Inverse HAMILTONIAN CYCLE and Inverse 3-D MATCHING Are coNP-Complete

Michael Krüger and Harald Hempel

Institut für Informatik, Friedrich-Schiller-Universität Jena
{krueger, hempel}@minet.uni-jena.de

Abstract. In this paper we show that the inverse problems of HAMILTONIAN CYCLE and 3-D MATCHING are coNP complete. This completes the study of inverse problems of the six natural NP-complete problems from [2] and answers an open question from [1]. We classify the inverse complexity of the natural verifier for HAMILTONIAN CYCLE and 3-D MATCHING by showing coNP-completeness of the corresponding inverse problems.

Keywords: computational complexity, coNP-completeness, inverse NP-problems, HAMILTONIAN CYCLE, 3-DIMENSIONAL MATCHING.

1 Introduction

The influential book by Garey and Johnson [2] lists six natural NP-complete languages: 3SAT, VERTEX COVER (VC), CLIQUE, HAMILTONIAN CYCLE (HC), 3-D MATCHING (3DM) and PARTITION. When it comes to studying the complexity of inverse NP problems it seems desirable to start by investigating the inverse problems of the above six examples. The inverse problems of 3SAT, VC, CLIQUE, and PARTITION have been shown to be coNP-complete in [4, 1].

In this paper we show that the inverse problems for the remaining two problems HC and 3DM, are coNP-complete. This settles an open question from [1] and contributes to the growing knowledge about the complexity of inverse NP problems [4, 1, 3]. In particular we show that inverting the natural verifiers of HC and 3DM, is complete for the class coNP.

The complexity class NP is often referred to as the class of problems having polynomial-time verifiers. A polynomial-time verifier V is a polynomial-time computable function mapping from $\Sigma^* \times \Sigma^*$ to $\{0, 1\}$ such that there exists a polynomial p such that for all $x, \pi \in \Sigma^*$, $V(x, \pi) = 1$ implies $|\pi| \leq p(|x|)$. The language $L(V)$ associated with a verifier V is defined as $L(V) = \{x \in \Sigma^* : (\exists \pi \in \Sigma^*)[V(x, \pi) = 1]\}$. It is well-known that $\mathrm{NP} = \{L(V) : V \text{ is a polynomial-time verifier}\}$. The inverse problem for a verifier V is given a set $\Pi \subseteq \Sigma^*$ (represented as a list) to decide if there exists a string x such that $\Pi = \{\pi \in \Sigma^* : V(x, \pi) = 1\}$. It appears that inverting verifiers has an Σ_2^p upper bound but this bound only holds for so called fair verifiers [1]. However, even though there do exist verifiers such that their inverse problems are

T. Asano (Ed.): ISAAC 2006, LNCS 4288, pp. 243–252, 2006.

Σ_2^p complete, the inversion of natural verifiers for some NP-complete languages is complete for the class coNP [4, 1, 3]. It is known that different verifiers for one and the same NP problem may have inverse problems of different complexity [1]. However for many NP-complete languages L there seems to exist a verifier that deserves to be called "the natural verifier" for L and we will focus on the inverse problems relative to those natural verifiers. We mention in passing that a different definition of the inverse problem for a verifier has been studied and a very general coNP-hardness result been proven in [3]. The difference between the two concepts is the representation of the input as a list (see [1] and above) or as a boolean circuit (see [3]).

Our paper is organized as follows. In Chap. 2 we define the basic concepts and some useful graph modules. The two main theorems of this paper are stated in Sect. 3. Due to space restrictions we only give a proof for the coNP-completeness of the inverse problem of HC. We mention that the proof idea can be modified to also work for the inverse problem of HC in directed graphs and thus coNP-completeness holds in the directed case as well.

2 Preliminaries

We assume the reader to be familiar with the basic definitions and notations from graph theory [6] and complexity theory [5]. Let $\Sigma = \{0, 1\}$ be our alphabet.

2.1 Inverse Problems

The class NP is also called the class of problems with short proofs. Essential for a definition that follows that informal description is the notion of a verifier.

Definition 1. *1. A (polynomial-time) verifier V is a polynomial-time computable function $V : \Sigma^* \times \Sigma^* \to \{0, 1\}$ such that there exists a polynomial p satisfying that for all $x, \pi \in \Sigma^*$, $(x, \pi) \in V \implies |\pi| \leq p(|x|)$.*

2. For a (polynomial-time) verifier V let $V(x)$ denote the set of proofs for a string $x \in \Sigma^*$, that is, for all $x \in \Sigma^*$, $V(x) = \{\pi \in \Sigma^* : V(x, \pi) = 1\}$.

3. The language $L(V)$ accepted by a polynomial-time verifier V is defined as $L(V) = \{x \in \Sigma^* : V(x) \neq \emptyset\}$.

It is well known that NP is the class of languages that can be accepted by (polynomial-time) verifiers. Inverse problems are defined relative to a verifier V.

Definition 2 ([1]). *The inverse problem V^{-1} for a verifier V is defined as*

$$V^{-1} = \{\Pi \subseteq \Sigma^* : (\exists x \in L(V))[V(x) = \Pi]\}.$$

The inverse problem of a language $A \in$ NP can clearly only be defined relative to a verifier accepting A. We will study the inverse problems of the natural verifiers for 3SAT and HC.

The concept of a candidate function is a useful tool when studying the complexity of inverse problems.

Definition 3 ([1]). *Let V be a verifier. A polynomial-time computable mapping $c : \mathcal{P}(\Sigma^*) \to \Sigma^*$ is called a candidate function for V if and only if for all $\Pi \subseteq \Sigma^*$: if there exists a $z \in \Sigma^*$ such that $V(z) = \Pi$ then $V(c(\Pi)) = \Pi$.*

It is not clear if all verifiers do have candidate functions. However, many natural verifiers for NP-complete languages, such as 3SAT or VC, have candidate functions. Note that if a verifier V has a candidate function c then we have an obvious coNP upper bound for the complexity of V^{-1}, namely given Π, compute $c(\Pi)$, and then check if for all π such that $|\pi| \leq p(|c(\Pi)|)$ (where p is the polynomial that bounds the length of witnesses with respect to the verifier V) we have $\pi \in \Pi \iff V(c(\Pi), \pi) = 1$.

Observation 1. *If V is a verifier with a candidate function, then $V^{-1} \in$ coNP.*

2.2 HAMILTONIAN CYCLE

A cycle in a graph $G = (V, E)$ is assumed to be a set $C \subseteq E$ such that there exist pairwise different vertices $x_1, x_2, x_3, \ldots x_{k-1}, x_k$, $k \geq 3$, such that $C = \{\{x_1, x_2\}, \{x_2, x_3\}, \ldots, \{x_{k-1}, x_k\}, \{x_k, x_1\}\}$. A Hamiltonian cycle in a (simple and undirected) graph is a cycle in G that contains every vertex of G.

The NP-complete problem HC is defined as the set of all (simple and undirected) graphs that contain a Hamiltonian cycle.

Definition 4. *The verifier V_{HC} is defined as $V_{\mathrm{HC}}(G, C) = 1$ if G is a simple, undirected graph and C is a Hamiltonian cycle in G and $V_{\mathrm{HC}}(G, C) = 0$ otherwise.*

Clearly, V_{HC} is a verifier for HC and since it appears to be the most natural verifier for HC we will choose Invs-HC as a more intuitive notation for V_{HC}^{-1}.

Following a more general concept from [1] we will call a collection of sets of edges $\Pi = \{C_1, C_2, \ldots, C_k\}$ well-formed if and only if $C_1, C_2, \ldots, C_k$ are cycles over a common vertex set V such that $||C_1|| = ||C_2|| = \cdots = ||C_k|| = ||V||$. It is not hard to see that non well-formed sets Π can not be in Invs-HC. Obviously, testing if an instance Π is well-formed can be done in polynomial time.

Looping back to the notion of a candidate function (see Definition 3) it is not hard to see that the verifier V_{HC} has a candidate function c_{HC}. Given a well-formed collection of sets of edges $\Pi = \{C_1, C_2, \ldots, C_k\}$ let $c_{\mathrm{HC}}(\Pi)$ be the graph induced by $C_1, C_2, \ldots, C_k$, that is $c_{\mathrm{HC}}(\Pi) = (V, E)$ such that $V = \{v : (\exists u)[\{u, v\} \in C_1]$ and $E = C_1 \cup C_2 \cup \cdots \cup C_k$. The following corollary is an immediate consequence of Observation 1.

Corollary 1. *The problem Invs-HC is in* coNP.

2.3 3-DIMENSIONAL MATCHING

The NP-complete problem 3-DIMENSIONAL MATCHING(3DM) is defined as follows.

Proc. 17th ISAAC-06, Vol. 4288 in LNCS, pp. 243-252, Springer, 2006

3-DIMENSIONAL MATCHING

Input: A 4-tuple (S, X, Y, Z) of sets such that S is a subset of $X \times Y \times Z$ and X, Y and Z have the same number of elements.

Question: Does S contain a 3D-Matching, i.e., a subset $M \subseteq S$ such that $|M| = |X|$ and no two elements of M agree in any coordinate.

As many other NP-complete problems 3DM has a natural verifier.

Definition 5. *The verifier V_{3DM} is defined via $V_{\mathrm{3DM}}((S, X, Y, Z), M) = 1$ if S is a subset of $X \times Y \times Z$, X, Y and Z are sets, having the same number of elements and M is a 3D-Matching for S. Otherwise $V_{\mathrm{3DM}}((S, X, Y, Z), M) = 0$.*

It is easy to see that V_{3DM} is a verifier for 3DM. Since we feel that V_{3DM} is the most natural verifier for 3DM we let Invs-3DM denote the language V_{3DM}^{-1}.

2.4 3-SATISFIABILITY

One of the standard NP-complete problems is 3-SATISFIABILITY (3SAT), the set of all satisfiable boolean formulas in 3-conjunctive normal form (3-CNF), that is any clause has at most three literals. 3SAT will play an important role in the proofs of our main theorems.

A natural verifier for 3SAT is the following: $V_{\mathrm{3SAT}}(F, \alpha) = 1$ if F is a boolean formula in 3-CNF and α is a satisfying assignment for the variables of F, and $V_{\mathrm{3SAT}}(F, \alpha) = 0$ otherwise. The inverse problem V_{3SAT}^{-1} also has been denoted by Invs-3SAT or $\mathrm{3SAT}^{-1}$ [1, 4]. Throughout this paper we will use Invs-3SAT to denote V_{3SAT}^{-1}.

As it has been the case for V_{HC} there are easy to check properties that a proof set for V_{3SAT} has to have in order to be in Invs-3SAT. Since any assignment for an n-variable boolean formula is represented by a string from $\{0, 1\}^n$, the notion of well-formed proof sets Π with respect to V_{3SAT} only requires that all strings from Π have the same length.

Theorem 2 **([4]).** *Invs-3SAT is* coNP*-complete.*

A concept that will be useful for our purposes as well was defined in [4].

Definition 6 **([4]).** *Let Π be a set of boolean assignments for $x_1, x_2, \ldots, x_n$.*

1. *An assignment α for $x_1, \ldots, x_n$ is said to be $\{x_i, x_j, x_k\}$-*compatible *with Π, $1 \leq i < j < k \leq n$, if and only if there exists an assignment $\beta \in \Pi$, such that α and β assign the same truth values to x_i, x_j, x_k.*
2. *An assignment for $x_1, \ldots, x_n$ is called* 3-compatible *with Π if and only if it is $\{x_i, x_j, x_k\}$-compatible with Π for each triplet x_i, x_j, x_k of variables, $1 \leq i < j < k \leq n$.*

The notion of 3-compatibility leads to a useful characterization of Invs-3SAT.

Theorem 3 **([4]).** *A well-formed set of proofs Π for V_{3SAT} is in Invs-3SAT if and only if it is closed under 3-compatibility, i. e., if and only if for each assignment α it holds, that if α is 3-compatible with Π then $\alpha \in \Pi$.*

Proc. 17th ISAAC-06, Vol. 4288 in LNCS, pp. 243-252, Springer, 2006

2.5 Some Helpful Graph Modules for Hamiltonian Cycles

For the proof of our main result we need two simple but very useful graph modules, that will help us to direct a Hamiltonian cycle through a given graph. The first module is the parity-module, introduced in [5].

The parity module "connects" two edges e and f of a graph G and forces each Hamiltonian cycle to either enter and leave the parity module through the endpoints of the original edge e or enter and leave the parity module through the endpoints of the original edge f, but not both, i.e., each Hamiltonian cycle either "uses" e or f but not both. In all forthcoming figures we will use the symbol shown in Fig. 1 to express that a pair of edges is connected by a parity-module.

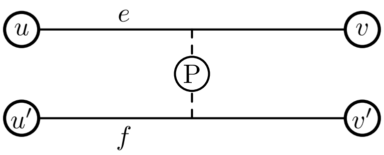

Fig. 1. Symbolic depiction of two parity-connected edges $e = \{u, v\}$ and $f = \{u', v'\}$

It is not hard to extend this idea to connecting f with several edges $e_1, e_2, \ldots,$ e_k via parity-modules. We obtain a module that relates the edges $e_1, e_2, \ldots, e_k$ of a graph in such a way that each Hamiltonian cycle of the modified graph uses all of the edges $e_1, e_2, \ldots, e_k$ or none of them.

Lemma 1. *Let G be an undirected graph with an edge f, that is used by each Hamiltonian cycle of G and let $e_1, e_2, \ldots, e_k$ be pairwise different edges of G. The graph G is modified to G' by first inserting an edge f' that connects the endpoints of f and then inserting parity modules between f' and each of $e_1, e_2, \ldots, e_k$.*

Each Hamiltonian cycle of G' uses either all of $e_i, 1 \leq i \leq n$ or none of them.

The proof is omitted. We will call edges that are connected in the sense of the above Lemma 1 *all-connected* and symbolize this connection as done in Fig. 2. The symbolization does not include the connection to the edge f. In the forthcoming proof it will always be clear which edge will play the role of the edge f.

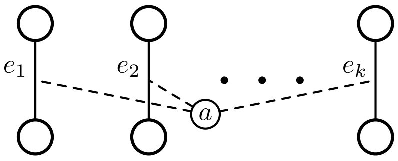

Fig. 2. Symbolization for all-connected edges $e_1, e_2, \ldots, e_k$

Proc. 17th ISAAC-06, Vol. 4288 in LNCS, pp. 243-252, Springer, 2006

3 Main Results

We now state the main results of this paper. However, due to space restrictions we will only prove the coNP-completeness of Invs-HC by giving a reduction from Invs-3SAT in the remainder of this section.

Theorem 4. *Invs-3DM is* coNP*-complete.*

Theorem 5. *Invs-HC is* coNP*-complete.*

Proof of Theorem 5: Due to Corollary 1 it suffices to give a $\leq_m^p$-reduction from the coNP-complete problem Invs-3SAT (Theorem 2) to Invs-HC. If a proof set Π_{3SAT} is not well-formed (with respect to V_{3SAT}) then it is not in Invs-3SAT and hence we will map it to a fixed non-member of Invs-HC. Given a well-formed proof set Π_{3SAT} for V_{3SAT}, we will construct a graph $G_{\Pi_{\text{3SAT}}}$ that will contain a Hamiltonian cycle for each assignment (via an one-to-one correspondence) that is 3-compatible with Π_{3SAT}. The proof set Π_{HC} can then be easily extracted from $G_{\Pi_{\text{3SAT}}}$, and it will contain exactly those Hamiltonian cycles from $G_{\Pi_{\text{3SAT}}}$ that correspond to the assignments in Π_{3SAT}. Recall that by Theorem 3 for any proof set Π_{3SAT} it holds that Π_{3SAT} is in Invs-3SAT if and only if Π_{3SAT} is closed under 3-compatibility. Our construction will ensure that the latter is the case if and only if the candidate graph of Π_{HC} contains exactly the Hamiltonian cycles from Π_{HC}.

Now let Π_{3SAT} be well-formed. Hence, there exist $n, m \in \mathbb{N}$ such that $\Pi_{\text{3SAT}} = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ and for all i, $1 \leq i \leq m$, $\alpha_i \in \{0,1\}^n$. The strings α_i will be interpreted as assignments to n boolean variables $y_1, y_2, \ldots, y_n$ in the canonical way, and we will write $\alpha_i(y_j)$ to denote the jth bit of α_i. Recall that by Theorem 3 $\Pi_{\text{3SAT}} \in$ Invs-3SAT if and only if Π_{3SAT} is closed under 3-compatibility, that is, if and only if Π_{3SAT} contains all assignments, that are 3-compatible with Π_{3SAT}.

We will now construct a graph $G_{\Pi_{\text{3SAT}}}$ in stages that will contain exactly one Hamiltonian cycle for each assignment that is 3-compatible with Π_{3SAT}.

Construction Step 1. *The construction starts with a simple cycle C_l, $l = n+2\binom{n}{3}+1$. Fix n consecutive edges $e_1, e_2, \ldots, e_n$ in C_l and for each i, $1 \leq i \leq n$, add one new edge f_i to C_l that connects the endpoints of e_i (and so produce a chain of n double edges). Let $G'_{\Pi_{\text{3SAT}}}$ be the graph constructed so far.*

Even though $G'_{\Pi_{\text{3SAT}}}$ is not a simple graph the upcoming stages of the construction will ensure that the final graph $G_{\Pi_{\text{3SAT}}}$ is simple.

The chain of "double" edges e_i, f_i, $1 \leq i \leq n$ will be called n-chain in the following. For each i, we associate the edges e_i, f_i with the variable y_i. Note that the n-chain induces exactly 2^n Hamiltonian cycles in $G'_{\Pi_{\text{3SAT}}}$, one for each possible assignment of the variables $y_1, y_2, \ldots, y_n$. The edges e_i (f_i), $1 \leq i \leq n$, will also be called 0-edges (1-edges) referring to the desired correspondence between Hamiltonian cycles and assignments. The usage of a 0-edge e_i (1-edge f_i) in a Hamiltonian cycle will represent assigning the boolean value 0 (1) to y_i. Hence any Hamiltonian cycle traversing the n-chain canonically corresponds to an assignment for $y_1, y_2, \ldots, y_n$ and vice versa.

The remaining part of the construction of $G_{\Pi_{3\mathrm{SAT}}}$ consists of the insertion of subgraphs into $G'_{\Pi_{3\mathrm{SAT}}}$ in order to restrict the set of Hamiltonian cycles to those that correspond to assignments that are 3-compatible with $\Pi_{3\mathrm{SAT}}$. Recall that an assignment β is called 3-compatible with a set of assignments Π (over the same variable set) if for each three-element set of variables $\{y_{i_1}, y_{i_2}, y_{i_3}\}$ the assignment β is $\{y_{i_1}, y_{i_2}, y_{i_3}\}$-compatible with Π.

We will now define gadgets H_i, $1 \leq i \leq \binom{n}{3}$, one gadget for each three-element set of variables $\{y_{i_1}, y_{i_2}, y_{i_3}\}$, that will eventually be subgraphs of the to be constructed graph $G_{\Pi_{3\mathrm{SAT}}}$. The structure of a subgraph H_i associated with the three variables $y_{i_1}, y_{i_2}, y_{i_3}$ and its connections to the remaining graph, in particular the n-chain, will ensure that every Hamiltonian cycle in $G_{\Pi_{3\mathrm{SAT}}}$ corresponds to an assignment that is $\{y_{i_1}, y_{i_2}, y_{i_3}\}$-compatible with $\Pi_{3\mathrm{SAT}}$.

The gadgets H_i, $1 \leq i \leq \binom{n}{3}$, will all have the same structure and so without loss of generality we will only describe the construction of the gadget for the three variables y_1, y_2, y_3, call it H_1. The gadget H_1 will also be constructed in stages.

First we define the define set

$$\Pi^1_{3\mathrm{SAT}} = \{a_1a_2a_3 \in \{0,1\}^3 : (\exists \alpha \in \Pi_{3\mathrm{SAT}})(\forall i : 1 \leq i \leq 3)[\alpha(y_i) = a_i]\}.$$

So, $\Pi^1_{3\mathrm{SAT}}$ is the set of pairwise different partial $\{y_1, y_2, y_3\}$-assignments in $\Pi_{3\mathrm{SAT}}$, i.e., assignments from $\Pi_{3\mathrm{SAT}}$ restricted to the variables y_1, y_2, and y_3. In other words $\Pi^1_{3\mathrm{SAT}}$ consists of those possible triples of values $(\beta(y_1), \beta(y_2), \beta(y_3))$ for an assignment β that is $\{y_1, y_2, y_3\}$-compatible with $\Pi_{3\mathrm{SAT}}$. Let k_1 denote the number of elements in $\Pi^1_{3\mathrm{SAT}}$ and note that $k_1 \leq 8$.

Construction Step 2a. *The construction of the gadget H_1 starts with a path of four edges. After "doubling" the first three edges, i.e., inserting new edges connecting their endpoints and so building a graph consisting of a chain of three double edges followed by a (single) edge, we obtain a graph K'. Connect k_1 copies of K', call them $K'_1, K'_2, \ldots, K'_{k_1}$ in a path-like manner by identifying the start and end vertices of the original path of four edges of consecutive copies of K' (see also Fig. 3).*

Each chain of (three) consecutive double edge will be called a 3-chain and every 3-chain will correspond to an element $a_1a_2a_3$ from $\Pi^1_{3\mathrm{SAT}}$. This correspondence will play an important role in the upcoming Construction Step 2b. Within each 3-chain of H_1 we will associate the first double edge with the variable y_1, the second with y_2 and the third with y_3, where one edge participating in a double edge will be called 0-edge while the other will be called 1-edge.

The next construction step for H_1 deals with the issue that H_1 is supposed to handle $\{y_1, y_2, y_3\}$-compatibility. Informally put, the traversal of a Hamiltonian cycle through the n-chain, i.e. the usage of 0- and 1-edges in the n-chain, will effect the traversal of that Hamiltonian cycle through the gadget H_1. So in the upcoming step we describe how that yet to be completely defined gadget H_1 is connected to the n-chain.

Construction Step 2b. *Let e' be a 0-edge in H_1. Suppose e' is part of a 3-chain that is associated with the partial assignment $a_1a_2a_3 \in \Pi^1_{\mathrm{3SAT}}$, $a_1, a_2, a_3 \in \{0, 1\}$, and let e' be associated with the variable y_i, $1 \leq i \leq 3$, within that 3-chain. Connect e' with f_i (the 1-edge in the n-chain that is associated with y_i) via a parity module if and only if $a_i = 1$ and connect e' with e_i (the 0-edge in the n-chain that is associated with y_i) via a parity module if and only if $a_i = 0$.*

Suppose that C is a Hamiltonian cycle in the to be constructed graph $G_{\Pi_{\mathrm{3SAT}}}$ that by its traversal through the n-chain defines an assignment β. Consider a 3-chain K in H_1 that is associated with a partial assignment $a_1a_2a_3$ and the 0-edge (edge) in that 3-chain associated with y_1. Observe that by the above insertion of the parity modules we have that C does not use the 0-edge in K that is associated with y_1 if and only if $a_1 = \beta(y_1)$.

It follows that a Hamiltonian cycle C corresponds to an assignment (the assignment defined by the Hamiltonian cycle's traversal of the n-chain) that is $\{y_1, y_2, y_3\}$-compatible with Π_{3SAT} if and only if there exists one 3-chain K in H_1 such that C does not use any of the three 0-edges in K.

The next step introduces three auxiliary vertices in H_1 that will force any Hamiltonian cycle in $G_{\Pi_{\mathrm{3SAT}}}$ to avoid all three 0-edges in at least one 3-chain K and so in light of the above comment force any Hamiltonian cycle in $G_{\Pi_{\mathrm{3SAT}}}$ to correspond to an assignment that is (y_1, y_2, y_3)-compatible with Π_{3SAT}.

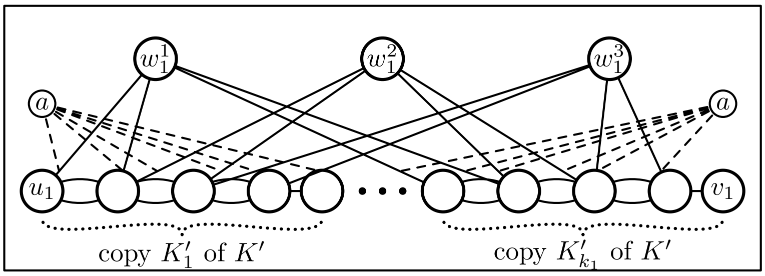

Fig. 3. The structure of the gadget H_1

Construction Step 2c. *Add three new vertices w_1^1, w_2^1, w_3^1 to the gadget H_1, that will be associated with the variables y_1, y_2, and y_3, respectively. For each i, $1 \leq i \leq 3$, and each 3-chain K in H_1 add edges from the endpoints of its y_i-double-edge to w_i^1. Furthermore, for each 3-chain K all-connect (see Lemma 1) the six edges between K and the vertices w_1^1, w_2^1, w_3^1* [1] *(see Fig. 3).*

In order to see that the introduction of w_1^1, w_2^1, w_3^1 has the desired effect note that the three new vertices have to be visited by each Hamiltonian cycle. Each edge leading from a 3-chain to one of the new vertices is all-connected to the

[1] Note that all-connection requires the existence of an auxiliary edge, that is used by each Hamiltonian cycle. However the 3-chain K contains an edge that has not been doubled in Construction Step 2a and so it can be used for exactly this purpose.

other five edges that connect this 3-chain with one of w_1^1, w_2^1, w_3^1. It follows that w_1^1, w_2^1, w_3^1 will be visited by any Hamiltonian cycle via edges coming from one and the same 3-chain.

Now it easy to see that each Hamiltonian cycle avoids all three 0-edges in at least one 3-chain, namely in the 3-chain, from which the Hamiltonian cycle visits w_1^1, w_2^1, w_3^1. If a Hamiltonian cycle C would use one of these 0-edges there would be a small cycle in C consisting of the 0-edge and those two edges between the 3-chain and w_1^1, w_2^1, w_3^1, that start at the endpoints of the 0-edge, a contradiction.

Observe that with respect to a potential Hamiltonian cycle C in the graph $G_{\Pi_{3\mathrm{SAT}}}$ and C's traversal of the n-chain the parity-connections between the n-chain and the gadget H_1 uniquely define C's way through the series of 3-chains inside H_1 and hence also the way the vertices w_1^1, w_2^1, w_3^1 are traversed. The latter is determined by the 3-chain in which C avoids all three 0-edges, in other words by the 3-chain that witnesses the $\{y_1, y_2, y_3\}$-compatibility of the assignment defined by C's traversal through the n-chain.

This completes the construction of H_1. It is obvious how the gadgets for other triples of variables have to be constructed. The overall structure of the gadgets $H_2, \ldots, H_{\binom{n}{3}}$ is identical to the structure of H_1 as shown in Fig. 3, except for the names of the vertices and the number of copies of K'. The connections of a gadget to the n-chain (not shown in Fig. 3) also differ between the gadgets.

This concludes the construction of the gadgets H_i and we will return to the construction of $G_{\Pi_{3\mathrm{SAT}}}$. Let for all i, u_i and v_i be the "first" and "last" vertices of H_i (see Fig. 3).

Construction Step 3. *Insert the gadgets $H_1, \ldots H_{\binom{n}{3}}$ into the graph $G'_{\Pi_{3\mathrm{SAT}}}$ as shown in Fig. 4. In particular, recall that the graph $G'_{\Pi_{3\mathrm{SAT}}}$ contains a simple path of $2\binom{n}{2} + 1$ edges. Replace any second edge of that simple path by one gadget (replacing an edge $\{u'', v''\}$ by H_i means removing the edge $\{u'', v''\}$ from $G'_{\Pi_{3\mathrm{SAT}}}$ and identifying the vertices u'' and v'' with the vertices u_i and v_i—the "first" and "last" vertices of H_i—respectively). Note that by inserting a gadget H_i any connections from H_i to the n-chain, i.e., the parity modules spoken of in Construction Step 2b, are also inserted in $G_{\Pi_{3\mathrm{SAT}}}$.*

This completes the construction of the graph $G_{\Pi_{3\mathrm{SAT}}}$. As mentioned at the beginning of the proof, $G_{\Pi_{3\mathrm{SAT}}}$ has the property that it contains exactly one Hamiltonian cycle for each assignment that is 3-compatible with $\Pi_{3\mathrm{SAT}}$.

Lemma 2. *Let $\Pi_{3\mathrm{SAT}}$ be a well-formed set of proofs for $V_{3\mathrm{SAT}}$. There is a bijective mapping between the set of assignments that are 3-compatible with $\Pi_{3\mathrm{SAT}}$ and the set of Hamiltonian cycles in $G_{\Pi_{3\mathrm{SAT}}}$.*

Note that the size of the constructed graph $G_{\Pi_{3\mathrm{SAT}}}$ is polynomial in the length n of the strings in $\Pi_{3\mathrm{SAT}}$ and thus in $|\Pi_{3\mathrm{SAT}}|$. Also, the construction of $G_{\Pi_{3\mathrm{SAT}}}$ can be done in time polynomial in $|\Pi_{3\mathrm{SAT}}|$. Furthermore, it follows from the construction of the graph $G_{\Pi_{3\mathrm{SAT}}}$ that given a well-formed set of proofs $\Pi_{3\mathrm{SAT}}$ for $V_{3\mathrm{SAT}}$ and an assignment $\alpha \in \Pi_{3\mathrm{SAT}}$ the Hamiltonian cycle that is associ-

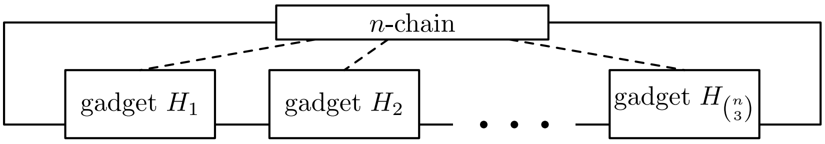

Fig. 4. The overall structure of $G_{\Pi_{\text{3SAT}}}$. The dashed lines represent the "connections" between the gadgets H_i and the n-chain that are realized by several parity-modules.

ated with α via the bijective mapping spoken of in the above Lemma 2 can be constructed in time polynomial in Π_{3SAT}.

We will now turn to formally define the function f that reduces Invs-3SAT to Invs-HC. As already mentioned at the beginning of this proof f maps non well-formed proof sets Π_{3SAT} to a fixed non-member of Invs-HC. For well formed proof-sets Π_{3SAT} we define $f(\Pi_{\text{3SAT}})$ to be the set of those Hamiltonian cycles in $G_{\Pi_{\text{3SAT}}}$ that correspond to the assignments from Π_{3SAT} via the mapping spoken of in Lemma 2. Note that any assignment from Π_{3SAT} is 3-compatible with Π_{3SAT} and hence there does indeed exist a Hamiltonian cycle in $G_{\Pi_{\text{3SAT}}}$ for every assignment from Π_{3SAT} (Lemma 2). Since checking whether a proof set is well-formed, constructing the graph $G_{\Pi_{\text{3SAT}}}$, and also extracting Hamiltonian cycles corresponding to given assignments can all be done in polynomial time it follows that f is computable in polynomial time.

It remains to show that for all Π_{3SAT} we have $\Pi_{\text{3SAT}} \in$ Invs-3SAT $\longleftrightarrow$ $f(\Pi_{\text{3SAT}}) \in$ Invs-HC. Due to space restrictions that part of the proof is omitted. □

Acknowledgement. We thank Tobias Berg for all the inspiring and helpful discussions and we thank the anonymous reviewers for their useful comments.

References

[1] H. Chen. Inverse NP problems. *Mathematical foundations of computer science (2003), 338–347, Lecture Notes in Comput. Sci., 2747, Springer, Berlin, 2003*

[2] M. Garey and D.Johnson. *Computers and Intractability: A Guide to the Theory of* NP*-completeness*,W.H.Freeman Company, 1979

[3] E. and L.Hemaspaandra, H. Hempel. All superlinear inverse schemes are coNP-hard. *Theoretical Computer Science 345 (2005), no. 2-3, 345–358.*

[4] D. Kavvadias and M. Sideri. The inverse satisfiability problem. *SIAM Journal on Computing 28(1) (1998), no. 1, pp. 152–163*

[5] C. H. Papadimitriou. *Computational Complexity*, Addison-Wesley, 1994

[6] D.B. West. *Introduction to Graph Theory*, Prentice Hall, 2001