

5. Hardware Modification Machines

Objektyp: **Chapter**

Zeitschrift: **L'Enseignement Mathématique**

Band (Jahr): **27 (1981)**

Heft 1-2: **L'ENSEIGNEMENT MATHÉMATIQUE**

PDF erstellt am: **10.08.2024**

Nutzungsbedingungen

Die ETH-Bibliothek ist Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Inhalten der Zeitschriften. Die Rechte liegen in der Regel bei den Herausgebern.

Die auf der Plattform e-periodica veröffentlichten Dokumente stehen für nicht-kommerzielle Zwecke in Lehre und Forschung sowie für die private Nutzung frei zur Verfügung. Einzelne Dateien oder Ausdrucke aus diesem Angebot können zusammen mit diesen Nutzungsbedingungen und den korrekten Herkunftsbezeichnungen weitergegeben werden.

Das Veröffentlichen von Bildern in Print- und Online-Publikationen ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Die systematische Speicherung von Teilen des elektronischen Angebots auf anderen Servern bedarf ebenfalls des schriftlichen Einverständnisses der Rechteinhaber.

Haftungsausschluss

Alle Angaben erfolgen ohne Gewähr für Vollständigkeit oder Richtigkeit. Es wird keine Haftung übernommen für Schäden durch die Verwendung von Informationen aus diesem Online-Angebot oder durch das Fehlen von Informationen. Dies gilt auch für Inhalte Dritter, die über dieses Angebot zugänglich sind.

Proof sketch: It suffices to show that SATISFIABILITY is in the second class given it is in the first class. Reasoning as above, the Turing machine M can check whether a candidate aggregate β'_n correctly tells whether a propositional formula F is satisfiable by making β'_n produce a satisfying assignment bit by bit, by plugging in partial truth assignments to F and asking β'_n about the result. The trouble is M cannot remember the partial assignments in small space. However, the problem of whether "the i -th bit is 1 in the lexicographically first assignment which β'_n says satisfies F " is in P . Thus by theorem 4.2 this bit can be determined in space $O(\log n \cdot \log \log n)$, and M can determine whether this assignment satisfies F in small space.

5. HARDWARE MODIFICATION MACHINES

As mentioned in the introduction, there is a need to define a parallel model which is more powerful than an aggregate, in that it can modify its circuits, but less powerful than existing parallel RAM models, in that each unit of hardware can only perform a bounded amount of work in one step. We shall call the new machine a *hardware modification machine* (HMM), since it is intended to be the parallel analog of the storage modification machine. An HMM consists of a finite collection of finite state machines connected together as in a conglomerate. At each step, each machine may, in addition to assuming a new state and transmitting output signals, modify its input connections. Specifically, it may detach any of its inputs and re-attach it to a new machine which it brings into the HMM, or it may re-attach it to an output of any machine which can be reached by a path of length at most two traced backwards from the input.

One advantage of an HMM over circuits, aggregates, and conglomerates is that there is no question of uniformity. The machine is uniform because it constructs itself.

An HMM can execute an algorithm like the one described in [FW] to simulate a deterministic S space bounded machine in time $O(S)$, and HMM time S can be simulated in deterministic space $O(S^2)$. Thus the inclusions (1.2) apply.

The theory of HMM's is developed in [D1].