

# MAX-PLANCK-INSTITUT FÜR INFORMATIK

The Circuit Subfunction Relations are  
 $\Sigma_2^P$ -complete

Bernd Borchert and Desh Ranjan

MPI-I-93-121

May 1993

  
INFORMATIK  
F C  
T I K

Im Stadtwald  
W 6600 Saarbrücken  
Germany

The Circuit Subfunction Relations are  
 $\Sigma_2^P$ -complete

Bernd Borchert and Desh Ranjan

MPI-I-93-121

May 1993

# The Circuit Subfunction Relations are $\Sigma_2^P$ -complete

Bernd Borchert  
Universität Heidelberg

Desh Ranjan  
Max-Planck-Institut, Saarbrücken

## Abstract

We show that given two Boolean circuits  $f$  and  $g$  the following three problems are  $\Sigma_2^P$ -complete: (1) Is  $f$  a c-subfunction of  $g$ , i.e. can one set some of the variables of  $g$  to 0 or 1 so that the remaining circuit computes the same function as  $f$ ? (2) Is  $f$  a v-subfunction of  $g$ , i.e. can one change the names of the variables of  $g$  so that the resulting circuit computes the same function as  $f$ ? (3) Is  $f$  a cv-subfunction of  $g$ , i.e. can one set some variables of  $g$  to 0 or 1 and simultaneously change some names of the other variables of  $g$  so that the new circuit computes the same function as  $f$ ? Additionally we give some bounds for the complexity of the following problem: Is  $f$  isomorphic to  $g$ , i.e. can one change the names of the variables bijectively so that the circuit resulting from  $g$  computes the same function as  $f$ ?

## 1 Introduction

The questions which we are concerned with in this paper can be very roughly stated as follows - given two Boolean circuits, what is the complexity of determining if the functions computed by them satisfy a certain relationship. For instance, given two boolean circuits  $f$  and  $g$ , how difficult is it to decide

if  $f$  computes the same function as  $g$ ? We are interested in relations that relate to equivalence and which are natural extensions of equivalence, for example the notion of subfunctions.

The paper is organized as follows. After the preliminaries we will give in section 3 the necessary background and definitions required to formulate the questions that we are interested in. In section 4 we prove the main results in this paper. Roughly stated, the first result shows that given two circuits  $f$  and  $g$  the problem of determining that  $f$  is a “restriction” of  $g$  is  $\Sigma_2^P$ -complete. The second result shows that allowing only the change of variables also results in a  $\Sigma_2^P$ -complete problem, and the third result shows that to determine if  $f$  results by a restriction and an additional change of the names of the variables of  $g$  is also  $\Sigma_2^P$ -complete. In section 5 we present some results about the complexity of the circuit isomorphism problem - the problem of determining if  $f$  results from  $g$  by a one-to-one renaming of the variables.

## 2 Preliminaries

### 2.1 Circuits

Let  $C = \{0, 1\}$  be the set of Boolean constants and let  $V$  be a countably infinite set of variables. Let CIR be the set of (finite Boolean) circuits with input variables from  $V$ , using the constants 0,1 and  $\wedge$ -gates,  $\vee$ -gates and  $\neg$ -gates for conjunction, disjunction and negation, respectively. Note that already a single variable or constant is a circuit. For a given circuit  $f$  we call a variable which is used by  $f$  as an input variable a variable *occurring in  $f$* .

A total assignment  $\tau$  is a function from  $V$  to  $C$ . An assignment as defined here is an infinite object. This is to facilitate the formulation of the problems that we are interested in. Given any total assignment a circuit evaluates either to 0 or 1 in the usual way. Clearly, for any circuit, only a finite part of a total assignment is relevant for evaluation.

Two circuits  $f$  and  $g$  are said to be *equivalent*, denoted  $f \equiv g$ , iff for all total assignments they evaluate to the same value. In other words, two circuits are equivalent if they compute the same function. We give two brief examples

to clarify the definitions.

**Examples:** The circuit  $(x_1 \vee \neg x_1)$  is equivalent to  $(x_2 \vee \neg x_2)$  though they use different variables.  $(x_1 \vee x_2)$  is *not* equivalent to  $(x_3 \vee x_4)$ , because a total assignment  $\tau$  with  $\tau(x_1) = 0, \tau(x_2) = 0, \tau(x_3) = 1, \tau(x_4) = 1$  evaluates the first circuit to 0 and the second to 1. Later we will define the notion of circuit isomorphism and we will see that these two circuits are isomorphic.

A circuit  $f$  is dependent on a variable  $x$  if there are two total assignments which differ only on  $x$  such that  $f$  evaluates to different values on these two assignments. It is clear that a circuit is independent of each variable not occurring in it, but also it may be independent of a variable occurring in it.

## 2.2 Parity

Given  $f_1, \dots, f_n \in \text{CIR}$  we define  $\oplus(f_1, \dots, f_n)$  as the circuit that computes parity of the outputs of  $f_1, \dots, f_n$ . Formally, one defines as follows:  $\oplus(f_1) := f_1$ ,  $\oplus(f_1, f_2) := ((f_1 \wedge \neg f_2) \vee (\neg f_1 \wedge f_2))$ ,  $\oplus(f_1, \dots, f_{n+1}) := \oplus(\oplus(f_1, \dots, f_n), f_{n+1})$ .

Besides properties like  $\oplus(0, f_1, \dots, f_n) \equiv \oplus(f_1, \dots, f_n)$ ,  $\oplus(g, g, f_1, \dots, f_n) \equiv \oplus(f_1, \dots, f_n)$  or  $\oplus(\neg g, f_1, \dots, f_n) \equiv \neg \oplus(g, f_1, \dots, f_n)$  we will often refer to the following properties of the parity-circuits:

(P1)  $\oplus(f_1, \dots, f_n) \equiv \oplus(f_{\sigma(1)}, \dots, f_{\sigma(n)})$  for every permutation  $\sigma$  of  $\{1, \dots, n\}$

If  $v_1, \dots, v_n \in V$  are all different from each other we also have:

(P2)  $\oplus(v_1, \dots, v_n)$  is dependent on each  $v_i$

(P3)  $\oplus(v_1, \dots, v_n, 1) \equiv \oplus(v_1, \dots, v_n, f) \iff 1 \equiv f$ .

## 2.3 Computational Complexity

We assume familiarity with the complexity classes within the Polynomial-time Hierarchy like P, NP, co-NP,  $\Sigma_k^P$  and also with the notion of polynomial-time many-one reducibility ( $\leq_m^P$ ), equivalence ( $\equiv_m^P$ ) and completeness.

We may assume from now on that circuits are defined in some usual way as strings over some fixed alphabet.

We know that  $\text{SAT} = \{f \in \text{CIR} \mid f \neq 0\}$  is NP-complete and that  $\text{TAUT} =$

$\{f \in \text{CIR} \mid f \equiv 1\}$  is co-NP-complete.

For a binary relation  $R$  on circuits let  $\langle\langle R \rangle\rangle$  denote the corresponding set of coded pairs  $\{\langle f, g \rangle \mid f, g \in \text{CIR} \text{ and } (f, g) \in R\}$ . With this notation **TAUT** and  $\langle\langle \equiv \rangle\rangle$  are polynomial-time many-one equivalent by the reductions  $f \rightarrow \langle 1, f \rangle$  and  $\langle f, g \rangle \rightarrow \neg(\oplus(f, g))$ .

Let  $\exists\forall\text{CIR}$  denote the set of quantified circuits  $\exists x_1 \dots \exists x_m \forall y_1 \dots \forall y_n f$  such that  $f$  is a circuit, the variables  $x_1, \dots, x_m, y_1, \dots, y_n$  are all different from each other and are exactly the variables occurring in  $f$ . Note that  $\exists\forall\text{CIR}$  is recognizable in polynomial time. We know from [2] that  $\mathbf{B}_2 := \{g \in \exists\forall\text{CIR} \mid g \text{ evaluates to true}\}$  is  $\Sigma_2^P$ -complete.

## 3 Replacements and Relations

### 3.1 Replacements

In general, a *replacement* is a function  $V \rightarrow \text{CIR}$ , and the set of all replacements is called  $\mathbf{R}$ . For a circuit  $f \in \text{CIR}$  and a given replacement  $\rho \in \mathbf{R}$  the *application* of  $\rho$  to  $f$  – denoted by  $f_\rho$  – represents the circuit resulting from  $f$  where every variable  $v$  is replaced by  $\rho(v)$ . Of course, for a circuit  $f$  and a replacement  $\rho$  the construction of  $f_\rho$  only depends on the  $\rho$ -values of the variables occurring in  $f$ .

**Example:** Let  $f$  be the circuit  $((x \vee y) \wedge x)$ , and let  $\rho$  be a replacement with  $\rho(x) = (x \wedge z)$  and  $\rho(y) = 1$ . Then  $f_\rho$  is the circuit  $((x \wedge z) \vee 1) \wedge (x \wedge z)$ .

Often we use  $f_{\rho_1, \dots, \rho_n}$  to denote application of  $\rho_n$  to  $f_{\rho_1, \dots, \rho_{n-1}}$ .

Note that application of a replacement respects equivalence in a double sense: (1)  $f \equiv g \implies f_\rho \equiv g_\rho$  for all  $f, g \in \text{CIR}, \rho \in \mathbf{R}$  and (2) for all  $f \in \text{CIR}$ , if for two replacements  $\rho, \sigma \in \mathbf{R}$   $\rho(y) \equiv \sigma(y)$  for all  $y \in V$  then  $f_\rho \equiv f_\sigma$ .

### 3.2 Special Sets of Replacements

We already defined the set of total assignments  $\mathbf{R}_t = \{\rho \in \mathbf{R} \mid \rho(V) \subseteq C\}$  in the preliminaries. We now define some more sets of replacements:

- the set of *c-mappings* or *partial assignments*  $\mathbf{R}_c = \{\rho \in \mathbf{R} \mid \rho(V) \subseteq C \cup V \text{ and if } \rho(v) \in V \text{ then } \rho(v) = v\}$
- the set of *v-mappings*  $\mathbf{R}_v = \{\rho \in \mathbf{R} \mid \rho(V) \subseteq V\}$
- the set of *cv-mappings*  $\mathbf{R}_{cv} = \{\rho \in \mathbf{R} \mid \rho(V) \subseteq C \cup V\}$
- the set of *renamings*  $\mathbf{R}_r = \{\rho \in \mathbf{R} \mid \rho(V) \subseteq V \text{ and } \rho \text{ is bijective}\}$

A c-mapping – which will always call partial assignment – allows to set some of the variables to constants, leaving the other variables unchanged. A v-mappings allows to change the names of variables. A cv-mapping combines the power of c-mappings and v-mappings by allowing for each variable either to set it to a constant or to change its name. A renaming is a bijective v-mapping. Note that  $\mathbf{R}_f \subset \mathbf{R}_c \subset \mathbf{R}_{cv}$  and  $\mathbf{R}_r \subset \mathbf{R}_v \subset \mathbf{R}_{cv}$ .

### 3.3 Relations on Circuits

Given the these four sets of replacements we uniformly define corresponding relations  $\ll_c, \ll_v, \ll_{cv}$  and  $\sim$  on circuits:

- $f \ll_c g \iff \exists \rho \in \mathbf{R}_c : f \equiv g_\rho$  (*f is a c-subfunction of g*)
- $f \ll_v g \iff \exists \rho \in \mathbf{R}_v : f \equiv g_\rho$ . (*f is a v-subfunction of g*)
- $f \ll_{cv} g \iff \exists \rho \in \mathbf{R}_{cv} : f \equiv g_\rho$ . (*f is a cv-subfunction of g*)
- $f \sim g \iff \exists \rho \in \mathbf{R}_r : f \equiv g_\rho$ . (*f is isomorphic to g*)

In other words: *f* is a c-subfunction of *g* iff there is way of setting some variables of *g* to 0-1 such that the resulting circuit computes the same function as *f*. Similarly, *f* is a v-subfunction of *g* iff there is a way of changing the names of variables (we allow to give different variables the same new name) of *g* such that the resulting circuit *g'* is equivalent to *f*. Likewise, *f* is a cv-subfunction of *g* iff there is a way of setting some variables of *g* to 0-1 and simultaneously changing the names of some other variables of *g* so that the resulting circuit *g'* is equivalent to *f*. And *f* is isomorphic to *g* if *f* is

equivalent to the circuit  $g'$  which results from  $g$  after a bijective renaming of the variables  $V$ .

Obviously we have  $f \equiv g \implies f \ll_c g \implies f \ll_{cv} g$ , and  $f \equiv g \implies f \sim g \implies f \ll_v g \implies f \ll_{cv} g$ . By the following examples these implications do not hold for the opposite direction.

**Examples:** Let  $x, y, z$  be three different variables. (1) The circuit  $(x \wedge z)$  is a  $c$ -subfunction of  $((x \wedge y) \wedge z)$  by setting  $y$  to 1, though the two circuits are not equivalent. (2) The circuit  $x$  is a  $cv$ -subfunction but not a  $c$ -subfunction of the circuit  $(y \vee z)$ . (3) The circuits  $(x \wedge y)$  and  $\neg(\neg y \vee \neg z)$  are isomorphic but not equivalent. (4) The circuit  $x$  is a  $v$ -subfunction of  $(y \wedge z)$  by setting both  $y$  and  $z$  to  $x$ , but the two circuits are not isomorphic. (5) The circuit  $x$  is a  $cv$ -subfunction but not a  $v$ -subfunction of the circuit  $\oplus(y, z)$ .

Observe that for circuits  $f$  and  $g$  whether or not  $f \ll_c g$  ( $f \ll_v g$ ,  $f \ll_{cv} g$ ) depends only on the functions computed by  $f$  and  $g$ . This justifies the nomenclature "subfunction".

Note that the first three relations are preorders and the last is an equivalence relation.

In the next section we will investigate the computational complexity of the first three relations, leaving the last relation to section 5.

## 4 The Complexity of the Subfunction Relations

In this section we will locate the computational complexity of the relations  $\ll_c$ ,  $\ll_v$  and  $\ll_{cv}$ . The first observation is the following:

**Lemma 1**  $\langle\langle\ll_c\rangle\rangle, \langle\langle\ll_v\rangle\rangle, \langle\langle\ll_{cv}\rangle\rangle$  are in  $\Sigma_2^P$ .

**Proof:** Note that replacements for a given circuit  $f$  can be encoded by only coding the values for the variables occurring in  $f$ . To decide the question  $f \ll_c g$  for a given pair of circuits  $\langle f, g \rangle$ , just guess a code for a partial assignment  $\rho$  for  $g$ , and check for each code of a total assignment  $\tau$  for  $f$  if



$f_\tau$  evaluates to the same constant as  $g_{\rho,\tau}$ . Thus,  $\langle\langle\langle c \rangle\rangle\rangle$  is in  $\Sigma_2^p$ . The proof for  $\langle\langle\langle cv \rangle\rangle\rangle$  and  $\langle\langle\langle v \rangle\rangle\rangle$  works in an analog way.  $\square$

We shall now state and prove the main result in this paper.

**Theorem 1**  $\langle\langle\langle c \rangle\rangle\rangle, \langle\langle\langle v \rangle\rangle\rangle, \langle\langle\langle cv \rangle\rangle\rangle$  are  $\Sigma_2^p$ -complete.

Proof: Since we know that these problems are in  $\Sigma_2^p$  and that  $\mathbf{B}_2$  is  $\Sigma_2^p$ -complete, to show  $\Sigma_2^p$ -completeness it suffices to give for each problem a polynomial-time many-one reduction from  $\mathbf{B}_2$  to it.

In the proofs we will use the following equivalent definition of  $\mathbf{B}_2$ : a quantified circuit  $q = \exists x_1 \dots \exists x_m \forall y_1 \dots \forall y_n f \in \exists\forall\text{CIR}$  belongs to  $\mathbf{B}_2$  iff there is a partial assignment  $\chi$  which sets none of the variables  $y_1, \dots, y_n$  to constants and for which  $f_\chi$  is a tautology, i.e.  $f_\chi \equiv 1$ .

(a)  $\mathbf{B}_2 \leq_m^p \langle\langle\langle c \rangle\rangle\rangle$

Let  $q = \exists x_1 \dots \exists x_m \forall y_1 \dots \forall y_n f \in \exists\forall\text{CIR}$  be given.

The first approach for the reduction would be to construct from  $q$  something like the pair of circuits  $\langle 1, f \rangle$ , and in fact if  $q \in \mathbf{B}_2$  then  $1 \ll_c f$ , but for the other direction this construction is not correct: take  $q = \exists x \forall y (x \wedge y)$ , then  $1 \ll_c (x \wedge y)$  but  $q \notin \mathbf{B}_2$ . We observe that we have to guarantee that the variables  $y_i$  are not set to constants by the partial assignment. This is done with the help of the parity-function with its special property (P2):

Given  $q$  from above, construct the pair of circuits  $\langle g, h \rangle$  with

$$g = \oplus(y_1, \dots, y_n, 1)$$

$$h = \oplus(y_1, \dots, y_n, f)$$

This construction can be done in polynomial time. Hence, it remains to show that  $q \in \mathbf{B}_2 \iff g \ll_c h$ :

( $\implies$ ) Given that  $q \in \mathbf{B}_2$ , take the assignment  $\chi$  from the characterization of  $\mathbf{B}_2$  from above, so  $1 \equiv f_\chi$ . Because  $\chi$  does not change the variables  $y_1, \dots, y_n$  we have  $h_\chi = \oplus(y_1, \dots, y_n, f_\chi)$  and therefore  $h_\chi \equiv \oplus(y_1, \dots, y_n, 1)$ . Thus,  $g \equiv h_\chi$ .

( $\impliedby$ ) If there exists a partial assignment  $\rho \in \mathbf{R}_c$  such that  $g \equiv h_\rho$  then none of the variables of  $y_1, \dots, y_n$  can be mapped to a constant because otherwise

$h_\rho$  would be independent of some  $y_i$  while  $g$  is by (P2) dependent on  $y_i$ . Thus we have  $\oplus(y_1, \dots, y_n, f_\rho) = h_\rho \equiv g = \oplus(y_1, \dots, y_n, 1)$ , and by (P3) we conclude  $1 \equiv f_\rho$ . By the characterization of  $\mathbf{B}_2$  from above we have that  $q \in \mathbf{B}_2$ .

We will first prove the completeness of the third problem because then the proof of the second becomes more clear:

(c)  $\mathbf{B}_2 \leq_m^p \langle\langle cv \rangle\rangle$

Let  $q = \exists x_1 \dots \exists x_m \forall y_1 \dots \forall y_n f \in \exists \forall \text{CIR}$  be given.

If we try to apply here the previous construction we still have that if  $q \in \mathbf{B}_2$  then  $g \ll_{cv} h$ , but the opposite direction is not correct: let  $q = \exists x \forall y (x \wedge y)$ , then  $\oplus(y, 1) \ll_{cv} \oplus(y, (x \wedge y))$  by the cv-mapping which sets  $x$  to  $y$  and  $y$  to 1, but  $q \notin \mathbf{B}_2$ . We observe that the problem is that a variable  $x_i$  can be mapped to a variable  $y_j$  and vice versa. The idea is to “blow up” the variables  $y_j$  by huge parity circuits which “swallow” the values for the variables  $x_i$  of a cv-mapping:

Given  $q$  from above, let  $X$  denote the set of variables  $\{x_1, \dots, x_m\}$  and let  $Y$  denote the set of variables  $\{y_1, \dots, y_n\}$ . Let  $s = 2m + 1$ , and choose  $(n \times s)$  new variables  $z_1^1, \dots, z_1^s, \dots, z_n^1, \dots, z_n^s$ . Call  $Z_i := \{z_i^1, \dots, z_i^s\}$ ,  $Z := Z_1 \cup \dots \cup Z_n$ . Take the replacement  $\omega$  with  $\omega(y_i) := \oplus(z_i^1, \dots, z_i^s)$  and  $\omega(v) := v$  otherwise. Then  $f_\omega$  looks like  $f$  besides that each variable  $y_i \in Y$  is replaced by the parity-circuit  $\omega(y_i)$ .

For the reduction, construct the pair of circuits  $\langle g, h \rangle$  with

$$g = \oplus(z_1^1, \dots, z_1^s, \dots, z_n^1, \dots, z_n^s, 1)$$

$$h = \oplus(z_1^1, \dots, z_1^s, \dots, z_n^1, \dots, z_n^s, f_\omega)$$

This construction can be carried out in polynomial time. We show that  $q \in \mathbf{B}_2 \iff g \ll_{cv} h$ .

( $\implies$ )

If  $q \in \mathbf{B}_2$  then take the partial assignment  $\chi$  from the characterization of  $\mathbf{B}_2$  above. Note that we can assume that  $\chi$  only sets variables from  $X$  to constants. We have that  $f_\chi$  is a tautology, so also  $f_{\chi, \omega}$  is a tautology. But  $f_{\chi, \omega} = f_{\omega, \rho}$  because  $\chi$  does not change the variables from  $Z$  and  $\omega$  does not change the variables from  $X$ . Therefore  $1 \equiv f_{\omega, \chi}$  and so  $h_\chi = \oplus(z_1^1, \dots, z_n^s, f_{\omega, \chi}) \equiv \oplus(z_1^1, \dots, z_n^s, 1) = g$ . Because  $\chi$  is a cv-mapping we have

$g \ll_{cv} h$ .

( $\Leftarrow$ )

Let  $g \equiv h_\rho$  with  $\rho \in \mathbf{R}_{cv}$ . One first recognizes that  $Z \subseteq \rho(X \cup Z)$  because by (P2)  $g$  depends on each variable of  $Z$  and in  $h$  occur only variables from  $X$  and  $Z$ . Now we have the following simple combinatorial conclusion: because  $s > m$  there are two subsets  $A := \{a_1, \dots, a_n\}, B := \{b_1, \dots, b_n\}$  of  $Z$  such that  $A$  and  $B$  have exactly  $n$  elements,  $a_i \in Z_i, \rho(a_i) = b_i$  and each variable  $b_i \in B$  is neither the  $\rho$ -image of a variable in  $X$  nor the  $\rho$ -image of a variable in  $Z$  different from  $a_i$ .

We will define a partial assignment  $\gamma$  distinguishing two cases (I) and (II):

(I)  $\rho(Z \setminus A) = Z \setminus B$ : then  $\rho$  permutes  $Z$  and so by (P1) and (P3)  $1 \equiv f_{\omega, \rho}$ . Let  $\gamma$  be the partial assignment which sets every variable to 0 except the variables of  $B$ . Then we have still  $1 \equiv f_{\omega, \rho, \gamma}$ .

(II)  $\exists v \in Z \setminus B : v \notin \rho(Z \setminus A)$ : Let  $\beta$  be the partial assignment which sets every variable to 0 except  $v$  and the variables of  $B$ .  $g_\beta$  is equivalent to either  $\oplus(v, b_1, \dots, b_n, 1)$  or its negation, and  $h_{\rho, \beta}$  is equivalent either to  $\oplus(b_1, \dots, b_n, f_{\omega, \rho, \beta})$  or its negation. Extend  $\beta$  to a partial assignment  $\gamma$  by mapping  $v$  to that constant which makes (P3) applicable so that we can conclude  $1 \equiv f_{\omega, \rho, \gamma}$ .

In both cases we have that  $1 \equiv f_{\omega, \rho, \gamma}$  and for each  $z_i$  we see that  $\omega(z_i)_{\rho, \gamma}$  is equivalent either to  $b_i$  or to  $\neg b_i$ , because every variable  $z_i^j \in Z_i$  besides  $a_i$  is finally mapped to a constant.

Let  $\pi$  be the following partial assignment setting all variables from  $X$  to constants and leaving the other variables unchanged:

$$\pi(v) := \begin{cases} v & \text{if } v \notin X \\ \rho(v) & \text{if } v \in X \text{ and } \rho(v) \in C \\ \gamma(\rho(v)) & \text{else} \end{cases}$$

Now compare  $f_\pi$  and  $f_{\omega, \rho, \gamma}$ : the variables  $x_i \in X$  are by definition of  $\pi$  in both circuits replaced by the same constants, and instead of a variable  $y_i$  in  $f_\pi$  we find in  $f_{\omega, \rho, \gamma}$  a circuit equivalent to either  $b_i$  or  $\neg b_i$ . Thus the two circuits are "very similar" in the following precise sense: let  $\tau$  be a total assignment and define the total assignment  $\tau'$  by

$$\tau'(v) := \begin{cases} \tau(v) & \text{if } v \notin B \\ 0 & \text{if } v = b_i \in B, \tau(y_i) = 0 \text{ and } \omega(y_i)_{\rho,\gamma} \text{ is equivalent to } b_i \\ 1 & \text{if } v = b_i \in B, \tau(y_i) = 1 \text{ and } \omega(y_i)_{\rho,\gamma} \text{ is equivalent to } b_i \\ 1 & \text{if } v = b_i \in B, \tau(y_i) = 0 \text{ and } \omega(y_i)_{\rho,\gamma} \text{ is equivalent to } \neg b_i \\ 0 & \text{if } v = b_i \in B, \tau(y_i) = 1 \text{ and } \omega(y_i)_{\rho,\gamma} \text{ is equivalent to } \neg b_i \end{cases}$$

Then  $f_{\omega,\rho,\gamma}$  evaluates with the total assignment  $\tau'$  to the same constant  $f_\pi$  evaluates to with the total assignment  $\tau$ .

Assume that  $f_\pi$  is not a tautology. Then there is a total assignment  $\tau$  which lets  $f_\pi$  evaluate to 0. But then  $f_{\omega,\rho,\gamma}$  evaluates with  $\tau'$  also to 0, contradicting the fact that it is a tautology. Thus  $f_\pi$  is a tautology, where  $\pi$  is a partial assignment which does not change the variables from  $Y$ . So  $q \in \mathbf{B}_2$  by the characterization of  $\mathbf{B}_2$  from above.

And finally we come to part (b):

(b)  $\mathbf{B}_2 \leq_m^p \langle\langle v \rangle\rangle$

First consider the following reduction from SAT to  $\langle\langle v \rangle\rangle$ :

Given a circuit  $f$  construct the pair of circuits  $\langle g, h \rangle$  with

$$g = \oplus(r_0, r_1, 1)$$

$$h = \oplus(r_0, r_1, d) \text{ with}$$

$$d = (r_0 \vee \neg r_1 \vee f)$$

where  $r_0, r_1$  are two different variables not occurring in  $f$ .

We show that  $f \in \text{SAT} \iff g \ll_v h$ :

( $\implies$ ) if  $f$  evaluates to 1 for a total assignment  $\tau$ , then construct the following v-mapping  $\delta$ : let  $\delta(r_0) := r_0$ ,  $\delta(r_1) := r_1$ , and else if  $\tau(v) = 0$  then define  $\delta(v) := r_0$ , and if  $\tau(v) = 1$  then define  $\delta(v) := r_1$ . Now  $d_\delta$  is a tautology: if for a total assignment  $\xi$   $\xi(r_0) = 1$  or  $\xi(r_1) = 0$  then  $d_\delta$  evaluates to 1, and if  $\xi(r_0) = 0$  and  $\xi(r_1) = 1$  then  $f_\delta$  evaluates with  $\xi$  to 1 the same way like  $f$  evaluates with  $\tau$  to 1, thus  $d_\delta$  is a tautology. We conclude  $g \equiv h_\delta$ . Thus  $g \ll_v h$ .

( $\impliedby$ ) if  $g \equiv h_\rho$  for a v-mapping  $\rho$ , then we can conclude that  $\rho(\{r_0, r_1\}) = \{r_0, r_1\}$ : assume that neither  $r_0$  nor  $r_1$  is mapped to  $r_0$  by  $\rho$ . Then define the

partial assignment  $\pi$  which only sets  $\rho(r_0)$  to 1 and leaves all other variables unchanged. Then  $g_\pi$  is still dependent on  $r_0$  while  $h_{\rho\pi}$  is not, what contradicts the equivalence of  $g$  and  $h_\rho$ . Together with the analog reasoning for  $r_1$  we have that  $\rho(\{r_0, r_1\}) = \{r_0, r_1\}$ . With (P1) and (P3) we conclude that  $d_\rho$  is a tautology. Because  $\rho(r_0) \neq \rho(r_1)$  we can define the total assignment  $\xi$  with  $\xi(\rho(r_0)) = 0$ ,  $\xi(\rho(r_1)) = 1$  and  $\xi(v) = 0$  else.  $f_\rho$  with the total assignment  $\xi$  has to evaluate to 1, showing that  $f$  is in SAT.

Now back to the reduction from  $\mathbf{B}_2$  to  $\langle\langle\ll v\rangle\rangle$ :

Given a word  $q = \exists x_1 \dots \exists x_m \forall y_1 \dots \forall y_n f \in \exists \forall \text{CIR}$ , construct the pair of circuits  $\langle g, h \rangle$  with

$$g = \oplus(z_1^1, \dots, z_1^s, \dots, \dots, z_n^1, \dots, z_n^s, r_0, r_1, 1)$$

$$h = \oplus(z_1^1, \dots, z_1^s, \dots, \dots, z_n^1, \dots, z_n^s, r_0, r_1, r_0 \vee \neg r_1 \vee f_\omega)$$

where  $r_0, r_1$  are chosen like above, and where  $z_i^j$  and  $\omega$  are chosen as in the proof of part (c).

We have  $q \in \mathbf{B}_2 \iff g \ll_v h$  with a proof which combines the arguments of the proof of (c) and the proof of the reduction from SAT above.  $\square$

## 5 Circuit Isomorphism

In this section we will investigate the equivalence relation  $\sim$  from section 3. First we show that  $\sim$  could be defined by means of  $\ll_{cv}$  and  $\ll_v$ :

The relations  $\ll_c, \ll_{cv}$  and  $\ll_v$  are reflexive and transitive, so let us define the corresponding equivalence relations  $\approx_c, \approx_{cv}$  and  $\approx_v$  by

- $f \approx_c g \iff f \ll_c g$  and  $g \ll_c f$ ,
- $f \approx_v g \iff f \ll_v g$  and  $g \ll_v f$ ,
- $f \approx_{cv} g \iff f \ll_{cv} g$  and  $g \ll_{cv} f$ .

**Proposition 1**  $\forall f, g \in \text{CIR}$

$$f \equiv g \iff f \approx_c g, \text{ and } f \sim g \iff f \approx_v g \iff f \approx_{cv} g.$$

Proof:

$$f \equiv g \implies f \approx_c g:$$

If  $f \equiv g$  then the identity on the variables  $id : V \longrightarrow V : v \longrightarrow v$  is a partial assignment with  $f_{id} \equiv g \equiv f \equiv g_{id}$ .

$$f \approx_c g \implies f \equiv g:$$

If  $g \equiv f_\rho$  and  $f \equiv g_\sigma$  with  $\rho, \sigma \in \mathbf{R}_c$  then  $f \equiv g_\rho \equiv f_{\rho, \sigma}$ . In  $f_{\rho, \sigma}$  the variables which were set by  $\rho$  to constants don't occur any longer so that  $f_{\rho, \sigma, \rho} = f_{\rho, \sigma}$  and therefore  $f \equiv f_{\rho, \sigma} \equiv f_{\rho, \sigma, \rho} \equiv f_\rho \equiv g$ .

$$f \sim g \implies f \approx_v g:$$

If there exists a bijection  $\rho : V \longrightarrow V$  such that  $f \equiv g_\rho$  then  $g \equiv f_{\rho^{-1}}$  because  $g = g_{\rho, \rho^{-1}} \equiv f_{\rho^{-1}}$ , and therefore  $f \ll_v g$  and  $g \ll_v f$ .

$$f \approx_v g \implies f \sim g:$$

Let  $g \equiv f_\rho$  and  $f \equiv g_\sigma$  with  $\rho, \sigma \in \mathbf{R}_v$ . We define  $f^1 := f, g^n := f_\rho^n, f^{n+1} := g_\sigma^n$ . Then, for all  $n$ ,  $f \equiv f^n$  and  $g \equiv g^n$  because if  $f \equiv f^n$  and  $g \equiv g^n$  then  $f^{n+1} \equiv g_\sigma^n \equiv g_\sigma \equiv f$  and  $g^{n+1} \equiv f_\rho^n \equiv f_\rho \equiv g$ . Let  $\text{Occ}(f)$  denote the set of variables occurring in  $f$  and consider the sequence of natural numbers  $|\text{Occ}(f^1)| \geq |\text{Occ}(g^1)| \geq |\text{Occ}(f^2)| \geq |\text{Occ}(g^2)| \geq |\text{Occ}(f^3)| \geq |\text{Occ}(g^3)| \dots$ . Because  $|\text{Occ}(f)|$  is finite this sequence must have two consecutive members with the same value, say that  $|\text{Occ}(f^n)| = |\text{Occ}(g^n)|$ . We can conclude that  $\sigma$  restricted to  $\text{Occ}(f^n) \longrightarrow \text{Occ}(g^n)$  is a bijection and that there is some bijection  $\alpha : \text{Occ}(g^n) - \text{Occ}(f^n) \longrightarrow \text{Occ}(f^n) - \text{Occ}(g^n)$ . We know that  $f^n$  is independent of all variables in  $V - \text{Occ}(f^n)$ ; therefore define the bijection  $\beta$  by  $\beta(v) := \rho(v)$  for  $v \in \text{Occ}(f^n)$  and  $\beta(v) := \alpha(v)$  for  $v \in \text{Occ}(g^n) - \text{Occ}(f^n)$  and  $\beta(v) := v$  otherwise. Now we have  $g \equiv g^n = f_\rho^n = f_\beta^n \equiv f_\beta$ . Thus  $f$  and  $g$  are isomorphic.

The proof for  $\approx_{cv}$  works the same way like that for  $\approx_v$ . □

From the proposition above, it follows that  $\langle\langle \approx_c \rangle\rangle$  is co-NP-complete. We shall call the problem  $\langle\langle \approx_v \rangle\rangle = \langle\langle \approx_{cv} \rangle\rangle = \langle\langle \sim \rangle\rangle$  the Circuit Isomorphism problem, short **CI**.

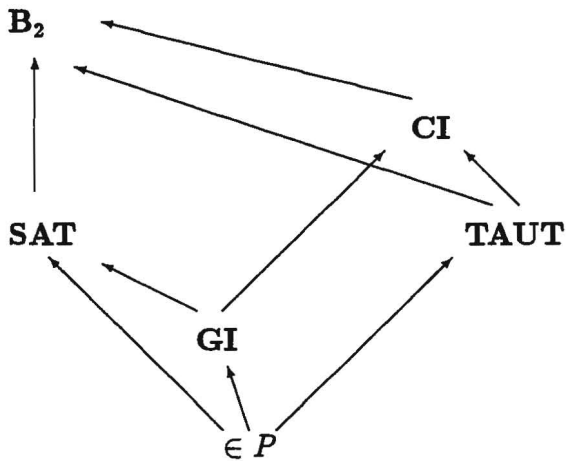
Regarding its complexity we have the following proposition where **GI** denotes the Graph Isomorphism Problem (for its definition see [1], p.155):

**Proposition 2**  $\text{CI} \in \Sigma_2^p$ ,  $\text{TAUT} \leq_m^p \text{CI}$  and  $\text{GI} \leq_m^p \text{CI}$ .

Proof: With a proof similar to that for  $\langle\langle\langle_p\rangle\rangle\rangle$  we have that  $\text{CI} \in \Sigma_2^p$ . The reduction from  $\text{TAUT}$  to  $\text{CI}$  is by  $f \rightarrow \langle f, 1 \rangle$ .

For the Graph Isomorphism problem, let  $h_G$  for a graph  $G = (V, E)$  be the circuit defined as follows: for every vertex  $i \in V$  in  $G$  choose a different variable  $v_i$ ; then  $h_G = \bigvee_{(i,j) \in E} (v_i \wedge v_j)$ . Now, it is not difficult to see that  $G_1$  and  $G_2$  are isomorphic if and only if  $h_{G_1} \sim h_{G_2}$ .  $\square$

The following picture places  $\text{CI}$  graphically. The arrows denote the knowledge of the existence of a  $\leq_m^p$ -reduction:



## 6 Remarks and Open Questions

We would like to remark that analogous results can be stated for formulas instead of circuits.

Note the analogy in the definition of  $\text{GI}$  and  $\text{CI}$ : Two graphs are isomorphic iff they are the “same” graphs – modulo node names. Similarly, two circuits are isomorphic iff they compute the “same” function – modulo variable names. Note that Graph Isomorphism like Circuit Isomorphism can be defined as

the corresponding equivalence relation of a preorder, namely the preorder Subgraph Isomorphism, which is NP-complete, see [1], p. 202.

Some open questions of interest that remain are:

- What exactly is the complexity of CI? Is it a  $\Sigma_2^P$ -complete problem? If not, is it possibly in  $\Delta_2^P$ ?
- Could one define a  $\Pi_3^P$ -complete or a  $\Sigma_3^P$ -complete problem based on one of the  $\Sigma_2^P$ -complete problems defined here?
- The relations  $\ll$  and  $\ll_t$  obtained via the most general and most restrictive replacement schemes  $\mathbf{R}$  and  $\mathbf{R}_t$  are in co-DP and DP respectively and hence unlikely to be  $\Sigma_2^P$ -complete. It will be interesting to study the question of complexity for replacement schemes other than the ones investigated here. In this regard we would like to mention that the problem  $\langle\langle\ll_1\rangle\rangle$ , where  $f \ll_1 g$  if  $f$  can be obtained from  $g$  via a replacement that maps variables to literals (*literal mapping*), is  $\Sigma_2^P$ -complete.

## 7 Acknowledgements

The notion of Circuit Isomorphism and the reduction from Graph Isomorphism to it was observed together with Richard Chang.

## References

- [1] M.R.Garey, D.S.Johnson (1978) *Computers and Intractability*, Freeman, San Francisco
- [2] L.Stockmeyer (1977) *The Polynomial Time Hierarchy*, Theoretical Computer Science **3**,1-22



