Wednesday
10th May

Dear Tony,

Attached is your draft on satisfiability. The most general class of expression that I know is easily evaluated is that generated by the construction [Q, S] where Q is a ~~sequence~~ set of expressions (each of which can be evaluated efficiently) and S is a set of natural numbers. Expression [Q, S] is true precisely when the number of elements of Q that are true is a member of S. Expression [Q, S] in turn can be quickly evaluated provided that no two elements of Q have any variable in common.

$$(E \triangleleft F \triangleright G) = \left[[\langle E, F\rangle, 2]\ [\langle G, \bar{F}\rangle, \{2\}], \{1,2\}\right]$$

Special cases

$$\mathit{false} \equiv [\langle\rangle, \{\}]$$

$$\mathrm{maj}(\langle E, F, G\rangle)$$

$$\mathit{true} \equiv [\langle\rangle, \{0\}]$$

$$= [E, F, G, \{2,3\}]$$

$$\neg E \equiv [\langle E\rangle, \{0\}]$$

$$E \wedge F \equiv [\langle E, F\rangle, \{2\}]$$

$$E \vee F \equiv [\langle E, F\rangle, \{1,2\}]$$

$$E \Leftrightarrow F \equiv [\langle E, F\rangle, \{0,2\}]$$

The elements of Q can be variables, or recursively expressions of the form [Q′, S′], or else expressions that are monotone/antimonotone in every variable.

Bryan Todd

Efficient evaluation of satisfiability

The conditional is a ternary propositional function defined

$$X \triangleleft B \triangleright Y = X \wedge B \vee (\neg B) \wedge Y.$$

Its value can be computed (in linear time or better) from the laws

$$X \triangleleft T \triangleright Y = X$$

$$X \triangleleft F \triangleright Y = Y$$

where T and F are the truth values.

Every truth function can be expressed by nested conditionals, because.

$$\neg X = F \triangleleft X \triangleright T$$

and $X \wedge Y = Y \triangleleft X \triangleright F$

Without loss of generality, the condition (the middle operand) can be restricted to a single propositional variable that does not appear in either of the other operands. This is because any truthfunction $G(b)$ can be expressed

$$G(T) \triangleleft b \triangleright G(F).$$

The test is in the worst case linear in the length of the formula; which only shows that any formula (expressed as conditionals) can be exponentially longer than its more conventional truthfunctional representation. However, experience with Binary Decision Diagrams suggests that there are useful applications which do not suffer exponential expansion.

Let us confine attention to (slightly more general) formulae, expressed as conditionals in which the condition shares no variables with either of the other operands. We wish to determine quickly whether such formulae are tautologies (+), contradictions (−) or contingencies (?). This may be done by simple computation, using a three-valued truth-table derived from the laws:

$T = +$, $F = -$,

propositional variables are ?

$$X \triangleleft + \triangleright Y = X$$

$$X \triangleleft - \triangleright Y = Y$$

$$X \triangleleft ? \triangleright Y = X \quad \text{if } X = Y$$

$$= ? \quad \text{otherwise.}$$

of the other operands. In gen
of the conditional is the negation of the other:

$$(X \triangleleft B \triangleright \neg X) = (B \equiv X)$$

$$(? \equiv X) = ?$$

The calculation is significantly more efficient than the general case when B is ?

Whenever the condition is a tautology or a contradiction, the threevalued calculation omits (shortcircuits) evaluation of one of the other operands. In general, this cannot be done when the condition is a contingency; but certain special cases may be worth extra attention. For example, one of the limbs of the conditional may be the negation of the other:

$$(X \triangleleft B \triangleright \neg X) = (B \equiv X)$$
$$(\neg X \triangleleft B \triangleright X) = (B \not\equiv X)$$
$$(? \equiv X) = (? \not\equiv X) = ?$$

In this case, evaluation is in constant time, rather than requiring evaluation of two operands.

Here is another attempt at optimisation.

Suppose we are interested only in two cases, say contradictions($-$) and non-contradictions ($?+$)

Suppose also that for a conditional

$$X \triangleleft B \triangleright Y$$

evaluation of X is expensive and it is known that

$$\vdash X \Rightarrow Y.$$

ie, that the combination $(X = ?+ , Y = -)$ is impossible. Then we can use the evaluation rules

$$X \triangleleft - \triangleright Y = Y$$

$$X \triangleleft ?+ \triangleright Y = \begin{cases} - & \text{if } Y = - \\ X & \text{otherwise} \end{cases}$$

This will sometimes avoid calculation of X, with good savings on the average.

The optimisation is available for any truthfunction $G(b)$ which is monotone (or antimonotone) in b.

But is it really worth it?

Turning to the general case,

Let $G(x, y, \ldots, z)$ be an arbitrary truthfunction; ~~let us~~ and suppose we can accept restriction of its use to arguments which are pairwise disjoint (no variable appears in more than one of them).

G can always be expanded to conditional form by repeated application of the rule:

$$G(T, y, \ldots, z) \lhd x \rhd G(F, y, \ldots, z),$$

thus "defining" the appropriate procedure for three-valued evaluation. However the evaluation may be done without actually expanding the formula G to its conditional form, thus avoiding the possible explosion in space, even when the tautology test is exponential in time. Of course, the technique depends on prior knowledge of the function G, which is sufficiently expressive even when its arguments are required to be disjoint.

Consider, for example the parity function, defined by

$$P_0(\ ) = T$$

$$P_0(x, y, \ldots, z) = \neg P_0(y, \ldots z) \lhd x \rhd P_0(y, \ldots, z)$$

Notoriously, this leads to worst case expansion for BDDs, (because y appears twice, ..., and z appears, say, 2^{26} times). But the calculation of the tautology test can be done in linear time, by directly executing the recurrence displayed above. Indeed, using the optimisation

$$x \not\equiv P_0!(y, \ldots, z)$$

the execution can terminate on first encounter of a ?

In greater generality, let P_s be a counting function which tests whether the number of it true operands is in the set s.

$$P_{\{s\}}(\langle x\rangle) = ? \quad \text{if } 0 \in s \vee 1 \in s$$
$$= F \quad \text{otherwise}$$

$$P_{\{\}}(\langle x, \dots \rangle) = F$$
$$P_s(\langle\rangle) = T \triangleleft 0 \in s \triangleright F$$
$$P_s(x, y, \dots) = P_{s-1}(y, \dots) \triangleleft x \triangleright P_s(y, \dots)$$

where $s-1 = \{k \mid k+1 \in s\}$

Interesting special cases are:

$\wedge$	$s = \{\text{length of sequence}\}$
$\vee$	$s = \{k \mid k \neq 0\}$
parity	$s =$ even numbers.
$\equiv(x, y)$	$s = \{0, 2\}$
$\not\equiv(x, y)$	$s = \{1\}$
$P_{\{k\}}$	$s =$ exactly k true
$P_{\leq k}$	not more than k

G, H are ~~a~~ lists of hypotheses, ~~where~~ where a hypothesis has the form

$$x{:}T \qquad y{:}F \qquad z{:}?$$

(x is true, y is false, z is a contingency)

s is a set of natural numbers

$s+t = \{m+n \mid m \in s \,\&\, n \in t\}$.

~~H,~~ P, Q are list of propositions

$H \mathrel{?}\vdash_s P$ means ~~the~~

under hypotheses H, it is ~~pos~~ not a contradiction that the number of true propositions in list P is a member of the sets s

Axioms. $\cancel{H?}\ x{:}T \mathrel{?}\vdash_{s} x$ if $1 \in s$.

$x{:}F \mathrel{?}\vdash_s x$ if $0 \in s$

$x{:}? \mathrel{?}\vdash_s x$ if $0 \in s$ or $1 \in$

$$\frac{G \mathrel{?}\vdash_s P \qquad H \mathrel{?}\vdash_t Q}{G, H \mathrel{?}\vdash_{s+t} P, Q}$$

$$\frac{H \vdash_s P}{H \vdash_{t} (\vdash_s P)} \quad \text{if } 1 \in t$$

$P_{\{\}}(s) = \{F\}$ $P_s(\langle\rangle) = \underset{T}{\{0\in S\}}$ if $0 \in S$; F otherwise

$\cancel{P_{\forall}(s) = \{T\}}$

$P_s(\langle ?\rangle ^\wedge t) = P_{s\cap(s-1)}(t)$ for tautology

$\cup P_{(s-1)\cup s}(t)$ $s\cup(s-1)$ for contingency.

$P_s(\langle T\rangle ^\wedge t) = P_{s+1}(t)$

$P_s(\langle F\rangle ^\wedge t) = P_s(t)$

$P_{\{0\}}(\langle ?\rangle ^\wedge y) =$

can be true

$P(O,$ H

$P.O.x =$

$($

$$\frac{H \vdash_s P \quad G \vdash_t Q}{H, G \vdash_{s+t} P, Q}$$

$H \Vdash P$ P can be true

$H \nVdash P$ P can be false

$x : T \Vdash_{s} x$ $0 \in s$

$x : T \Vdash_{\{1\}\cup s} x$

$$\frac{H \Vdash P}{H \nVdash \neg P}$$

$\cancel{H \vdash P_s}$

$$\frac{\dfrac{H \vdash_{s-1} P}{x : T, H \vdash_s x, P}}{\cancel{H \vdash_{s-1} P}}$$

$x : F \Vdash_{\{0\}\cup s} x$ ✓

$x : ? \vdash_{\cancel{\{0,1\}\cup s}} x$

$0 \in s \vee 1 \in s$

$$\frac{x : F, H \vdash_s x, P}{1}$$

$H \vdash_s P$

$H \vdash_{\{1\}} f_s(P)$

$$\frac{G \Vdash X}{G \Vdash \neg X} \qquad \frac{G \Vdash X \qquad H \ominus Y}{G, G \Vdash X \equiv Y \qquad H, G \Vdash X \not\equiv Y}$$

$\neg X$

$$\frac{H \Vdash X \equiv Y}{G \vdash Y \equiv X}$$ omit?

$\nvdash \neg X \qquad \vdash X \vee Y \qquad \vdash$

$$\frac{G \vdash X \qquad H \ominus Y}{G, H \vdash X \vee Y}, \qquad G \nvdash \neg X$$

$x : Y \ominus x : Y$

$$\frac{G \ominus X \qquad H \ominus Y}{G, H \ominus X \oplus Y}$$

$\cancel{G, H} \vDash X$

We explore a fragment of propositional logic in which tautologies and contradictions can be tested in linear time. The valid formulae are defined inductively.

1. true, false, and propositional letters are valid.
2. If P is valid so is $\neg P$
3. If P is valid and the letter b does not occur in P, then $(b \equiv P)$ is valid
4. If P and Q are valid, and do not contain the letter c, and further, $\vdash Q \Rightarrow P$ ~~$\vdash P \Rightarrow Q$~~ is a tautology, then $(c \wedge P \vee Q)$ and $(\neg c \wedge P \vee Q)$ are valid.
5. If P and Q share no letters, $(P \wedge Q)$ and $(P \vee Q)$ are valid.

[actually, in rule 5, P and Q can share a variable c, introduced to both of them under the same option of clause 4] and after all applications of 3.

Valid formulae can be tested for tautology by the following properties

true is a tautology

propositional letters and false are not

$(b \equiv P)$ is not

$(c \wedge P \vee Q)$ and $(\neg c \wedge P \vee Q)$ are tautologies

iff Q is   because $Q \Rightarrow P$

$(P \wedge Q)$, ~~R~~ is a tautology

iff both P and Q are

$(P \vee Q)$ is a tautology

iff P or Q is.

$\neg P$ is a tautology

iff P is a contradiction.

The properties of a contradiction are dual to these