List of Formula Families - v1

Noel Arteche

October 2019

1 List of Formula Families

1. Chen Formulas of Type 1 (CNF)
2. Chen Formulas of Type 2 (circuits)
3. Generalized Chen Formulas of Type 2 (circuits)
4. QParity formulas (CNF / circuits)
5. Chromatic Formulas (circuits)
6. Janota Formulas (CNF)
7. KBKF Formulas (CNF)

2 Definitions

2.1 Chen Formulas of Type 1

These formulas have linear size proofs in the QU-Resolution proof system. These are defined in CNF. They constitute the paradigmatic example of a CNF definition by the explicit declaration of variables, quantifier blocks and conjunction of clauses.

Definition 1 (Chen Formulas of Type 1) *For every positive natural number n and every $i \in \{0\} \cup [n]$, let X_i be the set of variables $\{x_{i,j,k} \mid j,k \in \{0,1\}\}$. Analogously, we have the set $X'_i = \{x_{i,j,k} \mid j,k \in \{0,1\}\}$, except for $i = 0$, when X'_i is not defined. Additionally, we have, for every positive i, a variable y_i. With these variables, we define $\vec{P_n}$ to be the quantifier prefix $\exists X_0 \exists X'_1 \forall y_1 \exists X_1 ... \exists X'_n \forall y_n \exists X_n$. Now, we define the following sets of clauses:*

- $B = \{(\neg x_{0,j,k}) \mid j,k \in \{0,1\}\} \cup \{(x_{n,j,0} \vee x_{n,j,1}) \mid j \in \{0,1\}\}$
- *For every $i \in [n]$ and every $j \in \{0,1\}$,* $H_{i,j} = \{(\neg x'_{i,0,k} \vee \neg x'_{i,1,l} \vee x_{i-1,j,0} \vee x_{i-1,j,1}) \mid k,l \in \{0,1\}\}$

- *For every $i \in [n]$, $T_i = \{(\neg x_{i,0,k} \vee y_i \vee x'_{i,0,k} \mid k \in \{0,1\}\} \cup \{(\neg x_{1,i,k} \vee \neg y_i \vee x'_{i,1,k}) \mid k \in \{0,1\}\}$*

For every positive natural number n, a QBF instance Φ is a Chen Formula of Type 1 *when $\Phi = \vec{P_n} : \phi$, where $\vec{P_n}$ is the prefix vector defined before and ϕ is the Boolean formula obtained from the conjunction of all the clauses in B, $H_{i,j}$ and T_i for every $i \in [n]$ and every $j \in \{0,1\}$.*

2.2 Chen Formulas of Type 2

Definition 2 (Chen Formulas of Type 2) *Let n be a positive natural number, and let $\vec{P_n}$ be the quantifier prefix $\exists x_1 \forall y_1 \ldots \exists x_n \forall y_n$. Now, we consider Boolean circuits $\phi_{n,j}$ such that $\phi_{n,j}$ is true if and only if $j + \sum_{i=1}^{n}(x_i + y_i) \not\equiv n \pmod 3$. A QBC $\Phi_n = \vec{P_n} : \phi_{n,0}$ is called a* Chen Formula of Type 2.

Chen Formulas of Type 2 can be built as circuits in size linear in n. Their representation as such is as follows:

For every $k \in \{1, \ldots, n\}$ and every $m \in \{0, 1, 2\}$, we will now consider the auxiliary circuits μ_m^k over variables $(x_1, y_1, \ldots, x_k, y_k)$, which are defined so that they verify the following property:

$$\mu_m^k = 1 \Leftrightarrow \sum_{i=1}^{k}(x_i + y_i) \equiv m \pmod 3$$

For $k = 1$, the μ-circuits are:

$$\mu_0^1 = \neg x_1 \wedge \neg y_1$$
$$\mu_1^1 = x_1 \oplus y_1$$
$$\mu_2^1 = x_1 \wedge y_1$$

If we have circuits μ_m^k for every k up to $n-1$, we can easily obtain the ones for $k = n$:

$$\mu_0^n = (\mu_0^{n-1} \wedge \neg x_n \wedge \neg y_n) \vee (\mu_1^{n-1} \wedge x_n \wedge y_n) \vee (\mu_2^{n-1} \wedge (x_n \oplus y_n))$$

$$\mu_1^n = (\mu_0^{n-1} \wedge (x_n \oplus y_n)) \vee (\mu_1^{n-1} \wedge \neg x_n \wedge \neg y_n) \vee (\mu_2^{n-1} \wedge x_n \wedge y_n)$$

$$\mu_2^n = (\mu_0^{n-1} \wedge x_n \wedge y_n) \vee (\mu_1^{n-1} \wedge (x_n \oplus y_n)) \vee (\mu_2^{n-1} \wedge \neg x_n \wedge \neg y_n)$$

Now, we can easily express our formula ϕ as

$$\phi = \neg \mu^n_{n \bmod 3}$$

We can use a Tseitin transformation a convert ϕ into CNF by adding the auxiliary μ-circuits as variables. Depending on where they are added in the quantifier proof length may vary.

2.3 Generalized Chen Formulas of Type 2

Chen Formulas of Type 2 are defined via a precise $\pmod 3$ modular property. We can, however, extend this definition to any $\pmod p$ in general, for any $p \geq 3$ (for $p = 2$ the generalization does not apply, as the formulas are actually satisfiable because they would contain no universal quantifiers; for $p = 1$ the formulas are not defined, as there would be $(p-1)n = (1-1)n = 0$ variables). Note that for $p = 3$ we had a total of $2n$ variables. In general, for the case p we will have $(p-1)n$ variables. This is because at each level the sum of the bits should add up to one of the possible values of the equivalence classes of the $\pmod p$ relation. For $p = 3$ the possible remainders (this is, the possible values of the sum) are $\{0, 1, 2\}$. Thus, in general, we need enough variables to add up from 0 to $p-1$, thus the $(p-1)$ variables at each level. Since there are n levels, that accounts for the $(p-1)n$ variables. The prefix vector will consist of an alternation of existential and universal quantifiers for each level-block, starting with $\exists$.

Generalized Chen Formulas are interesting because now the number of variables and intermediate gates depends not only on n but on a second parameter p. In fact, we could consider Chen's third parameter as well, k, although this this is not of much use in practice and we always consider formulas with $k = 0$.

Definition 3 (Generalized Chen Formulas of Type 2) *Let n and p be two positive natural numbers, $p \geq 3$ and let $\vec{P_{n,p}}$ be the quantifier prefix*

$$\exists x_{1,0} \forall x_{1,1} \ldots \exists x_{1,p-1} \ldots \exists x_{n,0} \forall x_{n,1} \ldots \exists x_{n,p-1}$$

Now, we consider Boolean circuits $\phi_{n,p,k}$ such that $\phi_{n,p,k}$ is true if and only if

$$k + \sum_{i=1}^{n} \sum_{j=0}^{p-1} x_{i,j} \not\equiv n \pmod p$$

A QBC $\Phi_{n,p} = \vec{P_{n,p}} : \phi_{n,p,0}$ is called a General Chen Formula of Type 2.

We can again use μ-circuits to build any general version of Chen Formulas of Type 2.

For every $k \in \{1, \ldots, n\}$ and every $m \in \{0, \ldots, p-1\}$, we will now consider the auxiliary circuits μ_m^k over variables $(x_{1,0}, \ldots, x_{1,p-1}, \ldots, x_{k,0}, \ldots, x_{k,p-1})$, which are defined so that they verify the following property:

$$\mu_m^k = 1 \Leftrightarrow \sum_{i=1}^{k} \sum_{j=0}^{p-1} x_{i,j} \equiv m \pmod p$$

2.4 QParity formulas

The QParity formulas were first introduced in [9] and later used in [7] to show that *extended Q-Resolution* can find short proofs of them, while *weak extended Q-Resolution* needs exponential size to show their unsatisfiability.

They are defined as follows:

Definition 4 (QPARITY formulas) *Let n be a positive natural number. We define the quantifier prefix $\vec{P_n} = \exists x_1 \ldots \exists x_n \forall z$.*

We define a basic auxiliary circuit t_2 as $t_2 = x_1 \oplus x_2$ and for $i \in \{3, \ldots, n\}$ we define auxiliary t-circuits to be:

$$t_i = t_{i-1} \oplus x_i$$

Then we define $\rho_n = (z \vee t_n) \wedge (\neg z \vee \neg t_n)$. The QBF instance will be

$$\text{QPARITY}_n = \vec{P_n} : \rho_n$$

The formulas express that there exists an assignment to the x variables such that $x_1 \oplus \cdots \oplus x_n$ is neither 0 nor 1, an obvious contradiction.

Note that there is an alternative definition of this formulas, directly written in CNF. The t-circuits expressing the $\oplus$ operation and assignment can be expressed as the conjunction, for $i \in \{3, \ldots, n\}$:

$$(\neg t_{i-1} \vee \neg x_i \vee \neg t_i) \wedge (t_{i-1} \vee x_i \vee \neg t_i) \wedge (\neg t_{i-1} \vee x_i \vee t_i) \wedge (t_{i-1} \vee \neg x_i \vee t_i)$$

Then, we could define a new quantifier prefix with this additional variables:

$$\exists x_1 \ldots \exists x_n \forall z \exists t_2 \ldots \exists t_n$$

Of course, this is nothing else than a Tseitin transformation in disguise.

From the QPARITY we can extend the QINNERPRODUCT where, each variable x_i is now interchanged with the conjunction of two new variables, y_i and z_i. This family is interesting because it provides a clear example of how circuits are converted into CNF and vice versa.

2.5 Chromatic Formulas

The Chromatic Formulas encode the *Chromatic Number Problem*: give a graph G and a positive natural number k, does the graph G have chromatic number k? In other words, is k the mimimum number for which G is k-colorable?

This problem can be encoded using a QBF instance. These formulas are actually dependent on three parameters: the number of nodes n, the value k we want to check and the actual graph G, whose adjancy matrix is needed for knowing when two nodes are adjacent. In our case, we will consider the case of a complete graph G with n nodes, which has $n(n-1)/2$ edges. We consider it this way because a complete graph's chromatic number is n, so for any $k < n$ the formula encoding the problem will be false.

Regarding the definition's structure, it is interesting because it has two parameters, n and k, which are used to define the number of variables. The definition is extracted from [11].

Definition 5 (Chromatic formulas) *Let n and k be positive natural numbers. We define variables $x_{i,j}$ for $i \in [n]$ and $j \in [k]$ and $y_{i,j}$ for $i \in [n]$ and $j \in [k-1]$ (semantically, any of these variables is set to 1 if and only if node i is set to have colour j). Then we define the following two subformulas:*

$$\Gamma = \bigwedge_{i \in [n]} (x_{i,1} \vee \cdots \vee x_{i,k}) \wedge \bigwedge_{\substack{i \in [n] \\ j \neq j' \in [k]}} (\neg x_{i,j} \vee \neg x_{i,j'}) \wedge \bigwedge_{\substack{(i,i') \in [n]^2, i \neq i' \\ j \in [k]}} (\neg x_{i,j} \vee \neg x_{i',j})$$

$$\Delta = \bigvee_{i \in [n]} (\neg y_{i,1} \wedge \cdots \wedge \neg y_{y,k-1}) \vee \bigvee_{\substack{i \in [n] \\ j \neq j' \in [k-1]}} (y_{i,j} \wedge y_{i,j'}) \vee \bigvee_{\substack{(i,i') \in [n]^2, i \neq i' \\ j \in [k-1]}} (y_{i,j} \wedge y_{i',j})$$

Semantically, Γ is true when the x variables form a legal k-coloring, while Δ is true only when the y variables do not form a legal $(k-1)$-coloring. Thus, the Chromatic Formula *is*

$$K_k^n = \exists x_{1,1} \dots x_{1,k} \dots x_{n,1} \dots x_{n,k} \forall y_{1,1} \dots y_{1,k-1} \dots y_{n,1} \dots y_{n,k-1} : \Gamma \wedge \Delta$$

Chromatic Formulas' definition is interesting because subformula Δ is actually in Disjunctive Normal Form, so it does not directly encode into into a CNF QBF. Thus, it provides a good example of formulas where the intermediate gates are not defined explicitly but they must be added to be able to construct them.

2.6 Janota Formulas

The Janota formulas consider a game played by the existential the universal player over the cartesian product of two sets $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_n\}$. They were presented in [F5].

Definition 6 (Janota formulas) *Let n be a positive natural number. We consider the sets of variables $X = \{x_{i,j} | i, j \in [n]\}$ and $L = \{a_i | i \in [n]\} \cup \{b_i | i \in [n]\}$. The* Janota formula *of size n is defined with the conjunction of the following clauses:*

$$x_{i,j} \vee z \vee a_i, i \in [n]$$

$$\neg x_{i,j} \vee \neg z \vee b_i, i \in [n]$$

$$\bigvee_{i \in [n]} \neg a_i$$

$$\bigvee_{i \in [n]} \neg b_i$$

The final QBF instance is:

$$\mathrm{J}_n = \exists X \forall z : \phi$$

where ϕ is the previous conjunction of clauses.

Janota formulas are not interesting because of their clauses, but because of the quantifier vector, as it provides a case were we define a vector based on a block and not on quantification over each one of the variables.

2.7 KBKF formulas

The Kleine-Büning-Karpinski-Flögel formulas, or KBKF for short, were first introduced in [F1] and have been later used for other purposes. There are several variants of this family based on for which proof system we want them to be difficult. Here we present the original definition, as formulated in [F3], as the variants do not presents any particular differences regarding their definition style.

KBKF formulas are interesting because of their somewhat cumbersome structure, but they show that, in essence, they are no more than a a conjunction of clauses giving rise to a CNF QBF.

Definition 7 (KBKF formulas) *Let t be a positive natural number. We define the prefix*

$$\vec{P}_t = \exists y_0 y_{1,0} y_{1,1} \forall x_1 \exists y_{2,0} y_{2,1} \forall x_2 \dots \forall x_{t-1} \exists y_{t,0} y_{t,1} \forall x_t \exists y_{t+1} \dots y_{t+t}$$

Now we define the following clauses:

$$C_{\_} = \neg y_0$$

$$C_0 = y_0 \vee \neg y_{1,0} \vee y_{1,1}$$

$$C_i^0 = y_{i,0} \vee x_i \vee \neg y_{i+1,0} \vee \neg y_{i+1,1}$$

$$C_i^1 = y_{i,1} \vee x_i \vee \neg y_{i+1,0} \vee \neg y_{i+1,1}, i \in [t-1]$$

$$C_t^0 = y_{t,0} \vee x_t \vee \neg y_{t+1} \vee \dots \vee \neg y_{t+t}$$

$$C_t^0 = y_{t,1} \vee x_t \vee \neg y_{t+1} \vee \dots \vee \neg y_{t+t}$$

$$C_{t+i}^0 = x_i \vee y_{t+i}$$

$$C_{t+i}^0 = \neg x_i \vee y_{t+i}, i \in [t]$$

The QBF is formed with the defined prefix $\vec{P}_t$ and the matrix obtained from the conjunction of all those clauses.

The KBKF family is interesting because a single parameter t defines a quite intricate quantifier vector, with some work on the prefix to perform. This is the first quantifier prefix were some of the subindices contain operations, such as $t + t$, rather than simple iteration over their range.