Dealing with Size Limits in a Hardware Encoding of Weighted Finite Automata

Mathieu Giraud and Dominique Lavenier
IRISA / CNRS / Université de Rennes 1
35042 Rennes Cedex, France
e-mail: mathieu.giraud@irisa.fr

We use Weighted Finite Automata (WFA) [7] to parse protein or nucleic banks for finding specific patterns. The weights of WFA enable arbitrary error counts or substitution costs [5]. A WFA $\mathcal{A}$ over the semi-ring $(\mathbb{Z}, +, \max)$ assigns to every word w a weight $P(w)$. With a threshold s_0, one can define the language recognized by $\mathcal{A}$ by $\mathcal{L} = \{w \,|\, P(w) \geq s_0\}$. Given a large word w (the bank), we address the problem of *continuous pattern matching*: one must find all the terminating positions j of matching subwords $w_i w_{i+1} \dots w_j \in \mathcal{L}$. We hardwire WFA in reconfigurable processors to accelerate this parsing.

Linear Encoding Scheme. Finite state machines with q states can be encoded in hardware either by the *logarithmic scheme* (a bit vector of size $\log_2 q$ stores the current state) or by the *linear scheme* which uses a bit vector of size q. Usually only one bit is set (this scheme is also named *one-hot*), but one can have multiple states active at the same time and thus simulate indeterminism. The linear scheme has other advantages [3, 8], and we showed that it can be generalized to parse WFA [5]. In this case, each state is translated into one flip-flop, and each set of transitions between two states is translated into a weight generator, an adder and an optional maximum operator (figures 1 and 3).

Size and Speed. The hardwired WFA has a surface area of $O(|\Sigma| \cdot p \cdot |\delta|)$, where p is the number of bits representing the weight and $|\delta|$ the number of transitions. One character is parsed on every clock cycle. The cycle time is in $O(p \cdot \log d_{\max})$, where $d_{\max}$ is the maximum incoming degree.

Prototype Implementation. Our practical implementation uses the *R-disk prototype*, a parallel architecture designed for mass data filtering [6]. Data is distributed among several nodes linked by an Ethernet network, and each node houses a hard disk drive and a reconfigurable processor (a FPGA) which filters data in a on-the-fly way. A host computer send queries and collects results (figure 2). With the current reconfigurable chip, the size constraint is $p \cdot |\delta| \leq 600$. Therefore, one can use WFA with 75 transitions and 8 bits weights. That covers common biological patterns like those of [1]. The speed constraint is less restrictive, as the clock runs at 40 MHz, and each board can filter data at 16 MB/s. That flow on a single board is more than *4 times faster* than software simulation of WFA [4] on a 2 GHz PC. Massive parallelism is achieved through parallelization of several boards.

Additional Transitions. The size limit becomes crucial in some applications, for example when WFA modelize insertions or deletions in patterns. The linear encoding scheme prevents chains of ϵ-transitions from being arbitrarily long (figure 4) because of the increasing critical path. Systolic techniques could realize ϵ-transitions, but they are only suitable for linear-shaped WFA. One solution is to add *pseudo-deletion transitions* for each possible path (figure 5). There are $O(|\delta|^2)$ such transitions in the general case, but only a subset of them is sufficient for practical implementation to fit into the available chip size.

We are currently assembling a prototype of this architecture with 48 boards, and we expect a maximum flow of 768 MB/s. Moreover, as the available size on reconfigurable processors evolves faster than the computing power of conventional sequential machines, this kind of space computing will be even more advantageous [2].

References

[1] P. Bucher and A. Bairoch. A Generalized Profile Syntax for Biomolecular Sequences Motifs and its Function in Automatic Sequence Interpretation. In *Second International Conference on Intelligent Systems for Molecular Biology (ISMB 94)*, pages 53–61, 1994.

[2] A. DeHon. Very Large Scale Spatial Computing. In *Third International Conference on Unconventional Models of Computation (UMC 02)*, pages 27–37, 2002.

[3] J. Dunoyer, F. Pétrot, and L. Jacomme. Stratégies de codage des automates pour des applications basse consommation : expérimentation et interprétation. In *Journées d'étude Faible Tension et Faible Consommation (FTFC 97)*, 1997.

[4] M. G. Eramian. Efficient Simulation of Nondeterministic Weighted Finite Automata. In *Fourth Workshop on Descriptional Complexity of Formal Systems (DCFS 02)*, 2002.

[5] M. Giraud and D. Lavenier. Réalisation matérielle d'automates pondérés pour la recherche de motifs génomiques. In *Symposium en Architecture et Adéquation Algorithme Architecture (SympAAA 03)*, pages 345–352, 2003.

[6] D. Lavenier, S. Guyetant, S. Derrien, and S. Rubini. A Reconfigurable Parallel Disk System for Filtering Genomic Banks. In *International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA 03)*, 2003.

[7] M. Mohri. Finite-State Transducers in Language and Speech Processing. *Computational Linguistics*, 23(2):269–311, 1997.

[8] R. Sidhu and V. Prasanna. Fast Regular Expression Matching Using FPGAs. In *IEEE Symposium on Field Programmable Custom Computing Machines (FCCM 01)*, april 2001.

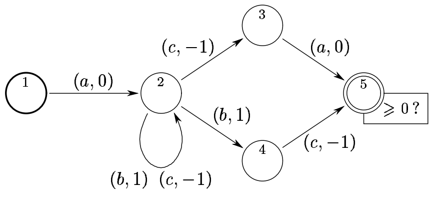

Fig. 1 – The WFA $\mathcal{A}_1$

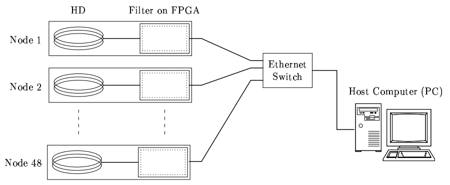

Fig. 2 – The R-disk Architecture

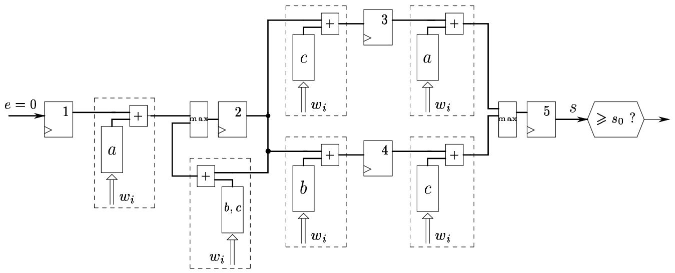

Fig. 3 – Linear Encoding Scheme for $\mathcal{A}_1$

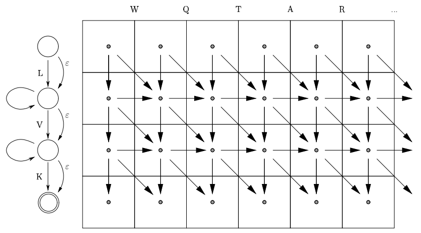

Fig. 4 – Deletions with ϵ-transitions

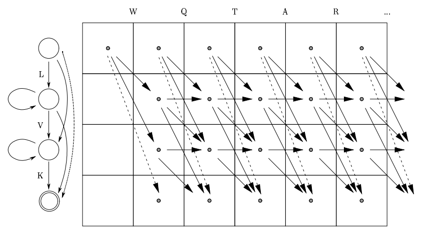

Fig. 5 – Pseudo-deletion transitions