

Autarkies for DQCNF

Oliver Kullmann*

Computer Science Department
Swansea University, UK
Email: O.Kullmann@Swansea.ac.uk

Ankit Shukla†

Institute for Formal Models and Verification
Johannes Kepler University, Austria
Email: ankit.shukla@jku.at

Abstract—Autarkies for SAT are partial assignments for boolean CNF, which either satisfy a clause or leave it untouched. We introduce the natural generalisation of autarkies for DQCNF (dependency-quantified boolean CNF), by generalising constant boolean functions 0, 1, as used in SAT, to arbitrary boolean functions assigned to existential variables, as allowed by the dependency-specification. We regard here DQCNF as a proper generalisation of QCNF (QBF with CNF), and all results naturally apply also to QCNF. We provide the most basic theory, considering confluence of autarky reduction (removing the clauses satisfied by some autarky), and the Autarky Decomposition Theorem, the unique decomposition of a DQCNF into the lean kernel (free from any autarky) and the clauses satisfiable by some autarky. Finding autarkies is NEXPTIME-hard (or PSPACE-hard, when restricting to QCNF), and so autarky systems are introduced, which allow for more feasible restricted notions of autarkies, while maintaining the basic properties. The two most basic autarky systems restrict either the number of existential variables assigned, or the number of universal variables used in the boolean functions assigned.

I. INTRODUCTION

QBF is the natural extension of SAT to a PSPACE-complete problem, by allowing quantifiers (for the basics of the theory see [5]). QBF can be seen as maturing, with solvers and underlying theory available. Its NEXPTIME-complete extension DQBF, which generalises the existential quantifiers by allowing arbitrary dependencies, is at an earlier stage. On the theory side, proof calculi get developed (see [1] for a basic pointer), and dependency-schemes naturally play an important role (see [12], [15] for basic pointers). On the solver side, our investigations are closer to pre- or inprocessing (see [14]). We introduce basic autarky theory, which can become a fabric for pre/inprocessing. The basic theory of autarkies for SAT can be found in [6]. An autarky for a CNF F is a partial assignment φ (with boolean values 0, 1) such that every clause of F , where some variable is assigned by φ , is satisfied by φ . Clauses satisfied by some autarky can be removed satisfiability-equivalently (and indeed these clauses can not be used by any resolution refutation ([9])). Restricted forms of autarkies yield “autarky systems”; an example is given by “matching autarkies” ([6, Subsection 11.11.2]), where selected variables used to satisfy clauses must be distinct.

The natural generalisation for a DQCNF F is to allow φ to assign existential variables v of F , using as values boolean functions as allowed by the dependency-set of v . Making a

clause `true` now means making it a tautology (as propositional formula in the universal variables). The basic autarky-systems A_0, A_1 allow the boolean functions to *essentially* (not just formally) depend on zero resp. one universal variable, while E_1 only uses one existential variable (we use “A” to denote universal variables, and “E” for existential variables). Let’s consider the example from [3, after Definition 4]:

$$\begin{aligned} F &:= \forall x_1, x_2, x_3 \exists y_1(x_1, x_2) \exists y_2(x_2, x_3) : F_0 \\ F_0 &:= (y_1 \vee x_1) \wedge (\neg y_1 \vee x_2) \wedge (\neg y_2 \vee \neg x_2 \vee x_3). \end{aligned}$$

y_1 depends (only) on x_1, x_2 , and y_2 depends (only) on x_2, x_3 . F has an E_1 -autarky for y_2 , which is also an A_0 -autarky, namely we can assign $y_2 \mapsto 0$, and obtain the satisfiability-equivalent QCNF $F' := \forall x_1, x_2 \exists y_1 : (y_1 \vee x_1) \wedge (\neg y_1 \vee x_2)$. By assigning $x_1, x_2 \mapsto 0$ one sees that F' is unsatisfiable. Since there is only one existential variable, it follows that there is no non-trivial autarky for F' , i.e., F' is “lean”. We call F' the *lean kernel* of F . If we add y_2 to the second clause of F , then $y_2 \mapsto \neg x_2$ becomes an A_1 -autarky, and the remaining clause can be satisfied by the E_1 -autarky $y_1 \mapsto 1$. Finding A_1 -autarkies in general is translated to SAT, where the fundamental idea is to *compile* for each clause the *minimal* possibilities to make it a tautology. Our work is related in two dimensions to existing literature: our “model”-based approach, starting so to speak with the Skolem functions (not extracting them as e.g. in [13]), is discussed in Subsection 1, while autarkies for QBF are discussed in Subsection 2.

1) *Starting with the boolean functions (models)*: Our approach in general is similar to [7], concerning the basic framework, while we generalise satisfying assignments (“models”) to autarkies. They consider “ K_1 -models”, which correspond to our A_1 -autarky system (for total assignments, as only studied by [7]). They show that every satisfiable Q2CNF (QCNFs with the matrix a 2CNF) has a K_1 -model (Theorem 3). We remark here that this generalises to DQ2CNF, and thus every satisfiable DQ2CNF is A_1 -satisfiable, but due to space restrictions we don’t prove this here. Furthermore they consider K_2 (conjunctions of positive literals), and mention K_3 , which is the closure of K_1 under conjunction; we will consider the corresponding autarky-systems in future work.

Our notion of an autarky system goes beyond the study of classes of boolean functions, since the autarkies may depend on the formula, as for example for matching autarkies.

2) *Restricted autarkies for QBF*: Autarkies for QBF in the literature only use boolean values (not boolean functions as

*Supported by EPSRC grant EP/S015523/1.

†Supported by Austrian Science Fund (FWF) project W1255-N23.

values, as we do). [8] uses A_0 (especially matching autarkies), and exploits that not having any such autarkies implies that the “deficiency” (the number of clauses minus the number of existential variables) is at least one. [11] consider the matrix F of a QCNF, and just any autarky for F , not differentiating between universal and existential variables. If only existential variables are involved, then this is the A_0 -case. While the main use of an autarky just involving universal variables is based on the following simple fact for CNFs F : if φ is an autarky for F with $v \in \text{var}(\varphi)$, then in the result of applying φ without v to F , the variable v must be pure (occurs only in one sign). And if a universal variable is pure, then it can be assigned (satisfiability-equivalently) so that its instances are (all) *false*. In [11] this is used in the DLL-framework (a simple backtracking algorithm) for QBF, considering at most two variables. We believe that in a DQCNF the only real variables are the existential variables, while the universal variables belong to the value-structure (as in constraint satisfaction). Thus we do not consider assigning the universal variables. For QBF that might be different, when exploiting *duality* (the negation of a QBF is again a QBF — PSPACE is closed under complement), but that would be a more complicated matter.

II. DQCNF AND AUTARKIES

We assume a standard framework for *boolean clause-sets*: a set \mathcal{LIT} of literals is given with a fixed-point free involution $x \in \mathcal{LIT} \mapsto \bar{x} \in \mathcal{LIT}$, the complementation. For a set $L \subseteq \mathcal{LIT}$ of literals we use element-wise complementation $\bar{L} := \{\bar{x} : x \in L\}$. The set of variables is $\mathcal{VA} \subseteq \mathcal{LIT}$ with $\mathcal{VA} \cap \bar{\mathcal{VA}} = \emptyset$ and $\mathcal{VA} \cup \bar{\mathcal{VA}} = \mathcal{LIT}$. A clause is a finite subset $C \subseteq \mathcal{LIT}$ without clashes ($C \cap \bar{C} = \emptyset$), and finally a clause-set is a finite set of clauses. For a clause C we set $\text{var}(C) := (C \cup \bar{C}) \cap \mathcal{VA}$, while for a clause-set F we define $\text{var}(F) := \bigcup_{C \in F} \text{var}(C)$. A **DQCNF** is a 4-tuple (A, E, F, D) , where

- A is the set of universal variables,
- E is the set of existential variables, with $A \cap E = \emptyset$,
- F is a clause-set over $A \cup E$ (i.e., $\text{var}(F) \subseteq A \cup E$),
- D is the dependency-map with $\text{dom}(D) = E$, mapping $v \in E \mapsto D(v) \subseteq A$, the variables on which v depends.

The DQCNFs (A, E, F, D) with empty $A = \emptyset$ correspond to (formal) boolean clause-sets, where E is then the set of (formal) propositional variables (i.e., they might not occur in F). A *universal clause* is a clause $C \in F$ with $\text{var}(C) \subseteq A$ (generalising the unique empty clause of boolean clause-sets). The example from the Introduction is now denoted by $F = (\{x_1, x_2, x_3\}, \{y_1, y_2\}, F_0, \{(y_1, \{x_1, x_2\}), (y_2, \{x_2, x_3\})\})$, with the clause-set $F_0 = \{\{y_1, x_1\}, \{\bar{y}_1, x_2\}, \{\bar{y}_2, \bar{x}_2, x_3\}\}$. We use here the standard definition of a map f as a set of pairs (x, y) with $x \in \text{dom}(f)$ and $y = f(x)$. We now define the semantics of DQCNF, generalising the standard semantics for boolean clause-sets, where now instead of constant boolean functions we consider arbitrary boolean functions as values of assignments. A total boolean assignment f for a set V of variables is a map $f : V \rightarrow \{0, 1\}$, the set of all $2^{|V|}$ many total assignments over V is denoted by $\mathcal{TASS}(V)$. As usual, $f(\bar{v}) = \bar{f}(v)$, with $\bar{0} = 1$ and $\bar{1} = 0$.

A **boolean function** over V is a map $b : \mathcal{TASS}(V) \rightarrow \{0, 1\}$, where we set $\text{var}(b) := V$. For $V' \supseteq V$ and $f' \in \mathcal{TASS}(V')$ we let $b(f') := b(f'|V)$, using the restriction of f' to V . The set of essential variables, denoted by $\text{var}_{\text{es}}(b) \subseteq \text{var}(b)$, is the set of $v \in \text{var}(b)$, such that there exists $f, f' \in \mathcal{TASS}(\text{var}(b))$ with $b(f) \neq b(f')$ and $\forall w \in \text{var}(b) : f(w) \neq f'(w) \Leftrightarrow w = v$. A **total assignment** for a DQCNF (A, E, F, D) (indeed only depending on (E, D)) is a map Φ with $\text{dom}(\Phi) = E$ such that for each $v \in E$ the value $\Phi(v)$ is a boolean function over $D(v)$. Again, as usual, $\Phi(\bar{v}) = \bar{\Phi}(v)$, where for a boolean function b and $f \in \mathcal{TASS}(\text{var}(b))$ we have $\bar{b}(f) = \bar{b}(f)$. In [7, Section 1] our total assignments are called “proper assignments” (in the special case of QBF). Informally, a total assignment Φ **satisfies** (A, E, F, D) (“is a model of (A, E, F, D) ”) if Φ satisfies each clause $C \in F$. This means that for C , understood as disjunction of literals, after replacing the existential literals of C with the boolean functions as given by Φ (negated if the literal is negated), the disjunction of these boolean functions and the remaining universal literals is a tautology over A . Formally, that can be easily expressed as follows: The total assignment Φ for (A, E, F, D) satisfies $C \in F$, if for every $f \in \mathcal{TASS}(A)$ there exists a universal literal $x \in C$ (i.e., $\text{var}(x) \in A$) with $f(x) = 1$ or there exists an existential literal $y \in C$ (i.e., $\text{var}(y) \in E$) with $\Phi(y)(f) = 1$. As usual, we only need to consider $f \in \mathcal{TASS}(A \cap \text{var}(C))$. A DQCNF (A, E, F, D) is **satisfiable** (or “true”) if it has a satisfying (total) assignment, otherwise it is **unsatisfiable** (or “false”).

For boolean clause-sets, an *autarky* for a clause-set F generalises satisfying assignments of F . A partial assignment is a total assignment for some finite set V of variables. An autarky for F is a partial assignment φ , such that every clause $C \in F$ is either satisfied by φ or not “touched” at all, that is, $\text{var}(\varphi) \cap \text{var}(C) = \emptyset$ (using $\text{var}(\varphi) = \text{dom}(\varphi) = V$). See [6] for the basic theory of autarkies for boolean clause-sets. This is naturally generalised to DQCNF (A, E, F, D) . A partial assignment for (A, E, F, D) is a total assignment for some $V \subseteq E$. An **autarky** for (A, E, F, D) is a partial assignment satisfying every $C \in F$ with $\text{var}(\varphi) \cap \text{var}(C) \neq \emptyset$. The above definition that Φ satisfies C is generalised in the natural way here to the partial assignment φ , now requiring that for the existential literal $y \in C$ we have $\text{var}(y) \in \text{var}(\varphi)$.

The empty partial assignment is always an autarky for every DQCNF (never touching any clause), and also every satisfying assignment is an autarky (satisfying every clause). We note that the definitions of satisfying assignments and autarkies for DQCNFs are *semantical*, do not consider any representation of boolean functions (by circuits, say) other than the canonical representation by truth-tables. For each restricted notion of “autarky system” the representation has to be considered anew.

III. BASIC THEOREMS

Generalising [6, Subsection 11.8.3], we call a DQCNF (A, E, F, D) **lean** if for every autarky φ with $\text{var}(\varphi) \subseteq E$ we have $\text{var}(\varphi) = \emptyset$ (i.e., there are no non-trivial autarkies).

For two DQCNF $(A, E, F, D), (A', E', F', D')$ their **union** is defined if $A \cap E' = \emptyset, E \cap A' = \emptyset,$ and D, D' restricted to $E \cap E'$ coincide (i.e., $D|(E \cap E') = D'|(E \cap E')$). In this case we define $(A, E, F, D) \cup (A', E', F', D') := (A \cup A', E \cup E', F \cup F', D \cup D')$, using that the union of two compatible maps (equal on the intersection of their domains) is again a map. It is easy to see that the union of two lean DQCNFs is again lean, and thus every DQCNF has a largest lean sub-DQCNF, its **lean kernel**. Here we use for DQCNFs $(A, E, F, D) \subseteq (A', E', F', D')$ iff $A \subseteq A', E \subseteq E', F \subseteq F',$ and $D \subseteq D'$. We note that thus for the lean kernel (A_0, E_0, F_0, D_0) of (A, E, F, D) we have $A_0 = A,$ while E_0 is exactly the set of existential variables occurring in F_0 . Another trivial property is that for every universal $C \in F$ we have $C \in F_0$ (these clauses can neither be “touched” nor satisfied). We denote the lean kernel by $N_a(A, E, F, D)$ (“N” like “normal form”). The clause-set of $N_a(A, E, F, D)$ is empty iff (A, E, F, D) is satisfiable.

The algorithmic approach for the lean kernel of a DACNF (A, E, F, D) uses **autarky reduction**. For an autarky φ let $\varphi * (A, E, F, D) := (A, E \setminus \text{var}(\varphi), \{C \in F : \text{var}(\varphi) \cap \text{var}(C) = \emptyset\}, D|E)$ be the sub-DQCNF obtained by removing the clauses satisfied by φ . The most basic observations on autarkies is worth stating as a fundamental lemma:

Lemma III.1 *For an autarky φ of (A, E, F, D) the sub-DQCNF $\varphi * (A, E, F, D)$ is satisfiability-equivalent to it.*

Proof: A satisfying assignment of (A, E, F, D) satisfies also $\varphi * (A, E, F, D)$, since just clauses have been removed. And if Φ is a total satisfying assignment for $\varphi * (A, E, F, D)$, then $\varphi \cup \Phi$ is a (partial) satisfying assignment for (A, E, F, D) . \square

The basic fact now is that autarky reduction is confluent, and ends up with the lean kernel. For this it is convenient to allow for autarkies φ of (A, E, F, D) some variables outside of $A \cup E$, so that an autarky for a DQCNF is also an autarky for any sub-DQCNF. Thus if φ, ψ are autarkies for (A, E, F, D) , then φ is also an autarky for $\psi * (A, E, F, D)$. This already proves that if we apply autarky-reduction as long as possible, then the final result is uniquely determined (we have confluence of autarky-reduction). It is also instructive to note that for the result of this chain of reduction we have $\varphi * (\psi * (A, E, F, D)) = (\varphi \circ \psi) * (A, E, F, D)$. Here for partial assignments φ, ψ we define their composition $\varphi \circ \psi$ as the partial assignment with domain $\text{var}(\varphi) \cup \text{var}(\psi)$, which assigns to $v \in \text{var}(\psi)$ the value $\psi(v)$, and to $v \in \text{var}(\varphi) \setminus \text{var}(\psi)$ the value $\varphi(v)$. The basic observation regarding composition is that for autarkies φ, ψ of (A, E, F, D) also their composition $\varphi \circ \psi$ is an autarky of (A, E, F, D) , touching (satisfying) the clauses touched by φ plus the clauses touched by ψ : This is clear for the clauses touched by ψ , since the assignments by ψ are contained in $\varphi \circ \psi$, and additional assignments do not hurt in satisfying a clause. While for the remaining clauses, the clauses touched by φ and *not* touched by ψ , trivially all assignments of φ touching them are contained in $\varphi \circ \psi$.

So for the final result (A_0, E_0, F_0, D_0) of autarky-reduction of (A, E, F, D) , where actually $A_0 = A$ holds, there exists a single “maximal” autarky Φ such that $\Phi * (A, E, F, D) = (A, E_0, F_0, D_0)$, and where Φ can be obtained by a composition of all autarkies of F (in any order).

By definition (A, E_0, F_0, D_0) is lean. And indeed we have $(A, E_0, F_0, D_0) = N_a(A, E, F, D)$, since no clause of $N_a(A, E, F, D)$ can be touched by any autarky of (A, E, F, D) (which is also an autarky for the lean kernel – necessarily a trivial one). An *autark sub-DQCNF* of (A, E, F, D) is a sub-DQCNF (A', E', F', D') such that an autarky φ of (A, E, F, D) exists satisfying all clauses of F' . We see that $(A, E, F \setminus F_0, D)$ is the unique largest autark sub-DQCNF of (A, E, F, D) (satisfied by Φ). We can summarise the results of this section in the Autarky Decomposition Theorem:

Theorem III.2 *For an arbitrary DQCNF (A, E, F, D) we have the following. There exists a (unique) largest lean sub-DQCNF $N_a(A, E, F, D) = (A, E_0, F_0, D_0)$ (with every lean sub-DQCNF $(A', E', F', D') \subseteq (A, E_0, F_0, D_0)$) and a (unique) largest autark sub-DQCNF (A, E, F_1, D) (with every autark sub-DQCNF $(A', E', F', D') \subseteq (A, E, F_1, D)$), where we have $F = F_0 \cup F_1, F_0 \cap F_1 = \emptyset$. Furthermore $\text{var}(F_0) \cap E = E_0$ holds, and there exists an autarky Φ of (A, E, F, D) with $\text{var}(\Phi) = E \setminus E_0$. Every chain of autarky reductions starting with (A, E, F, D) can be extended to $N_a(A, E, F, D)$ (where it necessarily ends).*

IV. AUTARKY SYSTEMS

Due to the high complexity of general autarky-finding for DQCNF, it is vital to allow restricted notions of autarkies (with lower complexity). For that purpose we generalise the notion of “autarky systems” from [6, Section 11.11]. While for a DQCNF (A, E, F, D) we write $\text{Auk}(A, E, F, D)$ for the set of all autarkies, an **autarky system** \mathcal{A} now allows to consider subsets $\mathcal{A}(A, E, F, D) \subseteq \text{Auk}(A, E, F, D)$. There are two basic conditions which make \mathcal{A} an autarky system. First for DQCNFs (A, E, F, D) and $\varphi, \psi \in \mathcal{A}(F)$ it must always hold $\varphi \circ \psi \in \mathcal{A}(F)$ (closure under composition). And second for DQCNFs $(A, E, F, D) \subseteq (A', E', F', D')$ it must always hold $\mathcal{A}(A', E', F', D') \subseteq \mathcal{A}(A, E, F, D)$ (removal of clauses does not remove \mathcal{A} -autarkies). **\mathcal{A} -satisfiability** means satisfiability by a series of \mathcal{A} -autarkies, while **\mathcal{A} -leanness** means there are no nontrivial \mathcal{A} -autarkies.

In order for the results from Section III to generalise, we formulate five conditions for **normal autarky systems**; we formulate the conditions here only informally, since there are no real subtleties involved. First four trivial conditions on \mathcal{A} :

standardised: the presence or absence of formal variables in A, E (not actually occurring in F) is not of relevance

\perp -invariant: universal clauses in F are not of relevance

invariant under variable elimination: removing existential variables from the clauses of F does not affect autarkies of \mathcal{A} which do not use these variables

invariant under renaming: renaming variables (existential or universal) is respected by the autarky system.

These four conditions are always expected to hold. Finally there is a non-trivial condition:

iterative: if we apply an autarky of \mathcal{A} , and take another autarky of \mathcal{A} for the reduction result, then their composition also belongs to \mathcal{A} (for the original DQCNF).

All results of Section III hold for arbitrary normal autarky systems. Especially we have confluence of \mathcal{A} -autarky reduction, and we have the unique decomposition of any DQCNF in an \mathcal{A} -lean and an \mathcal{A} -satisfiable part. If \mathcal{A} is not iterative (but fulfils the other four conditions), then we can consider the unique closure \mathcal{A}^* of \mathcal{A} under iteration, which then is a normal autarky system, and which has exactly the same power of reduction, with the only difference that by \mathcal{A}^* we can always obtain every (iterated) reduction result also in a single step (applying a single \mathcal{A} -autarky – the composition of the autarkies used in the reduction chain). For two autarky systems $\mathcal{A}_1, \mathcal{A}_2$, we can consider their combination $\mathcal{A}_1 + \mathcal{A}_2$, which contains all possible compositions $\varphi \circ \psi, \psi \circ \varphi$ for $\varphi \in \mathcal{A}_1(A, E, F, D)$, $\psi \in \mathcal{A}_2(A, E, F, D)$. If $\mathcal{A}_1, \mathcal{A}_2$ are normal, then so is $\mathcal{A}_1 + \mathcal{A}_2$.

V. A- AND E-SYSTEMS

Consider a DQCNF (A, E, F, D) , and $k \in \mathbb{N}_0$. An \mathbf{A}_k -**autarky** for (A, E, F, D) is an autarky φ , such that for all $v \in \text{var}(\varphi)$ holds $|\text{var}_{\text{es}}(\varphi(v))| \leq k$ (i.e., all boolean functions used in assignments by φ use at most k (universal, essential) variables), while an \mathbf{E}_k -autarky is an autarky φ with $|\text{var}(\varphi)| \leq k$ (i.e., at most k (existential) variables are assigned by φ). \mathbf{A}_0 is just CNF-autarky, and we concentrate on \mathbf{A}_1 , which has some nontrivial power (can satisfy all satisfiable DQ2CNF) and has a nice SAT-translation, and on \mathbf{E}_1 , which has a natural polytime computation. We obtain the composition $\mathcal{E}_1 + \mathcal{A}_1$, which one might consider as the basic “clean-up autarky system”. An example for \mathbf{A}_1 -satisfiability and $\mathbf{A}_0 + \mathbf{E}_1$ -leanness is $\forall x_1, x_2 \exists y_1(x_1) \exists y_2(x_2) : (y_1 \vee y_2 \vee \bar{x}_1) \wedge (y_1 \vee y_2 \vee \bar{x}_2) \wedge (\bar{y}_1 \vee x_1) \wedge (\bar{y}_2 \vee x_2)$, which has the satisfying \mathbf{A}_1 -assignment $y_1 \mapsto x_1, y_2 \mapsto x_2$. An example for \mathbf{E}_1 -satisfiability and \mathbf{A}_1 -leanness is $\forall x_1, x_2 \exists y : (\bar{y} \vee x_1 \vee x_2) \wedge (y \vee \bar{x}_1) \wedge (y \vee \bar{x}_2)$, with the satisfying \mathbf{E}_1 -assignment $y \mapsto x_1 \vee x_2$. An example for $\mathbf{E}_1 + \mathbf{A}_1$ -satisfiability, while not satisfiable by $\mathbf{E}_1 + \mathbf{A}_0$ or \mathbf{A}_1 alone, is obtained by taking the union of the two after renaming y to y_3 .

Since the existence of an ordinary propositional autarky for a clause-set F is NP-complete (see [9]), which is covered by \mathbf{A}_0 -autarkies, deciding whether a DQCNF has a non-trivial \mathbf{A}_k -autarky is NP-hard for every $k \geq 0$ (boolean functions can be just represented by truth-tables here). Deciding the existence and finding some short \mathbf{E}_1 -autarky can be done in polynomial time, as we will discuss below, while for any fixed $k \geq 2$ we are not aware of results in the literature (see also below). We note here that there are no restrictions on the boolean functions used in \mathbf{E}_k autarkies, but we will see that for $k = 1$ a CNF- as well as a DNF-representation can be read off F in linear time. The restriction \mathbf{A}_k yields a normal autarky system for every $k \geq 0$, while the \mathbf{E}_k fulfil all requirements for a normal autarky system except for not being iterative.

1) \mathbf{E}_1 -autarkies: Consider a DQCNF (A, E, F, D) and $V \subseteq E$. Let $([A, E, F, D][V] := (A, V, F[V], D|V)$ be obtained by restriction to V , where $F[V] := \{C \cap (A \cup \bar{A} \cup V \cup \bar{V}) : C \in F \wedge \text{var}(C) \cap V \neq \emptyset\}$ is obtained from F by removing all clauses not containing some variable from V , and removing from the remaining clauses all existential literals with variables not from V . The existence of an autarky φ with $\text{var}(\varphi) = V$ is equivalent to the DQCNF $(A, E, F, D)[V]$ being satisfiable, where the satisfying assignments correspond to the autarkies. So searching for an \mathbf{E}_1 -autarky of (A, E, F, D) means solving for each v the one-existential-variable DQCNF $(A, E, F, D)[\{v\}]$. From any clause C we can remove all universal literals $x \in C$, where there is no existential variable in C depending on the underlying universal variable of x (this is called “universal reduction”). So we have a QCNF $\forall X \exists v : F'$ with exactly one existential variable v to solve. F' is equivalent to $A \rightarrow v \rightarrow B$, where A is a DNF over X , and B is a CNF over X . There is a solution for v iff every conjunct of A has a non-empty intersection with every disjunct of B , in which case either A or B can be chosen as a solution. We note here that for \mathbf{E}_2 we do not get a QCNF with two existential variables, but only such a DQCNF.

2) \mathbf{A}_1 -autarkies: The main strategy to solve the constraint-satisfaction problem of finding some non-trivial \mathbf{A}_1 -autarky for a given (A, E, F, D) is: 1. to explicitly list the possible boolean functions as values of the existential variables, 2. to *compile* for each clause $C \in F$ the minimal possibilities for C to become a tautology, and 3. to make it explicitly part of the SAT-translation that at least one of these *minimal* possibilities for C is fulfilled, if the selector-variable of C is true (a dedicated variable per clause, true iff the clause is touched). For $v \in E$ we thus have $2^{|D(v)|} + 2$ possible values (boolean functions depending on at most one of the variables of $D(v)$) to consider, the two constant boolean functions plus for each universal variable the two associated literals. And there are exactly three types of minimal possibilities for C to become a tautology:

- 1) some $y \in C \cap (E \cup \bar{E})$ is set to 1;
- 2) for some $y \in C \cap (E \cup \bar{E})$ there is $x \in C \cap (A \cup \bar{A})$ with $\text{var}(x) \in D(\text{var}(y))$, and y is set to \bar{x} ;
- 3) for some $y, y' \in C \cap (E \cup \bar{E})$, $y \neq y'$, there is $x \in D(\text{var}(y)) \cap D(\text{var}(y'))$, and y is set to x and y' to \bar{x} .

The translation into a SAT-problem is now in principle not difficult, but effort and experimentation is needed for the representation of AMO-constraints (at-most-one), and for encodings of nonboolean values (direct or logarithmic encodings). The results on the normalforms by the basic autarky system $\mathbf{E}_1 + \mathbf{A}_1$ on all known QCNF and DQCNF instances (as in QBFLIB), will be made available online; for the DQCNF instances of QBFEVAL'18 see [10].

VI. CONCLUSION

$\mathbf{E}_1 + \mathbf{A}_1$ establishes the basic autarky system. All the SAT-theory on autarkies can be combined with many interesting classes of boolean functions to yield autarky systems. The use of autarkies in pre- and inprocessing then is the next question.

REFERENCES

- [1] Olaf Beyersdorff, Leroy Chew, Renate A Schmidt, and Martin Suda. Lifting QBF resolution calculi to DQBF. In Creignou and Berre [4], pages 490–499. doi:10.1007/978-3-319-40970-2_30.
- [2] Armin Biere, Marijn J.H. Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, February 2009.
- [3] Uwe Bubeck and Hans Kleine Büning. Dependency quantified Horn formulas: Models and complexity. In Armin Biere and Carla P. Gomes, editors, *Theory and Applications of Satisfiability Testing - SAT 2006*, volume 4121 of *Lecture Notes in Computer Science*, pages 198–211. Springer, 2006. doi:10.1007/11814948_21.
- [4] Nadia Creignou and Daniel Le Berre, editors. *Theory and Applications of Satisfiability Testing - SAT 2016*, volume 9710 of *Lecture Notes in Computer Science*. Springer, 2016. doi:10.1007/978-3-319-40970-2.
- [5] Hans Kleine Büning and Uwe Bubeck. Theory of quantified boolean formulas. In Biere et al. [2], chapter 23, pages 735–760. doi:10.3233/978-1-58603-929-5-735.
- [6] Hans Kleine Büning and Oliver Kullmann. Minimal unsatisfiability and autarkies. In Biere et al. [2], chapter 11, pages 339–401. doi:10.3233/978-1-58603-929-5-339.
- [7] Hans Kleine Büning, K. Subramani, and Xishun Zhao. On boolean models for quantified boolean Horn formulas. In Enrico Giunchiglia and Armando Tacchella, editors, *Theory and Applications of Satisfiability Testing 2003*, volume 2919 of *Lecture Notes in Computer Science*, pages 93–104, Berlin, 2004. Springer. doi:10.1007/978-3-540-24605-3_8.
- [8] Hans Kleine Büning and Xishun Zhao. Computational complexity of quantified Boolean formulas with fixed maximal deficiency. *Theoretical Computer Science*, 407(1-3):448–457, 2008. doi:10.1016/j.tcs.2008.07.022.
- [9] Oliver Kullmann. Lean clause-sets: Generalizations of minimally unsatisfiable clause-sets. *Discrete Applied Mathematics*, 130:209–249, 2003. doi:10.1016/S0166-218X(02)00406-7.
- [10] Oliver Kullmann and Ankit Shukla. E1+A1 autarkies for the DQBF track of QBFEVAL’18. Webpage, August 2019. <http://cs.swan.ac.uk/~csoliver/Artikel/FMCAD2019-Autarkies.html>.
- [11] Jens Rühmkorf. Using autarky to evaluate quantified boolean formulae. In Wolfgang Gentzsch, Pascal Lorenz, and Oana Dini, editors, *ADVCOMP 2010: The Fourth International Conference on Advanced Engineering Computing and Applications in Sciences*, IARIA Conference, pages 154–159, 2019. Available from: http://www.thinkmind.org/index.php?view=article&articleid=advcomp_2010_8_10_20139.
- [12] Friedrich Slivovsky and Stefan Szeider. Quantifier reordering for QBF. *Journal of Automated Reasoning*, 56(4):459–477, April 2016. doi:10.1007/s10817-015-9353-1.
- [13] Karina Wimmer, Ralf Wimmer, Christoph Scholl, and Bernd Becker. Skolem functions for DQBF. In Cyrille Artho, Axel Legay, and Doron Peled, editors, *ATVA 2016: Automated Technology for Verification and Analysis*, volume 9938 of *Lecture Notes in Computer Science*, pages 395–411. Springer, 2016. doi:10.1007/978-3-319-46520-3_25.
- [14] Ralf Wimmer, Karina Gitina, Jennifer Nist, Christoph Scholl, and Bernd Becker. Preprocessing for DQBF. In Marijn H. Heule and Sean Weaver, editors, *Theory and Applications of Satisfiability Testing - SAT 2015*, volume 9340 of *Lecture Notes in Computer Science*, pages 173–190. Springer, 2015. doi:10.1007/978-3-319-24318-4_13.
- [15] Ralf Wimmer, Christoph Scholl, Karina Wimmer, and Bernd Becker. Dependency schemes for DQBF. In Creignou and Berre [4], pages 473–489. doi:10.1007/978-3-319-40970-2_29.