Complete Sets of Hamiltonian Circuits for Classification of Documents

Bernd Steinbach[1] and Christian Posthoff[2]

[1] Freiberg University of Mining and Technology, Institute of Computer Science,
D-09596 Freiberg, Germany `steinb@informatik.tu-freiberg.de`
[2] The University of The West Indies, St. Augustine Campus, Trinidad & Tobago
`Christian.Posthoff@sta.uwi.edu`

Abstract. The calculation of Hamiltonian Circuits is an NP-complete task. This paper uses slightly modified complete sets of Hamiltonian circuits for the classification of documents. The known solution method is based on a SAT-instance with a huge number of clauses which is flattening the knowledge about the problem. We suggest an even more compact model of Boolean equations that preserves the knowledge by summarizing restrictions and requirements. The presented *implicit two-phase SAT-solver* finds efficiently the solution using operations of the XBOOLE library. This solver can be included easily as signal processing unit into the device where the classification of the documents is required.

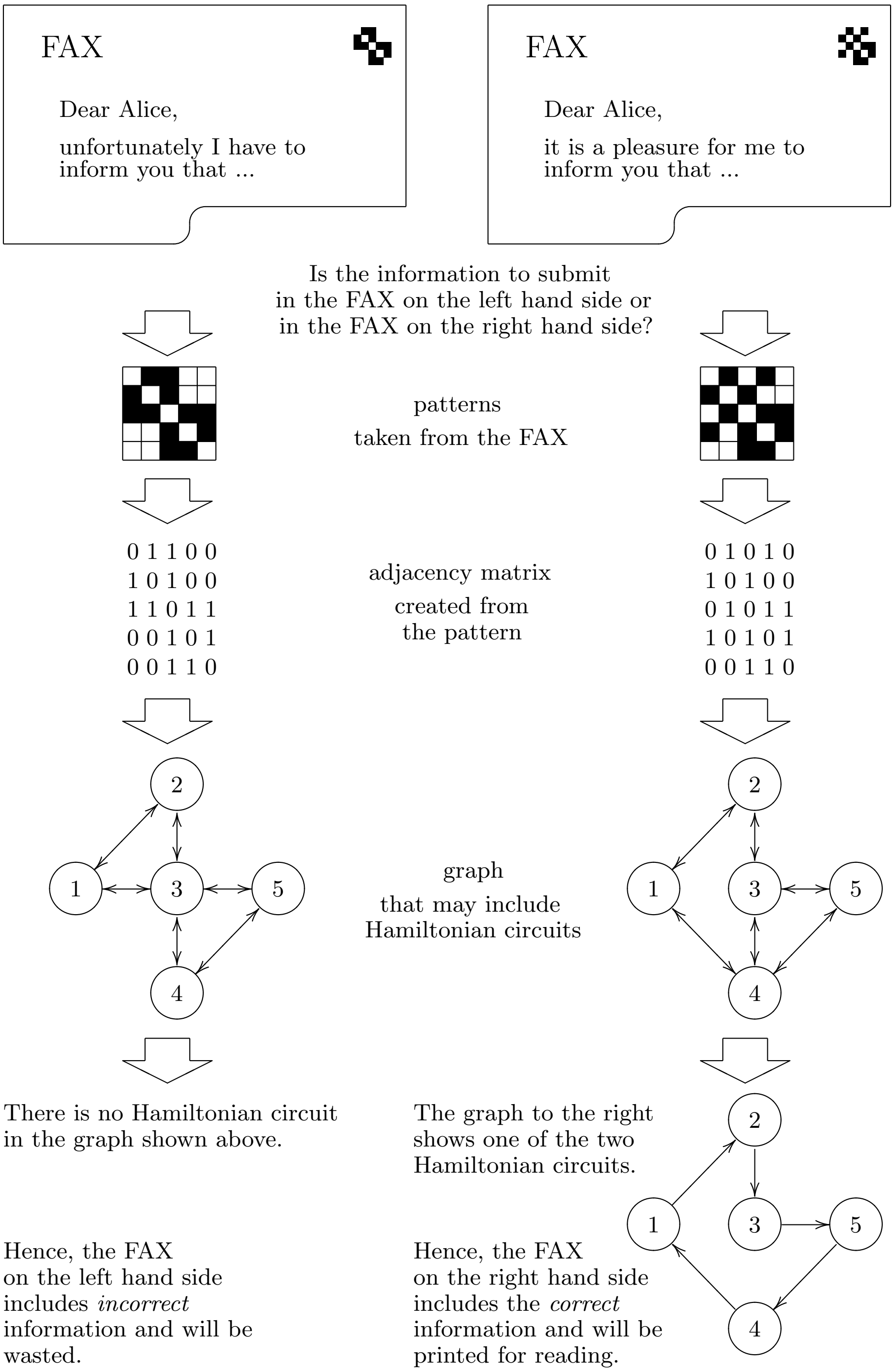
1 Introduction

A Hamiltonian path in a graph is a sequence of edges that uses each node precisely once. A Hamiltonian circuit [1], also called Hamiltonian cycle, is a cycle in the graph which visits each node exactly once and returns to the starting node. It is well-known that the problem of determining whether such paths or cycles exist is an NP-complete problem [4].

We suggest a slight modification of this problem and apply it to the classification of documents by means of an efficient signal processing unit. Basically our approach can be applied to each document that is transmitted by a sequence of bits. As an example we motivate our approach in the context of FAX transmissions. In spite of several new possibilities the FAX transmission is popular for the fast exchange of textual and graphic data because only simple equipment is required. A disadvantage of the FAX transmission from Alice to Bob is that Eve (an eavesdropper) can see the transmitted information. An easy way to reduce the value of the information seen be Eve is that both *good* and *bad* information are transmitted by FAX. It is necessary that the receiver Bob can select the *good* FAX, and a signal processing unit throws away the *bad* information immediately.

A couple of bits of the transmitted document is used for this detection These bits may form an $n \times n$ adjacency matrix of a graph. Figure 1 illustrates the suggested procedure.

Reflexive edges that begin and end on the same node are ignored. Instead of a single Hamiltonian circuit a set of Hamiltonian circuits will now be allowed. A



Dear Alice,

unfortunately I have to inform you that ...

Dear Alice,

it is a pleasure for me to inform you that ...

Is the information to submit in the FAX on the left hand side or in the FAX on the right hand side?

There is no Hamiltonian circuit in the graph shown above.

The graph to the right shows one of the two Hamiltonian circuits.

Hence, the FAX on the left hand side includes *incorrect* information and will be wasted.

Hence, the FAX on the right hand side includes the *correct* information and will be printed for reading.

Fig. 1. Principle to select the correct FAX information by Hamiltonian circuits

complete set of Hamiltonian circuits covers all nodes of the graph. Any received FAX without a complete set of Hamiltonian circuits will be rejected. The number of different complete sets of Hamiltonian circuits is an additional information for the receiver Bob, but hidden for Eve. Due to the missing number of nodes n and the position of the classification bits in the transmitted FAX the eavesdropper Eve is not able to distinguish between *good* and *bad* FAX transmissions.

2 Boolean Model

2.1 Coding the Edges of the Graph by Boolean Variables

Boolean variables are introduced for each edge and each direction as follows:

$$x_{i,j} = \begin{cases} 1 & \text{if the edge is used from node } i \text{ to node } j \\ 0 & \text{otherwise} \end{cases} \quad . \tag{1}$$

Hence, the number of variables needed to express all conditions of complete sets of Hamiltonian circuits is equal to the number of values 1 in the $n \times n$ adjacency matrix of a graph where values 1 on the main diagonal are replaced be values 0. By means of these variables all conditions for complete sets of Hamiltonian circuits can be expressed.

2.2 Simple SAT - Model

The selection of any edge, i.e. $(x_{i,j} = 1)$ determines for Hamiltonian circuits three conditions:

1. The reverse edge is forbidden:

$$x_{i,j} \wedge x_{j,i} = 0 \ . \tag{2}$$

2. Any edge from the same start node i to any destination node $d_l \neq j$ is forbidden:

$$x_{i,j} \wedge x_{i,d_l} = 0, \ \forall \ d_l \neq j \ . \tag{3}$$

3. Any edge that ends at the node j and begins at any source node $s_m \neq i$ is forbidden:

$$x_{i,j} \wedge x_{s_m,j} = 0, \ \forall \ s_m \neq i \ . \tag{4}$$

The final condition for complete sets of Hamiltonian circuits is determined by the nodes of the graph. In each node of the graph one of the outgoing edges must be an element of the complete sets of Hamiltonian circuits:

$$\bigvee_{d_l=1}^{d_{l\,\max}} x_{i,d_l} = 1 \ . \tag{5}$$

The system of equations defined by the formula given above can be solved directly using XBOOLE [2, 3] or transformed into a single equation where a

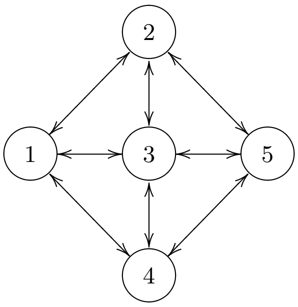


Fig. 2. Simple graph to analyze for complete sets of Hamilitonian circuits

conjunctive form on the left hand side is equal to 1 and solved by a SAT - solver like *zChaff* or *March* . The abbreviation SAT is used for the term *satisfiability.*

For the simple graph of Figure 2 we get the SAT - equation (6) that consists of 93 clauses that include a total of 16 variables.

$$\begin{aligned}
&(\overline{x}_{1,2} \vee \overline{x}_{2,1})\,(\overline{x}_{1,2} \vee \overline{x}_{1,3})\,(\overline{x}_{1,2} \vee \overline{x}_{1,4})\,(\overline{x}_{1,2} \vee \overline{x}_{3,2})\,(\overline{x}_{1,2} \vee \overline{x}_{5,2}) \wedge \\
&(\overline{x}_{1,3} \vee \overline{x}_{3,1})\,(\overline{x}_{1,3} \vee \overline{x}_{1,2})\,(\overline{x}_{1,3} \vee \overline{x}_{1,4})\,(\overline{x}_{1,3} \vee \overline{x}_{2,3})\,(\overline{x}_{1,3} \vee \overline{x}_{4,3})\,(\overline{x}_{1,3} \vee \overline{x}_{5,3}) \wedge \\
&(\overline{x}_{1,4} \vee \overline{x}_{4,1})\,(\overline{x}_{1,4} \vee \overline{x}_{1,2})\,(\overline{x}_{1,4} \vee \overline{x}_{1,3})\,(\overline{x}_{1,4} \vee \overline{x}_{3,4})\,(\overline{x}_{1,4} \vee \overline{x}_{5,4}) \wedge \\
&(\overline{x}_{2,1} \vee \overline{x}_{1,2})\,(\overline{x}_{2,1} \vee \overline{x}_{2,3})\,(\overline{x}_{2,1} \vee \overline{x}_{2,5})\,(\overline{x}_{2,1} \vee \overline{x}_{3,1})\,(\overline{x}_{2,1} \vee \overline{x}_{4,1}) \wedge \\
&(\overline{x}_{2,3} \vee \overline{x}_{3,2})\,(\overline{x}_{2,3} \vee \overline{x}_{2,1})\,(\overline{x}_{2,3} \vee \overline{x}_{2,5})\,(\overline{x}_{2,3} \vee \overline{x}_{1,3})\,(\overline{x}_{2,3} \vee \overline{x}_{4,3})\,(\overline{x}_{2,3} \vee \overline{x}_{5,3}) \wedge \\
&(\overline{x}_{2,5} \vee \overline{x}_{5,2})\,(\overline{x}_{2,5} \vee \overline{x}_{2,1})\,(\overline{x}_{2,5} \vee \overline{x}_{2,3})\,(\overline{x}_{2,5} \vee \overline{x}_{3,5})\,(\overline{x}_{2,5} \vee \overline{x}_{4,5}) \wedge \\
&(\overline{x}_{3,1} \vee \overline{x}_{1,3})\,(\overline{x}_{3,1} \vee \overline{x}_{3,2})\,(\overline{x}_{3,1} \vee \overline{x}_{3,4})\,(\overline{x}_{3,1} \vee \overline{x}_{3,5})\,(\overline{x}_{3,1} \vee \overline{x}_{2,1})\,(\overline{x}_{3,1} \vee \overline{x}_{4,1}) \wedge \\
&(\overline{x}_{3,2} \vee \overline{x}_{2,3})\,(\overline{x}_{3,2} \vee \overline{x}_{3,1})\,(\overline{x}_{3,2} \vee \overline{x}_{3,4})\,(\overline{x}_{3,2} \vee \overline{x}_{3,5})\,(\overline{x}_{3,2} \vee \overline{x}_{1,2})\,(\overline{x}_{3,2} \vee \overline{x}_{5,2}) \wedge \\
&(\overline{x}_{3,4} \vee \overline{x}_{4,3})\,(\overline{x}_{3,4} \vee \overline{x}_{3,1})\,(\overline{x}_{3,4} \vee \overline{x}_{3,2})\,(\overline{x}_{3,4} \vee \overline{x}_{3,5})\,(\overline{x}_{3,4} \vee \overline{x}_{1,4})\,(\overline{x}_{3,4} \vee \overline{x}_{5,4}) \wedge \\
&(\overline{x}_{3,5} \vee \overline{x}_{5,3})\,(\overline{x}_{3,5} \vee \overline{x}_{3,1})\,(\overline{x}_{3,5} \vee \overline{x}_{3,2})\,(\overline{x}_{3,5} \vee \overline{x}_{3,4})\,(\overline{x}_{3,5} \vee \overline{x}_{2,5})\,(\overline{x}_{3,5} \vee \overline{x}_{4,5}) \wedge \\
&(\overline{x}_{4,1} \vee \overline{x}_{4,1})\,(\overline{x}_{4,1} \vee \overline{x}_{4,3})\,(\overline{x}_{4,1} \vee \overline{x}_{4,5})\,(\overline{x}_{4,1} \vee \overline{x}_{2,1})\,(\overline{x}_{4,1} \vee \overline{x}_{2,1}) \wedge \\
&(\overline{x}_{4,3} \vee \overline{x}_{3,4})\,(\overline{x}_{4,3} \vee \overline{x}_{4,1})\,(\overline{x}_{4,3} \vee \overline{x}_{4,5})\,(\overline{x}_{4,3} \vee \overline{x}_{1,3})\,(\overline{x}_{4,3} \vee \overline{x}_{2,3})\,(\overline{x}_{4,3} \vee \overline{x}_{5,3}) \wedge \\
&(\overline{x}_{4,5} \vee \overline{x}_{4,5})\,(\overline{x}_{4,5} \vee \overline{x}_{4,1})\,(\overline{x}_{4,5} \vee \overline{x}_{4,3})\,(\overline{x}_{4,5} \vee \overline{x}_{2,5})\,(\overline{x}_{4,5} \vee \overline{x}_{3,5}) \wedge \\
&(\overline{x}_{5,2} \vee \overline{x}_{2,5})\,(\overline{x}_{5,2} \vee \overline{x}_{5,3})\,(\overline{x}_{5,2} \vee \overline{x}_{5,4})\,(\overline{x}_{5,2} \vee \overline{x}_{1,2})\,(\overline{x}_{5,2} \vee \overline{x}_{3,2}) \wedge \\
&(\overline{x}_{5,3} \vee \overline{x}_{3,5})\,(\overline{x}_{5,3} \vee \overline{x}_{5,2})\,(\overline{x}_{5,3} \vee \overline{x}_{5,4})\,(\overline{x}_{5,3} \vee \overline{x}_{1,3})\,(\overline{x}_{5,3} \vee \overline{x}_{2,3})\,(\overline{x}_{5,3} \vee \overline{x}_{4,3}) \wedge \\
&(\overline{x}_{5,4} \vee \overline{x}_{4,5})\,(\overline{x}_{5,4} \vee \overline{x}_{5,2})\,(\overline{x}_{5,4} \vee \overline{x}_{5,3})\,(\overline{x}_{5,4} \vee \overline{x}_{1,4})\,(\overline{x}_{5,4} \vee \overline{x}_{3,4}) \wedge \\
&(x_{1,2} \vee x_{1,3} \vee x_{1,4})\,(x_{2,1} \vee x_{2,3} \vee x_{2,5}) \wedge (x_{3,1} \vee x_{3,2} \vee x_{2,4} \vee x_{4,5}) \wedge \\
&(x_{4,1} \vee x_{4,3} \vee x_{4,5})\,(x_{5,2} \vee x_{5,3} \vee x_{5,4}) = 1
\end{aligned} \tag{6}$$

The disadvantages of this SAT – approach are on the one hand the large number of clauses that are required even for small graphs and on the other hand the loss of problem knowledge because the information with regard to single edges is distributed over a large number of clauses.

2.3 Compact Rule-based Model

We use the same encoding (1) to express which edges are included in a Hamiltonian circuit. As an example we take again the simple graph of Figure 2.

Basically there are two types of laws:

1. *restrictions* that describe prohibited states, and
2. *requirements* that describe necessary choices.

Table 1 show the three general restrictions for complete sets of Hamiltonian circuits and their concrete specification by implications for the edge from node 1 to node 2 in Figure 2.

Table 1. Restrictions for Hamiltonian circuits, expressed for the edge from node 1 to node 2 in Figure 2

Restrictions	Implications
An edge from node i to node j, i.e. $x_{i,j} = 1$, prohibits that the reverse edge is used, i.e $x_{j,i} = 0$.	$(x_{1,2} \Longrightarrow \overline{x}_{2,1}) = 1$
An edge from node i to node j, i.e. $x_{i,j} = 1$, prohibits all edges to other destination nodes d_l, i.e $x_{i,d_l} = 0$.	$(x_{1,2} \Longrightarrow \overline{x}_{1,3}) \wedge$ $(x_{1,2} \Longrightarrow \overline{x}_{1,4}) = 1$
An edge from node i to node j, i.e. $x_{i,j} = 1$, prohibits all edges from other source nodes s_m, i.e $x_{s_m,j} = 0$.	$(x_{1,2} \Longrightarrow \overline{x}_{3,2}) \wedge$ $(x_{1,2} \Longrightarrow \overline{x}_{5,2}) = 1$

The implications can be substituted by:

$$a \Longrightarrow b = \overline{a} \vee b \ , \tag{7}$$

so that the system of equations of Table 1 can be transformed into one single equation:

$$(\overline{x}_{1,2} \vee \overline{x}_{2,1})(\overline{x}_{1,2} \vee \overline{x}_{1,3})(\overline{x}_{1,2} \vee \overline{x}_{1,4})(\overline{x}_{1,2} \vee \overline{x}_{3,2})(\overline{x}_{1,2} \vee \overline{x}_{5,2}) = 1 \ . \tag{8}$$

It can be seen that each clause in the partial SAT formula (8) depends on exactly two negated variables where $\overline{x}_{1,2}$ appear in each clause. Applying the laws of the Boolean algebra to (8) we get:

$$(\overline{x}_{1,2} \vee x_{1,2}\, \overline{x}_{2,1}\, \overline{x}_{1,3}\, \overline{x}_{1,4}\, \overline{x}_{3,2}\, \overline{x}_{5,2}) = 1 \ . \tag{9}$$

The two conjunctions in formula (9) express two constraints with regard to the edge modeled by $x_{1,2}$:

1. if the edge from node 1 to node 2 is not used in a Hamiltonian circuit ($\overline{x}_{1,2}$), then there is no restriction for other edges, and
2. if the edge from node 1 to node 2 is used in a Hamiltonian circuit ($x_{1,2}$), then five edges indicated by the negated variables of the second conjunction in (9) can not be part of the Hamiltonian circuit.

Table 2. Requirement for Hamiltonian circuits, expressed for the node 1 in Figure 2

Requirement	Clause
There must be an edge starting from a node.	$(x_{1,2} \vee x_{1,3} \vee x_{1,4}) = 1$

Table 2 shows the requirement for complete sets of Hamiltonian circuits and their concrete specification by a disjunction for the node 1 in Figure 2.

Putting the restrictive laws and the requirement law together we get for node 1 in Figure 2:

$$\begin{aligned}(\overline{x}_{1,2} \vee x_{1,2}\, \overline{x}_{2,1}\, \overline{x}_{1,3}\, \overline{x}_{1,4}\, \overline{x}_{3,2}\, \overline{x}_{5,2})\wedge \\ (\overline{x}_{1,3} \vee x_{1,3}\, \overline{x}_{3,1}\, \overline{x}_{1,2}\, \overline{x}_{1,4}\, \overline{x}_{2,3}\, \overline{x}_{4,3}\, \overline{x}_{5,3})\wedge \\ (\overline{x}_{1,4} \vee x_{1,4}\, \overline{x}_{4,1}\, \overline{x}_{1,2}\, \overline{x}_{1,3}\, \overline{x}_{3,4}\, \overline{x}_{5,4}) \wedge (x_{1,2} \vee x_{1,3} \vee x_{1,4}) = 1 \ ,\end{aligned} \tag{10}$$

which can be simplified (using the distributive law) to

$$\begin{aligned}x_{1,2}\, \overline{x}_{2,1}\, \overline{x}_{1,3}\, \overline{x}_{1,4}\, \overline{x}_{3,2}\, \overline{x}_{5,2} \vee x_{1,3}\, \overline{x}_{3,1}\, \overline{x}_{1,2}\, \overline{x}_{1,4}\, \overline{x}_{2,3}\, \overline{x}_{4,3}\, \overline{x}_{5,3} \\ \vee x_{1,4}\, \overline{x}_{4,1}\, \overline{x}_{1,2}\, \overline{x}_{1,3}\, \overline{x}_{3,4}\, \overline{x}_{5,4} = 1 \ .\end{aligned} \tag{11}$$

Formula (11) describes completely the rule for node 1 in Figure 2. Each conjunction of (11) is a constraint associated to the edge indicated by the non-negated variable and describes completely **all the consequences**. For each edge of any graph such a constraint is defined uniquely.

For the simple graph of Figure 2 we finally get one equation that consists of five rules – one rule for each node, and 16 constraints – one constraint for each edge.

3 Implicit Two-Phase SAT Solver

The idea of our new approach is motivated in Section 2.3. Instead of processing the huge number of clauses (6) we solve an equation built by a much smaller number of rules of type (11).

The **first phase** covers the modeling of the problem by constraints and the calculation of partial solution sets implicitly. Therefore we call this new approach *Implicit Two-Phase SAT-Solver*. Algorithm 1 generates the matrix *mpss* that covers the partial solution sets of complete sets of Hamiltonian circuits of any graph completely.

As result of the first phase we get generally an $m \times m$-matrix for a graph of m edges. This matrix includes m partial solution sets. Each of them is the solution of one constraint and includes exactly one value 1 and so much values 0 as defined by the restrictive laws in Table 1 having the same variable of an edge as premise.

The **second phase** of the *implicit two-phase SAT Solver* is controlled by the requirement laws. In the lines 4 to 8 of Algorithm 2 the union operation

Algorithm 1 (1. Phase) Create the $m \times m$ matrix of partial solution sets from an $n \times n$ adjacency matrix: *mpss* `CPSS`(*Sizes* n, m, *AM am*)

```
Require: number $n$ of nodes in the graph
    number $m$ of edges in the graph
    adjacency matrix $am$ of the graph
Ensure: $m \times m$ matrix $mpss$ that includes for each edge the associated partial solution set
$mpss \leftarrow \emptyset$
for all $i$ such that $1 \leq i \leq n$ do {iterate over all rows}
  for all $j$ such that $1 \leq j \leq n$ do {iterate over all columns}
    if $am[i,j] = 1$ then {edge exists}
      $x \leftarrow$ vector of $n^2$ dashes
      $x[i * n + j] \leftarrow 1$ selected edge
      $x[j * n + i] \leftarrow 0$ {1. restrictive law: reverse edge}
      for all $c$ such that $1 \leq c \leq n$ do {iterate over the row}
        if $c \neq j$ then {other columns}
          if $am[i,c] = 1$ then {2. restrictive law:}
            $x[i * n + c] \leftarrow 0$ {outgoing edge from the start node}
          end if
        end if
      end for
      for all $r$ such that $1 \leq r \leq n$ do {iterate over all rows}
        if $r \neq i$ then {other rows}
          if $am[r,j] = 1$ then {3. restrictive law:}
            $x[r * n + j] \leftarrow 0$ {incoming edge to the end node}
          end if
        end if
      end for
      $mpss[\text{NumberOfRows}(mpss) + 1] \leftarrow \mathbf{x}$
    end if
  end for
end for
remove all columns from $mpss$ that include only dashes
return $mpss$ {the $m \times m$ matrix of partial solution sets}
```

UNI of XBOOLE [2, 3] to construct the solution of the rules for a selected node using the partial solution sets of the constraints generated in the first phase. The solution set s of Algorithm 2 is calculated in line 9 by the intersection operation *ISC* of XBOOLE, finally. This set s consists of all binary vectors of the length m that solve the SAT problem of complete sets of Hamiltonian circuits. There are 8 Hamiltonian circuits in the example of Figure 2.

4 Experimental Results

A small adjacency matrix of 5×5 bits with 12 values 1, for instance, is sufficient to sign a FAX as *good* or *bad* and deliver additional classification information.

Algorithm 2 Solve the Problem of Complete Sets of Hamiltonian Circuits: s `SCSHC`$(Size\ n, Vector\ g, MPSS\ mpss)$

```
Require: number $n$ of nodes,
    vector $g$ of the length $n$ that include the numbers of outgoing edges,
    matrix $mpss$ of partial solution sets as created in Algorithm 1
Ensure: set of all solution vectors of the length $m$ that indicate the used edges in
    complete sets of Hamiltonian circuits in the basic graph
$s \leftarrow FULL(mpss)$
$b \leftarrow 0$
for all $i$ such that $1 \leq i \leq n$ do {iterate over all nodes}
    $rule \leftarrow \emptyset$
    for all $j$ such that $1 \leq j \leq g[i]$ do {iterate over all outgoing edges of a node}
        $constraint \leftarrow mpss[b + j]$
        $rule \leftarrow UNI(rule, constraint)$
    end for
    $s \leftarrow ISC(s, rule)$
    $b \leftarrow b + g[i]$
end for
return $s${set of all solutions of complete sets of Hamiltonian circuits over $n$ nodes}
```

The power of such a signal processing unit for classification becomes visible when during the time of receiving a FAX this unit could find all existing 185,868 complete sets of Hamiltonian circuits in an adjacency matrix of 36×36 with 154 values 1.

5 Conclusions

The advantage of the new implicit two-phase approach in comparison with the known traditional SAT-model is that the assignment of one single value does not determine only one value of the solution, but additionally all values of the associated constraint which strongly restricts the remaining search space. For the simple example of Figure 2 the SAT - equation of 93 clauses is replaced by an equation of five rules over 16 constraints!

Basically this approach is not restricted to the detection of Hamiltonian circuits; it can easily be adapted to other tasks such as graph coloring or others.

References

1. Ore, O.: A Note on Hamiltonian Circuits. Amer. Math. Monthly. 67, 55 (1960)
2. Posthoff, Ch., Steinbach, B.: Logic Functions and Equations - Binary Models for Computer Science. Springer, Dordrecht, The Netherlands (2004)
3. Steinbach, B., Posthoff, Ch.: Logic Functions and Equations - Examples and Exercises. Springer Science + Business Media B.V. (2009)
4. Wegener, I.: Complexity Theory - Exploring the Limits of Efficient Algorithms. Springer, Dordrecht, The Netherlands (2005)