In sum, the book is recommended as an introduction to the more applied modal logic, especially Dutch-style modal logic. But the student who is more interested in the theory of modal logic will find the book too uninformative given that its title suggests that this is a book about modal logic as a whole.

(I have benefited from discussions with Patrick Blackburn, Maarten de Rijke, and Michael Zakharyaschev. The views expressed here are, however, solely my own.)

MARCUS KRACHT

II. Mathematisches Institut, Freie Universität Berlin, Arnimallee 3, D-14195 Berlin, Germany. kracht@math.fu-berlin.de.

MICHAEL ALEKHNOVICH, SAM BUSS, SHLOMO MORAN, and TONIANN PITASSI. *Minimum propositional proof length is NP-hard to linearly approximate*. ***The journal of symbolic logic***, vol. 66 (2001), pp. 171–191.

Let P be a *propositional proof system*, that is, any complete and sound propositional calculus. The traditional proof complexity mostly deals with the question of *existence* of short P-proofs for a given propositional tautology φ. It is, however, quite conceivable (especially for powerful proof systems) that even when short propositional proofs do exist, we nonetheless do not have the slightest clue as to how to find one efficiently. In many cases (and especially from the perspective of automated theorem proving), such a situation is little better than the sheer non-existence of a short proof. M. Bonet, T. Pitassi, and R. Raz in *On interpolation and automatization for Frege systems* (***SIAM journal on computing***, vol. 29 (2000), pp. 1939–1967) proposed a rigorous formulation of what it means to say that *searching* for a short propositional proof does not essentially contribute to the *inherent* complexity of a tautology.

Namely, denote by $S_P(\varphi)$ the minimum bit size of a P-proof of φ. They called a proof system P *automatizable* if there exists a *proof search algorithm* that on an input tautology φ works in time *polynomial in* $S_P(\varphi)$ and outputs a P-proof of φ (such a proof will necessarily be "nearly optimal," that is, its size will also be at most polynomial in $S_P(\varphi)$).

Since the paper under review is entirely devoted to *negative* results about the existence of efficient proof search algorithms, let me briefly summarize what was previously known on the subject.

First of all, we must have some sufficiently strong complexity assumption to start with: without assuming at least $\mathbf{P} \neq \mathbf{NP}$, we cannot rule out the existence of a polynomial time algorithm for any reasonable algorithmic problem, including ours. Then it turns out that the strength of negative results that can be established is determined by two major factors: the strength of this complexity assumption and the strength of the proof system P itself. If we for example assume that certain popular and widely used cryptographic protocols (like RSA) are secure (which is, of course, much stronger than merely $\mathbf{P} \neq \mathbf{NP}$), then really strong proof systems like Frege or extended Frege are known not to be automatizable. If, however, we allow ourselves only $\mathbf{P} \neq \mathbf{NP}$ as the hypothesis (i.e., try to prove ordinary **NP**-hardness results), then our knowledge becomes by far more limited. No propositional proof system is known to be non-automatizable solely under this assumption. Moreover, **NP**-hardness results were known only for the problem of constructing *exactly optimal* proofs (i.e., proofs of size *exactly* $S_P(\varphi)$).

The paper under review proves the first **NP**-hardness results for the problem of constructing efficient *approximate* proof search algorithms. In fact, all its results are applicable in the following more general situation. Note that every efficient algorithm which *produces a P-proof* of nearly optimal size $S_P(\varphi)$, can at the same time be used for approximating the *numerical value* of $S_P(\varphi)$ within the same accuracy. The paper under review rules out efficient algorithms even for the latter, more limited goal of *approximating the minimum propositional proof length*.