***Laboratoire de l'Informatique du Parallélisme***

École Normale Supérieure de Lyon
Unité Mixte de Recherche CNRS-INRIA-ENS LYON-UCBL n$^o$ 5668

# *Characterizing Valiant's algebraic complexity classes*

Guillaume Malod and Natacha Portier

Septembre 2005

Rapport de recherche N$^o$ 05-44

**École Normale Supérieure de Lyon**
46 Allée d'Italie, 69364 Lyon Cedex 07, France
Téléphone : +33(0)4.72.72.80.37
Télécopieur : +33(0)4.72.72.80.80
Adresse électronique : `lip@ens-lyon.fr`

CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

INRIA

# Characterizing Valiant's algebraic complexity classes

Guillaume Malod and Natacha Portier

Septembre 2005

### Abstract

Valiant introduced 20 years ago $VP$ and $VNP$: an algebraic analogue to the classes $P$ and $NP$. They are defined via non-uniform sequences of arithmetic circuits and are especially useful to study the complexity of polynomials families. In this paper we gather known results and new techniques under a unifying theme, namely the restrictions imposed upon the gates of the circuit, building a hierarchy from formulas to circuits. As a consequence we get a simpler proofs for known results such as the equality of the classes $VNP$ and $VNP_e$, the completeness of the determinant for $VQP$ or the equivalence of skew and weakly skew arithmetic circuits, and new results such as a characterisation of the classes $VP$ and $VQP$ or a full answer to a conjecture of Bürgisser. We also show that for circuits of polynomial depth and unbounded size these models all have the same expressive power and can be used to characterize a uniform version of $VNP$.

**Keywords:** Valiant's theory, algebraic complexity, arithmetic circuits

### Résumé

Il y a 20 ans, L. Valiant a défini des équivalents algébriques des classes $P$ et $NP$ : $VP$ et $VNP$. Ces classes sont définies via des suites non uniformes de circuits arithmétiques et sont particulièrement utiles dans l'étude de la complexité des suites de polynômes. Dans cet article nous utilisons des restrictions sur la structure des circuits pour présenter de manière unifiée des résultats connus ainsi que de nouvelles techniques et résultats.

**Mots-clés:** théorie de Valiant, complexité algébrique, circuits arithmétiques

# CHARACTERIZING VALIANT'S ALGEBRAIC COMPLEXITY CLASSES

GUILLAUME MALOD AND NATACHA PORTIER

Abstract. Valiant introduced 20 years ago an algebraic analogue to the classes P and NP. The classes VP and VNP are defined via non-uniform sequences of arithmetic circuits and are especially useful to study the complexity of polynomial families. In this paper we gather known results and new techniques under a unifying theme, namely the restrictions imposed upon the gates of the circuit, building a hierarchy from formulas to circuits. As a consequence we get a simpler proofs for known results such as the equality of the classes VNP and $\text{VNP}_\text{e}$, the completeness of the determinant for VQP or the equivalence of skew and weakly skew arithmetic circuits, and new results such as a characterization of the classes VP and VQP or a full answer to a conjecture of Bürgisser [Bür00]. We also show that for circuits of polynomial depth and unbounded size these models all have the same expressive power and can be used to characterize a uniform version of VNP.

## 1. Introduction

Circuits play an important part in complexity theory. Boolean circuits have been used to characterize several of the most important complexity classes. Arithmetic circuits provide an efficient representation of polynomials, and results by Kaltofen [Kal86] and von zur Gathen [vzG87] show that standard symbolic manipulations can be applied to these polynomials. Arithmetic circuits have also surfaced in recent results. Kabanets and Impagliazzo [KI03] link the de-randomization of Polynomial Identity Testing with super-polynomial arithmetic circuit lower bounds for the Permanent. Arithmetic circuits can also be Boolean inputs and define new problems with interesting consequences in complexity, as is shown by [ABKPM04], where the problem of deciding whether an arithmetic circuit computes a positive number is related to numerical analysis.

In this paper we will focus on the classes introduced by Valiant [Val79, Val82] and defined via arithmetic circuits. Their definition is very simple so that the combinatorial insights are unfettered by computational details (definitions are given in section 2). Moreover the reductions used are low level ($p$-projections, as used for example in [IL95]), thus retaining the algebraic character of problems. The completeness of the Permanent for the class VNP, the most famous result in this theory, is comparable to its completeness for $\sharp$P, yet avoids having to deal with integers or Boolean matrices and encodings. It shows the completeness of the permanent in a very strong sense.

We stress this combinatorial aspect by introducing restrictions on circuits to give a characterization of the class VP (theorem 1) and of the class VQP (theorem 3), defined to capture the complexity of the Determinant. The characterization of VP yields a simple and intuitive proof of one the main steps in the completeness proof of the Permanent, namely the equivalence of circuits and formulas under a Boolean sum. The characterization

*Date*: September 19, 2005.

1991 *Mathematics Subject Classification.* 03D15,68Q17.

*Key words and phrases.* algebraic complexity, Valiant's theory, polynomials, straight line programs, circuits, Permanent, Determinant, skew circuits, weakly skew circuits, formulas.

of VQP yields a full answer to a conjecture by Bürgisser [Bür00] stating that several operations of linear algebra are VQP-complete (theorem 6). The techniques used to study the class VQP are in fact similar to those used by Toda [Tod92]. We import his definition of a class capturing the complexity of the Determinant and suggest that it is better suited to the task than VQP. We show that by defining this class via weakly skew circuits, following a different order and proving a generic universality lemma we can get simpler proofs, for instance for the equivalence of skew and weakly skew circuits. We finally use similar circuit techniques to characterize a uniform version of VNP (theorem 10). We try to give the intuition behind the proofs, while technical details are omitted or left to the appendix.

## 2. Basic definitions

Valiant's complexity classes revolve around the representation of polynomials over a given field by arithmetic circuits. These polynomials are abstract, in the sense that they are defined by the sequence of their coefficients. One should remember to distinguish polynomials in this sense from polynomial functions, which are the functions defined by polynomials over a field. More details about Valiant's theory can be found in [vzG87] and [Bür00].

**Definition 1.** An *arithmetic circuit* is a finite acyclic directed graph with vertices of in-degree 0 or 2 and exactly one vertex of out-degree 0. Vertices of in-degree 0 are called *inputs* and labeled by a constant or a variable. The other vertices, of in-degree 2, are labeled by $\times$ or $+$ and called computation gates. We distinguish left and right arguments to a computation gate (i.e. our graph is implicitly labelled on the arrows). The vertex of out-degree 0 is called the *output*. The vertices of a circuit are commonly called *gates* and its edges *arrows*.

The polynomial represented by a circuit can easily be defined by induction. Circuits represent a computation where one can reuse partial results. If we do not allow this, that is if we require each argument to be computed especially for a given computation step, then the graph underlying the circuit must be a tree. Such circuits are called *expressions*, *arithmetic terms* or *formulas* (we shall use the latter).

**Definition 2.** The *size* of a circuit is its number of gates. The *depth* is the maximal length of a directed path from an input to an output. The *degree* of a gate is defined recursively: any input is of degree 1; the degree of a $+$ gate is the max of the incoming degrees; the degree of a $\times$ gate is the sum of the incoming degrees. The degree of the circuit is the degree of its output gate.

As usual in complexity theory we are interested in asymptotics, in this case the growth of the size of the circuits representing a sequence of polynomials. We give here the definitions of Valiant's classes and the reductions used. Note that the classes depend on a chosen field, but as we are interested in combinatorial techniques this will almost never play a role in this paper.

**Definition 3.** A sequence of polynomials $(f_n)$ belongs to VP if there exists a sequence of circuits $C_n$ of polynomially bounded size and degree such that $C_n$ represents $f_n$.
A sequence of polynomials $(f_n)$ belongs to VNP if there exists a polynomial $p$ and a sequence $g_n \in$ VP such that $f_n(\bar{x}) = \sum_{\bar{\epsilon}\in\{0,1\}^{p(|\bar{x}|)}} g_n(\bar{x}, \bar{\epsilon})$.
A polynomial $f$ is a *projection* of a polynomial $g$ if $f(\bar{x}) = g(a_1, \ldots, a_m)$, where the $a_i$ are elements of the field or variables among $x_1, \ldots, x_n$. A sequence $(f_n)$ is a *p-projection*

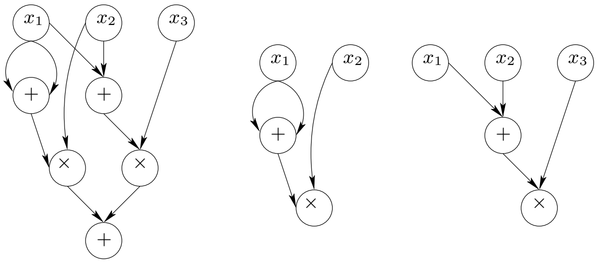

Figure 1. A multiplicatively disjoint circuit

of a sequence $(g_n)$ if there exists a polynomially bounded function $t(n)$ such that $f_n$ is a projection of $g_{t(n)}$ for all $n$.

It is obvious that VP is included in VNP. Valiant's hypothesis is that this inclusion is strict; it remains a major open problem of complexity theory. The definition of VP given here bounds both the degree and the size of the circuit representing a polynomial. The definition in [Bür00] bounds the degree of the represented polynomial and the size of the circuit. One can show that these definitions are equivalent. The following classes are defined using formulas in place of circuits and play an important part in the completeness proof of the permanent.

**Definition 4.** A sequence of polynomials $(f_n)$ belongs to the class $\mathrm{VP_e}$ if there exists a sequence of formulas $F_n$ of polynomially bounded size such that $F_n$ represents $f_n$.
A sequence of polynomials $(f_n)$ belongs to $\mathrm{VNP_e}$ if there exists a polynomial $p$ and a sequence $g_n \in \mathrm{VP_e}$ such that $f_n(\bar{x}) = \sum_{\bar{\epsilon}\in\{0,1\}^{p(|\bar{x}|)}} g_n(\bar{x}, \bar{\epsilon})$.

The main result in Valiant's theory is the completeness of the Permanent family of polynomials for the class VNP, over fields of characteristic different from 2. The permanent of a matrix of size $n$ with variables entries $z_{i,j}$ is defined as $\mathrm{PER}_n(z_{i,j}) = \sum_{\sigma\in S_n} \prod_{i=1}^n z_{i,\sigma(i)}$. In this definition, $S_n$ is the group of permutations of $\{1, \ldots, n\}$. This result stands in stark contrast to the fact that the Determinant family belongs to the class VP. The determinant is defined as the permanent but with positive and negative monomials depending on the sign $s(\sigma)$ of the permutation: $\mathrm{DET}_n(z_{i,j}) = \sum_{\sigma\in S_n} s(\sigma) \prod_{i=1}^n z_{i,\sigma(i)}$.

## 3. Characterizing VP

Whereas the class VNP captures the complexity of the Permanent and many other problem, there is no natural complete problem for the class VP, which is still not very well understood. We give here an intuitive characterization which we hope may provide better insight. For this purpose we introduce the following definition, exploiting the interplay between circuits and formulas in Valiant's theory.

**Definition 5.** Let $\alpha$ be a gate receiving arrows from gates $\beta$ and $\gamma$. We say that $\alpha$ is *disjoint* if the sub-circuits associated to $\beta$ and $\gamma$ are disjoint from one another. A circuit is *multiplicatively disjoint* (MD) if all its multiplication gates are disjoint.

The circuit in figure 1 is multiplicatively disjoint, as shown by the depiction of its multiplication gates and their respective sub-circuits. One can see MD circuits as intermediate between formulas and circuits. A circuit is a formula if and only if all its gates are disjoint. A multiplicatively disjoint circuit behaves like a formula for multiplications. Disjoint multiplications can be seen as a way to control the degree of the polynomial computed by a circuit, which links this technique to the retarded multiplication scheme used in [BF91]

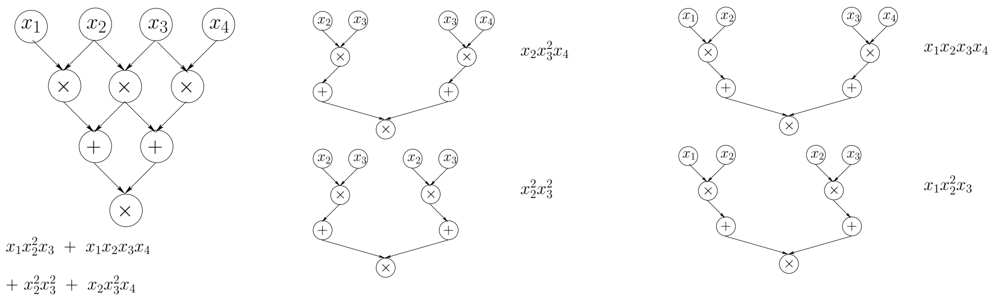

Figure 2. A circuit and its parse trees

to characterize the class ♯P. However it also provides combinatorial information which we will use in the next section. Let us now show that multiplicatively disjoint circuits enable us to characterize VP.

**Theorem 1.** *A sequence of polynomials* $(f_n)$ *belongs to* VP *if and only if there exists a sequence* $(C_n)$ *of multiplicatively disjoint circuits, of polynomially bounded size, such that* $C_n$ *represents the polynomial* $f_n$*.*

This theorem is an obvious consequence of lemma 1 and 2 below, the first of which can be shown by an easy induction on the size of the circuit while the second is proved in appendix A. The basic idea is comparable to the naïve transformation of a circuit into a formula by duplicating gates, however the degree of the circuit turns out to be a bound on the number of copies needed for a given gate, so that we avoid a potentially exponential growth in size.

**Lemma 1.** *If* $C$ *is a multiplicatively disjoint circuit of size* $t$*, its degree is less than* $t$*.*

**Lemma 2.** *If* $C$ *is a circuit of size* $t$ *and degree* $d$*, there exists a multiplicatively disjoint circuit* $C'$*, which computes the same polynomial and whose size is less than* $dt$*.*

Thus our characterization of VP uses circuits which seem to be the middle ground between formulas and circuits. It is therefore not so surprising that we should be able to use this characterization to compare the expressive power of both models.

## 4. Formulas and circuits

One major open question is whether circuits are more powerful than formulas at the polynomial level, i.e. whether the inclusion $\mathrm{VP_e} \subseteq \mathrm{VP}$ is strict or not. The first step of the completeness proof of the Permanent is to show that under a Boolean sum formulas and circuits have the same power. A technically involved proof of this can be found for example in [Bür00]. We use our characterization of VP to give a simpler and more intuitive proof.

**Theorem 2.** $\mathrm{VNP} = \mathrm{VNP_e}$ *over any field.*

The inclusion $\mathrm{VNP_e} \subseteq \mathrm{VNP}$ is obvious. It is easy to see that, in order to prove the inclusion $\mathrm{VNP} \subseteq \mathrm{VNP_e}$, we need only prove the inclusion $\mathrm{VP} \subseteq \mathrm{VNP_e}$. We therefore need to express the polynomial represented by a circuit as a sum of formulas. For a given circuit we will consider graphs called *parse trees*. These graphs appear under different names in several previous works [JS82, VT89, Ven92, AJMV98, Mal03]. We will use them in the context of arithmetic circuits, in the spirit of this quote from [JS82]: a parse tree is "a family tree which charts the generation of a particular monomial in the final result".

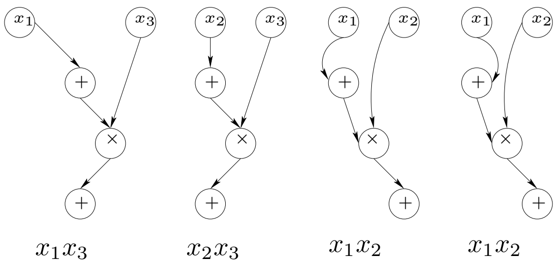

FIGURE 3. The parse trees of the circuit from figure 1

**Definition 6.** The set of parse trees of a circuit $C$ is defined by induction on it size:

- If $C$ is of size 1 it has only one parse tree, itself.
- If the output gate of $C$ is a + gate whose arguments are the gates $\alpha$ and $\beta$, the parse trees of $C$ are obtained by taking either a parse tree of $C_\alpha$ and the arrow from $\alpha$ to the output or a parse tree of $C_\beta$ and the arrow from $\beta$ to the output.
- If the output gate of $C$ is a × gate whose arguments are the gates $\alpha$ and $\beta$, the parse trees of $C$ are obtained by taking a parse tree of $C_\alpha$ and a parse tree of a disjoint copy of $C_\beta$ and the arrows from $\alpha$ and $\beta$ to the output.

We may also describe parse trees in the following manner. If $C$ is a multiplicatively disjoint circuit, a graph $T$ is an *expansion* of $C$ if the following conditions are met:

(1) $T$ is a sub-graph of $C$ which contains the output gate of $C$.
(2) If $\alpha$ is a multiplication gate in $T$ receiving arrows from gates $\beta$ and $\gamma$ in $C$, then the arrows $(\beta, \alpha)$ and $(\gamma, \alpha)$ both also appear in $T$.
(3) If $\alpha$ is an addition gate in $T$, it receives exactly one arrow in $T$.
(4) Only arrows and gates obtained in this way belong to $T$.

Figure 2 gives an example of a circuit and its parse trees. Each parse tree is identified with a monomial by computing the product of the values of the input gates. It turns out that the polynomial computed by the circuit is thus the sum of the values of its parse trees. This is true in general, and can easily be shown by induction. We write $\mathrm{PT}(C)$ for the set of parse trees of a circuit $C$ and $\mathrm{val}(T)$ for the value of parse tree $T$.

**Lemma 3.** *If $C$ represents the polynomial $f$ then $f(\bar{x}) = \sum_{T \in \mathrm{PT}(C)} \mathrm{val}(T)$.*

To prove the inclusion $\mathrm{VP} \subseteq \mathrm{VNP_e}$ we thus wish to write a polynomial in VP as a sum of formulas. We can use the previous lemma, but we need to show that we can indeed sum over all parse trees and compute the value of a parse tree. In other words we will in fact sum over all possible Boolean words of a given length, as in the definition of $\mathrm{VNP_e}$, therefore we need to have a formula to recognize when a word encodes a parse tree and to compute its value. This task is easier for MD circuits, thanks to the following proposition, which is not hard to prove (remember that we distinguish left and right arguments of a gate).

**Proposition 1.** *A circuit $C$ is multiplicatively disjoint iff any parse tree of $C$ is a sub-graph of $C$.*

For instance, figure 3 gives the parse trees of the circuit from figure 1 and one can see that they are sub-graphs of the circuit. McKenzie, Vollmer and Wagner study in [MVW00] the notion of parse trees (which they call proof trees) and the associated notion of proof

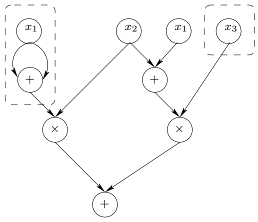

Figure 4. A weakly skew circuit

circuits, defined as sub-circuits satisfying conditions (1)–(4) above. They study the complexity of counting trees versus counting circuits. In their setting, one can see MD circuits as the circuits whose sets of parse trees and proof circuits are equal.

The useful implication for us is that in the case of MD circuits all parse trees are subgraphs. And since the circuit is of polynomial size, it is straightforward, if somewhat tedious, to recognize and compute the value of the parse trees of a circuit with a formula. This is done in appendix B.

## 5. The complexity of the determinant

5.1. **The class** VQP**.** The Determinant family is known to belong to the class VP. However it is not known to be VP-complete, nor is it thought to be. The class VQP, defined via circuits of quasi-polynomial size, was introduced to further study the complexity of the Determinant. Indeed one can find proofs of completeness of the Determinant for VQP in [vzG87, Bür00]. We give here a simple proof using a stronger restriction on multiplications than the one used to characterize VP. Note that the notion of reduciton is also changed in the definitions below.

**Definition 7.** A function $t$ from $\mathbb{N}$ to $\mathbb{N}$ is *quasi-polynomially bounded* if there exists two constants $a$ and $b$ such that $t(n) \leq n^{a \cdot \log^b n}$ for all $n \geq 2$.
A sequence of polynomials $(f_n)$ belongs to the class VQP if its number of variables and degree is polynomially bounded and if it is represented by a circuit of quasi-polynomially bounded size.
A sequence $(f_n)$ is a *qp-projection* of a sequence $(g_n)$ if there exists a quasi-polynomially bounded function $t$ such that for all $n$ $f_n$ is a projection of $g_{t(n)}$

The proof given in [Bür00] relies on a parallelization lemma [VSBR83] which states that a circuit of size $s$ and degree $d$ in $n$ variables can be parallelized to produce a circuit of size $O(d^6 s^3)$ and depth $O((\log ds) \log d + \log n)$. A stronger version of MD circuits was used in [Mal03] to prove the same completeness result without the need to parallelize. These so-called *strongly multiplicatively disjoint* circuits are in fact the *weakly skew* circuits of [Tod92]. This last work is extremely relevant to the complexity of the Determinant in Valiant's setting and it is surprising that this has not been noticed before. Much as MD circuits give us more information than the retarded programs of Babai and Fortnow, weakly skew circuits provide the necessary structural information when compared to the restricted programs introduced by Damm [Dam91]. Recall that a circuit is *skew* if all multiplications gate have at most one argument which is not an input gate. The condition is somewhat relaxed for weakly skew circuits.

**Definition 8.** A circuit is *weakly skew* if for any multiplication gate $\alpha$, receiving arrows from gates $\beta$ and $\gamma$, one of the two sub-circuits $C_\beta$ or $C_\gamma$ is only connected to the rest of the circuit by the arrow going to $\alpha$.

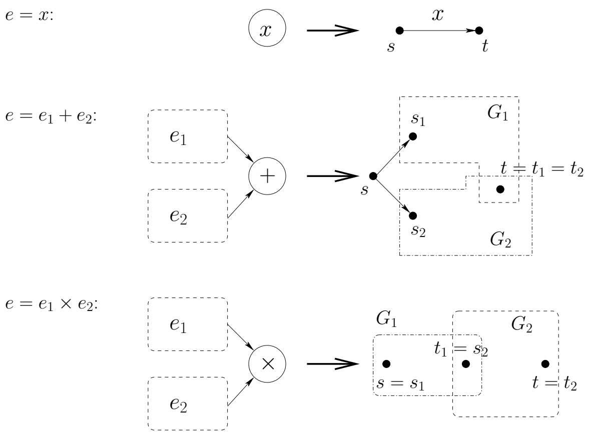

FIGURE 5. Universality for formulas

Formulas are circuits where arguments cannot be re-used, weakly skew circuits demand that at least one of the two arguments of a multiplication gate be computed just for that gate. Figure 4 gives an example of a weakly skew circuit and shows the independent argument of each multiplication gate. Weakly skew circuits characterize VQP.

**Theorem 3.** *A sequence of polynomials $(f_n)$ belongs to* VQP *if and only if there exists a sequence $(C_n)$ of weakly skew circuits, of quasi-polynomially bounded size and polynomially bounded degree, such that $C_n$ represents the polynomial $f_n$.*

Note that if one defines the class VQP with a quasi-polynomial bound on the degree instead of a polynomial bound, then the characterization is exact (we can omit the polynomial degree restriction). All the completeness results for VQP as defined above would hold for such a class. The previous theorem is a consequence of the following lemma, whose proof is given in appendix C.1.

**Lemma 4.** *If $C$ is a circuit of size $t$ and degree $d$, there exists a weakly skew circuit computing the same polynomial and of size less than $t^{\log d}$.*

The classical proof of the completeness of the Determinant is to show a so-called universality property for formulas, namely that a the polynomial computed by a formula $s$ is a projection of the Determinant or the Permanent of a matrix of size polynomial in $s$. This is shown by building weighted graphs with adequate properties. Let $G$ be an edge-weighted directed graph with two vertices $s$ and $t$, the weight of a path from $s$ to $t$ is the product of the weights of the edges appearing in the path. The weight of $(s, t)$ in $G$ is the sum of the weights of all paths from $s$ to $t$. To prove the universality lemmas one starts by building a graph whose weight is the polynomial computed by a formula (cf. [Bür00] for example). We show here that the same construction can be done for weakly skew circuits.

**Lemma 5.** *Let $C$ be a weakly skew circuit of size $m$, there exists an acyclic directed graph $G$, with two distinguished vertices $s$ and $t$, such that: $G$ is of size $m + 1$ and the weight of $(s, t)$ in $G$ is the polynomial computed by $C$.*

The complete proof of the lemma is given in appendix C.2. When proving the lemma for formulas, one can build the graph by induction in the following manner, illustrated in figure 5: an input gate becomes an edge weighted with the corresponding variable or constant; for an addition gate we place the graphs corresponding to the arguments in parallel; for a multiplication gate we place them in series. Going from a formula to a weakly skew circuit one must strengthen the property being proved so that it applies to circuits with multiple output gates. Therefore we wish to build a graph which has a vertex $t_\alpha$ for several gates $\alpha$ in the circuit such that the weight of the graph between $s$ and $t_\alpha$ is the polynomial represented by $\alpha$. The induction steps above still work except that by placing the circuits in series when we multiply we may change the weight of the gates of the second circuit. The weakly skew condition guarantees that we will not need the values of these gates later in the circuit, so that the construction goes through. Indeed, one can see weakly skew circuits as the most expressive circuits for which this construction can work, in the sense that any polynomial which is the weight of a graph of size $s$ can be computed by a weakly skew circuit of size polynomial in $s$ (this is a consequence of the completeness results which follow). From this construction we can show the universality of the Determinant for weakly skew circuits (cf. appendix C.3).

**Lemma 6.** *If $f$ is a polynomial computable by a weakly skew circuit of size $m$, $f$ is a projection of* $\mathrm{DET}_{m+1}$.

From this lemma and the fact that the Determinant is in VP and therefore in VQP one can show the completeness of the Determinant.

**Theorem 4.** *The Determinant is* VQP*-complete over any field.*

This gives us the following algebraic characterization of whether VNP is included in VQP, as noted in [vzG87] (it is shown in [Bür00] that VQP is not included in VNP).

**Theorem 5.** $\mathrm{VNP} \subseteq \mathrm{VQP}$ *iff the Permanent is a qp-projection of the Determinant.*

Now consider the families of polynomials $(F_n)$, $(G_n)$ and $(H_n)$ defined by $F_n = \mathrm{Tr}(X^n)$, $G_n = \mathrm{Tr}(X_1 \cdots X_n)$ and $H_n = \mathrm{Tr}(\mathrm{DET}(X) \cdot X^{-1})$, where Tr is the trace, and $X$ or $X_i$ are matrices with $n^2$ variables. The same technique can be used to show their completeness for VQP. One just needs to show a universality result and that they can be computed by weakly skew circuits of quasi-polynomial size. We will only mention the steps for $(F_n)$, since it is the case missing from [Blä01], which gives a partial answer to the conjecture. It is easy to show from the inductive definition that matrix powering can be computed by weakly skew circuits, in fact one can show a stronger result using skew circuits (cf. [Tod92]). To show universality we use the generic construction of lemma 5 and then modify the resulting graph. The construction is much more involved than for the Determinant and is given in appendix C.5. The construction can be simplified if we just want to prove the completeness of computing the $(1,1)$-coefficient of the power of a matrix, and this would be our choice because the universality of matrix powering will be used later to show the equivalence of skew and weakly circuits. As it is we show the more result with the trace to answer Bürgisser's conjecture.

**Lemma 7.** *If $f$ is a polynomial computable by a weakly skew circuit of size $m$, $f$ is a projection of $F_{2m+3}$ or $F_{2m+5}$.*

The following theorem is thus a positive answer to conjecture 8.1 from [Bür00].

**Theorem 6.** *The families $(F_n)$, $(G_n)$ and $(H_n)$ are* VQP*-complete over any field.*

5.2. **The class** VDET**.** We have already said that [Tod92] gives an excellent account of the complexity of the Determinant, which can be immediately transposed into Valiant's setting. In fact, Toda defines the very natural class DET(poly) of polynomial families which can be expressed as the determinant of a sequence of matrices (with variable or constant entries) of polynomially bounded size. This class is shown to be characterized by skew arithmetic circuits, and equivalently by weakly skew arithmetic circuits. Let us rename this class VDET in Valiant's framework and define it directly by weakly skew circuits.

**Definition 9.** A sequence of polynomials $(f_n)$ belongs to the class VDET if it is represented by a sequence of weakly skew circuits of polynomially bounded size.

This class is a much more natural candidate to capture the complexity of the Determinant than VQP. The previous universality lemma together with a computation by weakly skew circuits (cf. appendix C.4) gives us a natural proof of the completeness of the Determinant. Note that this completeness is under standard $p$-projections, which is one reason we suggest this class be preferred to VQP.

**Theorem 7.** *The sequence* $(\mathrm{DET}_n)$ *is* VDET*-complete over any field.*

Defining VDET via weakly skew circuits puts it naturally between $\mathrm{VP_e}$ and VP. The proof of completeness for the Determinant is easier because it is simpler to show that it can be copmuted by weakly skew circuits than skew circuits. Moreover, if we follow this order, we can use the completeness of matrix powering for VDET (same proof strategy as for VQP), to get an immediate proof of the characterization of this class by skew arithmetic circuits, thus avoiding the more technical constructions in [Tod92]. Indeed, any family in VDET is a $p$-projection of $(F_n)$. As noticed before, $(F_n)$ can be computed by sequences of *skew* circuits of polynomial size. The strict nature of $p$-projections thus yields polynomial size skew circuits for any family in VDET, including the Determinant. One can also show the VDET-completeness of the families $(G_n)$ and $(H_n)$.

**Theorem 8.** *The families* $(F_n)$*,* $(G_n)$ *and* $(H_n)$ *are* VDET*-complete over any field.*

**Proposition 2.** *A sequence of polynomials* $(f_n)$ *belongs to the class* VDET *if it is represented by a sequence of skew circuits of polynomially bounded size.*

We also get an algebraic characterization of a weaker form of Valiant's hypothesis.

**Theorem 9.** *The permanent is a p-projection of the determinant iff* $\mathrm{VDET} = \mathrm{VNP}$*.*

To summarize, we have considered the increasing expressive power of the following sequence of models, when the size is polynomially bounded: formulas, skew circuits, weakly skew circuits, MD circuits. One of the reason VQP was considered a "good" class is that if we allow a quasi-polynomially bounded size, all these classes are equal (cf. [vzG87]). This is also the case if we polynomially bound the depth rather than the size. And, as we shall see in the next section, in the uniform case the resulting class characterizes VNP.

## 6. Characterizing VNP

The title of this section is a bit misleading as we will be characterizing a uniform version of VNP. We wish to compare the respective expressive power of Boolean sums in front of a circuit of polynomial size and degree (VNP) on the one hand and of circuits of polynomial depth and degree on the other. This is related to the characterization of ♯P via circuits of polynomial depth and degree in [Ven92]. We will show that a similar theorem holds for a uniform version of Valiant's algebraic classes.

At the non-uniform level it is easy to see that circuits of polynomial depth and degree are at least as powerful as VNP. Indeed a sequence in VNP is defined from a sequence in VP which is represented by circuits of polynomial size and degree, and therefore polynomial depth and degree. By computing in parallel all the values of these circuits for all Boolean strings of appropriate length and then summing, we get a circuit of polynomial depth and degree. The summation can be done in polynomial depth because there is a simply exponential number of gates to sum.

For the converse we would like to express the polynomial computed by a circuit of polynomial depth and degree as a sum of the values of a circuit of polynomial size and degree. This sounds eerily similar to saying that the value of the circuit is the sum of the values of its parse trees. Except that now there is no polynomial bound on the size of our original circuit. However, as noticed in [Ven92], the constraints on depth and degree give us a constraint on the size of the parse trees.

**Lemma 8.** *If $C$ is a circuit of depth $p$ and degree $d$, any parse tree of $C$ is of size less than $pd$.*

Thus a circuit of polynomial size and degree has parse trees of polynomial size. In fact this characterizes such circuits, because if all the parse trees of a circuit are of size bounded by $t$, then by computing all the values of the parse trees in parallel and adding the results we get a circuit of depth $2t$ and degree $t$; in the case of parse trees of polynomial size this means we get a circuit of polynomial size and degree. Note that we can also obtain in this manner a formula of polynomial depth and degree so that, as mentioned at the end of section 5, the different restrictions imposed on circuits all have the same power for polynomial depth and degree.

Let us go back to the converse inclusion. We know that the polynomial represented by a circuit of polynomial depth and degree is the sum of the values of its parse trees, which are of polynomial size and in simply exponential number. Thus the polynomial represented is a sum of simply exponential number of monomials. However to show that it is in VNP we need to be able to recognize efficiently whether a Boolean string encodes a parse tree or not. This does not seem true for non-uniform sequences of circuits of polynomial depth and degree in general (we will return to this point later). By adding a uniformity condition we can get exactly what we need. We will use the condition given in [Ven92]. Define the *direct connection language* of a sequence of circuits $C_n$ as the set of strings of the form $\langle n, g, y, p \rangle$ such that either (i) $g$ is an addition gate in $c_n$ and $y$ is an input of $g$, or (ii) $g$ is a multiplication gate in $c_n$ and $y$ is a left or right input of $g$ depending on $p$, or (iii) $g$ is a gate name in $c_n$ and $y$ is the type of $g$. A sequence of circuits $C_n$ is DLOGTIME-uniform if its direct connection language can be recognized by a deterministic Turing machine in time logarithmic in the size of the circuits. In our case, with circuits of exponential size, it means that we can get information on an arrow or a gate in polynomial time.

Let us now define the uniform classes we have mentioned. For Valiant's classes, uniformity is the most common notion, meaning that the circuit $C_n$ is produced by a Turing machine in polynomial time upon input of $n$ in unary.

**Definition 10.** A sequence of polynomials is in the class $\mathrm{VP_u}$ if it is represented by a P-uniform sequence of circuits of polynomial size and degree.

A sequence of polynomials $(f_n)$ belongs to $\mathrm{VNP_u}$ if there exists a polynomial $p$ and a sequence $g_n \in \mathrm{VP_u}$ such that $f_n(\bar{x}) = \sum_{\bar{\epsilon} \in \{0,1\}^{p(|\bar{x}|)}} g_n(\bar{x}, \bar{\epsilon})$.

**Theorem 10.** *A sequence of polynomials $(f_n)$ belongs to* $\mathrm{VNP_u}$ *iff it can be represented by a* DLOGTIME-*uniform sequence of circuits of polynomial depth and degree.*

The proof of this theorem follows the sketch given above. The remainder is technical details which we will not give here. Note the similarity of this characterization with the characterization of ♯P by Venkateswaran [Ven92]. In both cases the class characterized is uniform. In our description of the proof strategy we emphasize the role played by uniformity. What happens in the non-uniform case?

We will use a converse of Valiant's criterion (cf. [Bür00]) to sketch an answer to this question. Valiant gave a criterion for showing that specific sequences of polynomials belong to VNP, the rough idea being that sequences whose coefficient function is in $\sharp\mathrm{P}/\mathit{poly}$ belong to VNP. One can show a converse of this theorem by using the coefficient function (we will not give details here, this is noted in [Pér04] and can be proved using techniques in [Mal03]). Such a converse states that if we have a family of functions $(f_n)$, where $f_n : \{0,1\}^n \to \big[0, \ldots, 2^{p(n)}\big]$, and if we define $g_n(x_1, \ldots, x_n) = \sum_{\bar{\epsilon}\in\{0,1\}^n} f_n(\bar{\epsilon})\ x_1^{\epsilon_1} \cdots x_n^{\epsilon_n}$, then $(g_n) \in \mathrm{VNP}$ implies $(f_n) \in \sharp\mathrm{P}/\mathit{poly}$.

Now suppose that the characterization of $\mathrm{VNP_u}$ holds in the non-uniform case. Consider any sequence of functions $(f_n)$, with $f_n : \{0,1\}^n \to \big[0, \ldots, 2^{p(n)}\big]$. View this sequence as the coefficient function of a polynomial sequence $g_n(\bar{x}) = \sum_{\bar{\epsilon}} f_n(\bar{\epsilon}) x_1^{\epsilon_1} \cdots x_n^{\epsilon_n}$.

We can compute in polynomial depth these monomials and the integer coefficients and just sum them, so that the sequence $(g_n)$ can be computed by a sequence of circuits of polynomial depth and degree. If the hypothesis is true in this non-uniform case, then $(g_n)$ belongs to VNP. Thus $(f_n)$ belongs to $\sharp\mathrm{P}/\mathit{poly}$. This would show that counting with polynomial advice is more powerful than any integer function.

## 7. Conclusion

We have shown in this paper that different classes in Valiant's framework can be defined via a hierarchy of circuits of polynomial size, from formulas to weakly skew circuits to multiplicatively disjoint circuits, and that all these restrictions become equivalent for polynomial depth (in the uniform case) and define the class VNP. These characterizations came with new results and new proofs of old results. In our view, one important aim of this paper is to bring attention to the work of Toda [Tod92] and suggest the adoption of the class VDET.

To stress the importance of this class we could like to find other complete polynomials. For any polynomial family which is shown to be in VP we should check if one can show that it is in VDET and complete. For instance the generating function of trees (cf. [Bür00] chapter 3) is reduced to the Determinant and thus belongs to VDET. It would be interesting to know whether it is complete.

As for the characterization of VP, it could help us find a natural complete problems. If we were in the Boolean setting we would have *MD circuit value* as a complete problem, but in Valiant's framework it is more complicated. We would need a "universal" MD circuit. One can be built ad hoc from the VP-complete polynomial family given in [vzG87] or in [Bür00] by applying lemma 2, but that is not a natural family of polynomials. Perhaps one could use the fact that the parse trees of MD circuits are sub-graphs of the circuit.

One last obvious question is the separation of VP and VDET, or in other words whether an MD circuit can be transformed into a weakly skew circuit without an exponential blow-up in size. If the answer is positive, then the classes VP and VDET are equal, the determinant is VP-complete, and interestingly the theorem stating that $\mathrm{VNP} = \mathrm{VNP_e}$ is not necessary to prove the completeness of the permanent, thus considerably simplifying the proof. On the other hand, if sequences in VP do not admit sequences of weakly skew circuits of polynomial size, than the classes $\mathrm{VP_e}$ and VP are distinct, an answer to a major open question, and the determinant is not VP-complete. Thus the restrictions imposed on multiplications seems to be a crucial point to investigate in order to better understand Valiant's complexity classes, although answers to the above questions will be hard to come by.

## References

[ABKPM04] Eric Allender, Peter Bürgisser, Johan Kjeldgaard-Pedersen, and Peter Bro Miltersen. On the complexity of numerical analysis. *Electronic Colloquium on Computational Complexity (ECCC)*, (037), 2004.

[AJMV98] Eric Allender, Jia Jiao, Meena Mahajan, and V. Vinay. Non-commutative arithmetic circuits: depth reduction and size lower bounds. *Theoret. Comput. Sci.*, 209(1-2):47–86, 1998.

[BF91] László Babai and Lance Fortnow. Arithmetization: a new method in structural complexity theory. *Comput. Complexity*, 1(1):41–66, 1991.

[Blä01] Markus Bläser. Complete problems for valiant's class of qp-computable families of polynomials. In *COCOON '01: Proceedings of the 7th Annual International Conference on Computing and Combinatorics*, pages 1–10, London, UK, 2001. Springer-Verlag.

[Bür00] Peter Bürgisser. *Completeness and reduction in algebraic complexity theory*, volume 7 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, 2000.

[Dam91] Carsten Damm. DET $= L^{\#L}$. Informatik-Preprint 8, Humboldt-Universität zu Berlin, 1991.

[IL95] Neil Immerman and Susan Landau. The complexity of iterated multiplication. *Inf. Comput.*, 116(1):103–116, 1995.

[JS82] Mark Jerrum and Marc Snir. Some exact complexity results for straight-line computations over semirings. *J. ACM*, 29(3):874–897, 1982.

[Kal86] E Kaltofen. Uniform closure properties of p-computable functions. In *STOC '86: Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 330–337, New York, NY, USA, 1986. ACM Press.

[KI03] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 355–364, 2003.

[Mal03] Guillaume Malod. *Polynômes et coefficients.* PhD thesis, Université Claude Bernard Lyon 1, 2003.

[MVW00] Pierre McKenzie, Heribert Vollmer, and Klaus W. Wagner. Arithmetic circuits and polynomial replacement systems. In Sanjiv Kapoor and Sanjiva Prasad, editors, *FSTTCS*, volume 1974 of *Lecture Notes in Computer Science*, pages 164–175. Springer, 2000.

[Pér04] Sylvain Périfel. Polynômes donnés par des circuits algébriques et généralisation du modèle de Valiant. Master's thesis, École Normal Supérieure de Lyon, France, June 2004.

[Tod92] S. Toda. Classes of arithmetic circuits capturing the complexity of computing the determinant. *IEICE Transactions on Information and Systems*, E75-D:116–124, 1992.

[Val79] L. G. Valiant. Completeness classes in algebra. In *STOC '79: Proceedings of the eleventh annual ACM symposium on Theory of computing*, pages 249–261, New York, NY, USA, 1979. ACM Press.

[Val82] L. G. Valiant. Reducibility by algebraic projections. In *Logic and Algorithmic: an International Symposium held in honor of Ernst Specker*, volume 30 of *Monographies de l'Enseignement Mathémathique*, pages 365–380, 1982.

[Val92] L. G. Valiant. Why is Boolean complexity theory difficult? In *Boolean function complexity (Durham, 1990)*, volume 169 of *London Math. Soc. Lecture Note Ser.*, pages 84–94. Cambridge Univ. Press, Cambridge, 1992.

[Ven92] H. Venkateswaran. Circuit definitions of nondeterministic complexity classes. *SIAM J. Comput.*, 21(4):655–670, 1992.

[VSBR83] L. G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast parallel computation of polynomials using few processors. *SIAM J. Comput.*, 12(4):641–644, 1983.

[VT89] H. Venkateswaran and Martin Tompa. A new pebble game that characterizes parallel complexity classes. *SIAM J. Comput.*, 18(3):533–549, 1989.

[vzG87] Joachim von zur Gathen. Feasible arithmetic computations: Valiant's hypothesis. *J. Symb. Comput.*, 4(2):137–172, 1987.

## Appendix A

**Proof of lemma 2.** As will happen in several other proofs, we allow circuits to have several output gates. We shall build a sequence of multiplicatively disjoint circuits $C_f$, with $f$ ranging from 1 to $d$ such that for any gate $\alpha$ of $C$ which is of degree $e$ less than $f$ the following conditions hold:

- $C_f$ has gates $\alpha_1, \ldots, \alpha_{d+1-e}$ which compute in $C_f$ the polynomial computed by $\alpha$ in $C$; gate $\alpha_k$ is called the clone of $\alpha$ of index $k$.
- the gates in the sub-circuit of $C_f$ associated with the clone $\alpha_k$ are clones whose index lies between $k$ and $k+e-1$ included.

Circuit $C_1$ is made of $d$ copies of the sub-circuit of $C$ containing only gates of formal degree 1. Therefore it does not contain any multiplication gate and is thus multiplicatively disjoint. Each gate $\alpha$ of $C$ of degree 1 has $d$ clones and the gates of the sub-circuit associated with $\alpha_k$ are clones of index $k = k+1-1$. The aforementioned conditions are met.

Suppose now that the circuits $C_f$ have been built up to $e-1$. We start by adding multiplication gates. Let $\alpha$ be a multiplication gate in $C$ of degree $e$, receiving arrows from gates $\beta$ and $\gamma$ of degree $e_1$ and $e_2$ respectively (with $e = e_1 + e_2$). We add the clones $\alpha_1, \ldots, \alpha_{d+1-e}$. For $i$ ranging from 1 to $d+1-e$, $\alpha_i$ receives an arrow from the clone $\beta_i$ and an arrow from the clone $\gamma_{i+e_1}$ of $C_{e-1}$ (these clones exist because $1 \leq i \leq d+1-e$ and $e_1+1 \leq i+e_1 \leq d+1-e_2$). Since each clone of $\beta$ in $C_{e-1}$ computes the same polynomial as $\beta$ in $C$, and similarly for $\gamma$, each clone of $\alpha$ in $C_{e-1}$ computes the polynomial computed by $\alpha$ in $C$. In order to show that the resulting circuit is multiplicatively disjoint, one need only check that each gate $\alpha_i$ is disjoint. But the gates in the sub-circuit associated with $\beta_i$ are clones whose index lies between $i$ and $i+e_1-1$ and that the gates in the sub-circuit associated with $\gamma_{i+e_1-1}$ are clones whose index lies between $i+e_1$ and $i+e_1+e_2-1$. The two sub-circuits which send an arrow to $\alpha_i$ are therefore disjoint. Finally one can check the last required property: the sub-circuit associated with $\alpha_i$ is the union of the sub-circuit associated with $\beta_i$ with the sub-circuit associated with $\gamma_{i+e_1-1}$. The gates are therefore clones of index ranging from $i$ to $i+e_1+e_2-1 = i+e-1$.

We then add the addition gates, following an order such that when we clone a gate, each gate from which it receives an arrow has already been cloned. Let $\alpha$ be an addition gate in $C$ of degree $e$, receiving arrows from gates $\beta$ and $\gamma$ of respective degree $e$ and $e'$ (with $e' \leq e$). We add the clones $\alpha_1, \ldots, \alpha_{d+1-e}$. For $i$ ranging from 1 to $d+1-e$, $\alpha_i$ receives an arrow from the clone $\beta_i$ and an arrow from the clone $\gamma_i$. Since we are adding an addition gate, the circuit stays multiplicatively disjoint. Each clone of $\alpha$ computes the adequate polynomial. And the gates of the sub-circuit associated with $\alpha_i$ are clones whose index lies between $i$ and $i+e-1$, because $e'$ is less than $e$.

Let $C'$ be the associated sub-circuit for the output gate of $C$ in $C_d$. By construction this circuit is multiplicatively disjoint and computes the same polynomial as $C$. Each gate in $C$ has been cloned at most $d$ times, so the size of $C'$ is less than $dt$.

## Appendix B

**End of the proof of** $\mathrm{VNP} = \mathrm{VNP_e}$**.** Let us label the gates of $C$ with the numbers from 1 to $t$. We then partition the set $\{1, 2, \ldots, t\}$ in three sets $I$, $M$, $A$ which respectively contain the labels for input gates, multiplication gates and addition gates and let us suppose that $t$ labels the output gate. For $i$ in $E$, let $V_i$ be the variable for the input gate $i$. a parse tree $D$ shall be encoded by the variables $a_{i,j}$ for $i$ and $j$ ranging from 1 to $t$ and such that the arrow $(i, j)$ belongs to $C$, with the idea that this variable is 1 if the arrow $(i, j)$ is in

$D$ and 0 otherwise, and by the variables $p_i$ for $i$ raging from 1 to $t$, this variable being 1 if gate $i$ is in $D$ and 0 otherwise.

We shall compute the product of the following polynomials, each being used to meet one of the requirements in the definition of a parse tree of a circuit.

We start by demanding that if an arrow is in $D$ then the gates it links must belong to $D$:

$$\prod_{(i,j)\in C} (a_{i,j}p_ip_j + 1 - a_{i,j}).$$

(i) To ensure that $D$ contains the output gate of $C$:

$$p_t.$$

(ii) To ensure that for any multiplication gates in $D$, both arrows it receives are also in $D$:

$$\prod_{\substack{i\in M \text{ and } j,k \text{ such that}\\ (j,i)\in C \text{ and } (k,i)\in C}} (p_i a_{j,i} a_{k,i} + (1 - p_i)).$$

(iii) To ensure that for any addition gate in $D$, it receives exactly one arrow in $D$:

$$\prod_{\substack{i\in A \text{ and } j,k \text{ such that}\\ (j,i)\in C \text{ and } (k,i)\in C}} (\ p_i\ (a_{j,i}(1 - a_{k,i}) + a_{k,i}(1 - a_{j,i}))\ +\ 1 - p_i\ ).$$

(iv) To ensure that any gate in $D$ which is not the output gate sends at least one arrow toward another gate in $D$ (do note that if a sub-graph of a multiplicatively disjoint satisfies conditions (ii) and (iii) then any of its gates sends at most one arrow, and we can thus write a disjunction as a sum):

$$\prod_{1\leq i<t} \left( p_i \cdot \left( \sum_{\substack{j \text{ such that}\\ (i,j)\in C}} a_{i,j} \right) + 1 - p_i \right).$$

At last, after having checked that $\bar{a}, \bar{p}$ does encode a parse tree of $C$, to compute the associated monomial:

$$\prod_{i\in E} (p_i \cdot V_i + 1 - p_i).$$

These polynomials can clearly be computed by arithmetic formulas of polynomial size with regard to the number of gates in the multiplicatively disjoint circuit $C$.

## Appendix C

C.1. **Proof of lemma 4.** We will consider circuits with multiple output gates. The degree of such a circuit $C$ is the maximal degree of a gate in $C$. If a circuit is weakly skew, for any multiplication gate one of the argument sub-circuit is independent from the rest of the circuit, in the sense that the values computed by its gates are not used elsewhere. A gate will be called *reusable* if it does not belong to the independent sub-circuit of a multiplication gate. In the case of figure 4, all gates are reusable except the leftmost input gate ($x_1$), the addition gate to which it is connected and the rightmost input gate ($x_3$).

Let us show by induction on $n$ that for any integer $d$ such that $2^n \leq d \leq 2^n + 1$, for any (multiple output) circuit of size $t$ and degree $d$, there exists a a weakly skew circuit $C'$ such that:

- the size of $C'$ is at most $t^{\log d}$.

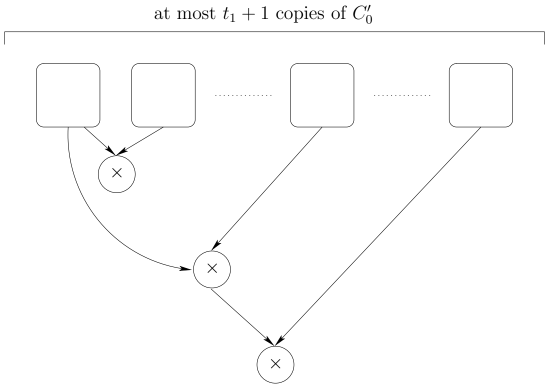

FIGURE 6

- for any gate $\alpha$ of $C$, there exists a reusable gate in $C'$ which computes the polynomial computed by $\alpha$ in $C$.

If $n$ is 0, the degree of $C$ is 1 so that there are no multiplication gates in $C$. Thus $C$ is weakly skew circuit and the property is true.

Suppose now that the property is true for all $k$ strictly less than $n$, with $n \geq 1$. Consider $C$ a circuit of size $t$ and degree $d$, with $2^n \leq d < 2^{n+1}$. Call $C_0$ the circuit obtained by removing all gates of degree strictly greater than $\lfloor d/2 \rfloor$. Let $t_0$ be the size of $C_0$ and $t_1$ the number of gates of $C$ of degree strictly greater than $\lfloor d/2 \rfloor$. We apply the induction hypothesis to $C_0$. This yields a circuit $C'_0$ of size at most $t_0^{\log\lfloor d/2 \rfloor}$. For any gate of $C_0$ there exists a reusable gate in $C'_0$ computing the same polynomial. Consider now a multiplication gate of $C$ of degree strictly greater than $\lfloor d/2 \rfloor$:

- if both its arguments are of degree at most $\lfloor d/2 \rfloor$, we add to $C'$ a multiplication gate receiving arrows from a reusable gate of the first copy of $C'_0$ and from a reusable gate of a new copy of $C'_0$ (cf. figure 6).
- otherwise, since at least one of the arguments is of degree at most $\lfloor d/2 \rfloor$, the other having already been computed by a gate of $C'$, we add to $C'$ a multiplication gate receiving arrows from the gate of degree greater than $\lfloor d/2 \rfloor$ and from a reusable gate of a new copy of $C'_0$.

Addition gates are easy to deal with, we just connect them to reusable gates computing their arguments. The resulting circuit is weakly skew and satisfies the required conditions. Since $t = t_0 + t_1$, one can bound the size of $C'$ as follows:

$$\begin{aligned} & (t_1 + 1) \cdot t_0^{\log\left\lfloor \frac{d}{2} \right\rfloor} + t_1 \\ \leq \;& t \cdot t^{\log\left\lfloor \frac{d}{2} \right\rfloor} \\ \leq \;& t^{\log\left(2 \cdot \left\lfloor \frac{d}{2} \right\rfloor\right)} \\ \leq \;& t^{\log d}. \end{aligned}$$

C.2. **Proof of lemma 5.** We will show a stronger result in the case of circuits with multiple outputs. We also keep the notion of reusable gates for a weakly skew circuit.

Let us show by induction on circuit size $m$ that for any multiple output weakly skew circuit $C$ there exists an acyclic directed graph $G$ with a distinguished vertex $s$, satisfying the following conditions:

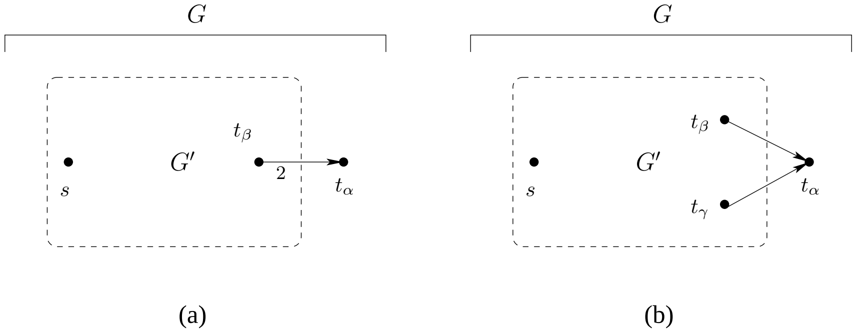

FIGURE 7. Addition gate

- $G$ is of size at most $m+1$.
- for any reusable gate $\alpha$ of $C$ there exists a vertex $t_\alpha$ in $G$ such that the weight of $(s, t_\alpha)$ in $G$ is the polynomial computed by $\alpha$ in $C$.

A circuit of size $m = 1$ is made of one gate $\alpha$ with a (constant or variable) label $u$. The graph $G$ with two vertices $s$ and $t_\alpha$ and an edge $(s, t_\alpha)$ of weight $u$ meets our requirements.

Suppose the above property is true for all integers strictly less than $m$ ($m \geq 2$). Let $C$ be a weakly skew circuit of size $m$ and $\alpha$ one of its output gates.

If $\alpha$ is an input gate labeled $u$ we just need to apply the induction hypothesis to the circuit $C'$ with $\alpha$ removed. This yields a graph $G'$ to which we add a new vertex $t_\alpha$ with an edge from $s$ to $t_\alpha$ of weight $u$. Clearly the graph $G$ thus obtained satisfies the necessary conditions.

If $\alpha$ is an addition gate, let $C'$ be the circuit $C$ without gate $\alpha$. By induction hypothesis there exists a graph $G'$ of size at most $m$. If $\alpha$ receives both its incoming arrows from one (necessarily reusable) gate $\beta$, there exists a vertex $t_\beta$ in $G'$ such that the weight of $(s, t_\beta)$ in $G'$ is the polynomial computed by $\beta$ in $C$. We add a new vertex $t_\alpha$ and the edge $(t_\beta, t_\alpha)$ with weight 2 (cf. figure 7 (a)). If $\alpha$ receives an arrow from two distinct gates $\beta$ and $\gamma$, both necessarily reusable, there exist vertices $t_\beta$ and $t_\gamma$ in $G'$ such that the weights of $(s, t_\beta)$ and $(s, t_\gamma)$ in $G'$ are the polynomials computed in $C$ by $\beta$ and $\gamma$ respectively. We then add a new vertex $t_\alpha$ to $G'$ and the edges $(t_\beta, t_\alpha)$ and $(t_\gamma, t_\alpha)$ with weight 1 (cf. figure 7 (b)). In both cases, the resulting graph $G$ is of size at most $m+1$ and satisfies the conditions.

If $\alpha$ is a multiplication gate, consider the distinct gates $\beta$ and $\gamma$ from which $\alpha$ receives an arrow. Suppose the sub-circuit $C_\gamma$ is independent from the rest of the circuit, then the circuit $C'$ obtained by removing $\alpha$ is composed of two disjoint circuits $C_\beta$ and $C_\gamma$, of size $m_\beta$ and $m_\gamma$ such that $m = m_\beta + m_\gamma + 1$. Applying the induction hypothesis separately to $C_\beta$ and to $C_\gamma$ yields two graphs $G_\beta$ and $G_\gamma$. In the first there are two vertices $s$ and $t_\beta$ such that the weight of $(s, t_\beta)$ in $G_\beta$ is the polynomial computed by $\beta$ in $C$. In the second there are two vertices $(s_\gamma, t)$ such that the weight of $(s_\gamma, t)$ in $G_\gamma$ is the polynomial computed by $\gamma$ in $C$. We obtain $G$ by identifying the vertices $t_\beta$ and $s_\gamma$ (cf. figure 8). $G$ is of size at most $m_\beta + 1 + m_\gamma + 1 - 1 = m + 1$. The weight of $(s, t)$ in $G$ is clearly the product of the weight of $(s, t_\beta)$ in $G_\beta$ and of the weight of $(s_\gamma, t)$ in $G_\gamma$, i.e. the polynomial computed by $\alpha$ in $C$. We have changed the value of the weights $(s, v)$ for all vertices $v$ in $G_\gamma$, but since these vertices were associated to the circuit $C_\gamma$ whose gates are *not* reusable, the necessary properties still hold.

C.3. **Proof of lemma 6.** From the graph $G$ built in lemma 5 we build a graph $G'$ by identifying the vertices $s$ and $t$ and adding a loop to each vertex except $s = t$. Now consider the graph $G''$ obtained from $G'$ by changing the weight of every edge which is

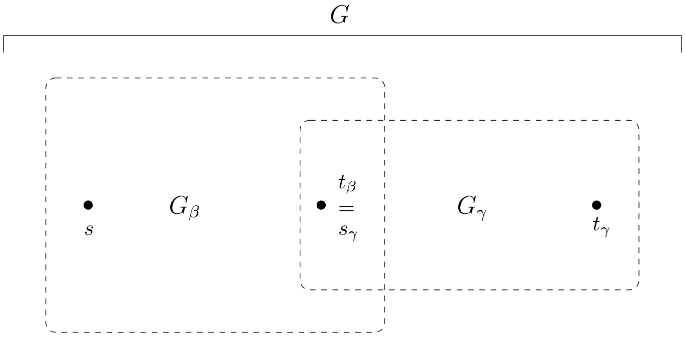

FIGURE 8. Multiplication gate

not a loop into its opposite: if $A = (a_{i,j})$ is the matrix representing $G'$, than the matrix representing $G''$ is the matrix $B$ defined by $b_{i,j} = -a_{i,j}$ if $i \neq j$ and $b_{i,i} = a_{i,i}$ for all $i$. It can be shown that the determinant of $B$ is the polynomial $-f$. One need just add a last row and last column full of 0 except for the value in the bottom right hand corner which is $-1$. The determinant of the resulting matrix is $f$.

### C.4. **Computing the Determinant.**

**Proposition 3.** (DET$_n$) *can be computed by a sequence of weakly skew circuits of polynomial size.*

*Proof.* We will recall here the algorithm given in [Val92] to compute the determinant of a matrix $X$ and show that this computation can be done by a weakly skew circuit of polynomial size. Let $B_k$ $(0 \leq k \leq n-1)$ be the principal $(n-k) \times (n-k)$ minor of $X$. Define the $(n-k) \times 1$ matrix $C_k$ and the $1 \times (n-k)$ matrix $D_k$ for $1 \leq k \leq n-1$ as follows:

$$B_{k-1} = \begin{pmatrix} B_k & C_k \\ D_k & X_{n-k+1,n-k+1} \end{pmatrix}$$

For each $k$ $(1 \leq k \leq n)$ define $T_k$ as the following $(n+2-k) \times (n+1-k)$ matrix:

$$(T_k)_{i,j} = \begin{cases} 0 & \text{if} \quad i > j+1 \\ -1 & \text{if} \quad i = j+1 \\ X_{n-k+1,n-k+1} & \text{if} \quad i = j \\ D_k B_k^{j-i-1} C_k & \text{if} \quad i < j \end{cases}$$

Then the coefficients of the characteristic polynomial are given by the $(n+1) \times 1$ matrix $\prod_{k=1}^{n} T_k$, where the $(1,1)$ coefficient is the determinant.

Because a product of matrices can be computed by a weakly skew circuit of polynomial size, each $T_k$ can be computed by an weakly skew circuit of size polynomial in $n$ (each entry of a given $T_k$ is computed separately). In computing the product of the $T_k$ we will need at most $(n+1)$ distinct copies of each $T_k$, so the overall size of the weakly skew circuit stays polynomial in $n$. □

C.5. **Proof of lemma 7.** Let $f$ be a polynomial computed by a weakly skew circuit $C$. Define a *walk* of length $k$ in a directed graph as a sequence of vertices $(t_1, \dots, t_k)$ such that the edges $(t_i, t_{i+1})$ and the edge $(t_k, t_1)$ belong to the graph. A walk may go through a given vertex several times. The vertex $t_1$ is called the *origin* of the walk. The weight of a walk is the product of the weights of its edges. The $k$-weight of a graph $G$ is the sum of the weights of all walks of length $k$.

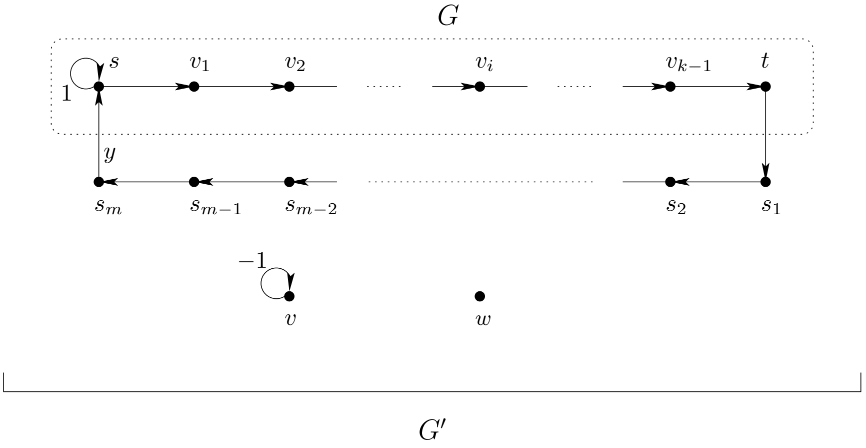

FIGURE 9

Let $X$ be the matrix with entries $x_{i,j}$ $(1 \leq i, j \leq n)$, it is easy to show that the polynomial $\mathrm{Tr}(X^n)$ is equal to:

$$\sum_{1\leq k_1,\ldots,k_n\leq n} x_{k_1,k_2}\cdots x_{k_{n-1},k_n}x_{k_n,k_1}.$$

If we interpret the matrix $X$ as the adjacency matrix of a graph $G$, we can see that $\mathrm{Tr}(X^n)$ is the $n$-weight of $G$. We therefore wish to build a graph of size $l$ whose $l$-weight is the polynomial $f$.

Lemma 5 yields an acyclic directed graph $G$ of size $m+1$, with vertices $s$ and $t$ such that the weight of $(s,t)$ is the polynomial $f$. We start by adding $m$ vertices $s_1,\ldots,s_m$, and the edges $(t,s_1)$, $(s_i,s_{i+1})$, all with weight 1, and finally the edge $(s_m,s)$, whose weight is a new variable $y$. We also add a loop of weight 1 to the vertex $s$, a vertex $v$ with a loop of weight $-1$ and an isolated vertex. (cf. figure 9). The size of the resulting graph $G'$ is $2m+3$. Let us study the walks of $G$ of length $2m+3$.

There is a unique walk of length $2m+3$ which consists in looping on the vertex $v$. Because $2m+3$ is an odd integer, its weight is $-1$.

There is a unique walk of length $2m+3$ which consists in looping on the vertex $s$. Its weight is 1.

Let $\tau = s, v_1, \ldots, v_{k-1}, t$ be a path from $s$ to $t$ of length $k$ in $G$ (and therefore in $G'$). The vertex $v_i$ is the origin of a unique walk of length $2m+3$ going through $\tau$. It consists in going to $t$ via $\tau$ (length $k-i$), then going to $s$ via the vertices $s_i$ (length $m+1$), looping $m+2-k$ times in $s$ (one can check that $m+2-k \geq 0$) finally returning to $v_i$ via $\tau$ (length $i$). The total length is $2m+3$. The path $\tau$ yields a unique walk for each of the vertices $t$, $s_1,\ldots,s_m$. For each of the vertices, any other walk would include going around twice, thus its length would be at least $2(m+2)$, which is strictly greater than $2m+3$. There are also $m+3-k$ walks with $s$ as origin, of length $2m+3$ and going through $\tau$, depending on whether one loops $0,1,\ldots$ or $m+2-k$ times in $s$ before going on $\tau$. All these walks have the same weight, namely the weight of $\tau$ multiplied by $y$. there are thus $2m+3$ walks of length $2m+3$ associated with $\tau$. The $(2m+3)$-weight of $G'$ is therefore:

$$(-1) \ + \ 1 \ + \sum_{\substack{\tau \text{ path} \\ \text{from } s \text{ to } t}} y(2m+3)\cdot \mathrm{weight}(\tau).$$

In characteristic 0, we just have to substitute $(2m+3)^{-1}$ for the variable $y$ to get the polynomial $f$.

In characteristic $p > 0$, we need to be a little more careful. For a fixed $m$ the same construction can be done if $p$ does not divide $2m+3$, and then use the inverse of $2m+3$ as above. If $p$ divides $2m+3$, then $p$ is strictly greater than 2, and $p$ does not divide $2m+5$. We follow the above construction but add two vertices $s_{m+1}$ and $s_{m+2}$.

Kyoto University, Japan
*E-mail address*: malod@kuis.kyoto-u.ac.jp

École Normale Supérieure de Lyon, France
*E-mail address*: Natacha.Portier@ens-lyon.fr