Arch. math. Logik 23 (1983), 27–36

A LOWER BOUND FOR THE COMPLEXITY OF CRAIG'S INTERPOLANTS IN SENTENTIAL LOGIC*

Daniele Mundici

Abstract

For any sentence α (in sentential logic) let d_α be the delay complexity of the boolean function f_α represented by α. We prove that for infinitely many d (and starting with some $d<620$) there exist valid implications $\alpha \to \beta$ with $d_\alpha, d_\beta \leqq d$ such that any Craig's interpolant χ has its delay complexity d_χ greater than $d+(1/3)\cdot\log(d/2)$. This is the first (non-trivial) known lower bound on the complexity of Craig's interpolants in sentential logic, whose general study may well have an impact on the central problems of computation theory.

0. Introduction

Craig's interpolation theorem yields, for any valid implication $\alpha \to \beta$, an interpolant χ, i.e. a sentence such that both $\alpha \to \chi$ and $\chi \to \beta$ are valid, while χ only uses the primitive notions, viz. the nonlogical symbols, (or the variables, if we are in sentential logic) used by α and β simultaneously (see [Sm]). The importance of Craig's interpolation far transcends sentential or first-order logic, due to syntactic and semantic merits (see, e.g., [Mu 1, 2, 6, 7, 10, 11]). In [Mu 3] the present author studied the complexity of interpolation, by showing that in first-order logic the length $\|\chi\|$ (i.e. the number of symbols in χ) of the shortest interpolant for $\alpha \to \beta$ grows as fast as some Π_1-function of $\|\alpha\| + \|\beta\|$. In the same paper it is proved that for each $m = 1, 2, \ldots$, we can write down a valid first-order implication $\alpha \to \beta$ with $\|\alpha\|, \|\beta\| < 1100 + 15m$ such that, whenever χ is an interpolant for $\alpha \to \beta$ we have that

$$\|\chi\| > 2^{\cdot^{\cdot^{\cdot^{2}}}} \Big\updownarrow\, 2m+1 \text{ twos}.$$

For the practical aspects of this result (when $m=3$) see [Mu 4, 5]. As a corollary of [Fr, Theorem 1, p. 22] one has that in first-order logic there is no recursive upper bound for $\|\chi\|$ in terms of $\|\alpha\| + \|\beta\|$. This is an example of an asymptotic bound, only dealing with suitably long sentences.

---

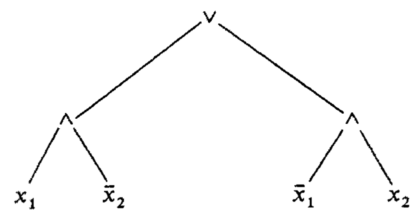
* Eingegangen am 25.3.1981, Revisionen am 2.11.1981.

As for sentential logic, in [Mu 3] it is proved that an interpolant χ can always be found satisfying
$$\|\chi\| \leqq -11+6\cdot 2^{(\|\alpha\|+\|\beta\|+6)/8}.$$

In this paper we pursue the study of the complexity of Craig's interpolation in sentential logic, in search of lower bounds for the size of interpolants. In Sections 1 and 2 we shall limit attention to the *delay complexity* d_α of sentence α, i.e. the delay complexity of the boolean function f_α represented by α, see [Sa]. Intuitively, the delay complexity of f_α measures the time required for inputs to propagate to the output, in the fastest logic circuit computing f_α. Our Theorem 2.5 states that for infinitely many d (and starting with some $d<620$), there exist valid implications $\alpha\to\beta$ with $d_\alpha, d_\beta \leqq d$, such that any interpolant χ has a delay complexity d_χ satisfying the following inequality:
$$d_\chi > d+(1/3)\cdot\log(d/2).$$
This is a first nontrivial lower bound on the complexity of Craig's interpolation in sentential logic: already it is remarkable the fact that in sentential logic an interpolant χ for $\alpha\to\beta$ may happen to be more complex than both α and β: intuitively, the time needed by the fastest logic circuit to decide if a sequence of truth values satisfies χ is greater than the time needed by the fastest logic circuit to decide if a sequence satisfies α, or β.
On the other hand, one must expect stronger results and lower bounds, as soon as computation theorists recognize the central importance of Craig's interpolation theorem for the deepest problems of computation theory, as briefly discussed in the final section of this paper. For further information see [Mu 3, 8, 9].

1. Preliminaries

Throughout this paper $\omega=\{0,1,\ldots\}$, logarithms are to the base two, $\lceil\log\rceil x$ (resp., $\lfloor\log\rfloor x$) is the smallest $y\in\omega$ such that $y\geqq\log x$ (resp. the greatest $y\in\omega$ such that $y\leqq\log x$). A map $f:\{0,1\}^a\to\{0,1\}$ is called an a-ary boolean function ($a=1,2,\ldots$). Important boolean functions are the *conjunction* $\wedge$ given by $x\wedge y=\min(x,y)$, the *disjunction* $\vee$ given by $x\vee y=\max(x,y)$, and the *negation* $^-$, given by $\bar{x}=1-x$. We let $\Omega_0=\{\wedge,\vee,^-\}$. We refer to [Sa, 2.2] for the notion of a *chain* φ *for the boolean function* f *over basis* Ω. Throughout this paper we shall only consider basis Ω_0; here is an example of a chain over Ω_0 for addition mod 2:



The *delay complexity* d_f of function f, [Sa, 2.3.2], is the *depth* (i.e. the length of the longest path) of the smallest depth chain for f. We shall only consider chains having *fan out* 1 [Sa, 2.3.1]. This does not affect the value of d_f. Also, given a chain φ for f one can directly construct another chain φ^* for f in which the negations are only applied to the variables $x_1, \ldots, x_a$, with φ^* not deeper than φ, (apply the De Morgan rules). Thus, without loss of generality, we shall limit our attention to such chains as φ^* only. We shall naturally incorporate the negation into the *data set* which is thus given by $X = \{x_1, \ldots, x_a, \bar{x}_1, \ldots, \bar{x}_a\}$; on the other hand, there is no need to include 0 or 1 into the data set, if $a \geqq 1$. Notice that by incorporating the negation in X the depth of φ^* and, in general, the delay complexity of f, are both decreased by one, with respect to the usual definition. In definitive, a chain for f is a finite binary tree having either $\wedge$ or $\vee$ on each of its nodes, and having two elements from X attached at the (bottom) end of each branch. A chain φ for f is *complete* iff all its branches have equal length: the chain in the above figure is complete, and each branch has length 2, by our stipulations about negations. If φ is complete and d is its depth then $b = 2^d$ is the number of its branches (= the number of occurrences of variables or negated variables from X at the bottom of φ), and $b-1$ is the number of its (binary) nodes (i.e., the number of occurrences of $\wedge$ or $\vee$).

In sentential logic, upon identifying 1 with "true" and 0 with "false", one defines as usual the notion of a sentence $\alpha(x_1, \ldots, x_n)$ being satisfied by a sequence $(c_1, \ldots, c_n) \in \{0, 1\}^n$. Thus α canonically determines a boolean function $f_\alpha : \{0, 1\}^n \to \{0, 1\}$ given by

$$f_\alpha(c_1, \ldots, c_n) = 1 \quad \text{iff} \quad (c_1, \ldots, c_n) \text{ satisfies } \alpha.$$

One then naturally defines the *delay complexity* d_α of sentence α as the delay complexity of f_α. Our main result below shows that there are infinitely many valid implications $\alpha \to \beta$ such that if χ is any interpolant (as given by Craig's interpolation theorem) then the delay complexity d_χ of χ is greater than the delay complexity of both α and β, in fact greater than $\max(d_\alpha, d_\beta) + (1/3) \cdot \log(d_\alpha/2)$.

The importance of lower bounds on Craig's interpolation in sentential logic might justify a deeper study of the other structural properties of interpolants (e.g., formula size), as further discussed in [Mu 3, 8, 9] and in Section 3 below.

The author wishes to express his gratitude to the referee.

2. Some Facts About Delay Complexity

2.1. Proposition. *Let $a \geqq 1$ be an arbitrary natural number. Then there exists an a-ary boolean function*

$$f : \{0, 1\}^a \to \{0, 1\}$$

with delay complexity $d_f > a - \log\log 4a$. In fact, such f are the majority of the a-ary boolean functions.

Proof. Let F be the set of all a-ary boolean functions g with delay complexity $d_g \leqq a - \log\log 4a$; let γ be a chain for g with depth d_g; by suitably adding branches and increasing the depth of γ we can construct a complete chain $\tilde{\gamma}$ with depth $\tilde{d} = a - \ulcorner\log\urcorner \log 4a$ and with $\tilde{\gamma}$ still computing function g. Then the cardinality $|F|$ of F is smaller than the cardinality $|\Phi|$ of the set Φ of complete chains of depth $\tilde{d}$; but any such chain has $b=2^{\tilde{d}}$ branches and $b-1$ nodes (without counting the variables); at the bottom end of each branch we can find either of $x_1, \ldots, x_a, \bar{x}_1, \ldots, \bar{x}_a$ and in each node there is either $\wedge$ or $\vee$. Also observe that $|F|<|\Phi|$, since there are at least two different chains in Φ representing the same function (as one can see by permuting two different branches in some circuit). Now we have

$$
\begin{aligned}
|F| &< \Phi \\
&= (2a)^b \cdot 2^{b-1}, \text{ by definition of } \Phi \\
&= (4a)^{2^{\tilde{d}}}/2 \\
&= 2^{-1+2^{\tilde{d}}\cdot\log 4a} \\
&= 2^{-1+(\log 4a)\cdot 2^{a-\lceil\log\rceil\log 4a}} \\
&\leqq 2^{-1+(\log 4a)\cdot 2^{a-\log\log 4a}} \\
&= 2^{-1+2^a} \\
&= 2^{2^a}/2 .
\end{aligned}
$$

This shows that $|F|$ is strictly smaller than half the number of possible a-ary boolean functions (i.e. 2^{2^a}), and completes the proof of the proposition. $\square$

2.2. We let now f be an arbitrary but fixed a-ary boolean function with delay complexity

$$d_f > a - \log\log 4a$$

and we let φ be a complete chain for f with depth d_f. We are going to prune φ from below, so that we can reduce its depth by an amount d ($d<d_f$); see the following figure, where each $\diamond$ represents either $\wedge$ or $\vee$, and each x represents either of $x_1, \ldots, x_a, \bar{x}_1, \ldots, \bar{x}_a$ (depending on its position):

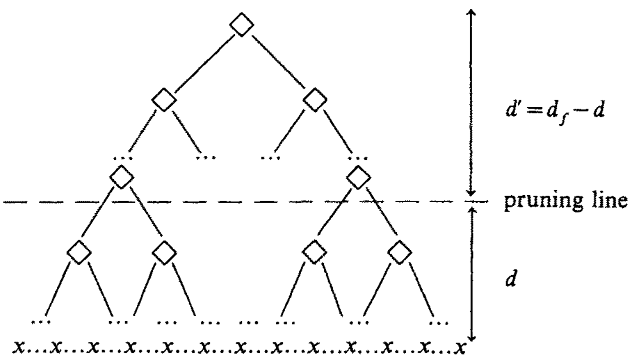


Fig. 1

Below the pruning line one can find $2^{d'}=2^{d_f-d}$ subchains of φ, but it may well occur that two subchains are equal: as a matter of fact, every such subchain has a depth d and is complete, hence it has 2^d branches, at whose bottom there is attached either of $x_1, \ldots, x_a, \bar{x}_1, \ldots, \bar{x}_a$. By arguing as in the proof of Proposition 2.1 we see that the maximum number n of *different* subchains of depth d below the pruning line in Fig. 1 satisfies the following condition:

$$n<(2a)^{2^d}\cdot 2^{-1+2^d}=(4a)^{2^d}/2. \tag{1}$$

We display these n subchains as follows:

$$\sigma_1, \sigma_2, \ldots, \sigma_n.$$

We prepare a set $Y=\{y_1, \ldots, y_n\}$ of *new* variables (i.e., other than $x_1, \ldots, x_a$); we let φ' be the chain obtained from φ by replacing every σ_j by y_j ($j=1, \ldots, n$); φ' is complete, has its depth $d'=d_f-d$ and each branch of φ' has a variable of Y at its bottom.

Finally, we prepare a (definitional) new chain δ, to take care of the intended equivalence between y_j and σ_j. To this purpose, let $\bar{\sigma}_j$ be the *dual* of σ_j, i.e. the tree of depth d obtained from σ_j by writing $\wedge$ instead of $\vee$, and vice versa, and x_i instead of $\bar{x}_i$, and vice versa. $\bar{\sigma}_j$ is still complete, has depth d and clearly expresses the negation of σ_j. Our definitional tree is defined in two stages: in the first we express the fact that $y_j \leftrightarrow \sigma_j$; in the second we collect together our definitions. More precisely we set:

Stage One. For each σ_j construct chain δ_j of depth $d+2$ as follows:

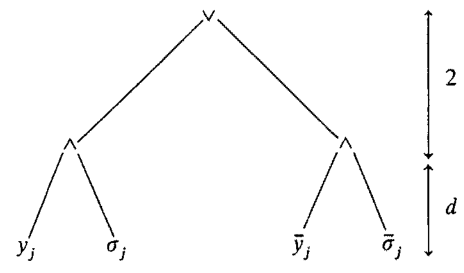


Fig. 2

Stage Two. Let δ be the conjunction of the δ_j constructed in stage one, as follows:

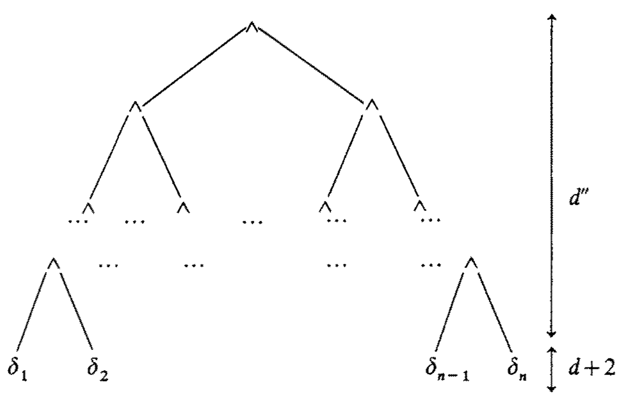


Fig. 3

In the light of inequality (1) we can say that

$$d'' < 1 + \log n < 1 + \log((4a)^{2^d}/2) = 2^d \log 4a, \tag{2}$$

so that the depth d_δ of δ satisfies the following inequality

$$d_\delta = 2 + d + d'' < 2 + d + 2^d \log 4a. \tag{3}$$

Notice that δ is not complete. Let ψ be the conjunction of φ' and δ. The depth d_ψ of ψ satisfies the following conditions:

$$\begin{aligned} d_\psi &= 1 + \max(d', d_\delta) \\ &= 1 + \max(d_f - d, d_\delta) \\ &< 1 + \max(a - \log\log 4a - d, 2 + d + 2^d \log 4a) \end{aligned}$$

by our assumption about d_f and by (3). Therefore we can write

$$d_\psi \leqq \max(a - d - \log\log 4a, 2 + d + 2^d \log 4a). \tag{4}$$

From now on we shall specialize on values of d which nearly maximize the difference between d_f and d_ψ, and are easier to handle in computations. We state the following:

2.3. Theorem. *Let $a \geqq 16$. Assume that d satisfies the following condition:*

$$d = \lfloor \log a \rfloor - \lceil \log \rceil \log 4a - 1.$$

Let f and ψ, d_f and d_ψ be as above. Then we have

$$d_\psi < d_f - \log(a/(8 \log 4a)).$$

Proof. First of all, we make the following

Claim. $a - d - \log\log 4a > 2 + d + 2^d \log 4a$.

Proof of the Claim. Notice that for $a \geqq 16$ one has

$$a \geqq 4a^{1/2} \geqq 4 \log a$$

i.e.,

$$a/2 \geqq 2 \log a,$$

hence

$$a/2 > 2 \log a - \log\log 4a$$

and

$$a/2 - 2 - \log\log 4a > 2 \log a - 2 \log\log 4a - 2,$$

i.e.

$$\begin{aligned}
a-2-\log\log 4a &> a/2+2(\log a-\log\log 4a-1)\\
&=2\cdot(\log a-\log\log 4a-1)+(\log 4a)\cdot 2^{\log a-\log\log 4a-1}\\
&\geqq 2\cdot(\lfloor\log\rfloor a-\lceil\log\rceil \log 4a-1)+(\log 4a)\cdot 2^{\lfloor\log\rfloor a-1-\lceil\log\rceil\log 4a}\\
&=2d+(\log 4a)\cdot 2^{d}, \text{ which establishes our claim.}
\end{aligned}$$

We now end the proof of the theorem as follows:

$$\begin{aligned}
d_\psi &\leqq \max(a-\log\log 4a-d, 2+d+2^{d}\log 4a), \text{ by (4),}\\
&=a-d-\log\log 4a, \text{ by claim 1}\\
&<d_f-d, \text{ by definition of } d_f\\
&=d_f-\lfloor\log\rfloor a+\lceil\log\rceil \log 4a+1, \text{ by definition of } d,\\
&<d_f-(\log a-1)+(\log\log 4a+1)+1\\
&=d_f+3-\log a+\log\log 4a
\end{aligned}$$

which completes the proof of our theorem. □

2.4. Corollary. *Adopt the notation of Theorem 2.3. Assume further* $a\geqq 609$ *and* $d=\lfloor\log\rfloor a-\lceil\log\rceil \log 4a-1$. *Then we have*

$$d_\psi<d_f-(1/3)\cdot\log(d_f/2), \textit{ hence, a fortiori,}$$
$$d_f>d_\psi+(1/3)\cdot\log(d_\psi/2).$$

Proof. In the light of Theorem 2.3, to prove the first inequality it is sufficient to show that

$$(1/3)\cdot\log(d_f/2)\leqq\log(a/(8\log 4a)). \tag{5}$$

Now it is well known that the following holds:

$$d_f<a+1+\log a \tag{6}$$

(just think of a chain for f in disjunctive normal form, see [Sa] for details, if necessary). Therefore, to prove (5) it suffices to prove that for each $a\geqq 609$ we have

$$(1/3)\cdot\log((a+1+\log a)/2)\leqq\log(a/(8\log 4a)). \tag{7}$$

This can be verified by a direct inspection.
Having thus proved the first inequality, the second immediately follows, by noting that $d_\psi<d_f$. □

We now apply the above results to sentential logic.

2.5. Theorem. (i). *For infinitely many $d \in \omega$ there exists in sentential logic a valid implication $\alpha \to \beta$ with both α and β having their delay complexity $\leqq d$, such that any interpolant χ has its delay complexity $d_\chi > d + (1/3) \cdot \log(d/2)$.*
(ii). *The phenomenon described in* (i) *above already occurs for some $d < 620$.*

Proof. Let $a \geqq 609$; let f be an a-ary boolean function with delay complexity $d_f > a - \log\log 4a$, as in Proposition 2.1; let φ be a complete chain for f with depth d_f; let ψ, n and $Y = \{y_1, \ldots, y_n\}$ be as in the discussion following the proof of Proposition 2.1. Let ψ' be obtained from ψ by writing y'_j instead of y_j, where $Y' = \{y'_1, \ldots, y'_n\}$ is a set of variables other than $x_1, \ldots, x_a$, and with $Y' \cap Y = \emptyset$.
We let α be a sentence with variables $\{x_1, \ldots, x_a, y'_1, \ldots, y'_n\}$ such that the boolean function represented by ψ' equals the boolean function f_α represented by α (see Section 1).
To construct β, let $g = 1 - f$. One can see that $d_g = d_f$ (by applying the De Morgan rules). Let γ be a complete chain for g with depth d_g; let ψ_g be obtained from γ via some (definitional) addition of variables $y_1, \ldots, y_n$ exactly as ψ is obtained from φ in 2.2. Let ψ'' be obtained from ψ_g by writing y''_j instead of y_j, where $Y'' = \{y''_1, \ldots, y''_n\}$ is a set of variables other than $x_1, \ldots, x_a$ and with $Y'' \cap Y = Y'' \cap Y' = \emptyset$. Let α'' be a sentence with variables $\{x_1, \ldots, x_a, y''_1, \ldots, y''_n\}$ such that the boolean function represented by ψ'' equals the boolean function represented by α''.
Let finally χ_0 be a sentence with variables $x_1, \ldots, x_a$ representing function f.

Claim. *$\alpha \wedge \alpha''$ is inconsistent, viz., there is no sequence of truth values $(c_1, \ldots, c_a, c'_1, \ldots, c'_n, c''_1, \ldots, c''_n)$ satisfying $\alpha \wedge \alpha''$ (recall that truth values are coded by* 0 *and* 1*).*

Proof of Claim. Deny. Then $(c_1, \ldots, c_a, c'_1, \ldots, c'_n)$ satisfies α, hence, by definition of α, $(c_1, \ldots, c_a)$ satisfies χ_0. On the other hand $(c_1, \ldots, c_a, c''_1, \ldots, c''_n)$ satisfies α'', hence $(c_1, \ldots, c_a)$ also would satisfy $\neg\chi_0$, by definition of α'', a contradiction.
After the proof of our claim, we let

$$\beta \underset{\text{def}}{=} \neg\alpha''.$$

Then $\alpha \to \beta$ is valid, so let χ be any interpolant, as given by Craig's interpolation theorem in sentential logic. Sentence χ only uses the common variables of α and β, i.e. $x_1, \ldots, x_a$, and both $\alpha \to \chi$ and $\chi \to \beta$ are valid.
Now, every sequence $(c_1, \ldots, c_a)$ satisfying χ_0 can be expanded to a sequence $(c_1, \ldots, c_a, c'_1, \ldots, c'_n)$ satisfying α, hence $\chi_0 \to \chi$ is valid (as $\alpha \to \chi$).
Conversely, every sequence satisfying $\neg\chi_0$ can be expanded to a sequence satisfying α'', hence $(\neg\chi_0) \to (\neg\chi)$ is valid (as $\chi \to \beta$, i.e. $\alpha'' \to \neg\chi$).
Therefore we must have $\chi_0 \leftrightarrow \chi$, and the delay complexity d_χ of χ must be equal to d_f. But the latter is strictly larger than $d_\psi + (1/3) \cdot \log(d_\psi/2)$, by the last inequality of Corollary 2.4. In addition, $d_\alpha = d_\beta \leqq d_\psi$. Now let $d = d_\psi$, in order to obtain the first part of the theorem.

To prove (ii), adopting the notation used in the proof of (i), we have

$$\begin{aligned} d &= d_\psi \\ &< d_f - (1/3)\cdot\log(d_f/2), \text{ by Corollary 2.4,} \\ &< d_f \\ &< a + 1 + \log a, \end{aligned}$$

see remark after (6) in Corollary 2.4. Now the proof of (*i*) holds for arbitrary $a \geqq 609$, thus in particular for $a = 609$. This yields the desired conclusion, and completes the proof of the theorem. □

3. Sentential Interpolation and Computation Theory

The above lower bound for sentential interpolants, together with the upper bound given in [Mu 3] and described in the introduction of this paper, do not settle the problem whether sentential interpolants *grow polynomially*, i.e., there exists a polynomial q such that whenever $\alpha \rightarrow \beta$ is valid in sentential logic, one can find an interpolant χ with $\|\chi\| \leqq q(\|\alpha\| + \|\beta\|)$. This problem is connected with the important problem of relating Turing and circuit complexity. See, e.g. [Sc] for background. For $g: \{0,1\}^\infty \rightarrow \{0,1\}$, let g_n be the restriction of g to $\{0,1\}^n$, where $\{0,1\}^\infty = \{0,1\} \cup \{0,1\}^2 \cup \{0,1\}^3 \cup \dots$; thus, g_n is the restriction of g to inputs of length equal to n. Then we have the following.

3.1. Theorem. *Assume sentential interpolants grow polynomially. Then for every function g having a Turing machine M which computes each g_n in time polynomial in n, there also exists a sequence of circuits $G_1, G_2, \dots, G_n, \dots$ with G_n computing g_n such that, for some $c > 0$,*

$$\operatorname{depth}(G_n) \leqq c \cdot \log_2 n \qquad (\textit{for all } n > 1).$$

Proof. See [Mu 3, 8]. See [Sa, 2.2] for details about circuits. □

As a consequence of the above theorem, consider the Transitive Closure (TC) of an $n \times n$ boolean matrix: TC is computable in polynomial Turing time; on the other hand, the circuit depth of TC grows faster than $(\log_2 n)^2$ in all *existing* circuits – as of *today*. If this fact were to hold for *any possible* circuit, then TC would have superpolynomial formula size, hence, by 3.1, sentential interpolants, too, would grow superpolynomially. This, in turn, would have an impact on the problem of the existence of "natural" deduction systems for propositional logic (see also [Co]), as follows: Recall that from any Gentzen-style proof P of $\alpha \rightarrow \beta$ one can quickly (in P) extract an interpolant χ (see [Sm] for details). Now, if $P = NP$ is true, let M be a Turing machine deciding the validity of sentential implications $\alpha \rightarrow \beta$ in polynomial time; one might not be content with a mere "yes" or "no" concerning the validity of $\alpha \to \beta$; one might require that M also gives out some sort of "souvenir" of its computation of $\alpha \to \beta$. For example, one might naturally require that (i) if $\alpha \to \beta$ is not valid, then M exhibits a counterexample, i.e. a sequence $(c_1, \ldots, c_m) \in \{0, 1\}^m$ satisfying $\alpha \wedge \neg \beta$, and (ii) if $\alpha \to \beta$ is valid, then M exhibits an interpolant χ. Let us agree to say that M then provides a *Craig deduction system* for the propositional calculus.

Then, using Theorem 3.1 we see that if TC turns out not to have circuit depth proportional to $\log_2$ (Turing time) then no Craig deduction system operating in polynomial time exists. This is a bit weaker than $P \neq NP$ but still gives an indication of what might be expected concerning the practical decidability of sentential logic. See [Mu 9] for further information.

REFERENCES

[Co] Cook, S.A., Rechkow, R.A.: The relative efficiency of propositional proof systems. J. Symb. Logic **44**, 36–50 (1979).

[Fr] Friedman, H.: The complexity of explicit definitions. Adv. Math. **20**, 18–29 (1976).

[Mu 1] Mundici, D.: Robinson's consistency theorem in soft model theory. Trans. Am. Math. Soc. **263**, 231–241 (1981).

[Mu 2] Mundici, D.: Compactness = JEP in any logic. Fund. Math. **116** (to appear) (1982).

[Mu 3] Mundici, D.: Complexity of Craig's interpolation. Ann. Soc. Math. Pol., Series IV: Fundamenta Informaticae **V.3/4**, 261–278 (1982).

[Mu 4] Mundici, D.: Natural limitations of algorithmic procedures in logic. Rendiconti of the Nat. Acad. of Lincei (Rome), Serie VIII, Vol. LXIX 3/4, 101–105 (1980).

[Mu 5] Mundici, D.: Irreversibility, uncertainty, relativity, and computer limitations. Il Nuovo Cimento, Europhysics Journal **61**B, No. 2, 297–305 (1981).

[Mu 6] Mundici, D.: Duality between logics and equivalence relations. Trans. Am. Math. Soc. **270**, 111–129 (1982).

[Mu 7] Mundici, D.: Quantifiers: an overview. In: Barwise, J., Feferman, S. (eds.): Abstract model theory and strong logics. Omega series. Berlin, Heidelberg, New York: Springer (to appear) 1983.

[Mu 8] Mundici, D.: Craig's interpolation theorem in computation theory. Rendiconti of the Nat. Acad. of Lincei (Rome), Serie VIII, Vol. **LXX.1**, 6–11 (1981).

[Mu 9] Mundici, D.: NP and Craig's interpolation theorem. In: Proceedings of the ASL Logic Colloquium '82 in Florence. Amsterdam: North-Holland (to appear) 1983.

[Mu 10] Mundici, D.: Compactness, interpolation, and Friedman's third problem. Ann. Math. Logic **22**, 197–211 (1982).

[Mu 11] Mundici, D.: Interpolation, compactness, and JEP in soft model theory. Arch. math. Logik **22**, 61–67 (1982).

[Sa] Savage, J.E.: The complexity of computing. New York: Wiley 1976.

[Sc] Schnorr, C.P.: The network complexity and the Turing machine complexity of finite functions. Acta Informatica **7**, 95–107 (1976).

[Sm] Smullyan, R.M.: First-order logic. Berlin, Heidelberg, New York: Springer 1971.

Daniele Mundici (National Research Council)
Loc. Romola N. 76
I-50060 Donnini (Florence)
Italy