vee \neg z) \wedge (\neg z \vee \neg x)$$

By the Negated Substitution Lemma, $B' \in \mathrm{Gen}^+(\{R, [x \neq y],[x \vee y]\})$. It is easy to check that [B'] = {(1,0,0),(0,1,0),(0,0,1)}. That is, [B'] is the relation "exactly one of x,y,z." []

Lemma 3.5. Let R be the logical relation "exactly one of x,y,z." Then Rep({R}) is the set of all logical relations.

Proof. Define

$$A := (\exists u_1,u_2,u_3,u_4,u_5,u_6)(R(x,u_1,u_4) \wedge R(y,u_2,u_4)$$

$$\wedge R(u_1,u_2,u_5) \wedge R(u_3,u_4,u_6) \wedge R(z,u_3,0)$$

$$B := R(x,y,0)$$

It is straightforward to verify that A is logically equivalent to $(x \vee y \vee z)$ and that B is logically equivalent to $x \neq y$.

Let a logical relation Q be given and let Q=[C] for some standard propositional formula C. By introducing a new existentially quantified variable for each binary logical connective of C, one can form a formula $(\exists y_1,\ldots,y_m)$ D, equivalent to C, where D is a conjunction of clauses each involving at most 3 variables (and hence D can be expanded to CNF form with at most 3 literals per conjunct.) Details of this process can be found in [St, Lemma 6.4] or [BBFMP]. It is now straightforward, using the formulas A and B, to convert D into an equivalent formula in Gen({R}). It follows that $Q \in \mathrm{Rep}(\{R\})$.

Thus Rep({R}) is the set of all logical relations. []

Note. Condition (b) of Lemma 3.1B can also be expressed in the following pleasantly symmetric form:

(b') For all $s_1,s_2,s_3 \in \mathrm{Sat}(A)$, $(s_1 \vee s_2) \wedge (s_2 \vee s_3) \wedge (s_3 \vee s_1) \in \mathrm{Sat}(A)$.

This is derived from condition (b) by setting $s_1=s$, $s_2 = s \oplus K_{1,V_1}$, $s_3 = s \oplus K_{1,V_2}$ and observing that $((s_2 \oplus s_1) \wedge (s_3 \oplus s_1)) \oplus s_1$ is equivalent to $(s_1 \vee s_2) \wedge (s_2 \vee s_3) \wedge (s_3 \vee s_1)$.

Proof of Theorem 3.0. First we show that if S does not satisfy any of the conditions (a)-(d) of Theorem 3.0, Rep(S) is the set of all logical relations.

Assume that S does not satisfy any of (a)-(d). Then S contains some relation R_1 which is not weakly positive, some relation R_2 which is not weakly negative, some relation R_3 which is not affine, and some relation R_4 which is not bijunctive. By Corollary 3.2.1, $[x \neq y] \in \mathrm{Rep}(\{R_1,R_2\})$. Now by Lemma 3.3, $[x \vee y] \in \mathrm{Rep}(\{R_1,R_2,R_3\})$. Hence, by Lemma 3.4, $\mathrm{Rep}(\{R_1,R_2,R_3,R_4\})$ contains the relation "exactly one of x,y,z," and hence is the set of all logical relations, by Lemma 3.5. Thus, Rep(S) is the set of all logical relations.

It remains to show that if S satisfies one of the conditions (a)-(d), so does Rep(S). The proof of this part does not involve any new techniques, and we leave it as an exercise for the reader. (This part is not needed in the proof of the Dichotomy Theorem.) []

4. PROOF OF DICHOTOMY THEOREM

This section finishes the proof of the Dichotomy Theorem (Theorem 2.1).

Lemma 4.1. (Dichotomy Theorem for Satisfiability-with-Constants) Let S be a finite set of logical relations. If S satisfies one of the conditions (a)-(d) of Theorem 3.0, then $\mathrm{SAT}_C(S)$ is polynomial-time decidable. Otherwise, $\mathrm{SAT}_C(S)$ is log-complete in NP.

Proof. (a) Suppose that every relation in S is weakly positive. Then SAT(S) is decidable using the following algorithm:

1. Given an S-formula A, replace each conjunct of A by an equivalent CNF formula A' having at most one negated variable in each conjunct.
2. If every conjunct of A' contains an unnegated variable, ACCEPT.
3. Otherwise, let $(\neg\xi)$ be a conjunct of A'. If (ξ) is also a conjunct of A', REJECT. Otherwise, drop every conjunct in which $\neg\xi$ occurs and drop ξ from every conjunct in which ξ occurs unnegated. (If A' becomes empty, ACCEPT.)
4. Go to step 2.

We leave verification to the reader.

(b) The case where every relation in S is weakly negative is similar to (a).

(c) Suppose that every relation in S is affine. Then to decide whether a given S-formula A is satisfiable, convert A to an equivalent system of linear equations over {0,1} and solve the system by Gaussian elimination. (Eliminate one variable at a time until either all variables have been eliminated or 0=1 has been deduced.) This is a well-known polynomial-time algorithm.

(d) Suppose that every relation in S is bijunctive. Then to decide whether a given S-formula is satisfiable, convert it to an equivalent CNF formula with at most 2 literals per conjunct and use the Davis-Putnam procedure [DP], which as noted in [C] decides satisfiability of such formulas in polynomial time.

In (a)-(d) above, we have sketched polynomial-time algorithms for SAT(S). For $SAT_C(S)$, that is if the formulas contain constants, it is obvious how to modify the algorithms.

Assume now that S does not satisfy any of the conditions (a)-(d). We will show that $SAT_C(S)$ is NP-complete by showing SAT3 $\leq_{\log}$ $SAT_C(S)$, where SAT3 is the set of satisfiable CNF formulas having at most 3 literals per conjunct, a known NP-complete problem [C].

Let R_0, R_1, R_2, R_3 be the 3-place logical relations defined by $\underset{\sim}{R}_0(x,y,z) \equiv (x \vee y \vee z)$, $\underset{\sim}{R}_1(x,y,z) \equiv (\neg x \vee y \vee z)$, $\underset{\sim}{R}_2(x,y,z) \equiv (\neg x \vee \neg y \vee z)$, $\underset{\sim}{R}_3(x,y,z) \equiv (\neg x \vee \neg y \vee \neg z)$. For i=0,1,2,3, let $F_i(x,y,z)$ be a formula in Gen(S) which is logically equivalent to $\underset{\sim}{R}_i(x,y,z)$. Such formulas exist by Theorem 3.0.

Given a CNF formula A with at most 3 literals in each conjunct, form an equivalent formula A' by replacing each conjunct of A by one of the formulas F_i with appropriate variables substituted. Form A" from A' by deleting all quantifiers, after making sure that all quantified variables are distinct from each other and from all free variables. Observe that A" is satisfiable iff A is satisfiable. It is not hard to show that A" is log-space computable from A. This proves that SAT3 $\leq_{\log}$ $SAT_C(S)$.

Since clearly $SAT_C(S) \in NP$, it follows that the problem $SAT_C(S)$ is log-complete in NP. []

We define "no-constants" analogues of Gen(S) and Rep(S), as follows. Let $Gen_{NC}(S) := \{A \in Gen(S)$: no constants occur in A$\}$, $Rep_{NC}(S) := \{[A] : A \in Gen_{NC}(S)\}$.

The logical relation R is <u>complementive</u> if it is closed under complement, that is, if for all $(a_1,\ldots,a_m) \in R$, $(1-a_1,\ldots,1-a_m) \in R$.

The following easily-proved lemma clarifies the relation between SAT(S) and $SAT_C(S)$.

<u>Lemma 4.2</u>. If $Rep_{NC}(S)$ contains [x] and $[\neg x]$, then $Rep(S) = Rep_{NC}(S)$, and hence $SAT_C(S)$ and SAT(S) have the same complexity.

As the next lemma shows, the hypothesis of Lemma 4.2 fails only under very restricted conditions. Thus, for "most" sets S, SAT(S) and $SAT_C(S)$ have the same complexity.

<u>Lemma 4.3</u>. Let S be a set of nonempty logical relations. Then at least one of the following holds:

(a) Every relation in S is 0-valid.
(b) Every relation in S is 1-valid.
(c) [x] and $[\neg x]$ are contained in $Rep_{NC}(S)$.
(d) $[x \not\equiv y] \in Rep_{NC}(S)$.

Moreover, if (c) fails and (d) holds, every relation in S is complementive.

<u>Proof</u>. Assume as noted above that all relations in S are nonempty. Assume that (a) and (b) fail. We will show that (c) or (d) holds.

<u>CASE 1</u>. Every relation in S is 0-valid or 1-valid. In this case, since (a) and (b) fail, there is some $R_0 \in S$ that is 0-valid but not 1-valid, and some $R_1 \in S$ that is 1-valid but not 0-valid. Letting $A_i := \underset{\sim}{R}_i(x_1,\ldots)$, we have $[A_i[^{\mathrm{Var}(A_i)}_{x}]] = \{(i)\}$ for i=0,1; hence, [x] and $[\neg x]$ are in $Rep_{NC}(S)$, so (c) holds.

<u>CASE 2</u>. S contains some nonempty relation R that is neither 0-valid nor 1-valid. In this case, let $A := \underset{\sim}{R}(x_1,\ldots)$ and choose $s \in Sat(A)$. Set

$$B := A[^{s^{-1}(\{0\})}_{x_0}, {}^{s^{-1}(\{1\})}_{x_1}]$$

Since R is neither 0-valid nor 1-valid, x_0 and x_1 both occur in B and [B] is either $\{(0,1)\}$ or $\{(0,1),(1,0)\}$. In the former case, $[(\exists x_0)B] = [x]$ and $[(\exists x_1)B] = [\neg x]$, so (c) holds. In the latter case, $[B] = [x \not\equiv y]$, so (d) holds.

Thus (c) or (d) holds in all cases.

Now assume (c) fails and (d) holds. We claim that $[x] \notin Rep_{NC}(S)$. For if $[x] \in Rep_{NC}(S)$, we would have $[(\exists x)(x \wedge x \not\equiv y)] = [\neg y]$, contradicting (c) failing. Similarly, $[\neg x] \notin Rep_{NC}(S)$. Assume for sake of contradiction that $R \in S$ and R is not complementive. Let $A := \underset{\sim}{R}(x_1,\ldots)$ and choose $s \in Sat(A)$ such that $\bar{s} \notin Sat(A)$, where $\bar{s}$ is the complementary assignment of s. Define the formula B as in CASE 2. Now x_0 must occur in B, or else [B] = [x]. Similarly x_1 must occur in B. Thus $Var(B)=\{x_1,x_2\}$ and [B] contains (0,1) but not (1,0). Now [B] must contain (0,0), or else $[(\exists x_0)B]$ is [x]. And [B] must contain (1,1), or else $[(\exists x_1)B]$ is $[\neg x]$. Thus, [B] is $[x \rightarrow y]$. But then $[(\exists x)((x \rightarrow y) \wedge (x \not\equiv y))]$ is [y], and this contradiction completes the proof that every relation in S is complementive. []

Finally, we are able to prove the Dichotomy Theorem for SAT(S).

<u>Proof of Theorem 2.1</u>. Assume that not every relation in S is 0-valid and not every relation in S is 1-valid (that is, conditions (a) and (b) of Theorem 2.1 fail). By Lemma 4.3 there are two cases to consider.

<u>CASE 1</u>. [x] and $[\neg x]$ are in $Rep_{NC}(S)$.

In this case, it is easy to replace any given S-formula with constants by an equivalent S-formula without constants, so the conclusion follows by Lemma 4.1.

<u>CASE 2</u>. $[x \not\equiv y] \in Rep_{NC}(S)$, and every relation in S is complementive.

In this case, let an S-formula with constants, A be given. Let A' be $A'' \wedge (y_0 \not\equiv y_1)$, where A" is formed from A be replacing each occurrence of 0 by y_0 and each occurrence of 1 by y_1, and y_0, y_1 are new variables. Thus, A' is an S-formula without constants, and, by complementarity, A' is satisfiable iff A is satisfiable. So again the conclusion follows by Lemma 4.1. []

5. REFINEMENT TO LOG-SPACE EQUIVALENCE

This section classifies the complexity of $SAT_C(S)$ up to log-space equivalence. Results are presented without proof. The definition of $(\leq k)$-weakly positive is found later in this section.

<u>Classification Theorem for $SAT_C(S)$</u>

<u>Theorem 5.1</u>. Let S be a finite set of logical relations. Then $SAT_C(S)$ lies in one of seven log-space equivalence classes, as follows: ($\overline{SAT_C}(S)$ denotes the complement of $SAT_C(S)$.)

L1. If every relation in S is (≤ 0)-weakly positive, then $SAT_C(S)$ is decidable deterministically in log space.

L2. If every relation in S is (≤ 1)-weakly positive, and S contains some non-(≤ 0)-weakly positive relation, then either
   (a) $\overline{SAT_C}(S)$ is log-equivalent to the graph reachability problem (given a graph G and nodes s,t, do s and t lie in the same connected component of G?).
   or (b) $\overline{SAT_C}(S)$ is log-equivalent to the digraph reachability problem (given a digraph G and nodes s,t, is there a directed path from s to t?), and hence is log-complete in nondeterministic log space.

L3. If every relation in S is weakly positive, and S contains some relation that is not (≤ 1)-weakly positive, then Rep(S) is the set of all weakly positive relations, and $SAT_C(S)$ is log-complete in P.

Statements L1-L3 also hold when "negative" is substituted for "positive."

L4. If S contains some relation that is not weakly positive and some relation that is not weakly negative, and every relation is S is affine and bijunctive, then Rep(S)=Rep([$x \not\equiv y$]), and $SAT_C(S)$ is log-equivalent to the problem of deciding whether a graph is bipartite.

L5. If S contains some relation that is not weakly positive, some relation that is not weakly negative, and some relation that is not bijunctive, and every relation in S is affine, then Rep(S) = Rep([$x \oplus y \oplus z = 0$]), and hence $SAT_C(S)$ is log-equivalent to the problem of deciding whether an arbitrary system of linear equations over the field {0,1} is consistent.

   Remark. The latter problem is log-equivalent to its own complement, since a set of linear equations is inconsistent iff there is some subset of them which sums to 0=1, a condition which itself can be written as a set of linear equations. Thus for this case, any class in which $SAT_C(S)$ is complete is closed under complement.

L6. If S contains some relation that is not weakly positive, some relation that is not weakly negative, and some relation that is not affine, and every relation in S is bijunctive, then Rep(S)=Rep([$x \to y$]) and $\overline{SAT_C}(S)$ is log-complete in nondeterministic log space.

L7. If S contains some relation that is not weakly positive, some relation that is not weakly negative, some relation that is not bijunctive, and some relation that is not affine, then Rep(S) is the set of all logical relations, and $SAT_C(S)$ is log-complete in NP.

Corollary 5.2. SAT3W and NOT-EXACTLY-ONE SATISFIABILITY (defined in Section 1) are log-complete in P. (This follows from statement L3.)

The proof of L3 relies on the result of Jones and Laaser [JL] that the problem UNIT (the set of CNF formulas that yield a contradiction from unit resolution) is log-complete in P. Although [JL] does not give any explicit syntactic characterization, we believe that UNIT is just the set of unsatisfiable CNF formulas having at most one positive variable in each conjunct.

Actually, the proof that SAT3W is complete in P can be given in a manner that completely parallels Cook's proof [C] that SAT3 (the set of satisfiable CNF formulas with 3 literals per conjunct) is complete in NP. Cook's proof takes a sequence of instantaneous machine descriptions and writes a CNF formula that describes the sequence. All that is necessary to get a completeness result in P is to observe that, if the machine involved is deterministic, the CNF formula constructed has at most one positive variable in each conjunct. (This requires certain modifications of Cook's argument, which are carried out in the UNIT proof of [JL].) The reduction to formulas with 3 literals per conjunct again completely parallels Cook's proof -- one has just to observe that this reduction preserves the property of having at most one positive variable per conjunct. (One can then reverse the negativity of all variables so as to get at most one negated variable per conjunct, i.e. SAT3W.)

The proof of statement L2(b) uses the result of Savitch [Sav] (later refined by Jones [J]) that the digraph reachability problem is complete in nondeterministic log space. Jones [J] asks whether the undirected graph reachability problem is also complete in nondeterministic log space.

The completness of the satisfiability problem for bijunctive formulas (cf. L6), which follows from the completeness of the digraph reachability problem, was noted in [JL].

Definitions

Let k be a nonnegative integer or ∞. The relation R is ($\leq k$)-weakly positive iff $\underline{R}(x_1,\ldots)$ is equivalent to some formula $B \wedge (\xi_1 \equiv \xi_1) \wedge \ldots \wedge (\xi_n \equiv \xi_n)$, where $\xi_1,\ldots,\xi_n$ are variables and B is a CNF formula in which each conjunct has at most one negated variable and every conjunct which has a negated variable has at most k unnegated variables. (Note: The clauses $(\xi_i \equiv \xi_i)$ serve no function except to make the variable ξ_i occur vacuously in the formula. Actually this is necessary only if k=0.)

Note that "weakly positive" is the same as "($\leq\infty$)-weakly positive."

Let A be a formula, $V \subseteq Var(A)$, k as above. We define $Cl^{\leq k}_{0,A}(V) := \{\xi : \exists W \subseteq V, |W| \leq k$, such that $\forall s \in Sat(A), s|W \equiv 0 \Rightarrow s(\xi)=0\}$. Thus $Cl^{\leq\infty}_{0,A}(V)$ is the same as $Cl_{0,A}(V)$. We say V is ($0,\leq k$)-closed for A if $Cl^{\leq k}_{0,A}(V) = V$.

(Similar definitions are given with "negative" instead of "positive" and "1" instead of "0".)

The analysis of weakly positive relations in Theorem 5.1 is based on the following lemma, which generalizes Lemma 3.1W.

Lemma 5.3. Let R be a logical relation and let $A := \underline{R}(x_1,\ldots)$ and let k be a nonnegative integer or ∞. Then R is ($\leq k$)-weakly positive iff whenever $V \subseteq Var(A)$ is 0-consistent and ($0,\leq k$)-closed for A, $K_{0,V} \in Sat(A)$.

Lemma 5.4. If R is affine and bijunctive, then $\underline{R}(x_1,\ldots)$ is logically equivalent to a CNF formula composed of conjuncts of the forms $(x \equiv y)$, $(x \not\equiv y)$, x and $\neg x$.

6. EXTENSION TO POLYNOMIAL SPACE

This section gives a polynomial-space analogue of the Dichotomy Theorem, involving quantified formulas. The result is presented without proof.

A <u>quantified S-formula with constants</u> is a member of the smallest set of formulas T such that (a) for each $R \in S$, $\underset{\sim}{R}(x_1,\ldots) \in T$, and (b) whenever $A,B \in T$ and ξ,η are variables, the following are in T: $A \wedge B$, $(\exists\xi)A$, $(\forall\xi)A$, $A[{\xi \atop \eta}]$, $A[{\xi \atop 0}]$, $A[{\xi \atop 1}]$. Define

$$QF_C(S) := \{A : A \text{ is a quantified S-formula with constants, } Var(A)=\phi, \text{ and } A \text{ is true}\}.$$

<u>Theorem 6.1</u>. Let S be a finite set of logical relations. If one of the four conditions (c)-(f) of Theorem 2.1 holds, $QF_C(S)$ is polynomial-time decidable. Otherwise, $QF_C(S)$ is log-complete in polynomial space.

The proof relies on the result of Stockmeyer and Meyer [StM] that the problem $B_\omega \cap$ 3CNF (decide the truth of a quantified CNF formula with 3 literals per conjunct) is log-complete in polynomial space, (See also [St]).

7. APPLICATIONS

The results presented here are potentially very useful in expediting NP-completeness proofs, for the reason that they give one a much broader "target cross-section" for use in reductions. Traditionally, a researcher has had to aim his reduction at a specific NP-complete problem, such as the CNF satisfiability problem. By virtue of Theorem 2.1, the researcher's aim no longer has to be so specific. Once he has set up the framework for simulating conjunctions of clauses, he has great latitude regarding the specific content of those clauses.

To illustrate this idea, we prove that the TWO-COLORABLE PERFECT MATCHING problem, defined in Section 1, is NP-complete. With the help of Theorem 2.1, the proof is rather simple, whereas previously the author had tried without success to prove this problem NP-complete.

<u>Theorem 7.1</u>. TWO-COLORABLE PERFECT MATCHING is log-complete in NP.

<u>Comment</u>. With additional arguments, which we do not give here, it can be shown that this problem restricted to planar cubic graphs in also NP-complete.

<u>Proof</u> (sketch). Consider the graph shown in Figure 1(a), three of whose nodes are labeled with the variables x,y,z. Any coloring of this graph with the colors "0" and "1" can be interpreted as assigning truth values to the variables x,y,z. The requirement that the coloring be a solution to the 2-colorable perfect matching problem is thus interpreted as imposing a certain relation on these 3 variables. It is straightforward to verify that this relation is $[(x \vee y \vee z) \wedge (\neg x \vee \neg y \vee \neg z)]$ -- the only values the triple (x,y,z) cannot assume are (0,0,0) and (1,1,1). Call this relation R and observe that SAT({R}) is the NOT-ALL-EQUAL SATISFIABILITY problem, which, as noted in Section 2, is NP-complete as a consequence of Theorem 2.1.

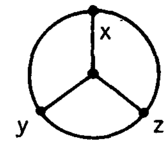

Figure 1(a)

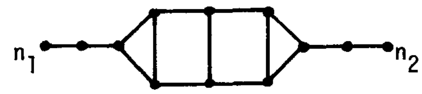

Figure 1(b)

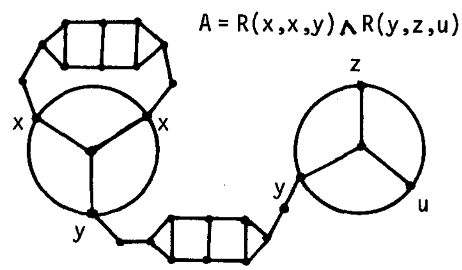

Figure 1(c)

We will reduce SAT({R}) to the 2-colorable perfect matching problem.

Let an {R}-formula A be given. Construct a graph G as follows. Let G_0 be the union of disjoint copies of Figure 1(a), one copy for each conjunct of A. On each copy label the nodes "x","y" and "z" with the names of the variables occurring in the corresponding conjunct of A. Then, for each pair n_1,n_2 of nodes that are labeled with the same variable, join n_1 to n_2 by means of the structure shown in Figure 1(b). It may be verified that this structure forces n_1 and n_2 to have the same color. Call the resulting graph G. Figure 1(c) shows a simple example of this construction. It can be seen that G has a two-colorable perfect matching iff A is satisfiable. Hence, TWO-COLORABLE PERFECT MATCHING is NP-complete. []

The point we wish to make is that, in the above proof, "almost any" graph could have been used for Figure 1(a). We suspect that if one simply randomly generated a graph having 10 or 15 nodes, within a certain range of arc probability, the result would be, with very high probability, a graph representing a relation satisfying the conditions of Theorem 2.1 for NP-completeness, which would therefore serve just as Figure 1(a) for a proof of NP-completeness.

This raises the intriguing possibility of computer-assisted NP-completeness proofs. Once the researcher has established the basic framework for simulating conjunctions of clauses, the relational complexity could be explored with the help of a computer. The computer would be instructed to randomly generate various input configurations and test whether the defined relation was non-affine, non-bijunctive, etc. The fruitfulness of such an approach remains to be proved: the enumeration of the elements of a relation on 10 or 15 variables is surely not a light computational task.

8. PROBLEMS FOR FURTHER RESEARCH

1. Generalize the Dichotomy Theorem from 2-valued variables to k-valued variables (i.e., a relation of rank n is a subset of $\{0,1,\ldots,k-1\}^n$). An analogue for k=3 would imply the (already known) NP-completeness of the graph 3-colorability problem. It would be interesting to see if any new polynomial-time decidable cases arise, which are not obvious generalizations of the case k=2.

2. Study the complexity of deciding whether a relation is affine, bijunctive, or weakly positive. From Lemmas 3.1A and 3.1B, it can be seen that it can be decided in polynomial time whether a given relation, presented as a list of its elements, is affine or bijunctive. But we do not know of any efficient algorithm for recognizing weakly positive relations.

9. CONCLUSION

We have studied the complexity of an infinite class of satisfiability problems and obtained classification theorems which include and extend various previous complexity results, thereby unifying these earlier results within a larger framework.

By way of exploring this problem, we were led to a fairly rich theory of classification of logical relations, which is independent of, although motivated by, complexity-theoretic notions. It seems likely that this theory will lead to a heightened understanding of the inherent complexity of various classes of logical relations.

Since Cook's NP-completeness proof [C], the standard CNF satisfiability problem has become a kind of canonical NP-complete problem, being probably used more widely in reductions than any other NP-complete problem. We feel that the usefulness of this problem for reductions is a property which is probably shared to some degree by other conjunctive satisfiability problems, such as those we have considered. Thus we feel that problems such as ONE-IN-THREE SATISFIABILITY and NOT-ALL-EQUAL SATISFIABILITY will likewise prove to have wide applicability in completeness proofs.

APPENDIX

Complexity-Theoretic Definitions

Inputs to all decision problems are assumed to be presented as strings of symbols from some fixed finite alphabet Σ.

A log-space bounded Turing machine is a Turing machine, having a two-way read-only input tape, a one-way write-only output tape, and a single work tape, which, on any input $w \in \Sigma^*$, never visits more than $c \log(|w|)$ frames of its work tape, for some constant c depending on the machine. The machine is assumed deterministic unless otherwise stated.

A function $f:\Sigma^* \to \Sigma^*$ is log-space computable if there is some log-space bounded Turing machine which on input $w \in \Sigma^*$ outputs f(w) and halts.

Let $A,B \subseteq \Sigma^*$. Then $A \leq_{\log} B$ ("A is log-space reducible to B") iff there is a log-space computable function f such that for all $w \in \Sigma^*$, $w \in A$ iff $f(w) \in B$. This is a transitive relation. A is log-equivalent to B if $A \leq_{\log} B$ and $B \leq_{\log} A$. A set B is log-complete in a class C if $B \in C$ and for all $A \in C$, $A \leq_{\log} B$.

Log space (resp. nondeterministic log space) is the class of all sets $A \subseteq \Sigma^*$ such that there is some deterministic (resp. nondeterministic) log-space bounded Turing machine that recognizes A.

Polynomial time (denoted P), nondeterministic polynomial time (denoted NP) and polynomial space (denoted PSPACE) are the classes of sets that are recognized by deterministic and nondeterministic polynomial-time bounded and polynomial-tape bounded Turing machines respectively. (These are standard one-tape Turing machines, and the bounds are expressed as a function of the length of the input string.) In this paper, "NP-complete" means "log-complete in NP."

More detailed explanations can be found in [K],[StM],[St].

ACKNOWLEDGEMENT

I am indebted to my thesis advisor, Professor Richard M. Karp, for discussing parts of this work with me.

REFERENCES

[BBFMP] M. Bauer, D. Brand, M. Fischer, A. Meyer and M. Paterson, A note on disjunctive form tautologies, SIGACT News April 1973, pp. 17-20.

[C] S.A. Cook, The complexity of theorem-proving procedures, Third Annual ACM Symposium on Theory of Computing, 1971, pp. 151-158.

[DP] M. Davis and H. Putnam, A computing procedure for quantification theory, JACM 7 (1960) pp. 201-215.

[J] N.D. Jones, Space-bounded reducibility among combinatorial problems, J. Comp. Sys. Sci. 11 (1975) pp. 68-85.

[JL] N.D. Jones and W.T. Laaser, Complete problems for deterministic polynomial time, Sixth Ann. ACM Symp. on Theory of Comp., 1974 pp. 40-46.

[K] R.M. Karp, Reducibility among combinatorial problems, in Complexity of Computer Computations, R.W. Miller and J.W. Thatcher eds., Plenum Press, New York, 1972, pp. 85-104.

[P] E.L. Post, The two-valued iterative systems of mathematical logic, Annals of Math. Studies, vol. 5, Princeton, 1941.

[Sav] W. Savitch, Relationships between nondeterministic and deterministic tape complexities, J. Comp. Sys. Sci. 4 (1970), pp. 177-192.

[Sch] T. J. Schaefer, Complexity of decision problems based on finite two-person perfect-information games, 8th Ann. ACM Symp. on Theory of Comp., 1976, pp. 41-49.

[St] L.J. Stockmeyer, The polynomial-time hierarchy, Theor. Comp. Sci. 3 (1976), pp. 1-22.

[StM] L.J. Stockmeyer and A.R. Meyer, Word problems requiring exponential time: preliminary report, 5th Annual ACM Symp. on Theory of Computing, 1973, pp. 1-9.