A circuit uniformity sharper than DLogTime

Guillaume Bonfante, Virgile Mogbil

A circuit uniformity sharper than DLogTime

(rapport interne LIPN - Mai 2012)*

Guillaume Bonfante[1] and Virgile Mogbil[2]

[1] LORIA – UMR7503, CNRS – Université de Lorraine
[2] LIPN – UMR7030, CNRS – Université Paris 13

Abstract. We consider a new notion of circuit uniformity based on the concept of rational relations, called Rational-uniformity and denoted *Rat*. Our goal is to prove it is sharper than DLogTime-uniformity, the notion introduced by Barrington et al. [3], denoted DLT, that is: 1) Rational-uniformity implies DLogTime-uniformity, 2) we have $\mathrm{NC}^0_{Rat} \subsetneq \mathrm{NC}^0_{DLT} \subsetneq \mathrm{NC}^1_{Rat}$, 3) we have $\forall k \geqslant 0,\ \mathrm{NC}^k_{DLT} \subseteq \mathrm{NC}^{k+1}_{Rat}$, 4) Rational-uniformity preserves separation results known with DLogTime-uniformity. In other word, we obtain an interleaved hierarchy:

$$\mathrm{NC}^0_{Rat} \subsetneq \mathrm{NC}^0_{DLT} \subsetneq \mathrm{NC}^1_{Rat} \subseteq \cdots \subseteq \mathrm{NC}^k_{Rat} \subseteq \mathrm{NC}^k_{DLT} \subseteq \mathrm{NC}^{k+1}_{Rat} \subseteq \cdots \subseteq \mathrm{NC}$$

which implies $\mathrm{NC}_{Rat} = \mathrm{NC}_{DLT}$. We also prove that $\mathrm{Reg} \neq \mathrm{NC}^0_{Rat}$. In other words, circuit build by rational relations compute relations not computable by rational relations. Finally, we consider circuit with unbounded fan-in, and we prove the standard result $\mathrm{NC}^k_{Rat} \subseteq \mathrm{AC}^k_{Rat} \subseteq \mathrm{NC}^{k+1}_{Rat}$ for all $k \geqslant 0$.

1 Introduction

In [10], Cook has described boolean circuits as a computational model of parallelism. In these views, the depth of a circuit corresponds to the running time, while its size corresponds to the number of processors. The class NC, that is functions computed by circuits of polynomial size and poly-logarithmic depth, plays a major role: it corresponds to the case for which there is an exponential speed-up (up to a polynomial) between the parallel algorithm and the relative sequential algorithm.

One of the issues with boolean circuits is that they intrinsically depend on the number of bits of their inputs. Thus, since problems handle inputs of arbitrary size, they are not computed by a single circuit but by a family of circuits. Without restrictions on the family, circuits may compute problems even not computable by a Turing Machine. In other words, the computational power of the model lies in the complexity of the family (the *external* computational power), not in the circuit's capabilities (the *internal* computational power). To tame this

* Work partially supported by the french project Complice (ANR-08-BLANC-0211-01,3)

awkward external computational power, circuit families are described by a "simple" mechanism depending on the number of inputs as its size measure, they are said to be *uniform*.

With time, the notion of uniformity considered by the authors has been sharper and sharper. It corresponds to polynomial time in Borodin [6], then logarithmic space in Cook [10], then alternating logarithmic time (ALogTime) in Ruzzo's seminal paper [21] and finally deterministic logarithmic time (DLogTime) by Barrington, Immermann and Straubing [3] after Buss's notion of problem reduction [8].

We propose to go one step further in the restrictions of the construction of circuit families, and to consider a uniformity condition based on the notion of (length preserving) rational relations. Predecessor gates are computed by a multi-tape one-way finite state automaton. Let us make two remarks to motivate this notion of uniformity. First, length preserving rational relations are quite robust: they are closed by union, intersection, complement, composition. Thus, many manipulation of addresses can be done harmlessly which means that the uniformity condition itself—though very sharp—is quite robust. Second consideration, in [5], we have characterized each layers of the hierarchy $(\mathrm{NC}^k)_{k\geqslant 2}$ by means of some recursive schemas. The core recursion scheme, that is mip-recursion (mutual in place recursion), involves computations by rational uniform circuit of constant depth. The layers NC^k are obtained by iterations of these constant depth circuits. The present work is a first step toward a characterization of rational uniform circuit families.

Our contribution splits in four parts. In a first step, we define the notion of rational-uniformity of a family of circuit and we justify the definition. In particular, we discuss its robustness. One of the main point of the definition is that the size parameter of the connection language refers to the number of inputs n written in binary, not to a witness word of size n. Such a choice already appears in [2], it is however mandatory for rational uniformity due to the low computability power of finite state automata. Then, we propose some alternative descriptions for circuit based on a finite fan-in basis or constant depth circuits, the latter being used for a separation theorem proof. In a second step, to relate the rational uniform hierarchy to the deterministic logarithmic time hierarchy, we establish the inclusion $\mathrm{NC}^k_{Rat} \subseteq \mathrm{NC}^k_{DLT} \subseteq \mathrm{NC}^{k+1}_{Rat}$ for all $k \geqslant 0$. It is based on the internalization of the construction of the circuits within the computation, an argument already present in Ruzzo's work [21]. In a third step, we compare the internal computational capabilities of rational circuit families with respect to the external ones, we prove $\mathrm{NC}^0_{Rat} \neq \textsc{Reg}$. At the same time, we show that at the first level there is a separation theorem: $\mathrm{NC}^0_{Rat} \subsetneq \mathrm{NC}^0_{DLT}$. Finally, in a last step, we consider rational circuit families based on unbounded fan-in gates, and we prove the standard theorem $\mathrm{NC}^k_{Rat} \subseteq \mathrm{AC}^k_{Rat} \subseteq \mathrm{NC}^{k+1}_{Rat}$ for all $k \geqslant 0$. To sum up, we establish the hierarchy:

$$\mathrm{NC}^0_{Rat} \subsetneq \mathrm{NC}^0_{DLT} \subsetneq \mathrm{NC}^1_{Rat} \subseteq \cdots \subseteq \mathrm{NC}^k_{Rat} \subseteq \mathrm{NC}^k_{DLT} \subseteq \mathrm{NC}^{k+1}_{Rat} \subseteq \cdots \subseteq \mathrm{NC}$$

with a separation theorem at the first levels. In the remaining of the paper, we study the hierarchy AC^k_{Rat}, $k \geqslant 0$, and we prove the expected result $\mathrm{NC}^k_{Rat} \subseteq$

$\mathrm{AC}^k_{Rat} \subseteq \mathrm{NC}^{k+1}_{Rat}$ for all $k \geqslant 0$. We relate the hierarchy to the standard scale with a proof that $\mathrm{AC}^k_{DLT} \subseteq \mathrm{NC}^{k+1}_{Rat}$ for all $k \geqslant 0$ with a separation theorem at the first level.

As a by product of this work, we provide a new description of the class NC, a class of problems which has been intensively studied in the past. Remarkably, NC is characterized according the three main lines of implicit computational complexity, that is in terms of descriptive complexity e.g. [3], with respect to the Curry-Howard paradigm [24, 16], or by means of recursion theory, see the work of Leivant [14], Oitavem [18] with Bellatoni [4].

2 Preliminaries

To begin with notations, $\bar{u}$ denotes a sequence $u_1, \ldots, u_n$. Usually, n is understood from the context. The complement set $\complement^S_R \triangleq \{x \in S \mid x \notin R\}$ is abbreviated to $\complement_R$ when S is clear from the context.

Given some *alphabet* Σ of *letters*, the set Σ^* denotes the set of words over Σ. The boolean alphabet is $\mathbf{B} = \{0, 1\}$. The empty string is written ε and $w \cdot w'$ denotes the concatenation of two words w and w'. Given a word w, the k-th concatenation w^k is defined by the equations: $w^0 = \varepsilon$, $w^{k+1} = w^k \cdot w$. The length of a word is written $|w|$. Finally, the subset of words in Σ^* of length smaller than $k \in \mathbf{N}$ is denoted by $\Sigma^{\leqslant k}$, those of length exactly k by Σ^k.

We let $\underline{n}$ be the usual binary encoding of the natural number n in $|\underline{n}| = \lfloor \log_2(n) \rfloor + 1$ bits. We also denote it $|n|$ when it is unambiguous. Actually, we do not specify the order of the bits: read left to right, they may be sequenced from less significant-bits to most significant ones (LH), or from most significant-bits to less significant ones (HL). This will not make difference since rational languages are closed by mirror. For $k > |\underline{n}|$, $\underline{n}_k$ denotes the padding of $\underline{n}$ with leading 0's so that $|\underline{n}_k| = k$. Naturally, for an encoding LH, the zero's appear on the right, for HL on the left.

An n-ary relation $R \subseteq (\Sigma^*)^n$ is said to be *length preserving* iff $(w_1, \ldots, w_n) \in R$ implies that $|w_1| = \cdots = |w_n|$. For a function $f : X \to Y$, $\mathrm{dom}(f)$ is its domain, that is the set of $x \in X$ such that $f(x)$ is defined. A function $f : \Sigma^* \to \mathbf{B}$ defines a language $\mathcal{L}(f) = \{w \in \Sigma^* \mid f(w) = 1\}$ called the relative language of the function.

2.1 Multi-tape finite state automata

We recall the some definitions concerning transducers and more generally multi-tape one-way finite state automata, and some well known facts used in the sequel. We refer the reader to the handbook [20] or to the book of Sakarovitch [22] for the proofs of the statements presented in this section. Some constructions are taken from Kaplan and Kay [13].

Definition 1 (Originally introduced by Rabin and Scott [19]). *A multi-tape one-way non writing finite state automaton (MONA) is a* 5*-tuple* $\langle \Sigma, Q, q_0, F, \delta \rangle$

with Σ an alphabet, Q a (finite) set of states, *$q_0 \in Q$ is called the* initial state, *$F \subseteq Q$ is the set of final states and $\delta \subseteq Q \times (\Sigma^*)^k \times Q$ is the transition relation. δ induces a relation $\delta^* \subseteq Q \times (\Sigma^*)^k \times Q$ defined as follows. $(q, w_1, \ldots, w_k, q') \in \delta^*$ iff $(q, w_1, \ldots, w_k, q') = (q, \varepsilon, \ldots, \varepsilon, q)$ or $(q, u_1, \ldots, u_k, q'') \in \delta^*$, $(q'', v_1, \ldots, v_k, q') \in \delta$ and $w_i = u_i \cdot v_i$ for all $i \leqslant k$.*

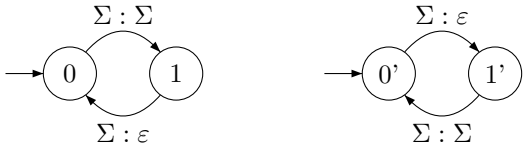
A MONA $T = \langle Q, \Sigma, \Gamma, q_0, F, \delta \rangle$ induces the relation $[T] \subseteq (\Sigma^*)^k$ to be $(w_1, \ldots, w_k) \in [T]$ iff there exists $q_f \in F$ such that $(q_0, w_1, \ldots, w_k, q_f) \in \delta^*$.

A relation $S \subseteq (\Sigma^*)^k$ is said to be *rational* if there is a MONA T such that $S = [T]$. It is a rational function if the relation is a function (the last component being the "output").

When $k = 1$, the definition corresponds to standard (non-deterministic) finite state automata, when $k = 2$, it corresponds to transducers which have a common input and output alphabet.

Example 1.

The left transducer $\mathcal{H}_0$, removes even letters from a word on some alphabet Σ, while right transducer $\mathcal{H}_1$ removes odd letters.



We recall well known facts about rational relations. They are justified by Nivat's Theorem. In the present setting, it is formulated as follows.

Theorem 1 (Nivat [17]). *Let Σ be a finite alphabet. $R \subseteq (\Sigma^*)^k$ is a rational relation iff there is a finite alphabet Γ, a regular language $S \subseteq \Gamma^*$ and k monoid morphisms $g_1 : \Gamma^* \to \Sigma^*, \ldots, g_k : \Gamma^* \to \Sigma^*$ and $R = \{(g_1(w), \ldots, g_k(w)) \mid w \in S\}$.*

First, rational relations are closed under union: given two MONA S and T, we let $S \cup T$ be the (a choice of a) MONA corresponding to the relation $[S] \cup [T]$. They are closed under composition:

Theorem 2. *Given two MONA S and T, let $(\bar{x}, \bar{z}) \in [S] \circ [T]$ iff $\exists \bar{y} \mid (\bar{x}, \bar{y}) \in [T] \wedge (\bar{y}, \bar{z}) \in [S]$. Then, there is a MONA $S \circ T$ such that $[S \circ T] = [S] \circ [T]$.*

Proposition 1. *Any projection of a MONA S is regular: that is for all $i \leqslant k$, $\{x_i \mid \bar{x} \in [S]\}$ is a regular language. For any regular sets $R_1, \ldots, R_{i-1}, R_{i+1}, \ldots, R_k$, the range set $\{x_i \mid \bar{x} \in [S] \wedge \forall j \neq i : x_j \in R_j\}$ is a regular language.*

Rational length-preserving relations are closed under intersection and complementation. Again, given two MONA S and T, let $S \wedge T$ denote the MONA $[S \wedge T] = [S] \wedge [T]$ and $[\complement_S] = \complement_{[S]}$. These two constructions can be actually justified by the following Proposition (see for instance Sakarovitch [22]):

Proposition 2. *For any rational length-preserving relation, there is a MONA $\langle \Sigma, Q, q_0, F, \delta \rangle$ such that $(w_1, \ldots, w_k) \in \delta$ only if $|w_1| = \cdots = |w_k| = 1$. Such a MONA is called a letter-to-letter MONA.*

Proposition 3. *Given a rational relation $R \subseteq (\Sigma^*)^k$ and a permutation $\sigma : \{1, \ldots, k\} \to \{1, \ldots, k\}$, the relation $R\sigma \triangleq \{\bar{w} \mid (w_{\sigma(1)}, \ldots, w_{\sigma(k)}) \in R\}$ is a rational relation.*

Proposition 4. *Consider a rational relation computed by a MONA $[T] \subseteq (\Sigma^*)^k$. If $R \subseteq \Sigma^*$ is a regular language then there is a transducer denoted by $T_{|R,i}$ such that $\bar{w} \in [T_{|R,i}]$ iff $\bar{w} \in [T]$ and $w_i \in R$.*

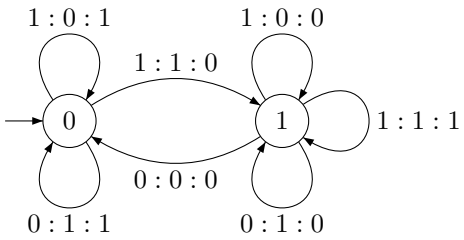
As a corollary of the closure by intersection and by complement, suppose R, S and T be three MONAs representing length-preserving relations, then, the following relation is rational (and length preserving): $U(\bar{w}) = \{\bar{w} \mid \bar{w} \in [R] \wedge [T]$ or $\bar{w} \notin [R] \wedge \bar{w} \in [S]\}$.

The relation U is computed by the MONA $(R \wedge T) \vee (\complement_R \wedge S)$ next denoted as follows: `if` $R(\bar{w})$ `then` $T(\bar{w})$ `else` $S(\bar{w})$.

Example 2. In particular, any singleton language is regular, thus if a MONA T computes a function $(\Sigma^*)^{k-1} \to \Sigma^*$, $a \in \Sigma^*$ and two MONAs U, V compute relations in $(\Sigma^*)^k$, the relation $R(\bar{w}) =$ `if` $[T](\bar{w}) = a$ `then` $[U](\bar{w})$ `else` $[V](\bar{w})$ is rational.

Example 3.

Addition (and thus subtraction by permutation) and comparison can be computed by length preserving MONA (as long as the two numbers are encoded by words of equal length and with an upper-bit equal to 0). We give the example of addition, the states contain the carry of the addition.



3 RATIONAL-uniform circuits

In this section, we define circuit families and circuit complexity classes. We refer the reader to the book of Vollmer [25] for a wide presentation of circuit complexity. In a large extent, definitions, notations are taken from the book.

A circuit is a labelled directed acyclic graph whose nodes, called *gates*, output values in booleans $\mathbf{B} = \{0, 1\}$. Labels, also called the *types*, are elements of the basis $\mathcal{B}_0 = \{\wedge, \vee, \neg, 0, 1, \mathbf{i}, \mathbf{o}\}$. Gate with type $\wedge, \vee$ have exactly two predecessors, gates of type $\neg$ and $\mathbf{o}$ (that is *output* gates) have one predecessor. Gates typed 0 (*rejecting* gates), gates typed 1 (*accepting* gates), and *input* gates (with type $\mathbf{i}$) have no predecessors.

Inputs (resp. outputs) are supposed to be ordered from 0 to $n-1$ (resp. from 0 to $m-1$), for some $n, m > 0$. Plugin the input $\bar{x} = x_0 \cdot x_1 \cdots x_{n-1} \in \{0, 1\}^n$ into a circuit C outputs an array of boolean of size m, denoted $C[\bar{x}]$. Such a circuit C computes the function $\{0, 1\}^n \to \{0, 1\}^m$ defined by $\lambda\bar{x}.C[\bar{x}]$. A circuit family is a sequence of circuits $(C_n)_{n>0}$ such that, for all $n \in \mathbf{N}$, the circuit C_n

has exactly n input gates. Such a family computes the function $\{0,1\}^* \to \{0,1\}^*$ defined by $\bar{x} \mapsto C_{|\bar{x}|}[\bar{x}]$. A circuit is said to be *boolean* if it outputs only one bit. A family of circuit is said to be boolean if all circuits of the family are boolean. In that case, the family defines a language, namely the set $\{\bar{x} \mid C_{|\bar{x}|}[\bar{x}] = 1\}$.

Functions computed by circuit family are length respecting, that is $|f(w)| = |f(w')|$ whenever $|w| = |w'|$. As argued by Vollmer, this is actually not a restriction, since it is possible to pad the output with leading zeros. So, from now on, all functions are supposed to be length respecting.

A path in a circuit is a pair (u, π) of a gate u and a word $\pi \in \{0,1\}^*$. Given a path (u, π), $u[\pi]$ denotes the gate defined as follows if it exists. $u[\varepsilon] = u$ and $u[\pi \cdot b]$ is the b-th predecessor gate of $u[\pi]$ if it exists.

For some circuit C, we denote by $|C|$ its size, that is the number of gates in C, and by $||C||$ its height – or its depth –, that is length of the longest path from an output gate to an input gate. The depth of a gate w is the maximal length of a path from w to an input gate.

3.1 RATIONAL-uniformity

We present a more restrictive notion of uniformity than DLOGTIME-uniformity which we call RATIONAL-uniformity. Gates are given identifiers, also called addresses, that is words in some given alphabet $\Sigma \supseteq \{0,1\}$.

All along, we suppose that circuits have polynomial size, there is a polynomial P such that $|C_n| \leqslant P(n)$ for all $n > 0$. Consequently there are constants $A, B \geqslant 0$ such that all the gates can be encoded in less than $|P(n)| \leqslant A \times |\underline{n}\,| + B$ bits. Accordingly, input addresses are encoded $\mathbf{i}\underline{0}_{\,A\times|\underline{n}\,|+B}, \ldots, \mathbf{i}\underline{n-1}_{\,A\times|\underline{n}\,|+B}$ where $\mathbf{i}$ is a letter different from 0,1. In the same way, output addresses are given addresses $\mathbf{o}\underline{0}_{\,A\times|\underline{n}\,|+B}, \ldots, \mathbf{o}\underline{m-1}_{\,A\times|\underline{n}\,|+B}$ with $\mathbf{o}$ a fresh letter. The addresses of the other gates are also supposed to have length $A \times |\underline{n}\,| + B$. The size witness is $\underline{n}_{\,A|\underline{n}\,|+B}$. Actually, since A and B are fixed by the encoding, we will shorten $\underline{n}_{\,A|\underline{n}\,|+B}$ in the following to $\underline{n}$.

This rather restrictive discipline on the syntax of addresses help us to provide some (more) precise descriptions of circuit families in the sequel. The syntax can be for sure generalized, but we let this issue for further research.

Definition 2. *A family of circuit $(C_n)_{n>0}$ is said to be* RATIONAL*-uniform iff:*

1. *let $\tau(-,-)$ be the function mapping any pair $(\underline{n}\,, w)$ such that w denotes an address of the circuit C_n to its type, that is in $\mathcal{B}_0$. Then, the function τ is rational.*
2. *the predecessor function φ mapping any triple $(\underline{n}\,, w, k)$ to the address of the k-th predecessor of gate w is a rational function. We take the convention that $\varphi(\underline{n}\,, w, k)$ is not defined if there are no such k-th predecessor gate.*

The underlying MONA corresponding to Item (1) is called the typing automaton and the one corresponding to Item (2) is the predecessor automaton.

The standard notion of uniformity is an efficient computation of a description of each C_n from 1^n (a word of size n). Formally this refers to the direct connection

language of the circuit family $(C_n)_n$, that is the set made of the tuples $(y, \underline{g}, p, \underline{b})$ such that: a) $|y| = n$, b) g is a gate address in C_n, c) $p \in \{\varepsilon, 0, 1\}$ and d) if $p = \varepsilon$, then $b \in \mathcal{B}_0$, otherwise b is the address of the p-th predecessor of g.

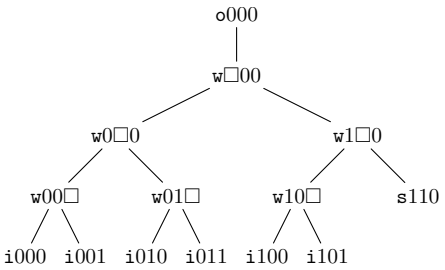
The major difference between the standard notion of uniformity and the current one is that we use a size witness which has size $O(\log(n))$ (that is the argument $\underline{n}$) and not n (the y's in the tuples). This choice is due to an intrinsic limitation on the relative size of components in rational relations. Indeed, consider a MONA running on the input (y, w). By the pumping Lemma, for $|w| > C$, $y = y_1 \cdot y_2 \cdot y_3$ and $w = w_1 \cdot w_2 \cdot w_3$ and for all $k \geqslant 0$, $(y_1 \cdot y_2^k \cdot y_3, w_1 \cdot w_2^k \cdot w_3)$ belongs to the relation. However, in the standard representation, since $|w| = O(\log(|y|))$, for all constant $A > 0$, there is a sufficiently large y such that $|y| > A \times |w|$. This contradicts $|w_2| > 0$. $|w_2| = 0$ means that $|y_2| > 0$ and $(y_1 \cdot y_3, w)$ is in the relation. Iterating this process, one finally gets a word y of smaller than C, in contradiction with the fact that $|y| \geqslant O(\log(|w|))$.

Second remark, thinking in terms of the direct connection language, Item (1) of Definition 2 corresponds to $p = \varepsilon$ and Item (2) to $p \in \{0, 1\}$. It is only for convenience that we split the mappings. From Proposition 4, it would be equivalent to take a unique transducer for both typing and predecessor functions.

Third remark, for our bounded fan-in function basis ($\mathcal{B}_0$), the second argument of φ is either 0 or 1. Again, by Proposition 4, it is equivalent to define φ by a pair φ_0, φ_1 of two rational functions. Then, the two functions are length preserving, thus can be described by a letter-to-letter MONA. For types, the function τ can be equivalently replaced by rational relations $\tau_t(-,-)$ for each type $t \in \mathcal{B}_0$ where $(x, w) \in \tau_t$ iff $\tau(x, w) = t$. Again, the MONA corresponding to τ_t can be supposed to be letter to letter.

Example 4.

Consider the addressing system for a circuit with 6 input bits and 1 output bit presented on the left. If one define gate addressed with a prefix `w` to have type $\vee$ and gates addressed with `s` to be rejecting gates, then, the circuit computes the function $\texttt{OR}^6$. Here, as length parameter, we have set $A = 1$ and $B = 1$. It is clear that the construction extends to a RATIONAL-uniform circuit family. Such circuits provide access to all the bits of the input word through a binary tree of depth $\log(n)$. To justify rationality, let us observe that the typing is immediate from the prefix letter and for the two predecessor functions, we work by case analysis.



$$\varphi_0(\underline{n}, \texttt{o}0^{|\underline{n}|}) = \texttt{w}\square 0^{|\underline{n}|-1}$$

$$\varphi_i(\underline{n}, \texttt{w}u\square 00^m) = \texttt{if } u \cdot i \cdot 0^{m+1} < \underline{n} \texttt{ then w} \cdot u \cdot i \cdot \square 0^m \texttt{ else s} \cdot u \cdot i \cdot 0^{m+1}$$

$$\varphi_i(\underline{n}, \texttt{w}u\square) = \texttt{if } u \cdot i < \underline{n} \texttt{ then i} \cdot u \cdot i \texttt{ else s} \cdot u \cdot i$$

with $u \in \{0,1\}^*, i \in \{0,1\}, m \geqslant 0$. Otherwise, φ_i, $i \in \{0,1\}$ is left undefined. The presentation entails the rationality of the predecessor functions.

We remind that a circuit family is said DLogTime-uniform (resp. ALogTime-uniform) if its direct connection language is in DLogTime (resp. ALogTime). As was to be expected we have the following property.

Theorem 3. *Any* Rational*-uniform circuit family is* DLogTime*-uniform.*

Proof. Let us suppose given a Rational-uniform circuit family $(C_n)_n$ with respective typing and successor functions $(\tau, \varphi_0, \varphi_1)$ on some alphabet Σ. As justified above, these relations are respectively computed by the letter to letter MONAs $(T_t)_{t\in\mathcal{B}_0}, \Phi_0, \Phi_1$. The main issue is that data are not presented in the same way for the two notions of uniformity. So, in a first step, we prepare the data to fit the format of the MONAs $(T_t)_{t\in\mathcal{B}_0}, \Phi_0$ and Φ_1.

Let us consider the Turing Machine $\mathcal{M}$ defined as follows. Remember that inputs for DLogTime-uniformity have the form $I = (y, \underline{g}, p, \underline{b})$. First, one computes $\log(y)$ on some working tape Y and pad it to length $A \times \log(y) + B$, this is $\underline{n}$. The computation of $\log(n)$ can be done in log-time as justified by Buss [8], the padding takes $A \times \log(n) + B$ steps, that is logarithmic time. Copying $\underline{g}$, p and $\underline{b}$ on respectively working tapes G, P and B is also done in log-time since $\underline{g}$ and $\underline{b}$ have logarithmic length. Words on the working tapes Y, G and B have the format required by the MONAs $(T_t)_{t\in\mathcal{B}_0}, \Phi_0$ and Φ_1.

If $p = \varepsilon$, one runs the MONAs $(T_t)_{t\in\mathcal{B}_0}$ in parallel on Y, G, B and we output the index of the succeeding MONA (the one reaching a final state). If $p = 0$, one runs Φ_0 on Y, G, B. Otherwise, one runs Φ_1. In each cases, since we took the precaution to use letter-to-letter MONA, the run time is in the size of the tapes, that is in $O(\log(n))$. To conclude, it is immediate to see that the machine $\mathcal{M}$ recognizes the connection language of the family, and that it runs in logarithmic time.

3.2 Constant-depth Rational-uniform circuit families

In this section, we present an alternative description of constant-depth Rational-uniform circuit families. It is used in Section 5 to prove the separation Theorem 5.

Let $\mathcal{C}^d$ be the finite set of circuits C such that $||C|| \leqslant d$ and C has a unique root, that is a unique gate which is an ancestor of all other gates. Each circuit C in $\mathcal{C}^d$ has clearly less than $W = 2^d$ input gates. Let $\Pi(C) = (\pi_1(C), \ldots, \pi_k(C))$ with $k \leqslant W$ be a sequence of all the paths from the root to the input gates of C. For each circuit $C \in \mathcal{C}^d$, for each $w \in C$, let $E(C, w)$ be the finite set of pairs of words (π_1, π_2) such that $w[\pi_1] = w[\pi_2]$.

A family of circuit $(C_n)_{n>0}$ is said to have constant depth whenever there is some $d > 0$ such that $||C_n|| \leqslant d$ for all $n > 0$. For each gate w in $(C_n)_n$, let us consider the sub-circuit made of w and its predecessors. It has depth at most d, and thus it is isomorphic to a circuit in $\mathcal{C}^d$, next called the index of w. By isomorphic we mean that the two circuits only differ by a renaming of addresses.

Proposition 5. *Given a* RATIONAL*-uniform circuit family of constant depth d, let θ be the function mapping $(\underline{n}\,, w)$ for $n > 0$ and an output gate address w in C_n to the index of w. The function θ is rational.*

In other words, up to some extra-copies of the circuit size, one may compute with a MONA the part of the circuit which participate to the computation of some gate w.

Proof. In a first step, we build a family of $d+2$-ary rational relations $(\theta_C)_{C \in \mathcal{C}^d}$ whose intention is the following: for some $n > 0$ and some gate address w, θ_C holds on $(\underbrace{\underline{n}\,, \ldots, \underline{n}}_{d+1 \text{ times}}, w)$ iff C is the index of w where d is the depth of the circuit C. Actually, the definition of the relations θ_C is done by induction on the depth of circuits $C \in \mathcal{C}^d$.

The circuits C of depth 0 in $\mathcal{C}^d$ are composed of one gate, either an input gate, an accepting gate or a rejecting gate. We define respectively $\theta_C \triangleq \tau_{\mathbf{i}}$, $\theta_C \triangleq \tau_1$, $\theta_C \triangleq \tau_0$.

Consider a circuit C of higher depth. By case analysis on the type t of the root gate r of C. If t corresponds to a unary gate, that is $\mathbf{o}$ or $\neg$, let s be the unique predecessor of r in C. As a matter of fact, s is the root of a circuit $C' \in \mathcal{C}^d$ of depth strictly smaller than C. We define accordingly $\theta_C(x_0, \ldots, x_d, w) \triangleq \tau_t(x_0, w) \wedge \theta_{C'}(x_1, \ldots, x_d, \varphi_0(x_0, w))$.

If t corresponds to a logical gate, $\wedge$ or $\vee$. Let s_0, s_1 be the two successors (possibly equal) of the unique root $r \in C$. The gate s_0 and s_1 are respectively the root of the circuits C_0, C_1 of depth strictly smaller than the depth of C. Let $\pi \in \{0,1\}^*$, we define by induction on π the rational function $\varphi_\pi(x_1, \ldots, x_n, w)$ by $\varphi_\varepsilon(x_1, \ldots, x_n, w) = w$ and $\varphi_{a \cdot \pi'}(x_1, \ldots, x_n, w) = \varphi_\pi(x_1, \varphi_a(x_2, \ldots, x_n, w))$. Then, $\varphi_\pi(\underline{n}\,, \ldots, \underline{n}\,, w)$ outputs $w[\pi]$ whenever the sequence $\underline{n}\,, \ldots, \underline{n}$ has length at least $|\pi|$. Now, given a circuit C', we claim that C' is isomorphic to C whenever (i) C' has a root r' labeled t, (ii) r' has two predecessor states s'_0 and s'_1 respectively roots of circuits isomorphic to C_0 and C_1, (iii) $E(C, r) = E(C', r')$. When defining $\theta_C(x_0, \ldots, x_d, w)$,

(i) is verified by: $(x_1, \ldots, x_d, w) \mapsto \tau_t(x_1, w)$,
(ii) is verified by:
$(x_1, \ldots, x_d, w) \mapsto \theta_{C_0}(x_1, \ldots, x_d, \varphi_0(x_0, w)) \wedge \theta_{C_1}(x_1, \ldots, x_d, \varphi_1(x_0, w))$,
(iii) is verified as follows. For all words π, π' of length smaller than d, if $(\pi, \pi') \in E(C, r)$, then: $(x_1, \ldots, x_d, w) \mapsto \varphi_\pi(x_1, \ldots, x_d, w) =_{\text{dom}} \varphi_{\pi'}(x_1, \ldots, x_d, w)$,
otherwise:$(x_1, \ldots, x_d, w) \mapsto \varphi_\pi(x_1, \ldots, x_d, w) \neq_{\text{dom}} \varphi_{\pi'}(x_1, \ldots, x_d, w)$,

where $\varphi_\pi(\bar{t}) =_{\text{dom}} \varphi_{\pi'}(\bar{t})$ iff either $\bar{t} \notin \text{dom}(\varphi_\pi) \wedge \bar{t} \notin \text{dom}(\varphi_{\pi'})$ or $\varphi_\pi(\bar{t}) = \varphi_{\pi'}(\bar{t})$. The three points (i), (ii), (iii) can be put all together by closure under intersection of length preserving rational relations.

To end the proof, observe that θ_C can be restricted to $\theta'_C(x_0, \ldots, x_d, w) \Leftrightarrow \theta(x_0, \ldots, x_d, w) \wedge x_0 = x_1 = \cdots = x_d$ which is a length preserving. Let $T'_C = (\Sigma, Q, q_0, F, \delta)$ be the corresponding letter-to-letter MONA. Then, θ is recognized by $(\Sigma, Q, q_0, F, \delta')$ with $(q, a, b, q') \in \delta'$ iff $(q, a, \ldots, a, b, q') \in \delta$. The trick

can be used for paths, and we let the MONA Ψ_π for all path π such that $[\Psi_\pi](\underline{n}, w) = \varphi_\pi(\underline{n}, \ldots, \underline{n}, w)$.

Actually, one may think of $\mathcal{C}^d$ as a (finite) function basis. Then, circuits of depth d on $\mathcal{B}_0$ can be seen as circuits of depth 1 on basis $\mathcal{C}^d$.

Proposition 6. *For any* RATIONAL*-uniform circuit family $(C_n)_{n>0}$ of constant depth d, there are MONA $(T_C)_{C\in\mathcal{C}^d}$ and MONA $\Phi_1, \ldots, \Phi_W$ such that for all $n > 0$, for all output gate w in C_n, $[T_C](\underline{n}\,, w)$ holds iff C is the index of w in C_n and $[\Phi_j](\underline{n}\,, w)$ outputs the address of the input gate in C_n corresponding to the j-th input gate in C, the index of w.*

Proof. The MONA $(T_C)_C$ have been constructed in the previous Lemma. The $W = 2^d$ MONA $\Phi_1, \ldots, \Phi_W$ are built as follows. Let the circuits $\mathcal{C}^d = (C_1, \ldots, C_r)$ and let $\mathrm{ar}(C)$ be the number of inputs of the circuit C. The MONA Φ_j is:

$$(\underline{n}\,, w) \mapsto \begin{cases} \texttt{if } T_{C_1}(\underline{n}\,, w) \wedge (j \leqslant \mathrm{ar}(C_1)) \texttt{ then } \Psi_{\pi_j(C_1)}(\underline{n}\,, w) \\ \texttt{else if } T_{C_2}(\underline{n}\,, w) \wedge (j \leqslant \mathrm{ar}(C_2)) \texttt{ then } \Psi_{\pi_j(C_2)}(\underline{n}\,, w) \\ \quad \ldots \\ \qquad \texttt{else } \bot \end{cases}$$

with Ψ_π be defined as in the previous Lemma.

4 Complexity hierarchy robustness

Definition 3 (Complexity classes). *For $k \in \mathbf{N}$, NC^k_{Rat} (resp. NC^k_{DLT} and NC^k_{ALT}) is the class of problems solvable by a* RATIONAL*-uniform (resp.* DLOGTIME*-uniform and* ALOGTIME*-uniform) family of circuit $(C_n)_{n>0}$ over basis $\mathcal{B}_0$, of respective depth $||C_n|| = O(log^k(n))$ and size $|C_n| = n^{O(1)}$ in the size of the input. We denote $NC = \cup_{k\in\mathbf{N}} NC^k_{DLT}$.*

Following Cook [10], NC is considered to be the class of problems that can be efficiently solved on a parallel computer. Proposition 8 proves that $\mathrm{NC}^k_{DLT} \subseteq \mathrm{NC}^{k+1}_{Rat}$ for all $k \geqslant 0$. Together with Theorem 3, we have the following hierarchy:

$$\mathrm{NC}^0_{Rat} \subseteq \mathrm{NC}^0_{DLT} \subseteq \mathrm{NC}^1_{Rat} \subseteq \mathrm{NC}^1_{DLT} \subseteq \cdots \subseteq \mathrm{NC}^k_{Rat} \subseteq \mathrm{NC}^k_{DLT} \subseteq \cdots \subseteq \mathrm{NC}$$

which implies that $\mathrm{NC}_{Rat} = \cup_{k\in\mathbf{N}} \mathrm{NC}^k_{Rat} = \mathrm{NC}$.

Lemma 1. *Given a* ALOGTIME*-uniform circuit family $(C_n)_n$, without loss of generality, one may suppose that the address of any gate is actually labeled by $t \cdot u$ where t denotes the type of the gate. We call such circuits explicitly typed circuits.*

Proof. Suppose that the circuit is not explicitly typed. Let M be the machine recognizing the connection language of $(C_n)_n$. Let $(C'_n)_n$ be exactly the same circuit family but with address $(t \cdot w)$ for any gate w of type t. It is also ALOGTIME-uniform. Indeed, verifying the type t of a gate w (y, w, ε, t) merely amounts to verify that t (whose size is bounded by some constant K) is the prefix of w so that the procedure takes constant time. Computing the i-th successor of a gate $(t \cdot u)$ is done as follows:

```
verify(y,w,i,u){
   //copies of types and address on some working tapes
   let (a,w') and (b,u') such that  a.w' = w and b.u' = u
   return (run M on (y,w',i,u')) and (run M on (y,u',epsilon,b))
}
```

Copying addresses of logarithmic size is done in logarithmic time. The copies of the types take constant time. Finally, the two last runs take logarithmic time.

4.1 Simulation of alternating random access Turing Machine

We recall some main facts about alternating random access Turing Machine (ARM) as described by Leivant and Marion [15], see also [9, 21]. A machine M is a tuple $\langle Q, q_0, \delta\rangle$ made of a set of states, an initial state and a transition function δ. States are classified as disjunctive or conjunctive (the action states), as accepting, rejecting or reading states. We recall that the operational semantics of the machine is a two step process: first generating the computation tree, second, evaluating the computation tree. A configuration is a tuple $(q, w_1, \ldots, w_k) \in Q \times \mathbf{B}^k$ of a state and k work stacks, the initial tuple being $(q_0, \varepsilon, \ldots, \varepsilon)$. A computation tree is a tree whose nodes are configurations and whose root is the initial configuration. For action states, depending on the state and on the bits at the top of the work stacks, one spawns a pair of successor configurations by pushing/popping letters on the work stacks. For a machine working in time $t(n)$, the tree is expanded up to depth $t(n)$. For the evaluation, the computations begin at the leaves of the tree. The output of accepting/rejecting states is respectively 1 and 0. The output of action state is given by the associated logical operator. Finally, since we will apply ARM on connection language, that is 4-tuples $(\underline{n}\,, w, b, u)$, to simplify reading, we suppose that the input is actually written as four independent words $t_0, \ldots, t_3$. To read the inputs, each word t_i, $i \leqslant 3$, is associated to a working tape indexed $\rho(i) \leqslant k$ and to a reading state $q_{\mathrm{read}(i)} \in Q$. The output of a configuration $(q_{\mathrm{read}(i)}, w_1, \ldots, w_k)$ is the $w_{\rho(i)}$-th bit of t_i.

To simulate ARM by means of Rational-uniform circuit family, we need the (rational) function `read` defined as follows. `read` takes inputs (w, u) with $w, u \in \{0,1\}^*$. On inputs of the shape $w = w' \cdot i$ with $i \in \{0,1\}$, `read` outputs $(w', \mathcal{H}_i(u))$, that is it removes the last bit i of w, and then apply transducer $\mathcal{H}_i$ on u, cf Example 1. Otherwise, that is if $w = \varepsilon$, $\mathtt{read}(w, u) = (w, u)$. By composition, `read` is clearly a rational function.

Given a circuit family $(C_n)_{n>0}$ whose direct connection language $\mathcal{L}$ is recognized by an ARM $M = \langle Q, q_0, \delta\rangle$ working in logarithmic time, given some 4-tuple (y, w, b, u), let us consider the circuits $C[M, \underline{n}\,, w, b, u]$ with $|y| = n$ as defined now. Gate addresses are tuples $(q, w_1, \cdots w_k, \underline{n}\,, w, b, u)$ made of a configuration of M and the content of the inputs of M. We call these addresses extended configurations. According to the state, we define the typing and the predecessor functions of $C[M, \underline{n}\,, w, b, u]$.

a) for any action state (of M), $(q, w_1, \cdots w_k, \underline{n}\ , w, b, u)$ has the corresponding logical type. Its two predecessors are the extended configuration $(q^i, w_1^i, \ldots, w_k^i, \underline{n}\ , w, b, u)$ for $i \in \{0, 1\}$, as given by the transition function.
b) Accepting/Rejecting states are typed 0 or 1.
c) for a reading state $q_{\mathrm{read}(i)}$, we build a chain of identity gates[3] with a last gate being a rejecting gate or an accepting one (actually depending on the bit read in t_i). The first identity gate has the address $(w_{\rho(i)}, t_i)$. Its predecessor is $\mathtt{read}(w_{\rho(i)}, t_i)$. After $|w_{\rho(i)}|$ application of the function `read`, we get a pair of words (ε, u) such that the last bit of u denotes the bit read at address $w_{\rho(i)}$ within t_i. The last gate of the gate is accepting or rejecting depending on this last bit.

Given the preceding description,

Proposition 7. *1. the circuit $C[M, \underline{n}\ , w, b, u]$ outputs 1 iff $(y, w, b, u) \in \mathcal{L}$,*
2. the circuit has depth $O(\max(|n|, |w|, |b|, |u|))$ and polynomial size,
3. the typing function, the two predecessor functions are rational.

Proof. 1. The simulation of action states, accepting states and rejecting states directly follows from the operational semantics. The simulation of reading state, though done in $O(\log(n))$ steps output the desired result.
2. Since w_j is a working tape, it has size $O(\log(n))$. The chain of identity gate having the same length, the depth is $O(\log(n))$. For the size, one observes that all words have length $O(\log(n))$. The result follows by enumeration.
3. Typing is immediate from the classification of states. For predecessor functions, for action states, since the next configuration depend only on the finite control of the top bits, computing the predecessor configurations is clearly rational. For reading states, computing (w_j, t_i) from $(q_{\mathrm{read}(i)}, w_0, \ldots, w_k, t_0, \ldots, t_3)$ is also rational and we have seen above that `read` is rational.

4.2 From DLogTime-uniformity to Rational-uniformity

Proposition 8. *For all $k \geqslant 0$, $NC^k_{DLT} \subseteq NC^{k+1}_{Rat}$.*

Proof. Actually, we prove the even more stronger result, that is $\mathrm{NC}^k_{ALT} \subseteq \mathrm{NC}^{k+1}_{Rat}$. Indeed, since DLogTime $\subseteq$ ALogTime, DLogTime-uniformity is a priori sharper than ALogTime-uniformity. Actually, the two notions coincide for $k \geqslant 1$ as shown by Barrington, Immerman and Straubing in [3]. So, consider a function computed by an ALogTime-uniform, explicitly typed family of circuits, $(C_n)_{n>0}$ of polynomial size and depth bounded by $c \times \log^k(n)$. Let M be the machine running in ALogTime which decides the direct connection language of $(C_n)_{n>0}$. We begin by some preliminary observations.

First, there is a constant K such that the machine M runs in time $K \times \log(n)$ and space $K \times \log(n)$ where n denotes the size of the input. Then accordingly, the size of any gate addresses in the circuit C_n is bounded by $K \times \log(n)$. So by a padding argument, one may suppose without loss of generality that every gate address (in C_n) has actually exactly the size $C \times \log(n)$ for any $n > 0$.

[3] On the basis $\mathcal{B}_0$, it is obtained as the composition of two gates: $x \mapsto x \vee 0$ for instance.

Second, circuits $(C[M, \underline{n}, w, b, u])_{\underline{n}, w, b, u}$ described for Proposition 7 have logarithmic depth whenever w, b, u have logarithmic size (in n).

We define now a RATIONAL-uniform circuit family $(C'_n)_n$ such that C'_i simulates C_i for all $i > 0$. The principle of our simulation is the following. To any gate w in C_n, we associate a corresponding gate w in C'_n with same logical operator. In a first step, we guess the addresses u_0 and u_1 of both successor gates of w. Given u_i, we verify in parallel that u_i is actually the i-th successor of w ($i \in \{0, 1\}$) by means of $C[M, \underline{n}, w, i, u_i]$ and we build recursively the circuit corresponding to the gate u_i. The process ends on input gates.

Let us describe the construction in more details. First, we suppose that the circuit family is typed (as justified by Lemma 1). Any address $(t \cdot w)$ corresponding to a logical gate of type t in C_n is mapped to a gate $(\underline{n}, (t \cdot w))$ in C'_n. From that gate, we build a small circuit as follows. Let A be the number of successors of w (one for the negation and two for the binary operators). $(\underline{n}, (t \cdot w))$ has A $\vee$-gate successors (t1) addressed $(\underline{n}, (t \cdot w), i, \varepsilon)$ with $0 \leqslant i < A$. Any such gate $(\underline{n}, t \cdot w, i, \varepsilon)$ is the root of the full binary tree of gates (t2) of depth $|w| = C \times \log(n)$ with leaves addressed $(\underline{n}, t \cdot w, i, u)$. Internal nodes of the binary tree are $\vee$-gates, the leaves having type $\wedge$.

The first successor (t3) of a gate $(\underline{n}, t \cdot w, i, u)$ is $C[M, \underline{n}, t \cdot w, i, u]$. The second successor (t4) is the gate corresponding to u itself (here the construction will recursively continue). Finally, any input gate within C_n is an input gate within C'_n.

It is clear that the family is RATIONAL-uniform: constructions (t1) and (t2) are obtained by concatenation of finitely many letters to the current address, (t3) refers to Proposition 7, and (t4) consists in erasing the arguments $t \cdot w$ and i.

The length of the path between gate w and gate u is $C \times \log(n) + 1$ (the depth of the binary tree (t2)). Thus, the depth $||w||$ of the circuit from gate w is

$$||w|| = \max(C \times \log(n) + ||u||, C \times \log(n) + K \times \log(n)) \quad (1)$$

$$\leqslant ||u|| + (C + K) \times \log(n). \quad (2)$$

where in Equation 1, the max function comes from the parallel choice (done at each leaf of (t2)) for the verifier (t3) as given by Proposition 7 and the depth of the circuit from that gate. Since the depth of the circuit C_n is bounded by $c \times \log^k(n)$, using recursively Equation 2, it follows that the depth of the circuit C'_n is bounded by $c \times \log^k(n) \times (C + K) \times \log(n) = O(\log^{k+1}(n))$. Remark that since all the addresses in circuit C'_n have size bounded by $O(\log(n))$, the size of the circuit C'_n is bounded by $n^{O(1)}$.

5 Separation theorems

We present classical separation proofs for the DLOGTIME-uniform hierarchy. We stress that these proofs are not restricted to the case of uniform circuit family!

Definition 4. *We define the standard following decision problems. Instance: A binary word* $w \in \{0,1\}^*$. *PARITY: Is the number of 1's in* w *odd? MAJORITY: Are the majority of input bits 1? PALINDROME: Are bits at positions* i *and* $|w|-i$ *equals, for all* $1 \leqslant i \leqslant |w|$*? OR: Is the disjunction over bits of* w *equals to 1? HALF: Is the middle bit of* w *equals to 1? The restriction of the previous problems to an instance of binary word with fixed length* n *are denoted respectively* $\mathtt{PARITY}^n$, $\mathtt{MAJORITY}^n$, $\mathtt{PALINDROME}^n$ *and* $\mathtt{HALF}^n$.

Lemma 2. *PARITY, MAJORITY, OR and PALINDROME are not in* NC^0_{Rat} *but in* NC^1_{Rat}.

Proof. Given an input of size n, all of these problems require circuit depth at least $\Omega(\log(n))$, just so all the input bits could effect the output gate. Thus no one belongs to NC^0_{Rat} (as well as NC^0_{DLT}). For NC^1_{Rat} membership, we have seen how to compute $\mathtt{OR}^n$ for a given n in Example 4. PARITY uses the same schema but with w-gates typed XOR (those gates being simulated by x `xor` $y = (x \vee \neg y) \wedge (\neg x \vee y)$ on basis $\mathcal{B}_0$) and s-gates typed 0. To compute PALINDROME, we take again the same schema, but with w-gates being typed $\wedge$, s-gates being typed 1 and input gates addressed $\mathbf{i} \cdot u$ are replaced by a XOR-gate addressed $\mathtt{c} \cdot u$ with two predecessors being $\mathbf{i} \cdot u$ and $\mathbf{i} \cdot (n-u-1)$. Since subtraction is rational, this latter address being obtained by equation $\varphi_1(\underline{n}\ , \mathtt{c} \cdot u) = \mathbf{i} \cdot \underline{n-u-1}$, the predecessor function φ_1 remains rational. A standard proof of membership for $\mathtt{MAJORITY}^n$ given n, is to repeatedly reduce the addition of three numbers to the addition of two numbers, computing carries without propagate them by using bitwise XOR gates. After $O(\log(n))$ depth, the resulting two $O(\log(n))$-bit numbers have to be compared with $n/2$. All of these computations may be done within RATIONAL-uniformity.

Lemma 3. $\mathtt{HALF} \in NC^0_{Rat}$.

Proof. Given n the middle bit address can be computed by a MONA: consider the circuit made of a unique output gate, with a predecessor function φ_0 defined only on the input: $\varphi(\underline{n}\ , \mathbf{o} \cdot 0^{|\underline{n}|}) = \underline{(n-1)/2}$. Subtraction is rational, division by 2 (a shift of the bits of $\underline{n-1}$) is rational, thus φ_0 is rational. φ_1 is the empty function, and typing is immediate.

We denote REG the class of decision problems solvable by deterministic finite automata. It is well known that REG = DSPACE$(O(1))$, the decision problems that can be solved in constant space [23].

Lemma 4. *PARITY, OR* $\in$ REG *and HALF, PALINDROME* $\notin$ REG.

Proof. As it is uncommon, we present that the language relative to the function HALF is not regular. Ad absurdum, by Pumping Lemma, suppose it is so. Let N be the bound above which $w \in \mathcal{L}(\mathtt{HALF})$ implies $w = w_1 \cdot u \cdot w_2$ with $|u| > 0$ and $w_1 \cdot u^k \cdot w_2 \in \mathcal{L}(\mathtt{HALF})$ for all $k \geqslant 0$. Let us apply it on $w = 0^N \cdot 1 \cdot 0^N$. Let the decomposition $w = w_1 \cdot u \cdot w_2$ as above. Either u contains the 1, but then $w_1 \cdot w_2$ contains no 1 and thus $w_1 \cdot w_2 \notin \mathcal{L}(\mathtt{HALF})$ in contradiction with the Lemma. Or,

$w_1 = 0^m$, $u = 0^j$ and $w_2 = 0^{N-m-j}10^N$ is the remaining word with $j > 0$ since $|u| > 0$. But, in that case, $w_1 \cdot u^2 \cdot w_2 = 0^{N+2j} \cdot 1 \cdot 0^N$ has no 1 in the middle, contradicting the Lemma. The last case, $w_1 = 0^N \cdot 1 \cdot N^k$, $u = 0^j$, $w_2 = 0^{N-j-k}$ is symmetric to the preceding one.

Theorem 4. REG $\neq NC^0_{Rat}$ *and* $NC^0_{DLT} \subsetneq NC^1_{Rat}$.

Proof. Inequality arises from previous lemmas: we have $\texttt{OR} \notin \mathrm{NC}^0_{Rat} \ni \texttt{HALF}$ and $\texttt{OR} \in \textsc{Reg} \not\ni \texttt{HALF}$. Finally $\mathrm{NC}^0_{DLT} \subsetneq \mathrm{NC}^1_{Rat}$ as $\texttt{OR} \notin \mathrm{NC}^0_{DLT}$ and $\texttt{OR} \in \mathrm{NC}^1_{Rat}$ by lemma 2.

Theorem 5. $NC^0_{Rat} \subsetneq NC^0_{DLT}$

Proof. Let $\tau : \{0,1\}^* \to \{0,1\}$ such that $\tau(w)$ is the $\log(|w|)$-th bit of w. Such a function is computable within NC^0_{DLT}. Given $n \in \mathbf{N}$, consider the circuit made of one output $\vee$ gate addressed 0 whose two successor gates are the input gate $\log(n)$. Typing is trivially DLOGTIME-uniform. To verify that the successors of the output gate can be computed in DLOGTIME, we have to verify that the relation $\{(y, 0, b, u) \mid b \in \{0,1\} \wedge u = \log(|y|)\}$ is computable in DLOGTIME. From Buss [8], for any word y of length n, for any boolean $b \in \{0,1\}$, computing $\log(|y|)$ from $(y, 0, b, u)$ is done in logarithmic time. Since $\log(|y|)$ has logarithmic size with respect to $(y, 0, b, u)$, the equality of $\log(|y|)$ with u is performed in logarithmic time.

We establish now ad absurdum that the predicate τ cannot be computed by a RATIONAL-uniform circuit family of constant depth. We propose a proof with an encoding based on size parameters $A = 1$ and $B = 0$. The proof can be extended to other cases. Suppose the existence of such a family $(C_n)_n$ computing τ. Let D be its depth.

Fact 1. There is a path from the output gate addressed $\mathbf{o} \cdot 0^n$ of the circuit C_n to the input gate addressed $\mathbf{i} \cdot \underline{\log(n)}$. Otherwise, the output of the circuit $C_n[w]$ on some word w would not depend on its $\log(n)$-th bit, that is $w_{\log(n)}$, which contradicts the definition of τ.

Fact 2. Without loss of generality, one may suppose that $w_{\log(n)}$ is actually the only input bit read on the input by the circuit C_n. Indeed, it is clear that τ does not depend on other bits, so that other input gates can be replaced by a (for instance) rejecting gate.

Fact 3. According to Proposition 6, there is a MONA Φ_1 such that Φ_1 outputs $\mathbf{i} \cdot \underline{\log(n)}_{|n|}$ on input $(\underline{n}, \mathbf{o} \cdot 0^{|\underline{n}|})$.

As we will see, the set of binary words $A = \{\mathbf{i} \cdot \underline{\log(n)}_{|n|} \mid n > 0\}$ is not regular. But the projection of a rational relation is a regular language (see Proposition 1). Since $\{u \mid (\mathbf{B}^*, \mathbf{o} \cdot 0^*, u) \in [\Phi_1]\} = A$, there is a contradiction.

Let us suppose that A is regular. Consider a transducer $[D]$ that removes the first letter of a word, then, according to Proposition 1, its range $A' = [D](A) = \{\underline{\log(n)}_{|n|} \mid n > 0\}$ is a regular language. We actually prove that A' is not regular. If it were the case, let L be the bound as given by the Pumping Lemma. Consider the smallest k such that $n = 2^{2^k}$ verifies $|\underline{n}| = 2^k + 1 > L$. The

word $\underline{\log(n)}_{|n|}$ has the shape $0^{2^k-k} \cdot 1 \cdot 0^k$, thus has length $2^k + 1$ Then, since, $\underline{\log(n)}_{|n|} \in A'$, there is a decomposition $\underline{\log(n)}_{|n|} = u \cdot v \cdot w$ such that for all $t \geqslant 0$, $u \cdot v^t \cdot w \in A'$. There are three cases, v is at the left or right of the 1, or it contains the 1. The reader can easily check the contradiction, that is, in the first cases $|w_t|$ is opposed to membership in A', and the last case is a simple computation.

6 Unbounded fan-in circuits

We provide same kind of results for RATIONAL-uniform circuit families, called with unbounded fan-in, over the basis $\mathcal{B}_1 = \{(\wedge^n)_{n\geqslant 2}, (\vee^n)_{n\geqslant 2}, \neg, 0, 1, \mathbf{i}, \mathbf{o}\}$.

Again we suppose that circuits have polynomial size, so there is a polynomial P such that $|C_n| \leqslant P(n)$ for all $n > 0$. Then each gate of C_n has at most $P(n)$ predecessors and we consider its k-th predecessor where we encode k also in less than $|P(n)|$ bits. We choose to describe gates over the basis $\mathcal{B}_1$ by giving its type in $\{\vee, \wedge, \mathbf{i}, \mathbf{o}, \neg\}$ and by describing all of its predecessors: this is exactly our RATIONAL-uniform definition 2. Since all the inputs have a common length, again, the MONAs can be considered to be letter-to-letter.

We define AC^k_{Rat} for $k \in \mathbf{N}$ exactly as NC^k_{Rat} but over basis $\mathcal{B}_1$. That is, for $k \in \mathbf{N}$ AC^k_{Rat} is the class of problems solvable by a RATIONAL-uniform family of circuit $(C_n)_{n>0}$ over basis $\mathcal{B}_1$, of respective depth $||C_n|| = O(log^k(n))$ and size $|C_n| = n^{O(1)}$ in the size of the input. Then by theorem 3, we have for all $k \geqslant 0$, $\mathrm{AC}^k_{Rat} \subseteq \mathrm{AC}^k_{DLT}$ but also the following.

Proposition 9. *For all $k \geqslant 0$, we have $NC^k_{Rat} \subseteq AC^k_{Rat} \subseteq NC^{k+1}_{Rat}$.*

Proof. First inclusion trivially comes from RATIONAL-uniform definition by describing AC^k_{Rat} with the same MONA as NC^k_{Rat}. Second inclusion is described as follows. Consider a circuit family $(C_n)_n$ in AC^k_{Rat}. There are letter-to-letter MONAs T and Φ computing τ and φ. Any gate w of C_n which is in $\mathcal{B}_0$ is left unchanged, any other gate w, say a $\wedge^n$, is replaced by a binary tree of gates $\wedge$ defined as follows. The root of the tree is $(q_0, \wedge, w, \varepsilon, \varepsilon)$ with q_0 being the initial state of Φ. Any state $(q, \wedge, a \cdot w', k', u')$ has type $\wedge$ and successors $(q', \wedge, w', b \cdot k', c \cdot u')$ for all $(q, a, b, c, q') \in \delta_\Phi$ with δ_Φ the transition relation of Φ. Since δ_Φ is finite, this is easily encoded in basis $\mathcal{B}_0$. Any state $(q, \wedge, \varepsilon, k, u)$ has type 1 if $q \notin F_\Phi$, the set of final states of Φ. Otherwise, the state has the type of u in C_n. From the construction, observe that u is the address of the k-th predecessor of w in C_n. Then, removing q, ε, k from the address by a MONA is easy and the definition continues from u. Since, the MONA Φ is letter-to-letter, the length of the path between gate w and gate u is $|w|$, that is $O(\log(n))$. Thus, the depth of the simulating circuit is $O(\log(n)^k) \times O(\log(n)) = O(\log(n)^{k+1})$. Since the addressing system simulates a running MONA, the construction is rational.

Proposition 10. *For all $k \geqslant 0$, we have $AC^k_{DLT} \subseteq NC^{k+1}_{Rat}$.*

Proof (Sketch). We revisit the proof of proposition 8 only for the three following points. 1) We consider a function computed by an ALogTime-uniform explicitly typed family of circuits $(C_n)_n$ of polynomial size and depth bounded by $c \times \log^k(n)$, but over $\mathcal{B}_1$. Then given n, each gate has at most $\log(n^{O(1)}) = O(\log(n))$ predecessors. 2) We will use the circuits $(C[M, \underline{n}, w, b, u])_{\underline{n}, w, b, u}$ described in Proposition 7 which have logarithmic depth whenever w, b, u have logarithmic size (in n). In Proposition 8, b was a constant, here, it has logarithmic size. But the proof of Proposition 7 remains valid in the present case. 3) When simulating a gate w, instead of spanning a full binary tree to find the predecessor u corresponding to w, we span a full binary tree whose leaves are pairs $(k, u) \in \Sigma^{A\times\log(n)+B} \times \Sigma^{A\times\log(n)+B}$. Such addresses having size $O(\log(n))$, the depth of the tree remains $O(\log(n))$ as in Proposition 8. At each leaf, instead of verifying that $(\underline{n}, t \cdot w, i, u) \in C[M, \underline{n}, w, i, u]$, one verifies that $(\underline{n}, t \cdot w, k, u) \in C[M, \underline{n}, w, k, u]$. The depth and the size bound are computed as in Proposition 8.

Lemma 5. $\mathit{PARITY} \notin AC^0_{Rat}$.

Proof. Standard and very nice proofs for $\texttt{PARITY} \notin \mathrm{AC}^0_{DLT}$ are valid for even non-uniform circuit family. E.g. they are based on the fact that every constant depth circuit computing $\texttt{PARITY}^n$ must have sub-exponential size (i.e. $exp(\Omega(n^{\frac{1}{d-1}}))$ for depth d circuit) [11, 1, 12]. This applies also with Rational-uniformity!

Theorem 6. $\textsc{Reg} \neq AC^0_{Rat} \subsetneq NC^1_{Rat}$

Proof. By proposition 9 for $k = 0$, we have $\mathrm{NC}^0_{Rat} \subseteq \mathrm{AC}^0_{Rat} \subseteq \mathrm{NC}^1_{Rat}$. By lemmas 4 and 3 , we have $\texttt{HALF}$ is not in Reg but in NC^0_{Rat}, thus in AC^0_{Rat}. Then $\mathrm{AC}^0_{Rat} \not\subseteq \textsc{Reg}$. By lemmas 2, 4 and 5, $\texttt{PARITY}$ is not in AC^0_{Rat} but in $\textsc{Reg} \cap \mathrm{NC}^1_{Rat}$. Then we have both $\textsc{Reg} \not\subseteq \mathrm{AC}^0_{Rat}$ and $\mathrm{NC}^1_{Rat} \not\subseteq \mathrm{AC}^0_{Rat}$, and the result holds.

7 Conclusion

Common questions remain to study w.r.t P and NP. Indeed our uniformity notion is a weak external computational power keeping the internal computational power for separation approaches. E.g. as $\texttt{MAJORITY} \in \mathrm{NC}^1_{Rat}$ we have for Constant-Depth Threshold Circuits $TC^0_{Rat} \subseteq \mathrm{NC}^1_{Rat}$, and one could ask whether $TC^0_{Rat} = \mathrm{NC}^1_{REG}$, or whether $TC^0 = NP$.

Finally, the relationship between Reg and AC^0 is a long standing issue. We refer to the work of [7] for a presentation of regular languages in NC^1. It is almost clear that $\textsc{Reg} \subseteq \mathrm{NC}^1_{Rat}$, but the question is renewed due to the fact that $\mathrm{AC}^0_{Rat} \neq \mathrm{AC}^0_{DLT}$ using again the argument of the $\texttt{HALF}$ function. Observe that $\mathrm{AC}^0_{DLT} \ni \texttt{HALF} \notin \mathrm{AC}^0_{Rat}$ and $\texttt{HALF} \notin \texttt{Rat}$ which could mean that AC^0_{Rat} could be the "correct" set to deal with Reg. In this branch of research, the relation between Reg and ACC^0_{Rat} should be also reconsidered since $\texttt{HALF} \notin \mathrm{AC}^0_{Rat}$.

References

1. M. Ajtai. $\sum_1^1$-formulae on finite structures. *APAL*, 24(1):48, 1983.
2. D. Barrington, K. Compton, H. Straubing, and D. Thérien. Regular languages in NC^1. *J. of computer and system sciences*, 44:478–499, 1992.
3. D. A. Mix Barrington, N. Immerman, and H. Straubing. On uniformity within NC^1. *Journal of Computer and System Sciences*, 41:274–306, 1990.
4. S. Bellantoni and I. Oitavem. Separating NC along the δ axis. *TCS*, 318:57–78, 2004.
5. G. Bonfante, R. Kahle, J.-Y. Marion, and I. Oitavem. Recursion Schemata for NC^k. In *CSL '08*, volume 5213 of *LNCS*. Springer, 2008.
6. A. Borodin. On relating time and space to size and depth. *SIAM Journal on Computing*, 6(4):733–744, 1977.
7. S. Buss, S. Cook, A. Gupta, V. Ramachandran, and Over Ac. An optimal parallel algorithm for formula evaluation. *SIAM J. Comput*, 21:755–780, 1990.
8. S. R. Buss. The boolean formula value problem is in alogtime. In *in Proceedings of the 19-th Annual ACM Symposium on Theory of Computing*, pages 123–131, 1987.
9. A. Chandra, D. Kožen, and L. Stockmeyer. Alternation. *J. ACM*, 28:114–133, 1981.
10. Stephen A. Cook. A taxonomy of problems with fast parallel algorithms. *Inf. Control*, 64(1-3):2–22, March 1985.
11. M. L. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.
12. J. Håstad. Almost optimal lower bounds for small depth circuits. In J. Hartmanis, editor, *STOC*, pages 6–20. ACM, 1986.
13. R. M. Kaplan and M. Kay. Regular models of phonological rule systems. *Computational Linguistics*, 20:331–378, 1994.
14. D. Leivant. A characterization of NC by tree recurrence. In *Foundations of Computer Science 1998*, pages 716–724. IEEE Computer Society, 1998.
15. D. Leivant and J.-Y. Marion. A characterization of alternating log time by ramified recurrence. *Theoretical Computer Science*, 236(1–2):192–208, 2000.
16. V. Mogbil and V. Rahli. Uniform circuits, & boolean proof nets. In *LFCS*, volume 4514 of *LNCS*, pages 401–421. Springer, 2007.
17. M. Nivat. Transduction des langages de chomsky. *Ann. Inst. Fourier*, 18:339–456, 1968.
18. I. Oitavem. Characterizing NC with tier 0 pointers. *Math. Logic Quarterly*, 50:9–17, 2004.
19. M. O. Rabin and D. Scott. Finite automata and their decision problems. *IBM J. Res. Dev.*, 3(2):114–125, April 1959.
20. G. Rozenberg and A. Salomaa. *Handbook of formal languages.* Springer-Verlag, 1991.
21. W. L. Ruzzo. On uniform circuit complexity. *J. of Comp. and Syst. Sci.*, 22:365–383, 1981.
22. J. Sakarovitch. *Elements of Automata Theory.* Cambridge University Press, 2009.
23. J. C. Shepherdson. The reduction of two-way automata to one-way automata. *IBM J. Res. Dev.*, 3(2):198–200, April 1959.
24. K. Terui. Proof nets and boolean circuits. In *LICS*, pages 182–191. IEEE, 2004.
25. H. Vollmer. *Introduction to Circuit Complexity: A Uniform Approach.* Springer, 1999.