Planar Hamiltonian problems and linear parsimonious reductions

Régis Barbanchon (regis@info.unicaen.fr)
GREYC, University of Caen, France

January 13, 2001

Abstract

We continue a series of results of [Gra96], [Cre95] (see also [LW99], [GS99]) which establish that quite a number of natural NP-complete problems are linearly reducible each other. The contribution of this paper is threefold:

1. A new very strict linear reduction, called "exact reduction" is studied. This is essentially a parsimonious transformation computable in linear time. One interest of such a reduction is that if some problem A is exactly reducible to some problem B and conversely, then problems A and B have exactly the same complexity according to: (a) the existence of a solution, (b) the number of their solutions, (c) the enumeration of their solutions (the same delay between two consecutive solutions, up to a multiplicative constant).
2. For planar (resp. non-planar) graphs we prove that nine natural variants of the Hamiltonian problem (directed/undirected, path/cycle, bounded/unbounded degree, etc) are exactly reducible each other.
3. We prove that the PLAN-SAT (resp. SAT) problem is exactly reducible to PLAN-HAM (resp. HAM) problem. This implies with (2) that (a) the many variants of problem #PLAN-HAM are #P-complete. (b) the many variants of problem UNIQUE-PLAN-HAM are DP-complete under random polynomial reduction, which fully answers a question asked by [HMRS98].

Results (2) and (3) are proved in a modular and systematic way by using a toolkit that is a series of highly reusable planar graph gadgets. Each gadget has nice local properties according to the Hamiltonian problem and acts as a logical or arithmetical gate in an electronic circuit. Finally, we present some results or conjectures which show the optimality of our results.

Keywords: planarity, Hamiltonian problems, satisfiability, parsimonious reductions, linear time.

1 Introduction

One of the most important questions in Computational Complexity is probably how non-determinism makes a machine stronger. In other words, how faster a given machine may solve a problem if at any step of the computation it is allowed to guess a bit or an integer. When dealing with decision problems, this question is usually asked as follows: Is a super-polynomial number of deterministic operations needed to simulate a polynomial number of (possibly non-deterministic) operations, that is, P $\neq$ NP? (the reference machine is then usually the multi-tape Turing Machine since this model is robust enough regarding polynomial time). Unfortunately, this question is still unsolved today, and some authors investigated weaker reformulations by focusing on non-deterministic linear time instead of polynomial non-deterministic time (see also [Sch78] and [GS89] for investigations of non deterministic quasi-linear time $O(n \log^{O(1)} n)$ on Turing Machine and RAM, respectively). The question then becomes: Is a non-linear number of deterministic operations needed to simulate a linear number of non deterministic operations on a given model?

When the underlying model is the Turing machine, this question is positively answered by the separation theorem NTIME(n) $\neq$ DTIME(n) due to Paul et al. (see [PPST83] and [BDG90]). However, the Turing machine is a rather weak model regarding deterministic linear time, in the sense that the set of problems solvable within this time on this machine is poor. This is due to the fact that in every step of a computation, the head is only allowed to move one cell left or right from its current position on the tape. This local behaviour of the head practically implies that only $O(1)$ passes on the tape are allowed to solve a problem if linear time is wanted.

In [Gra96] (see also [GO98], [Sch97] and [GS99]), Grandjean defined a robust model of computation for deterministic (DLIN) and non-deterministic (NLIN) linear time. The underlying machine is the RAM (Random access Machine), an abstraction of real computers, which has the ability to access any cell in time $O(1)$ by its index. Within this model, every integer involved in a computation must hold a value $O(n)$ where n is the size of the input. This includes the indexes (which implies linear space) and also the guessed values. On one hand, DLIN contrasts with DTIME(n) because it still contains interesting problems (such as Planarity, Horn satisfiability, Connectivity, $\ldots$) while still being a strict subset of P. On the other hand, while NLIN is a strict subset of NP (see [Coo73] and [SFM78] for a hierarchy of non-deterministic time), it still contains natural NP-complete problems (indeed most of them) such as the 3-colourability, few of them being also NLIN-complete (such as the function contractability problem). Thus, a proof of DLIN $\neq$ NLIN seems to be easier to find than a proof of P $\neq$ NP, and even if DLIN $\neq$ NLIN does not imply P $\neq$ NP, such a separation theorem would still be impressive.

In this paper, we focus on the Hamiltonian problem and its many variants (all globally denoted HAM) and also on the satisfiability problem (denoted SAT). Given a graph or digraph G, the HAM problem asks whether there exists a path in G visiting all its vertices exactly once. The HAM variants may relax or stress constraints on both the desired path (it may have unspecified distinct ends, specified distinct ends, or it may be constrained to be a cycle) and the graph (it may be directed, undirected, planar or not, degree-bounded or not). The SAT problem asks whether a propositional formula in conjunctive normal form (CNF) has a satisfying assignment of its variables. Such a formula may be seen as a bipartite graph problem by identifying a vertex to each variable and a vertex to each clause, and by creating an edge $e = (v, c)$ for each variable v occurring in a clause c. If the graph is planar, then the problem is called the planar satisfiability problem (denoted PLAN-SAT).

While SAT and HAM both belong to NLIN, none of them are believed to be NLIN-complete. This conjecture is motivated by the fact that the witnesses (i.e. the solutions) for a SAT or HAM instance are sublinear in the size of the input as explained below. As far as SAT is concerned, the n variables are basically represented as integers needing $\Theta(\log n)$ bits-long registers on the RAM, and for each variable we only need to decide its truth assignment – which is merely a 1-bit guess. So the total guess for SAT involves $O(n)$ bits whereas an input with m occurences involves $\Theta(m \log n)$ bits. The HAM problem does not require more non-determinism: for a graph G with n vertices and m edges, G needs a storage of $\Theta(m \log n)$ bits, and we only need to decide for each edge whether it belongs to an Hamiltonian path – which is also a mere 1-bit guess – and so, globally only $O(m)$ guessed bits are required. Indeed, in order to be a candidate for NLIN-completeness, it is conjectured to require a fully linear amount of non-determinism, that is, one should have to guess $\Theta(n)$ integers each holding $\Theta(n)$ values, encoding for example a satisfying projection or contraction of the input (see [Gra90]). So, if SAT or HAM ever came to be NLIN-complete, this would mean that the amount of non-determinism required to solve such problems could be surprisingly compressed by a logarithmic factor: More precisely, any instance of size s of these problems could be solved using a linear number of deterministic steps $O(s)$ and a sublinear, precisely $\Theta(s/\log s)$, number of non-deterministic steps.

There are clear reasons to study HAM and SAT in NLIN despite their expected non NLIN-completeness: First, HAM and SAT are among the most important NP-complete problems and they have been widely studied in NP (see [JP85]). It is also well-known that SAT and HAM remain NP-complete when the input graphs are planar (see [GJT76]) and that their counting counterparts remain #P-complete, even in the planar case (see [Lic82] and [HMRS98]). The set of problems that are linearly equivalent to SAT also form a large class (see [Cre95] and [Gra96]). Secondly, while SAT is linearly reducible to HAM, it is conjectured that the converse is not true: the constraints expressible by SAT being of a somewhat local nature, a SAT system linear in the size of a graph G does not seems to be able to distinguish between a partition in cycles from a Hamiltonian cycle in G (see the results of [LW99]). Such a distinction becomes possible if we use

$\Theta(\log n)$ variables per vertex, each one encoding a binary digit of the rank of the vertex along a solution cycle of the Hamiltonian problem. This leads to an $O(m \log n)$ reduction HAM $\leq$ SAT, and indeed, we believe this is also a lower bound. However, we cannot prove it, because if the non-existence of any linear reduction HAM $\leq$ SAT came to be proved, this would immediately imply DLIN $\neq$ NLIN.

Hence, we intuitively notice that the locale nature of SAT prevents it to express linearly the connectivity property inherent to HAM. In this paper, we examine what happens if we add an apparent locality feature to the Hamiltonian problem: planarity. As far as non-planar graphs are concerned, it is easy to prove that all non-planar HAM variants cited above are equivalent under linear reduction when no degree bound holds. This is not the case for the PLAN-HAM variants: in particular, the natural linear reductions from a directed HAM variant to an undirected HAM variant and from a HAM path variant to a HAM cycle variant are not planarity preserving. These two natural linear reductions clash with the locality factor introduced by the planarity in two different ways:

- When reducing a PLAN-HAM path to a PLAN-HAM cycle variant, the main difficulty is to connect both ends of a solution path together in order to form a cyclic solution (see Section 3.1 and Figure 5a), which is non trivial because the ends may not be on the boundary of the same face. The difficulty is increased if the ends are unspecified for the HAM path variant.
- When reducing a directed PLAN-HAM variant to an undirected PLAN-HAM variant, the locality factor is the clockwise ordering of the in-going and out-going arcs around a vertex; these two types of arcs may interleave around the vertex and the natural linear reduction does not preserve this ordering, creating crossings between the edges (see Section 3.1 and Figure 5b).

To our knowledge, no such planarity-preserving linear reductions exists till now and it is natural to ask whether the undirected PLAN-HAM variants and the PLAN-HAM cycle variants are more local than the directed PLAN-HAM variants and the PLAN-HAM path variants, respectively. Other natural questions are: while SAT is linearly reducible to HAM, is PLAN-SAT still linearly reducible to PLAN-HAM? Are (PLAN-)HAM variants without a degree bound linearly reducible to degree bounded (PLAN-)HAM variants?

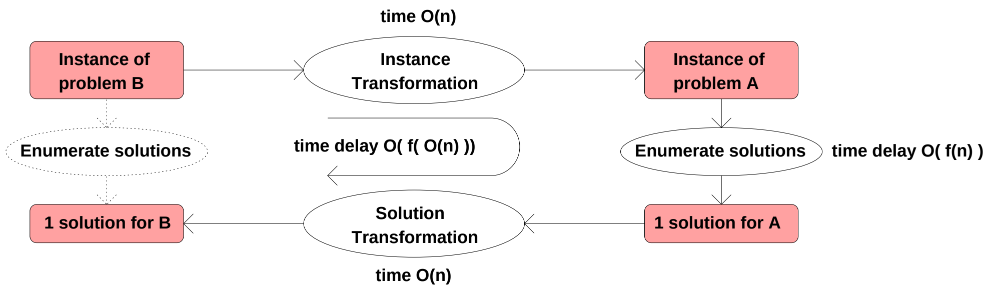


Figure 1: Exact reductions $B \leq A$ and enumeration of solutions

We answer these questions in a strong way within the following framework: we define the *expressiveness* of a given problem A as the set of all problems B "accurately and easily" reducible to it. To formalize this, we want two conditions to be met on the involved reductions from B to A. First, we require that there is a one-to-one correspondence between the solutions of an instance of B and those of the instance of A given by the reduction. This property, known as *parsimony*, shows that the problem A is suitable to capture exactly any instance of B and formalizes the "accuracy" of the reduction. The second requirement imposes that the reduction be computable in a time not only polynomial but *linear* in the size of the instance of B. Moreover, the one-to-one correspondence between the solutions must also be computable in linear time. The linearity formalizes how "easily" problem B can be expressed in terms of problem A. Such a linear parsimonious transformation is called an *exact reduction*. One property of an exact reduction $B \leq A$ is that the enumeration problem associated to B can be computed within the same complexity as the enumeration problem associated to A (see Figure 1) since there is the same delay (up to a constant factor) between the computation of two successive solutions. The results shown in this paper (partially summarized in Figure 2) are the following:

- All the undirected degree-unbounded variants of the non-planar HAM problem are exactly reducible to the degree-bounded non-planar variants with degree bound ≥ 3, and thus the degree-bounded variants and the degree-unbounded variants have the same expressiveness.
- All the variants of PLAN-HAM are exactly reducible each other, including the degree-bounded variants with degree bound ≥ 3, and thus they have the same expressiveness. This will be shown by designing the two types of exact reductions we lacked (from the directed to undirected planar variants, and from the path variants to the cycle variant).
- PLAN-SAT exactly reduces to (any variant of) PLAN-HAM.

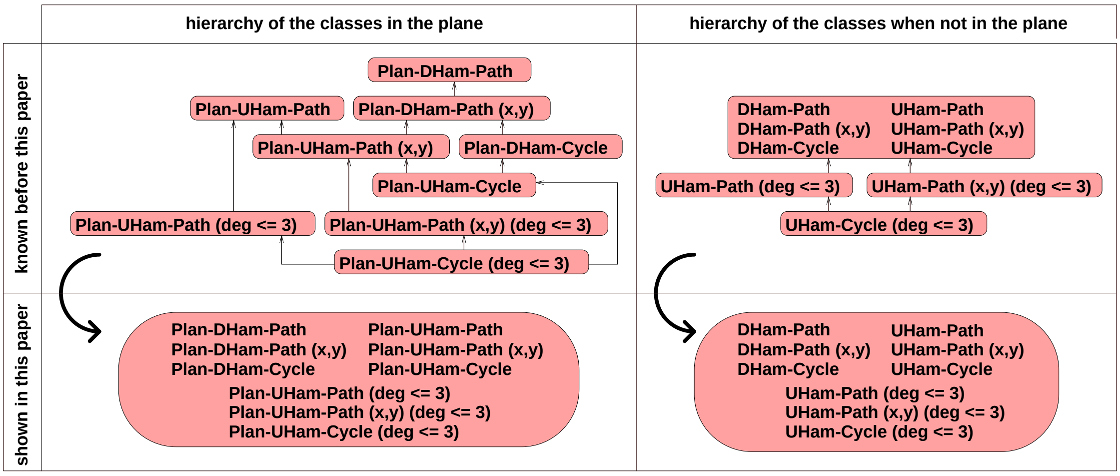


Figure 2: Equivalence under exact reduction between the variants of HAM

The precise definitions of all these variants of the HAM problems can be found in Section 2. As a byproduct of the parsimony, our reductions give alternative proofs for the #P-completeness of the #PLAN-HAM variants. Also, since UNIQUE-SAT is DP-complete under randomized reduction (see [VV86]) even when the instance is planar (see [HMRS98]), the last result shows that UNIQUE-PLAN-HAM is DP-complete under randomized reductions, which answers a question stated as open in [HMRS98].

The expressiveness of the HAM problems lies as much in the above results as in the way we reach them. To design our reductions, we use an unified system of graph gadgets to propagate information encoded as bits, whose design slightly differs from the gadgets usually found in the literature (e.g., gadgets from [JP85] and [Lic82]). Usually, each Hamiltonian gadget g is "relational", that is, it encodes a relation R between two boolean variables, say x and y; a path P assigns values to x and y by choosing to use or not to use some witness edges e_x and e_y, and if the relation xRy is not satisfied by these values, then the gadget g "clashes" in the sense that tP cannot simultaneously visit it and be Hamiltonian. Contrasting to these "relational" gadgets', most of the gadgets of our toolkit are "functional", in the sense that, whether the relation xRy is satisfied or not, an Hamiltonian path P is still able to visit the gadget g, but the status edge e_s ($s = 0$ or 1) by which P leaves g gives the right boolean value $s = R(x, y)$. A typical example of "relational" (resp. "functional") gadget is the XOR-EDGE gadget (resp. the AND gadget) presented in Section 4, and which encodes a binary XOR relation (resp. a binary AND function).

So, our functional gadgets are highly modular and easy to compose because the results of basic functions may be reused to build more complex ones that are direct translations of their formal definitions. In fact, once the three basic gadgets are designed (the AND, the NOT, and a crossover-box), the schemas of the other gadgets built over them are self-explanatory and almost need no proof. In a sense, this is where the expressiveness of the HAM problem shows its essence: it is able to express logic and arithmetic (in a limited way, since in order to keep things linear, we want to avoid any logarithmic factor introduced by the binary representation of integers) *together with the connectivity property.* As previously said, SAT is not believed to express this latter feature because of the locality of this problem. Thus, it is conjectured that the expressiveness of HAM is strictly larger than the expressiveness of SAT.

The paper is structured as follows: In Section 2, we introduce notational conventions for all the studied variants of HAM and define the vocabulary that will be used in all proofs. In Section 3, we recall the easy exact reductions between non-planar variants of HAM and explain how difficult it is to make some of them planarity-preserving. In Sections 8 and 9, we present the two hard reductions that are the core of this article, namely expressing the undirected planar Hamiltonian path problem with the undirected planar Hamiltonian cycle problem, and expressing the planar directed Hamiltonian path problem with the undirected planar Hamiltonian cycle problem. The Hamiltonian toolkit of gadgets are presented in Sections 4 (fundamental gadgets), 5 (crossover-boxes), 6 (derived logical gadgets) and 7 (derived arithmetical gadgets). These sections show how to encode logic (evaluating any boolean operator) and arithmetic (computing the sum of two integers) inside planar Hamiltonian configurations. As far as the undirected planar variants are concerned, Section 10 shows how to decrease the maximal vertex degree to 3 and still have the same expressiveness. Finally, in Section 11 we present an exact reduction from the PLAN-SAT problem to the planar undirected Hamiltonian cycle problem.

2 Notations, definitions

We define the following variants of the HAM problem:

- **UHAM-PATH.** Input: an undirected graph $G(V,E)$. Question: is there a Hamiltonian path in G starting and ending at (unspecified) distinct vertices, that is a path visiting each vertex $v \in V$ exactly once?
- **UHAM-PATH**(x,y)**.** Input: an undirected graph $G(V,E)$ along with two specified distinct vertices $x, y \in V$. Question: is there a Hamiltonian path in G whose ends are x and y?
- **UHAM-CYCLE.** Input: an undirected graph $G(V,E)$. Question: is there a Hamiltonian cycle in G?
- **DHAM-PATH.** Input: a directed graph $G(V,A)$. Question: is there a Hamiltonian path in G starting and ending at (unspecified) distinct vertices?
- **DHAM-PATH**(x,y)**.** Input: a directed graph $G(V,A)$ along with two specified distinct vertices $x, y \in V$. Question: is there a Hamiltonian path in G starting at x and ending at y? To simplify the presentation, we shall only consider the variant where both $x \longrightarrow^* y$ Hamiltonian paths and $y \longrightarrow^* x$ Hamiltonian paths are solutions. It is trivial to see that this problem is equivalent to the variant where only $x \longrightarrow^* y$ Hamiltonian paths are solutions (under exact planarity-preserving reduction).
- **DHAM-CYCLE.** Input: a directed graph $G(V,A)$. Question: is there a Hamiltonian circuit in G?

Furthermore, for all the Hamiltonian variants H above, we define **PLAN-**H as the restriction of H when the underlying graph G is required to be planar, and we similarly define $H(deg \leq k)$ when G is required to be undirected and have a bound k on the degree of all its vertices. If the question asks for the existence of a unique Hamiltonian solution, then **UNIQUE-**H denotes the associated problem (which belongs to the complexity class DP). In all proofs the following vocabulary will be used:

Definition 2.1 (plain and bold edges) *Let $G(V,E)$ a graph along with a simple path P and $e \in E$. If $e \in P$ then the edge e is said to be* bold. *Otherwise, it is said to be* plain. *The choice of these terms are simply motivated by the way we draw paths in all our figures.*

Definition 2.2 (gadget, pending edges) *A* gadget *$G'(V',E')$ is a graph whose purpose is to be a proper subgraph of another graph $G(V,E)$. A gadget has some selected edges called "pending edges" which are the ones used to connect G' and $G - G'$. One of the two extremities of a pending edge belongs to the gadget and the other does not. For this reason, in all the figures of this paper, the pending edges will be the ones with only one vertex explicitly drawn. If the gadget is planar, then it is an union of connected faces and the pending edges are required to be embedded on the outer-face of the gadget.*

Definition 2.3 (configuration) *Let $G(V, E)$ be a graph, $G'(V', E')$, a gadget in G, and P a simple path in G so that its ends are not in G'. The set of bold edges in G' is said to be a* configuration *in G'. Note that since G' is a proper subgraph of G, no bold cycle can occur in a configuration (otherwise P would not be connected). Also, the subpaths of P formed by the configuration in G' must all enter and leave G' at distinct pending edges.*

Definition 2.4 (Hamiltonian configuration) *For some gadget $G'(V', E')$, a configuration $H(V'', E'')$ is said to be a* Hamiltonian configuration *if $V' = V''$.*

The gadgets and their Hamiltonian configurations are tools to make proofs local and modular. Notice that, if G is a graph composed of several gadgets and H is a Hamiltonian cycle in G then all the gadgets of G must be in Hamiltonian configurations. This also holds if the ends x and y of all Hamiltonian paths H are known to be outside any gadget. This is the case when we have some control on x and y either because the problem – e.g. UHAM-PATH(x, y) – explicitly specifies them, or because the graph instance implicitly specifies them by setting their degree to 1. Whenever a problem – e.g. UHAM-PATH – lacks these two assumptions and we want to devise a reduction to it, we cannot rely on the correctness of our gadgets regarding their Hamiltonian configurations because an Hamiltonian path can start or end inside a gadget. Therefore, all the proofs in this paper will be based on gadgets and Hamiltonian configurations, except the proofs of reductions to Hamiltonian path variants with unspecified ends.

Definition 2.5 (parsimonious gadget) *A gadget G' is said to be* parsimonious *if it does not allow any two distinct Hamiltonian configurations H_1, H_2 with the same set of bold pending edges.*

Definition 2.6 (channels, wires, bits) *Let e_0 and e_1 be two chosen edges in a gadget. We design a model – inspired from electronic circuits – to represent information with this couple $W = (e_0, e_1)$. e_0 is called the* 0-channel *of the* wire W*, and e_1 is called its* 1-channel*. If exactly one of e_0 or e_1 is bold, then we say that the wire carries a* bit*, and if e_0 (resp e_1) is bold, then it is a* 0-bit *(resp. a* 1-bit*). If none of the edges are bold, the wire is said to be* mute*. If both edges are bold, then the wire is said to be* insane*. In this paper, all gadgets using wires will have the property to carry bits in any Hamiltonian configuration, that is, they will neither be mute nor insane. See Figure 3 for an intuition of the model. It will be used in all our "functional" gadgets, and in the reductions of Sections 8 and 9.*

Figure 3: Wires, channels, and representation of bits

Definition 2.7 (exact reduction) *An* exact reduction *from problem B to problem A is a parsimonious transformation from B to A that is computable in linear time on a deterministic RAM, and such that a solution of an instance of B can also be computed back in linear time from its image in the corresponding instance of A.*

All our exact reductions are based on (parsimonious) gadgets replacing locally the vertices and edges of the original instance graph. The direct access to memory provided by the RAM is essentially required to connect these gadgets together. A linear-sized transformation based on gadgets is generally computable in linear time on a RAM. Moreover, a parsimonious transformation $B \leq A$ that is based on gadget substitution provides by construction an injection from the objects in A to the ones of B, and thus getting back in linear time on a RAM a solution for B from its image in A is a trivial task. Therefore, as far as the correctness of exact reductions are concerned, we will prove their parsimony and their linear size, and we will omit the time issues.

3 The non-planar Hamiltonian variants

When the graphs are not required to be planar or degree-bounded, all Hamiltonian variants are easily reducible each other. In this section, we briefly review the set of the involved reductions, which is organized as in Figure 4a.

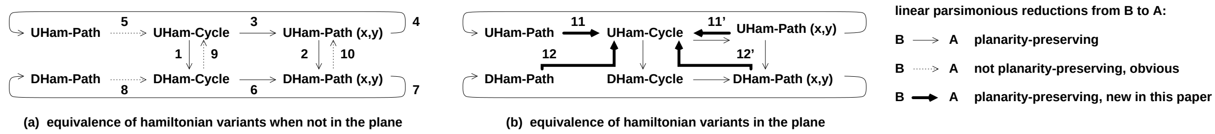


Figure 4: exact reductions between Hamiltonian variants

3.1 The easy reductions

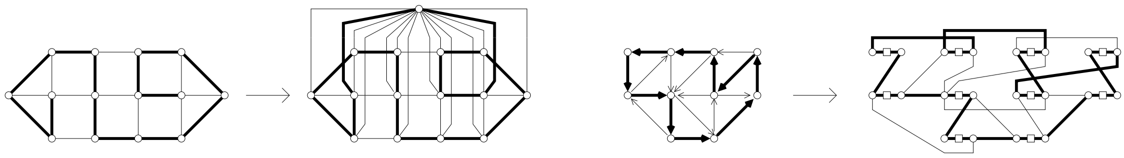
Reduction 3.1 (UHAM-CYCLE $\leq$ DHAM-CYCLE, ref. 1 in Fig. 4a) *This reduction is trivial if parsimony is not required. Let $G(V,E)$ be an undirected graph and k be the number of its Hamiltonian cycles. We build in linear time the graph $G'(V',A)$ as follows: Let $V'=V$ and for each $e=(i,j)\in E$, create two anti-parallel arcs (i,j) and (j,i) in A. The graph G' has clearly $2k$ Hamiltonian cycles since for every Hamiltonian cycle in G, we get the cycle represented twice in G' (one for each opposite direction). Yet, the duplicates are easily removed if G has a vertex of degree 2. Let v be this vertex and (u,v), $(v,w)\in E$ its two incident edges. In G', these edges generate the arcs $(u,v),(v,u),(v,w),(w,v)\in A$. We simply modify the reduction so that one direction is preferred on the other at vertex v by removing the arcs (v,u) and (w,v). We now get exactly k Hamiltonian cycles in G', and the (parsimonious) reduction is clearly planarity-preserving. However, it may happen that G has no vertex of degree 2. We show in Subsection 3.2 how to preprocess G so that such a vertex always appears in V without modifying the number of its Hamiltonian cycles.*

Reduction 3.2 (UHAM-PATH(x,y) $\leq$ DHAM-PATH(x,y), ref. 2 in Fig. 4a) *This reduction is basically the same as the previous one. Let $G(V,E)$ be an undirected graph, $x,y\in V$, $x\neq y$, and k be the number of its Hamiltonian paths from x to y. We build in linear time the graph $G'(V',A)$ as follows: Let $V'=V$ and for each $e=(i,j)\in E$, create two anti-parallel arcs (i,j) and (j,i) in A. For the same reason as before, the graph G' has $2k$ Hamiltonian paths of the form $x\longrightarrow^{*} y$ or $y\longrightarrow^{*} x$. However the duplicates paths are easier to remove: simply remove from A all arcs of the form (u,x) and (y,v). Since no arc enters x and no arc leaves y anymore, we now have only k Hamiltonian paths in G', all of the form $x\longrightarrow^{*} y$. The reduction is clearly planarity-preserving.*

Reduction 3.3 (UHAM-CYCLE $\leq$ UHAM-PATH(x,y), ref. 3 in Fig. 4a) *Let $G(V,E)$ be an undirected graph, and k be the number of its Hamiltonian cycles. The task is to manage to break the Hamiltonian cycles at some point. This is trivial if we know that the graph has a vertex v of degree 2, because every Hamiltonian cycle must use its two incident edges (u,v) and (v,w). We just have to remove v, (u,v) and (v,w) and rename u, w as x, y respectively. Clearly, the new graph has exactly k Hamiltonian paths with extremities x and y. However, it may happen that G does not have a vertex of degree 2. In this case, we preprocess G so that such a vertex appears without modifying the number of its Hamiltonian cycles, using again the trick shown in Subsection 3.2. The reduction is planarity-preserving.*

Reduction 3.4 (UHAM-PATH(x,y) $\leq$ UHAM-PATH, ref. 4 in Fig. 4a) *Let $G(V,E)$ be an undirected graph, $x,y\in V$, and k be the number of Hamiltonian paths from x to y. We just remove the Hamiltonian paths which do not end at x and y, by adding two new vertices x',y' and two new edges (x',x), (y',y). Since the degrees of x' and y' are 1, there are now k Hamiltonian paths, all of the form $(x',x,\cdots,y,y')$. The reduction is planarity-preserving.*

Reduction 3.5 (UHAM-PATH $\leq$ UHAM-CYCLE, ref. 5 in Fig. 4a) *Let $G(V,E)$ be an undirected graph, and k be the number of its Hamiltonian paths. By adding a new vertex v and a new edge (v,w) for* *every $w \in E$, any Hamiltonian cycle must use two edges (v, w_1) and (v, w_2) for some distinct $w_1, w_2 \in V$. The rest of the cycle defines a Hamiltonian path from w_1 to w_2. Thus, there are k Hamiltonian cycles in the new graph. Notice that this reduction* **is not planarity-preserving** *since all vertices of V do not necessarily lie on the boundary of the same face (see Figure 5a). Finding a more clever exact reduction that is still working in the plane is one of the main goals of this paper (ref. 11 on fig 4b).*



(a) from UHam-Path to UHam-Cycle (b) from DHam-Cycle to UHam-Cycle

Figure 5: The two main non planarity-preserving reductions

Reduction 3.6 (DHAM-CYCLE $\leq$ DHAM-PATH(x,y), ref. 6 in Fig. 4a) *The idea is the same as for the undirected version of this reduction (ref. 3 on fig. 4a), i.e. we want to break every Hamiltonian cycle into a path at some fixed point. We first use the trick presented in Subsection 3.2 to preprocess the original graph $G(V, A)$ so that a vertex v of degree 4 appears in it, with only incident arcs (x, v), (v, x), (v, y), (y, v) for some other vertices x, y, and so that the number of Hamiltonian cycles is not modified. Thus, in the modified graph $G'(V', A')$, all the Hamiltonian cycle must be of the form $v \longrightarrow y \longrightarrow^* x \longrightarrow v$ or $v \longrightarrow y \longrightarrow^* x \longrightarrow v$, each cycle determining a Hamiltonian path in $V' - \{v\}$ from x to y or from y to x. Therefore, by removing the vertex v and the arcs (x, v), (v, x), (v, y), (y, v), the (planarity-preserving) exact reduction is completed.*

Reduction 3.7 (DHAM-PATH(x,y) $\leq$ DHAM-PATH, ref. 7 in Fig. 4a) *Simply remove the Hamiltonian paths in $G(V, A)$ not ending at both x and y by renaming x,y into x', y' respectively, and by adding the new vertices x, y and the new arcs (x', x), (x, x'), (y', y), (y, y'). Since he new vertices x and y are of in-degree 1 and of out-degree 1, the only possible Hamiltonian paths end at x and y, and there is a one-to-one correspondence between the solutions of the original graph and the ones of the modified graph. The reduction is planarity-preserving.*

Reduction 3.8 (DHAM-PATH $\leq$ DHAM-CYCLE, ref. 8 in Fig. 4a) *This reduction is the basically same as its undirected variant (ref. 5 on fig. 4a). Let $G(V, A)$ the original graph. Create $G'(V', A')$ as follows: add a new vertex p in G, and for each vertex w in G two anti-parallel arcs (v, w) and (w, v). There is clearly a one-to-one correspondence between the Hamiltonian paths in G of the form $x \longrightarrow^* y$ for some $x, y \in V$ and the Hamiltonian cycles $p \longrightarrow x \longrightarrow^* y \longrightarrow p$ in G' so the reduction is parsimonious. For the same reason as for its undirected variant, this reduction is not planarity-preserving. In this paper, we won't replace it directly by a planarity-preserving one, but we will reduce DHAM-PATH to UHAM-CYCLE instead (ref. 12 on fig 4b).*

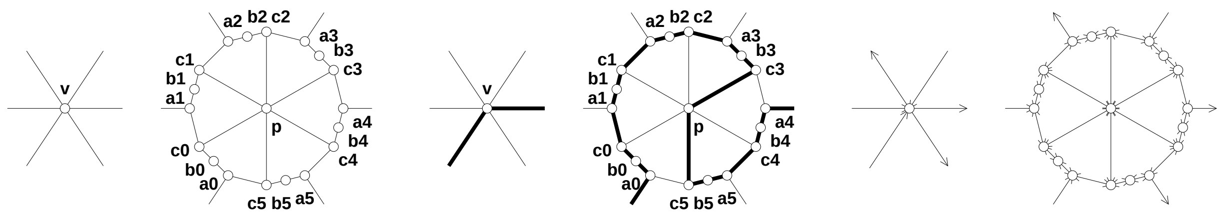
Reduction 3.9 (DHAM-CYCLE $\leq$ UHAM-CYCLE, ref. 9 in Fig. 4a) *The problem is to find how to simulate a vertex with in-going and out-going arcs by a vertex with undirected edges. Let $G(V, A)$ be the original directed graph. We compute $G'(V', E)$ in linear time as follows: for each vertex $v \in V$, create the vertices $i(v)$, $c(v)$, $o(v)$ and the two edges $(i(v), c(v))$,$(c(v), o(v))$ and for each arc $(u, w) \in A$, create an edge $(o(u)), i(w))$. Intuitively the control-vertex $c(v) \in V'$ of degree 2 represents $v \in V$, and since any Hamiltonian cycle in G' must use both $(i(v), c(v))$ and $(c(v), o(v))$, exactly one edge $(o(u), i(v))$ and one edge $(o(v), i(w))$ must also be used for some $u, v \in V$. These two edges represent respectively the in-going arc $(u, v) \in A$ and the out-going arc $(u, w) \in A$ and thus the vertex v has been simulated parsimoniously. However, this reduction is not planarity-preserving. Indeed, v may have its in-going and out-going arcs interleaved in the clockwise order around v. Yet, this is not the case for the corresponding edges in G' around the segment $(i(v), c(v), o(v))$ since the edges associated to the in-going arcs are grouped around $i(v)$, and the edges associated to the out-going arcs are grouped around $o(v)$. See Figure 5b for an intuition.*

As one can see on Figure 4a, the directed graph of the mutual reductions between all variants of the HAM problem is strongly connected when the planarity is not considered. Yet, if non planarity-preserving reductions (noted by dashed arcs) are removed, the graph is not strongly connected anymore. Figure 4b shows how we shall make this graph strongly connected again by designing new planarity-preserving exact reductions in the rest of this paper.

3.2 How to create a vertex of degree 2?

In this subsection, we present the mechanism needed by some of the previous reductions, that is, how to transform an undirected graph $G(V, E)$ with no vertex of degree 2, so that it contains such vertices without modifying the number of its Hamiltonian cycles. This is performed with the following gadget, namely *the BIKE-WHEEL*.

Gadget 3.10 (BIKE-WHEEL gadget) *Let v a vertex of degree d. The associated* BIKE-WHEEL gadget *of degree d is the planar Hamiltonian gadget shown in Figure 6a. The pending edges of the gadget are embedded the same way as the incident edges of v. The gadget is designed so that, in any Hamiltonian configuration, it simulates parsimoniously the vertex v being traversed by a path using exactly two of its incident edges. Its purpose is to provide artificially vertices of degree 2.*



(a) a vertex v of degree 6 and the associated bikewheel

(b) a path throw v along with corresponding hamiltonian configuration

(c) directed version of the bikewheel

Figure 6: A BIKE-WHEEL gadget of degree 6 along with a typical Hamiltonian configuration

Property 3.11 *A BIKE-WHEEL gadget of degree d allows only single paths as Hamiltonian configuration. Each one is free to start and end at any pair of pending edges, but is completely determined by the choice of these two edges. See Figure 6b.*

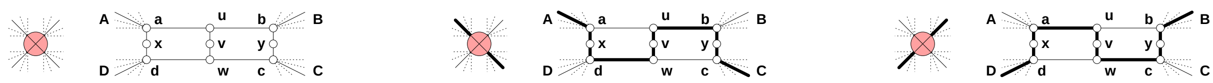
Proof: Let H be a Hamiltonian configuration in the BIKE-WHEEL. Since the central vertex p must be captured by H, exactly two edges incident to p must be bold. Also, because of the control-vertices b_i, $(0 \le i < d)$, all the edges (a_i, b_i) and (c_i, d_i) must be bold. Let I and J be the unique distinct indices so that (p, c_I) and (p, c_J) are bold. Because H cannot have vertices of degree > 2, the two edges (c_I, a_{I+1}) and (c_J, a_{J+1}) must be plain (with additions over indices performed modulo d), and thus the pending edges at a_{I+1} and a_{J+1} must be used. Since all edges (p, c_i) are plain for $i \neq I, J$, it follows that all edges (c_i, a_{i+1}) are bold. All vertices are now captured and the built path is uniquely defined by the two edges (p, c_I) and (p, c_J) we chose to be bold, or equivalently by the two pending edges at a_{I+1} and a_{J+1}. ■

Since a BIKE-WHEEL of degree $d > 2$ has d vertices b_i of degree 2, replacing any vertex v of degree d by this BIKE-WHEEL will make appear as many vertices of degree 2, while preserving both the planarity and the number of Hamiltonian cycles. For obvious reasons, the natural directed version of the BIKE-WHEEL (see Figure 6c) will generate at least d vertices b_i of degree 4 such that their incident arcs are of the form (a_i, b_i), (b_i, a_i), (c_i, b_i), (b_i, c_i) for some other vertices a_i, c_i.

4 The fundamental planar gadgets

In this section, we create several Hamiltonian gadgets useful for the two reductions of Sections 8 and 9. These gadgets are at the base of the more sophisticated logical and arithmetical gadgets of Sections 6 and 7. Each gadget are given under two forms in our figures: its symbolic notation and its implementation schema (e.g see Figures 7 and 9). Whenever a high level gadget uses a subgadget, the symbolic notation of the latter is used in the implementation schema of the former (e.g. see Figures 9, 10 and 16).

Gadget 4.1 (RIGIDITY gadget) *The* RIGIDITY gadget *is the Hamiltonian planar gadget shown in Figure 7a. It has four fans A, B, C and D of entry edges, entering the vertices a, b, c and d, respectively. The fan A is opposite the fan C, and the fan B is opposite the fan D. As it will be proved, this gadget allows only two types of Hamiltonian configurations. The first one consists of a path with ends on opposite fans A and C, and the second one consists of a path with ends on opposite fans B and D. Thus, the gadget may be seen as a planar crossroad where a Hamiltonian path is forced to go straight with no possibility to turn.*



(a) the rigidity gadget and its implementation (b) north-west to south-east configuration (c) north-east to south-west configuration

Figure 7: The RIGIDITY gadget along with its two Hamiltonian configurations

Property 4.2 *The RIGIDITY gadget allows only the two types of Hamiltonian configurations shown in Figure 7(b,c), i.e. any Hamiltonian path must exit the gadget via a fan lying the opposite side the fan it enters.*

Proof: Let H be an hamiltonian configuration in the BIKE-WHEEL. Note that because of control vertices v, x, and y, the edges (u, v), (v, w), (a, x), (x, d), (b, y) and (y, c) must all be bold. There are now two cases according to the status of the edge (a, u):

1. Suppose that (a, u) is plain (see Figure 7b). Then (u, b) and exactly one edge of fan A must be bold. Also none of the edges of fan B may be bold. (w, c) cannot be bold because this would close a cycle. This implies that (d, w) must be bold, that none of the edges of fan D may be bold, and that exactly one edge of fan C must be bold. All vertices are now captured and the configuration H is uniquely defined.
2. Suppose that (a, u) is bold (see Figure 7c). Then none of the edges of fans A may be bold. Also (d, w) cannot be bold because this would create a cycle. This implies that (w, c) is bold and none of the edges of fans C may be bold. All vertices are now captured and the path created must leave the gadget by exactly one edge of fan B and one edge of fan D. The configuration H is uniquely defined.

Gadget 4.3 (NOT gadget) *The* NOT gadget *is the Hamiltonian gadget equal to the RIGIDITY gadget where the fans A, B, C, D all degenerate to a single edge, respectively e_A, e_B, e_C, e_D. The couples (e_A, e_B) and (e_C, e_D) form the entry-wire and the exit-wire. Let H be a Hamiltonian configuration in the NOT gadget. It is easy to check on Figure 7 that the exit-wire (e_C, e_D) will carry a bit whose value is the negation of the bit carried in (e_A, e_B). Thus the NOT gadget may be seen as a planar mechanism to invert the value of the bit circulating in a wire (see Figure 8(a,b,c)).*

Gadget 4.4 (XOR-EDGE gadget) *The* XOR-EDGE *is a planar Hamiltonian gadget found in [JP85] (see also [Pap94]). It is symbolized by a dashed line with a $\oplus$-symbol binding two edges e_1 and e_2 of the graph. Its purpose is to enforce parsimoniously that, in any Hamiltonian configuration, exactly one edge among e_1 and e_2 is bold. This gadget is implemented by the ladder with four bars shown in Figure 9a. This seminal gadget will be widely used in this paper.*

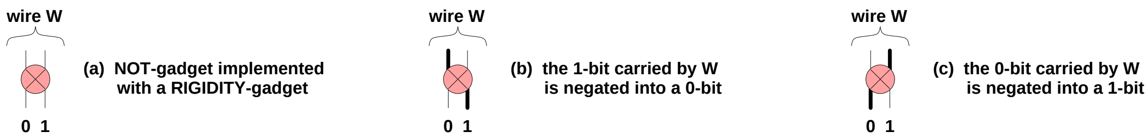


Figure 8: The NOT gadget implemented with a RIGIDITY gadget

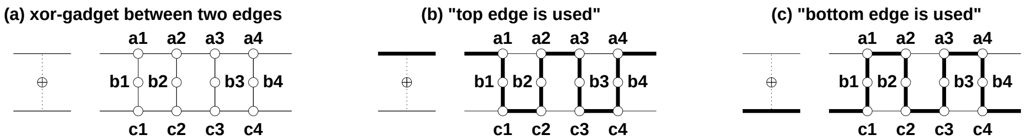


Figure 9: The XOR-EDGE gadget [JP85]

Property 4.5 *The XOR-EDGE allows exactly two Hamiltonian configurations, shown in Figure 9(b,c).*

Proof: Let H be a Hamiltonian configuration in the XOR-EDGE. Then, because of the control-vertices b_i, $(1 \leq i \leq 4)$, all the edges (a_i, b_i) and (b_i, c_i) must be bold. Now there are two cases according to the status of the edge (a_2, a_3):

1. Suppose that the edge (a_2, a_3) is bold (see Figure 9b). Then (c_2, c_3) must be plain because otherwise H would contain a cycle and could not be a configuration. Also a configuration cannot have paths ending at non-pending edges, thus (c_1, c_2) and (c_3, c_4) must be bold. It follows that the pending edges at a_1 and a_4 are used, and H is uniquely defined.

2. Suppose that the edge (a_2, a_3) is plain. Then by symmetry of the gadget, we obtain the same configuration, but vertically mirrored, as shown on Figure 9c. ■

Gadget 4.6 (AND gadget) *The* AND gadget *is the Hamiltonian gadget shown in Figure 10. The right side of Figure shows the implentation of the gadget, and the left side shows how we will symbolize it. It has two input-wires A, B called the* operands, *one output-wire $A \wedge B$ called the* result *and an edge called the* feed. *As we shall prove, it is designed so that, in any Hamiltonian configuration, A and B are enforced to carry bits and $A \wedge B$ carries the bit whose value is the "logical and" of the value carried by A and the value carried by B. As for all gadgets that will follow, there will be exactly one feed per bit of result, and the Hamiltonian configurations will be made of one path per operand wire, plus one path per result wire, each of them starting at its associated feed.*

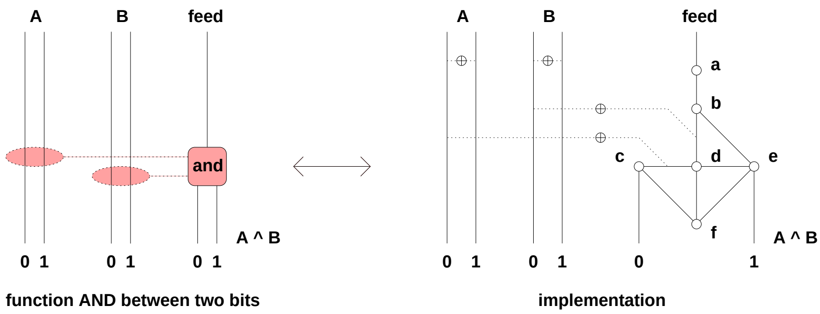


Figure 10: The AND-gadget

Property 4.7 *The AND gadget allows only the four Hamiltonian configurations shown in Figure 11, i.e. wires A and B necessarily carry bits, and wire $A \wedge B$ carries the bit with value the "logical and" between bits A and B.*

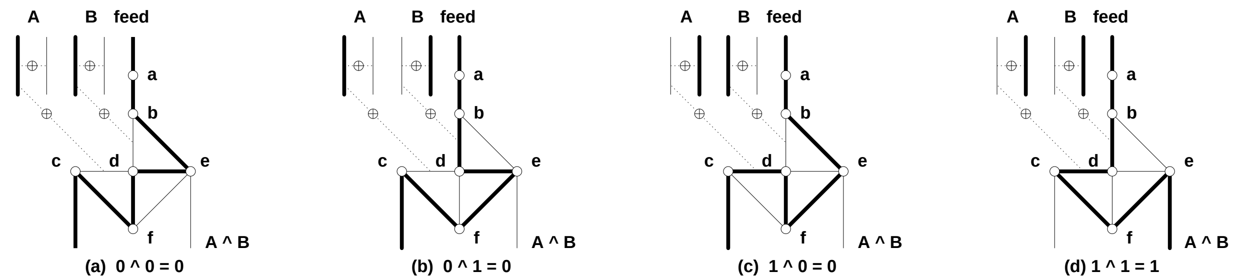


Figure 11: The four Hamiltonian configurations in the AND gadget

Proof: Let H be a Hamiltonian configuration in the AND gadget. The XOR-EDGE gadget between the channels of wire A implies that exactly one of its edges is bold. Thus A carries a bit. The same argument holds for wire B. Note that there are only 3 access-points to the gadget (namely the feed, and the two edges of the result wire). So this leaves only the place for one path to go through the gadget. Because of the control vertex a, the feed (a, b) must be bold. This implies that exactly one of the edges of the result wire is bold, in other words, $A \wedge B$ must carry a bit. The proof falls now into the study of the 4 possible combinations of bits that operand wires may carry:

1. Let H_1 be a Hamiltonian configuration where both A and B carry a 0-bit (see Figure 11a). Then the two XOR-EDGE gadgets imply that (c, d) and (b, d) are plain. This leaves only (b, e) to continue the path starting from the feed, and the edges (d, e) and (d, f) to capture vertex d. Also, this only leaves (c, f) and the 0-channel of wire $A \wedge B$ to capture c. and the edge (c, f) to capture vertex c. So all these edges must be bold. All vertices are now captured, and H_1 is uniquely defined by a path starting from the feed and ending in the 0-channel of $A \wedge B$.

2. Let H_2 be a Hamiltonian configuration where A carries a 0-bit and B carries a 1-bit. (see Figure 11b). Because of the two XOR-EDGE gadgets, (c, d) must be plain and (b, d) must be bold. This leaves only (c, f) and the 0-channel of wire $A \wedge B$ to capture vertex c, so these edges must be bold. Now, vertex e has still to be captured, and the only opportunity to do it is to make (d, e) and (e, f) bold. Thus H_2 is uniquely defined by a path starting from the feed and ending in the 0-channel of $A \wedge B$.

3. Let H_3 be a Hamiltonian configuration where A carries a 1-bit and B carries a 0-bit. (see Figure 11c). Because of the two XOR-EDGE gadgets, (c, d) must be bold and (b, d) must be plain. This leaves only (b, e) for the path starting from the feed to continue, so (b, e) must be bold. Exiting now via the 1-channel of $A \wedge B$ is impossible because only one path may go through the gadget, and we have still to go through (c, d). So the path will have to exit via the 0-channel of $A \wedge B$. This leaves now only (c, f) and (e, f) to capture vertex f, thus these edges must be bold. All vertices are now captured and H_3 is uniquely defined by a path starting from the feed and ending in the 0-channel of $A \wedge B$.

4. Let H_4 be a Hamiltonian configuration where both A and B carry a 1-bit (see Figure 11d). Then the two XOR-EDGE gadgets imply that (c, d) and (b, e) are plain. The path cannot exit now via the 0-channel of $A \wedge B$ because only one path may go through the gadget, and vertices e and f have still to be captured. The unique way to continue the path is now to make (c, f) and (e, f) bold and to exit via the 1-channel of wire $A \wedge B$. ■

Remark 4.8 *Note that the AND gadget is almost planar with the embedding chosen in Figure 10. The only non planar features are the crossings between several normal edges and some XOR-EDGE gadgets. This is not a problem since we shall be able to design a parsimonious crossover-box for this particular case. The details about this crossover are shown in Section 5. However, if one assumes the existence of this crossover, one may skip Section 5 and go directly to Section 6 where the other logical gadgets are developed.*

Remark 4.9 *In all this paper, we arbitrarily choose to systematically embed all the operators on the right side of their operands, and the result-wires beneath the operator, so that the gadgets always have their crossover-boxes at the same place.*

5 The crossover-box between a XOR-EDGE and several edges

In this section, we present the details of the crossover-box that we delayed in the previous sections. This crossover-box is needed to makes things planar everywhere the dashed line of a XOR-EDGE gadget crosses one or several normal edges (i.e. solid lines), as in Figures 12(a,b). In [JP85], a crossover-box which deals with the mutual crossing of two XOR-EDGE gadgets is presented, but it does not fit our needs because the dashed lines of our XOR-EDGES cross solid lines, and not simply the dashed lines of some other XOR-EDGES. Yet, we shall use a similar technique which echoes the wave circulating in the ladder of the XOR-EDGE gadget each time a crossed normal edge is bold (see Figure 12c for an intuition). For the presentation and the easiness of proofs, our crossover is built by concatenating several instances of another sub-gadget: the FLIP-FLAP.

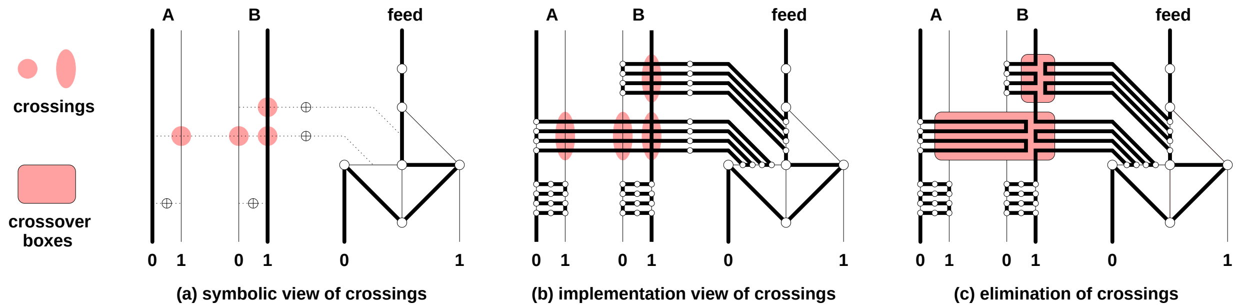


Figure 12: The crossings between XOR-EDGES and normal edges in the AND gadget

Gadget 5.1 *(FLIP-FLAP gadget) The* FLIP-FLAP *is the planar gadget shown in Figure 13a. The four segments* $(a_i, \cdots, d_i)$ $(1 \leq i \leq 4)$ *play the same rôle as the four bars in the ladder of the XOR-EDGE, namely the segments* $(a_i, \cdots, c_i)$ *in Figure 9. The pending edges at vertices* s *and* t *represent the edge that the ladder has to cross over.*

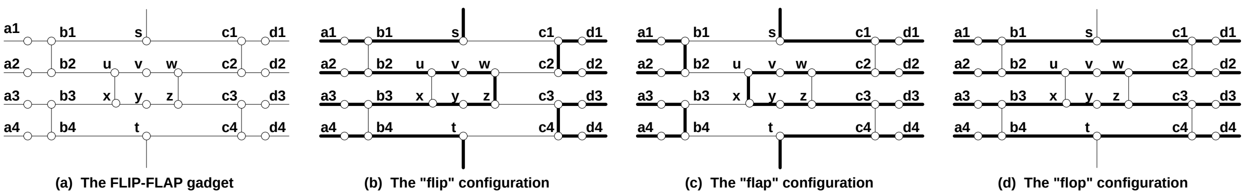


Figure 13: The FLIP-FLAP gadget, along with its three Hamiltonian configurations

Property 5.2 *The FLIP-FLAP allows only three Hamiltonian configurations, shown in Figure 13(b,c,d), namely the "flip"-configuration, the "flap"-configuration and the "flop"-configuration.*

Proof: Let H be a Hamiltonian configuration. Note that, because of control vertices a_i and d_i, $(1 \leq i \leq 4)$, all the edges (a_i, b_i), (c_i, d_i) and the corresponding pending edges must be bold. Also, because of control vertices v and y, the edges (u, v), (v, w), (x, y) and (y, z) must be bold. There are now three cases:

1. Suppose we make the pending edge at t bold and that we continue by making (b_4, t) bold (see Figure 13b). Then the propagation on the left is as follows: (b_3, b_4) is plain, (b_3, x) is bold, (u, x) is plain, (b_2, u) is bold, (b_1, b_2) is plain, (b_1, s) is bold. The propagation on the right is as follows: (t, c_4) is plain, (c_3, c_4) is bold, (z, c_3) is plain, (w, z) is bold, (w, c_2) is plain, (c_1, c_2) is bold. Finally, (s, c_1) is plain and the pending edge at s must be bold. This Hamiltonian configuration is called the *"flip"-configuration.*
2. Suppose now we make the pending edge at t bold and that we continue by making (t, c_4) bold. By symmetry of the gadget, we will obtain the previous configuration, but horizontally mirrored, as shown in Figure 13c. This Hamiltonian configuration is called the *"flap"-configuration.*

3. Finally, suppose that the pending edge at t is plain (see Figure 13d). Then both (b_4, t) and (t, c_4) must be bold, and this propagates as follows: $(b_3, , b_4)$ and (c_3, c_4) are plain, (b_3, x) and (z, c_3) are bold, (u, x) and (w, z) are plain, (b_2, u) and (w, c_2) are bold, (b_1, b_2) and (c_1, c_2) are plain. It turns out that (b_1, s) and (s, c_1) are both bold and the pending edge at s cannot be used. This Hamiltonian configuration is called the *"flop"-configuration.* ■

Gadget 5.3 (The crossover-box between XOR-EDGE and normal edges) *The crossover is designed as follows: Let e_1 and e_2 be the two edges to bind with a XOR relation on their boldness. Set four vertices $l_1, \cdots, l_4$ (resp. $r_1, \cdots, r_4$) onto the edge e_1 (resp. e_2). For each edge f_i $(1 \leq i \leq k)$ that the dashed line of XOR relation must cross over, create a FLIP-FLAP gadget on it and chain them sequentially as shown in Figure 14a by tying theit pending edge.*

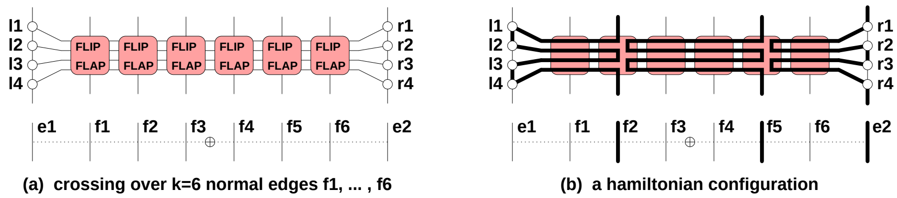


Figure 14: The crossover between a XOR-EDGE and several normal edges

Property 5.4 *In any Hamiltonian configuration H, for a XOR-EDGE between edges e_1 and e_2 with k FLIP-FLAPS over edges f_i, $(1 \leq i \leq k, k \geq 1)$, exactly one edge among e_1 and e_2 is used, while the edges f_i embedded between them may be bold or plain, independently. Moreover, H is uniquely defined by the boldness of e_1 (or equivalently e_2) and the boldness of all the f_i.*

Proof: We interpret the chain of k FLIP-FLAPS as a tape of length k where words may be written with the alphabet $\Sigma = \{flip, flap, flop\}$, and whose letters are the flip-configuration, flap-configuration and flop-configuration, respectively (see Figure 15a).

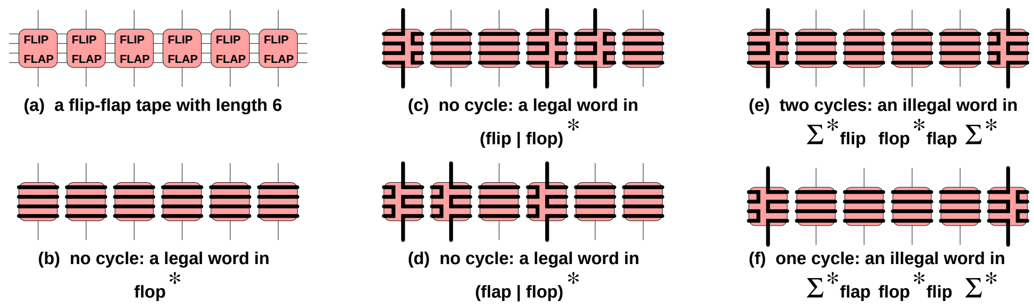


Figure 15: Hamiltonian configurations seen as words over the alphabet $\Sigma = \{flip, flap, flop\}$

We define the language $\mathcal{H} \subset \Sigma^k$ as the subset of words $h \in \Sigma^k$ that are Hamiltonian configurations. Recall that two conditions must be met for a set of paths to be a configuration: no cycle may appear, and all paths must end at some pending edge. The last condition is clearly met for all $w \in \Sigma^k$ because the FLIP-FLAPS have control-vertices on all pending edges used for their chaining. The "no cycle condition" however removes $\Sigma^* flip\ flop^* flap\ \Sigma^*$ from $\mathcal{H}$ because two cycles would arise, respectively on the two upper bars and two lower bars of the ladder of such words (see Figure 15e). Symmetrically $\Sigma^* flap\ flop^* flip\ \Sigma^*$ is not part of $\mathcal{H}$ because one cycle would arise on the two central bars of the ladder of such words (see Figure 15f). This proves that *flip* and *flap* may not appear thogether in a same word $h \in \mathcal{H}$. At this point, this leaves $(flip \mid flop)^k \bigcup (flap \mid flop)^k$ in the language. Clearly, the special case $flop^k$ belongs to $\mathcal{H}$ because it is formed by four disjoint horizontal paths (see Figure 15b). Moreover, we easily see on

Figures 15(c,d) that no cycle may appear in words $h_1 \in (flip \mid flop)^k$ and $h_2 \in (flap \mid flop)^k$. We conclude that $\mathcal{H} = (flip \mid flop)^k \bigcup (flap \mid flop)^k$, i.e. the edges f_i may be bold or plain, independently. We now prove that the gadget preserves the XOR relation between the use of edges e_1 and e_2; there are three cases:

1. $h = flop^k$ (see Figure 15b). Then none of the crossed edges f_i are used and since the configuration in the k FLIP-FLAPS is formed by four straight horizontal lines, The XOR relation between the boldness of edges e_1 and e_2 follows with the same two-cases proof as the simple XOR-EDGE gadget; the bars of the ladder are just longer.

2. $h \in (flip \mid flop)^k$, with at least one $flip$ letter (see Figures 15c and 14b). Then an edge f_i is used iff the i^{th} FLIP-FLAP is set to the "flip"-configuration. On the left side, the edge (l_2, l_3) iis necessarily plain, because otherwise a cycle would occur on the two central bars of the ladder, due to the leftmost $flip$ letter. It follows that edges (l_1, l_2) and (l_3, l_4) are bold, and thus edge e_1 is "used". On the right side, edges (r_1, r_2) and (r_3, r_4) are necessarily plain, because otherwise two cycles would occur, respectively on the two lower bars and two upper bars of the ladder, due to the rightmost $flip$ letter. It follows that (r_2, r_3) is bold as well as the pending edges at r_1 and r_4. Therefore, the edge e_2 is "used" and the XOR relation between the boldness of e_1 and e_2 is verified.

3. $h \in (flap \mid flop)^k$, with at least one $flap$ letter (see Figure 15d. This is the symmetrical case of 2, and we omit the detailed proof. The edge e_2 comes up to be used while e_1 cannot, thus XOR relation between the boldness of e_1 and e_2 is again verified.

We merely proved that either e_1 or e_2 has to be used but not both, and that the crossed edges f_i embedded between them may be bold or plain independently. Moreover, for each such choice, there is exactly one Hamiltonian configuration. ■

6 The derived logical gadgets

An immediate corollary of the previous crossover-box is the planarity of the AND gadget. Since the crossover is itself parsimonious, the parsimony of the AND gadget is also preserved. Now, recall that the AND and NOT operators form a complete logic, i.e. they can be combinated to generate all the boolean operators (such as OR, IMP, XOR, NAND, ...). We shall derive, in an electronic hardwire fashion, a long series of gadgets from the AND gadget and the NOT gadget. Two gadgets will be particularly useful as a first step: the OR gadget and the XOR gadget.

Gadget 6.1 (OR gadget) *The* OR gadget *is the planar Hamiltonian gadget symbolized and implemented in Figure 16. As for the AND gadget, it has two operand-wires A and B, one result-wire and one feed. It is designed to have a behaviour similar to the AND gadget, but along with the "logical OR" operator.*

Property 6.2 *For any Hamiltonian configuration in the OR gadget, A and B are enforced to carry bits, and a uniquely defined path goes through the gadget, starting from the feed and ending in the channel corresponding to the value of the "logical OR" of the bits carried by the operands.*

Proof: The operands are enforced to carry bits because they are the operands of the NOT and AND gadgets which have such a property. The correctness is a straightforward application of the "De Morgan law", namely: $A \vee B = \neg(\neg A \wedge \neg B)$. The path from the feed to the result-channel is uniquely defined because both the NOT gadget and the AND gadget are parsimonious. Note that two additional NOT gadgets lie at the bottom of the operand-wires to restore the initial values carried by A and B after the first inversion, so that potential further use of these operands remains possible. ■

Gadget 6.3 (XOR gadget) *The* XOR gadget *is the planar Hamiltonian gadget symbolized and implemented in Figure 16.*

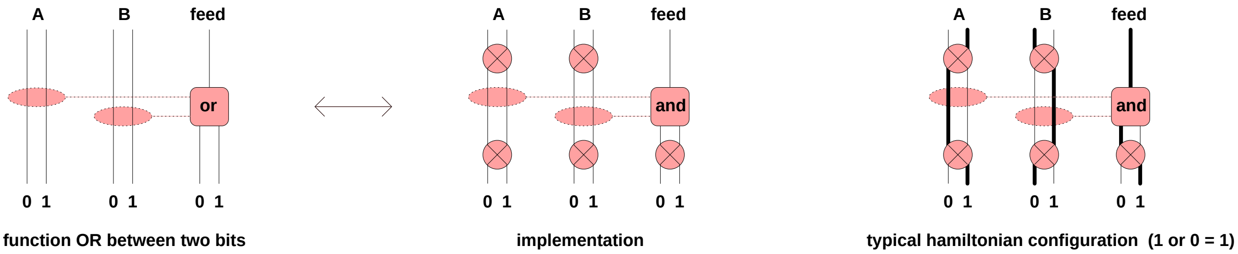


Figure 16: The OR gadget

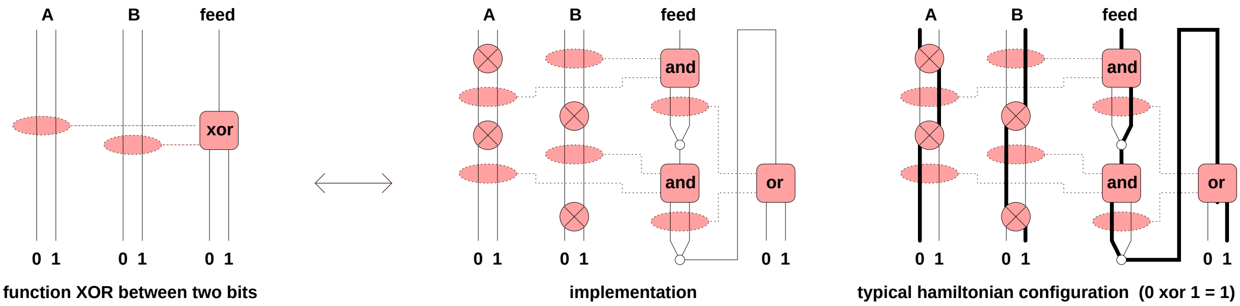


Figure 17: The XOR gadget

Property 6.4 *For any Hamiltonian configuration, the XOR gadget enforces its operand-wires A and B to carry bits, and the configuration is uniquely defined by a path starting from the feed and ending in the channel of the result-wire corresponding to the value of the "logical XOR" of the bits carried by A and B.*

Proof: The operands A and B are enforced to carry bits because they are also operands of some AND gadgets. Now, the correctness follows directly from the equality $A \oplus B = (A \wedge \neg B) \vee (\neg A \wedge B)$. The parsimony of the gadget is inherited from the parsimony of the AND gadget, the OR gadget and the NOT gadget. Note that the operand-wires are negated twice to keep their values unchanged at the end. ■

At this stage, we have a powerful Hamiltonian scheme to build reductions from logical problems to the HAM problem that are simultaneously linear, parsimonious and planarity-preserving. In Section 11, we shall easily devise such a reduction from the PLAN-SAT problem to the PLAN-UHAM-CYCLE problem. The straighforward corollary of the parsimony of this reduction will be that the UNIQUE-PLAN-HAM problem is DP-complete under randomized reduction.

7 The derived arithmetical gadgets

We go further in abstraction by creating arithmetical gadgets, that is, gadgets that are able to perform additions between integers represented in binary. Recall that if $(a_{n-1}a_{n-2}\cdots a_1a_0)$ and $(b_{n-1}b_{n-2}\cdots b_1b_0)$ are the binary representation on n bits of the unsigned integers A and B respectively, then the binary representations $(s_{n-1}s_{n-2}\cdots s_1s_0)$ of the sum $S = A + B$ (truncated on n bits) is computed with the propagation of the carries $c_0, c_1, \cdots, c_{n-2}, c_{n-1}$ from right to left as follows:

$$\left\{\begin{array}{lcl} c_0 & = & 0 \\ s_i & = & odd(a_i, b_i, c_i) \\ c_{i+1} & = & maj(a_i, b_i, c_i) \end{array}\right.$$

where $odd(a, b, c)$ is the *odd* function $\{0,1\}^3 \rightarrow \{0,1\}$ returning 1 iff an odd number of its arguments are set to 1, and where $maj(a, b, c)$ is the *majority* function $\{0,1\}^3 \rightarrow \{0,1\}$ returning 1 iff a majority of its arguments are set to 1.

Gadget 7.1 (ODD gadget) *The* ODD gadget *is the planar Hamiltonian gadget shown in Figure 18. It has three operand-wires A, B and C, one feed and one result-wire.*

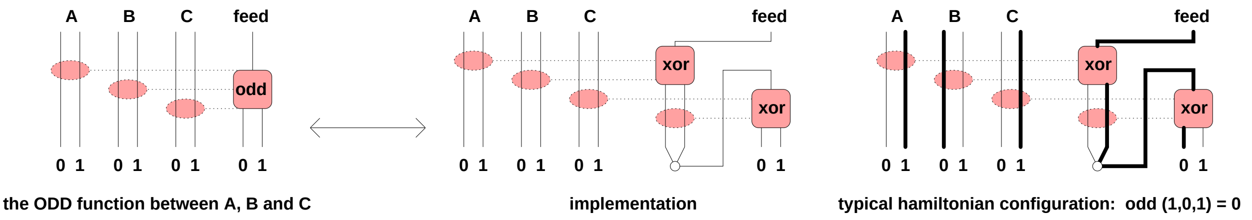


Figure 18: The ODD gadget

Property 7.2 *In any Hamiltonian configuration, the ODD gadget enforces its three operands A, B and C to carry bits, and a path is uniquely defined, starting from the feed and ending in the channel of the result-wire corresponding to the value $odd(a, b, c)$, where a, b, and c are the values of the bits carried by A, B and C, respectively.*

Proof: The operand-wires are enforced to carry bits because they are caught in some XOR gadgets which have this property. Now, the correctness is an immediate consequence of the equality $odd(a, b, c) = (a \oplus b) \oplus c$. The parsimony of the gadget is inherited from the parsimony of the XOR gadget. ■

Gadget 7.3 (MAJ gadget) *The* MAJ gadget *is the planar Hamiltonian gadget shown in Figure 19. It has three operand-wires A, B and C one feed and one result-wire.*

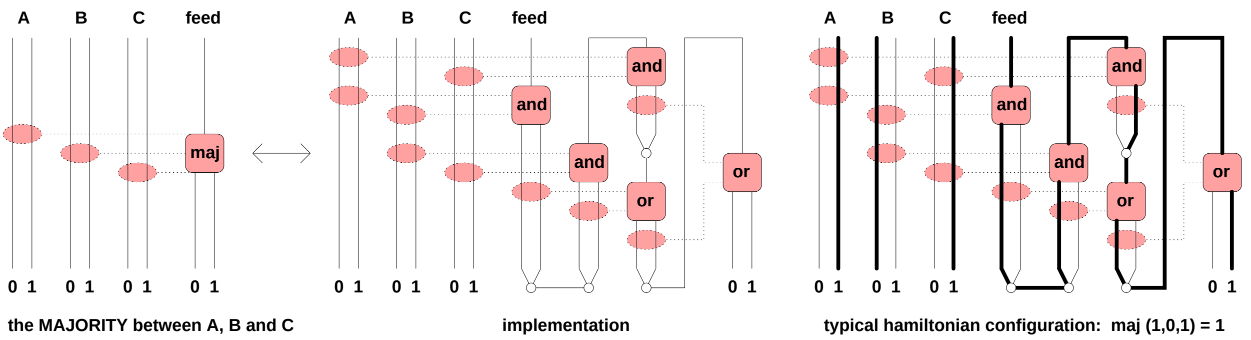


Figure 19: The MAJ gadget

Property 7.4 *In any Hamiltonian configuration, the MAJ gadget enforces its three operands A, B and C to carry bits, and a path is uniquely defined, starting from the feed and ending in the channel of the result-wire corresponding to the value $maj(a, b, c)$ where a, b, and c are the values of the bits carried by A, B and C, respectively.*

Proof: The operand-wires are enforced to carry bits because they are caught in some AND gadgets which have this property. Now, the correctness is an immediate consequence of the equality $maj(a, b, c) = (a \wedge b) \vee (b \wedge c) \vee (c \wedge a)$. The parsimony of the gadget is inherited from the parsimony of the AND gadget and the OR gadget. ■

Gadget 7.5 (ADD gadget) *The* ADD gadget *is the planar Hamiltonian gadget shown in Figure 20. It has two operands A and B, each of them being formed by n wires. It also has n feeds and n result-wires. It is designed so that in any Hamiltonian configuration, the operand-wires and the result-wires are forced to carry bits, so that the bits carried in the result-wires are the binary representation of the truncated sum on n bits of the binary representations held by A and B.*

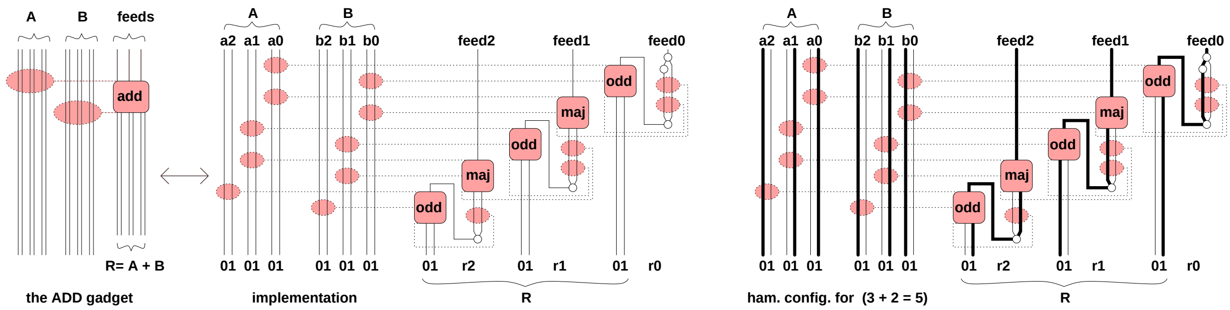


Figure 20: The ADD gadget for $n = 3$ bits

Property 7.6 *In any Hamiltonian configuration, the ADD gadget enforces the operands wires to carry bits, and n paths P_i are uniquely defined, starting from their feeds $feed_i$ and ending to the channel of their wires r_i so that the value of the bit carried by r_i equals the bit i of the sum of the integers held by A and B.*

Proof: The operand wires must carry bits because they are caught in some ODD gadgets and MAJ gadgets which have this property. Now the correction is a straightforward consequence of the formulas exposed previously for the truncated addition of two binary integers. The parsimony of the gadget is infered by the parsimony of the ODD gadget and the MAJ gadget. ∎

Remark 7.7 *All the previous gadgets (except the ADD gadget) have a constant cost even in the planar case because the size of their operands is fixed (i.e. 1 bit each), implying the use of a constant number of crossover-boxes (there is a constant number of crossings between the solid lines and the dotted lines connected to the ellipse, and each such dotted line hides a constant number of XOR-EDGE gadgets). Things are different with the ADD gadget because it deals with n bits long operands. Its complexity is $O(n)$ if one does not care about planarity. Otherwise, complexity grows up to $O(n^2)$, because each operand-wire uses $O(n)$ FLIP-FLAPS to access the operators it is connected to.*

8 The reduction PLAN-UHAM-PATH $\leq$ PLAN-UHAM-CYCLE

We now have all the tools needed to build exact reductions from PLAN-UHAM-PATH and PLAN-UHAM-PATH(x,y) to PLAN-UHAM-CYCLE. Here is the scheme. Figure 21 (left side) shows a graph G along with a Hamiltonian path. The graph G' that will be computed in linear time associates a special gadget to each original vertex, and a couple of edges to each original edge. The rôle of the special gadget is to enforce that the only possible Hamiltonian cycles in the reduced graph are of the form shown of Figure 21 (in the middle), i.e each such cycle somewhat the back-and-forth path of an Hamiltonian path in the original graph.

To do this, we need a special gadget which only only allows Hamiltonian configurations matching the pattern types 1 and 2 in Figure 21 (right side). What must be avoided is the pattern type 3, because then, it is clear that it would be possible to create a Hamiltonian cycle in the reduced graph in bijection to any spanning tree in the original graph. Moreover, note that in the case of the PLAN-UHAM-PATH(x,y) problem, the gadgets associated to vertices x and y must only allow the pattern type 1, whereas the gadgets associated to the other vertices must only allow the pattern type 2. Such versatile gadgets exist and we present an implementation: the CANDELABRA gadget.

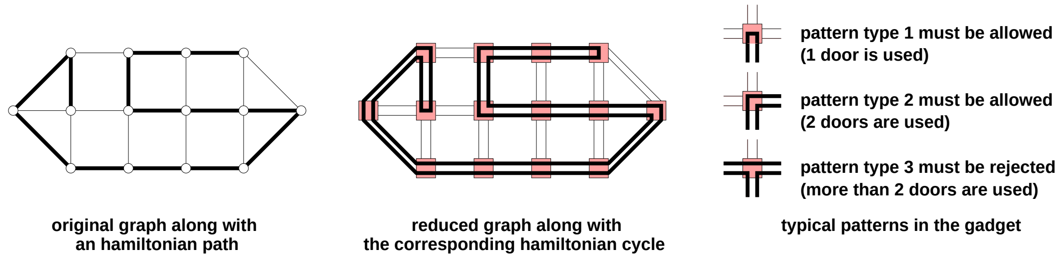


Figure 21: The scheme for the parsimonious reduction PLAN-UHAM-PATH $\leq$ PLAN-UHAM-CYCLE

Gadget 8.1 (CANDELABRA gadget) *The* CANDELABRA gadget *of degree d is the planar Hamiltonian gadget shown in Figure 22. It has d doors, each one being formed by two edges, and a black-box X that may be chosen among X_1, X_2 and X_3 from Figure 23. Its purpose is to implement the versatile gadget described previously. The versatility of the gadget is obtained by modifying what is contained in black-box X. Thus, with a certain black-box X_1, we will be able to make the gadget allow only pattern types 1, with another black-box X_2 only pattern types 2, and with another black-box X_3 both patterns types 1 and 2.*

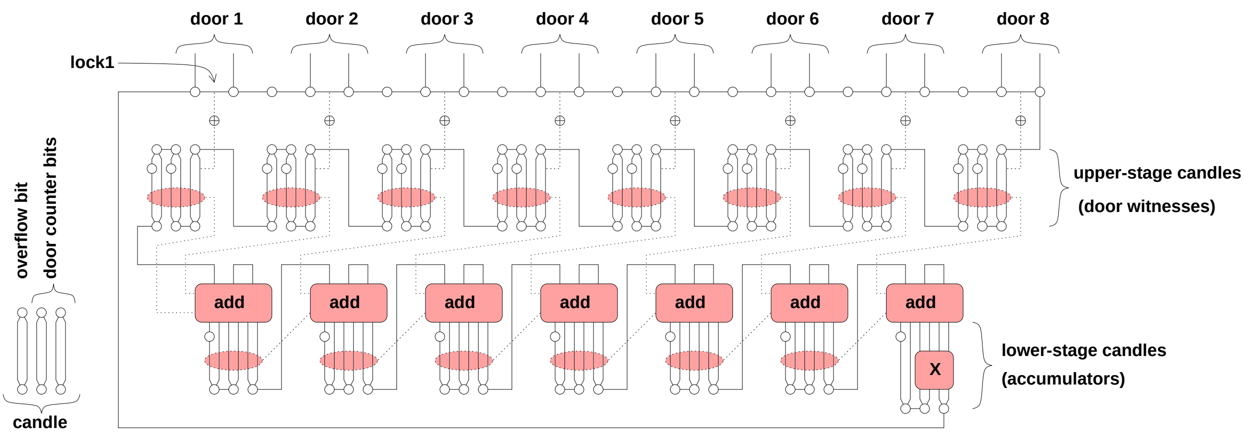


Figure 22: The CANDELABRA gadget (here of degree $d = 8$)

Property 8.2 *If the black-box X of the CANDELABRA is implemented by the gadgets X_1, X_2 or X_3 shown in Figure 23, then for any Hamiltonian configuration, the CANDELABRA allows exactly the pattern types shown in the same figure. Furthermore, the Hamiltonian configuration is uniquely defined by the doors it uses.*

Proof: The edge that lies between the two edges of each door is called a “lock”. Because of the control vertices surrounding each door, notice that either a lock is bold, or both edges of the associated door are bold (we say the door is used). Each lock is bound by a XOR-EDGE gadget to the least significant bit of a “candle” of the upper stage of the gadget. A candle is a set of 3 wires representing an unsigned number on 3 bits in binary. Thus the least significant bit of the candle is 1 (the candle is “lit”) iff the corresponding lock is plain (its door is used). The other wires of these candles are forced to carry 0-bit by a control vertex set on their 0-channel. As a consequence, the number held by a candle is 001 if the associated lock is plain, and is 000 if the lock is bold. Now, all the candles of the upper stage of the gadget are sequentially added from left to right with $n-1$ ADD gadgets. Observe that each result of an addition has its most significant bit forced to be 0 by a vertex set on its 0-channel. This is to prevent overflows in the intermediate computations. So the only values that may be held by the candles of the lower stage of the gadget are 000, 001, 010 and 011. Hence, if we ignore the black-box X, the number of plain locks must be either 0, 1, 2 or 3. It cannot be 0 because we would have a cycle in the gadget, which is not a configuration. There are 3 cases:

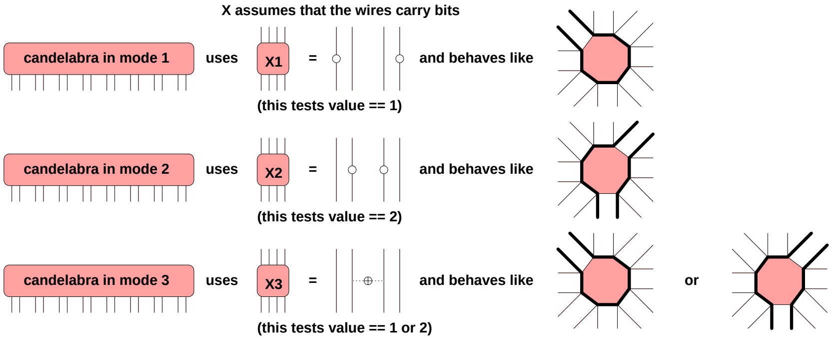


Figure 23: The different modes of use of the CANDELABRA gadget

1. The black box X is X_1 (mode 1). Then the final sum is forced to be 1, because of the control vertices set on the appropriate channels of the 2 least significant bits of the last candle. Thus, only one lock is plain, that is one door is used. The Hamiltonian configurations then follow the pattern type 1.
2. The black box X is X_2 (mode 2). Then the final sum is forced to be 2, because of the control vertices set on the appropriate channels of the 2 least significant bits of the last candle. Thus, exactly two locks are plain, that is two doors are used. The Hamiltonian configurations then follow the pattern type 2.
3. The black box X is X_3 (mode 3). Then the final sum is either 1 or 2, because of the XOR-EDGE gadget binding the two least significant bits of the last candle. Thus, exactly one or two locks are plain, that is one or two doors are used. The Hamiltonian configurations then follow either the pattern types 1 or 2. ■

Remark 8.3 *The parsimony of the gadget is inherited from the parsimony of the ADD gadget and the XOR-EDGE gadget. Moreover note that there are $O(d)$ additions and each addition is computed on a constant number (that is 3) of bits. Thus, although the complexity of the ADD gadget is $O(n^2)$ to handle n bits-long operands in the plane, the complexity to handle each door is $O(1)$, crossovers included, leading to a final complexity of $O(d)$ for the CANDELABRA gadget of degree d.*

Figure 24 make things clear by showing a typical Hamiltonian configuration in a CANDELABRA of degree $d = 8$ with the black-box X chosen to be X_2 or X_3. We now give the precise construction of the reduction and its proof.

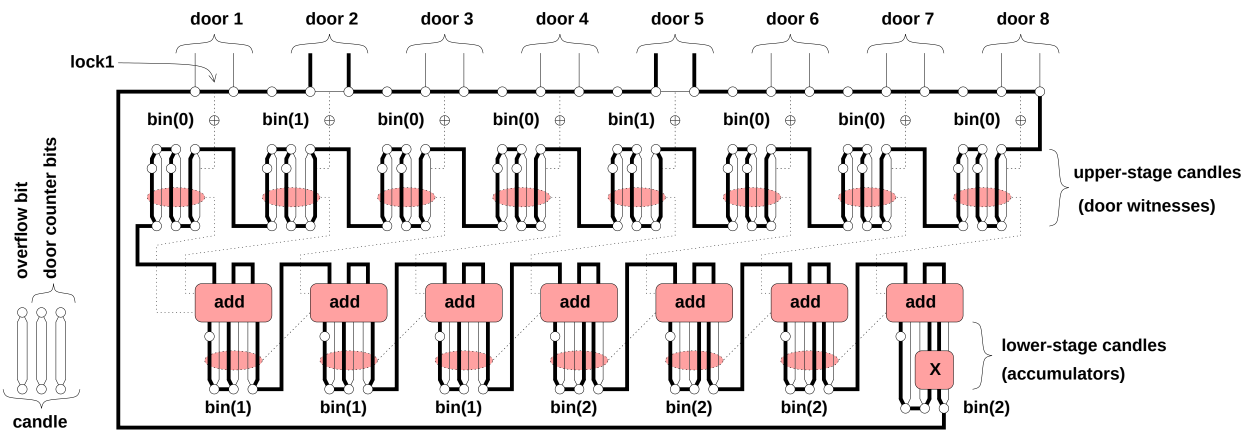


Figure 24: Typical Hamiltonian configurations in the CANDELABRA gadget

Reduction 8.4 (PLAN-UHAM-PATH $\leq$ PLAN-UHAM-CYCLE): *Let $G(V, E)$ be the original graph of the problem PLAN-UHAM-PATH. We create the reduced graph $G(V', E')$ for the problem PLAN-UHAM-CYCLE as follows: For each vertex $v \in V$, create a CANDELABRA $g(v)$ of the same degree, using the mode 3. Each edge in E incident to v is associated with a door of $g(v)$ with respect to its clockwise order in the planar embedding of G around v. For each edge $e = (x, y) \in E$, merge the doors associated to e in $g(x)$ and $g(y)$. We claim that the construction is an exact planarity-preserving reduction from PLAN-UHAM-PATH to PLAN-UHAM-CYCLE.*

Proof: For each vertex $v \in V$ of degree d, the CANDELABRA $g(v)$ has an $O(d)$ complexity, and each edge $e \in E$ contributes to two edges connecting the gadgets. Hence, the linearity follows. The construction is clearly planarity-preserving since the CANDELABRAS are internally planar, and they are arranged using the same embedding as the original graph. Let us prove the parsimony:

- Let P be a Hamiltonian path in G with ending at, say, vertices x and y. The associated Hamiltonian cycle C in G' is the following: Let $e(x)$ and $e(y)$ be the unique bold edges incident to x and y. Set $g(x)$ and $g(y)$ in the unique Hamiltonian configurations such that the locks associated to $e(x)$ and $e(y)$ are the only plain locks in $g(x)$ and $g(y)$, respectively. This is possible since of the CANDELABRAS are all in mode 3. For any vertex $v \in V - \{x, y\}$, with incident bold edges $e_1(v)$ and $e_2(v)$, set $g(v)$ in the unique Hamiltonian configuration such that the locks associated to $e_1(v)$ and $e_2(v)$ are the only plain locks in $g(v)$. This is possible since the modes of the CANDELABRAS are all mode 3. Thus, the path P uniquely defines the cycle C.

- Let C be a Hamiltonian cycle in G'. Note that exactly two gadgets $g(x)$ and $g(y)$, for some vertex $x, y \in V$ must be in a Hamiltonian configuration such that only one of their locks is plain (pattern type 1). If there were no gadget in such a configuration, then C could not be connected; if there were only one then C could not be a cycle, and if there were more than two, C could not be connected. The Hamiltonian path P in G associated to C is the following: Make bold the two edges $e(x), e(y) \in G$ associated with the plain locks of $g(x)$ and $g(y)$. Any other gadget $g(v)$ has been proved to be in a Hamiltonian configuration with exactly two plain locks. Let $e_1(v), e_2(v) \in E$ be the two edges incident to v associated with these locks. Make $e_1(v)$ and $e_2(v)$ bold. Thus, the cycle C uniquely defines the path P. ■

Reduction 8.5 (PLAN-UHAM-PATH(x, y) $\leq$ PLAN-UHAM-CYCLE): *Let $G(V, E)$ be the original graph of the problem PLAN-UHAM-PATH(x, y). We create the reduced graph $G(V', E')$ for the problem PLAN-UHAM-CYCLE as follows: For vertices x and y, create two CANDELABRAS $g(x)$ and $g(y)$ with the same degree, using mode 1. For each vertex $v \in V - \{x, y\}$, create a CANDELABRA $g(v)$ of the same degree, using mode 2. Each edge in E incident to v is associated with a door of $g(v)$ with respect to its clockwise order in the planar embedding of G around v. For each edge $e = (x, y) \in E$, connect with two edges the doors associated to e in $g(x)$ and $g(y)$. We claim that the construction gives an exact planarity-preserving reduction from PLAN-UHAM-PATH(x, y) to PLAN-UHAM-CYCLE.*

Proof: Same proof as above, except that the modes have been refined from mode 3 to modes 1 and 2, so that no Hamiltonian cycle in G' may be created in connexion with a Hamiltonian path in G not ending at x and y. ■

9 The reduction PLAN-DHAM-PATH $\leq$ PLAN-UHAM-CYCLE

In this section, we want to build in linear time a graph G' from any directed graph G so that there is a one-to-one correspondence between the Hamiltonian paths of PLAN-DHAM-PATH or PLAN-DHAM-PATH(x, y) in G and the Hamiltonian cycles in G'. To achieve this goal, we will use the same scheme as in the previous section by refining the CANDELABRA so that it has *entry doors* (simulating in-going arcs) and *exit doors* (simulating out-going arcs) instead of two-way doors (simulating undirected edges). The job is to enforce that no more than one entry door and one exit door per gadget be used in a Hamiltonian configuration by

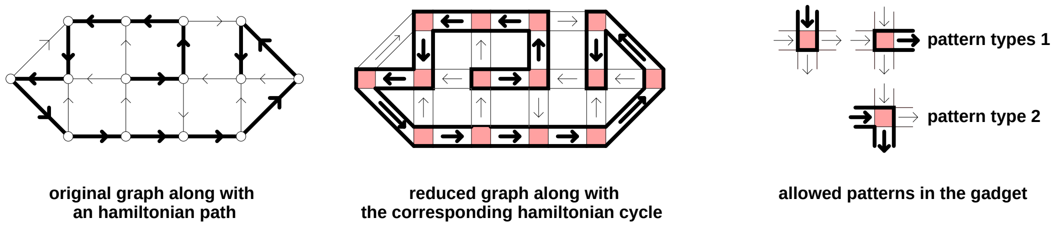


Figure 25: The scheme for the exact reduction PLAN-DHAM-PATH $\leq$ PLAN-UHAM-CYCLE

allowing only pattern types shown in Figure 25 in the gadgets associated to the vertices. In such a way, one simulates the directions of the arcs of G with the undirected edges of G'.

Gadget 9.1 (HEAVY-CANDELABRA gadget) *The* HEAVY-CANDELABRA *is the Hamiltonian planar gadget shown in Figure 26 and is a specialization of the CANDELABRA. Like the CANDELABRA, it has d doors, each door being formed by 2 pending edges, and a black-box X. The difference is that the candles are implemented with 4 bits instead of 3. If we want a door to behave like an entry door (resp. exit door), then we bind its lock via XOR-EDGE with the bit 0 (resp. 2) of its associated candle. Bits 1 and 3 in candles are overflow bits enforced to carry 0-bits, so that the counting of the used entry doors and the counting of the used exit doors neither interfere nor overflow).*

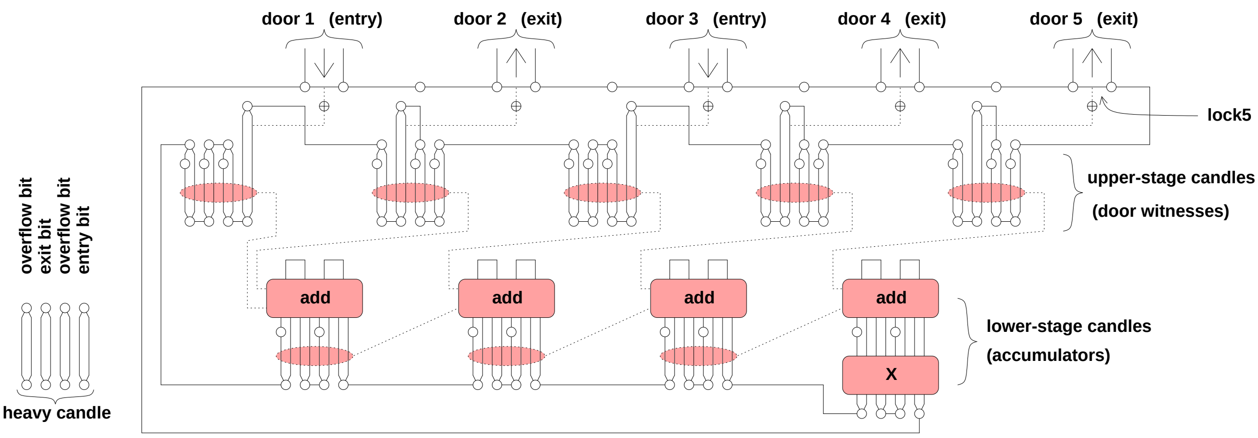


Figure 26: The HEAVY-CANDELABRA gadget (here of degree $d = 4$)

Property 9.2 *If the black box X of the HEAVY-CANDELABRA is implemented as one of the gadgets X_1, X_2 or X_3 shown in Figure 27, then the Hamiltonian configurations allowed by the CANDELABRA follow exactly the corresponding pattern types shown on the same figure. Furthermore, any Hamiltonian configuration is uniquely defined by the doors it uses.*

Proof: Because of the control vertices surrounding each door, either a lock is bold and the two edges of the associated door are plain, or the lock is plain and then the two edges of its doors are bold. Now, each lock is bound via XOR-EDGE to a wire of a "heavy candle". A heavy candle is a set of 4 wires representing an unsigned number in binary. The wire bound to the lock is either the bit 0 (the entry bit) or the bit 2 (the exit bit), according to the status (entry or exit) of the door of the lock. All the other wires of the candle are enforced to carry 0-bits by a control vertex set on their 0-channel. Hence, a candle holds the value 0001 if its door is used and 0000 otherwise. All the candle are sequentially summed via the ADD gadget. The wires of each intermediate result have the wires 1 and 3 (overflow bits) enforced to carry 0-bits by the control-vertices set on their 0-channel, in order to avoid overflows and unwanted propagation of a carry from the bit counting

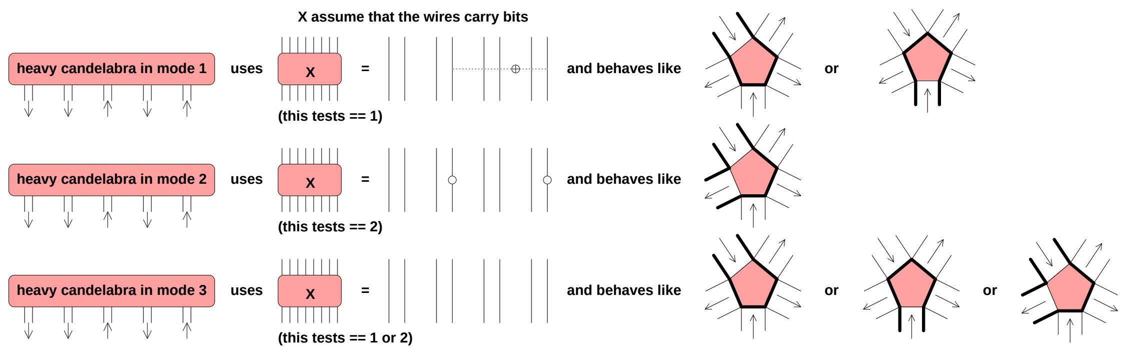


Figure 27: The different modes of use of the HEAVY-CANDELABRA gadget

entry-doors to the bit counting the exit-doors. It follows that the only values that the candles of the lower stage may hold are 0000, 0001, 0100 and 0101. As far as the last candle is concerned, the value 0000 is impossible, since it would mean that no door has been used and that the set of bold edges in the gadget is a cycle, which is not a configuration. The set of possible values in the last candle is further controlled by the nature of the X black-box:

1. The black-box X is X_1 (mode 1). Then the XOR-EDGE between the 1-channels of the wires 0 and 2 of the last candle implies that the allowed values of the final result are 0001 and 0100. Therefore, exactly one door must be used, either an entry-door or an exit-door.

2. The black-box X is X_2 (mode 2). Then the control-vertices set on the 1-channels of wires 0 and 2 of the last candle imply that the unique allowed value of the final result is 0101. Therefore, exactly one entry-door and one exit-door must be used.

3. The black-box X is X_3 (mode 3). This black-box hides nothing, so all values among 0001, 0100 and 0101 are allowed. Hence, the cases allowed by both previous black-boxes are allowed. ■

Remark 9.3 *The parsimony of the gadget is inherited from the parsimony of ADD gadget and XOR-EDGE gadgets. All the additions are made with a fixed number (that is 4) of bits, and thus each ADD gadget has complexity $O(1)$. There are $O(d)$ additions, leading to a complexity $O(d)$ for the HEAVY-CANDELABRA.*

Figure 28 makes things clearer by showing a typical Hamiltonian configuration in a HEAVY-CANDELABRA of degree 4 along with a black-box X set to X_2 or X_3. We now give the precise construction of the reduction and its proof.

Reduction 9.4 (PLAN-DHAM-PATH $\leq$ PLAN-UHAM-CYCLE) *Let $G(V, A)$ be the original graph of the problem PLAN-DHAM-PATH. We create the reduced graph $G(V', E')$ for the problem PLAN-UHAM-CYCLE as follows: For each vertex $v \in V$, create a CANDELABRA $g(v)$ with the same degree, using the liberal mode 3. Each arc of A incident to v is associated with a door of $g(v)$ with respect to its clockwise order in the planar embedding of G around v. If the arc is entering v then door must be an entry-door, and an exit-door otherwise. For each arc $a = (x, y) \in A$, merge the doors associated to a in $g(x)$ and $g(y)$. We claim that the construction gives an exact planarity-preserving reduction from PLAN-UHAM-PATH to PLAN-UHAM-CYCLE.*

Proof: The construction is clearly linear since the complexity of the HEAVY-CANDELABRA associated with a vertex of degree d is $O(d)$, and two edges of G' are associated to each arc of G. The planarity is straightforward since the HEAVY-CANDELABRA is internally planar, and they are arranged in G' with the same embedding as the vertices of G. We prove the parsimony:

- Let P be a directed Hamiltonian path in the original directed graph G. We create the corresponding undirected cycle in the reduced graph G' as follows. Let x and y be the starting vertex and ending

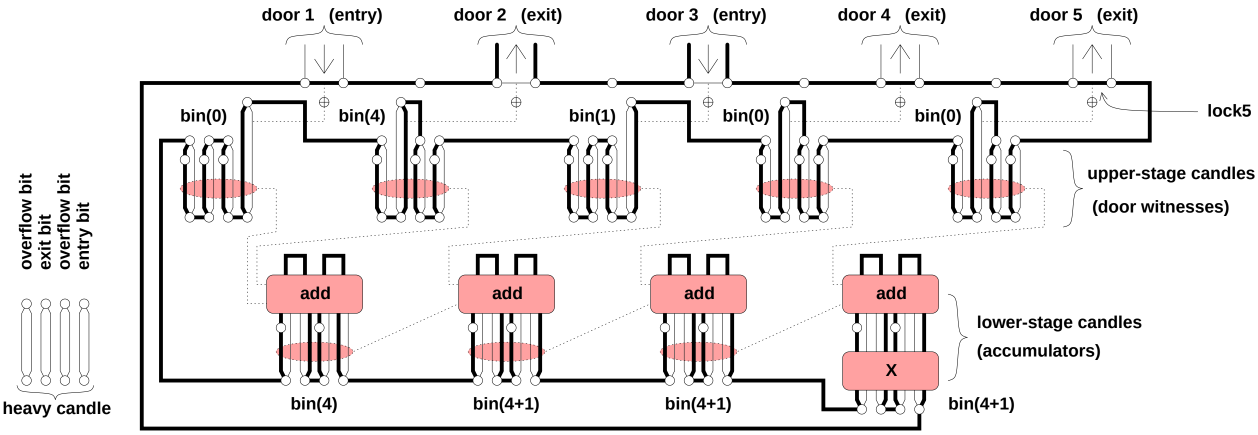


Figure 28: Typical Hamiltonian configuration in a HEAVY-CANDELABRA gadget

vertex in P, and $a(x), a(y) \in A$ their unique incident bold arcs. Set $g(x)$ in the unique Hamiltonian configuration so that the lock of exit-door associated to $a(x)$ is plain. Set $g(y)$ in the unique Hamiltonian configuration so that the lock of entry-door associated to $a(y)$ is plain. This is possible because all CANDELABRAS use the liberal mode 3. For all vertex $v \in V$, let $a_1(v), a_2(v) \in A$ be the only two arcs incident to v. Set $g(v)$ in the unique Hamiltonian configuration so that the only plain lock is the lock whose doors correspond to arcs $e_1(x), e_2(x)$. The cycle is uniquely defined.

- Let C be an undirected Hamiltonian cycle in the reduced graph G'. We claim that there must be exactly one gadget $g(x)$ for some $x \in V$ so that its configuration uses only one exit-door, and there must be exactly one gadget $g(y)$ for some $y \in V$ so that its configuration uses only one entry-door. Clearly, there must be at least one of them, let us say $g(x)$, because otherwise, with the only pattern type 2, C could not be connected. We continue the proof by induction, assuming the existence of $g(x)$ (the proof assuming the existence of $g(y)$ is symmetric). Let $g(v)$ be the gadget connected to $g(x)$ via the door of $g(x)$ whose lock is plain. Either $g(v)$ is the last remaining gadget which has not been set in some Hamiltonian configuration, and then we must have $v = y$ which ends the induction, or the gadget must be set in some configuration so that the lock of the entry-door connected to $g(x)$ is plain and the lock of some exit-door is also plain, so that the cycle may be continued further. The induction is continued by renaming v into x. The Hamiltonian path in G corresponding to C is formed by the sequence of vertices v whose gadgets $g(v)$ were involved in the induction. ■

Reduction 9.5 (PLAN-DHAM-PATH(x,y) $\leq$ PLAN-UHAM-CYCLE) *This is the same reduction as above, except that the modes of the gadgets are refined. the HEAVY-CANDELABRAS $g(x)$ and $g(y)$ associated to x and y are set in mode 1 instead of 3, and all other HEAVY-CANDELABRAS are set in mode 2, so that no Hamiltonian cycle may be created in the reduced graph G' in connexion to a Hamiltonian path of the form $s \longrightarrow^* t$ in G for some vertices s and t with $\{s,t\} \neq \{x,y\}$.*

10 The undirected planar variants with bounded degree

In this section, we show that any undirected variants of PLAN-HAM where the maximal vertex degree is bounded to 3 has the same expressiveness as the planar variants with no bound on the maximal vertex degree. As a first step, notice that all gadgets among AND, XOR-EDGE, XOR-EDGE crossover-box, NOT, OR, XOR, ODD, MAJ, ADD, CANDELABRA, HEAVY-CANDELABRA have maximal vertex-degree 4. In fact, all the vertices of degree 4 come from the AND. Thus, all the variants of PLAN-HAM may be expressed by PLAN-UHAM-CYCLE$(deg \leq 4)$. They may also be expressed by PLAN-UHAM-PATH$(x,y)(deg \leq 4)$, and by PLAN-UHAM-PATH($deg \leq 4$): first, reduce to PLAN-UHAM-CYCLE($deg \leq 4$) and then break the cycles at one of the many degree-2 vertices that the reduction introduces.

Similarly, if we manage to decrease the maximal vertex-degree of the AND gadget from 4 to 3, all PLAN-HAM variants also reduce to PLAN-HAM($deg \leq 3$). Two vertices are of degree 4 in the AND gadget, namely the central and east vertices. We show how to remove them by replacing them with gadgets of maximal vertex-degree 3 without changing the behaviour and the parsimony of the gadget.

Gadget 10.1 (CENTRAL and EAST gadgets) *The* CENTRAL gadget *is the gadget shown in Figure 29a. It is designed to simulate a vertex of degree 4 with only vertices of degree 2 and 3, and specifically the central vertex of the AND gadget (see Figures 29(c,d)). The CENTRAL gadget is not in itself parsimonious but its particular use inside the AND gadget will leave the AND gadget parsimonious. The* EAST gadget *is the gadget shown in Figure 29b. It is essentially the CENTRAL gadget where we add 2 control-vertices X and Y onto the edges (b, c) and (d, e). Thus, its set of Hamiltonian configurations is a subset of the CENTRAL gadget ones. It is designed to simulate the east vertex of the AND gadget. It is parsimonious and thus keeps the AND gadget parsimonious. Both CENTRAL and EAST gadget are used to decrease the maximal vertex-degree of the AND gadget from 4 to 3.*

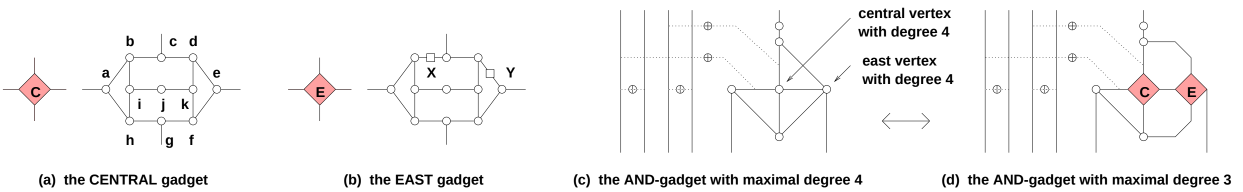


Figure 29: The CENTRAL and EAST gadgets and their use to decrease the degree of the AND gadget to 3

Property 10.2 *The CENTRAL gadget allows exactly the 8 Hamiltonian configurations shown in Figure 30. Each one is formed by only one path which may use any pair of pending edges. The patterns using two opposite pending edges are represented twice (cases 1a/2a and 1b/2b), and the other patterns are represented once (hence, the CENTRAL gadget is not parsimonious).*

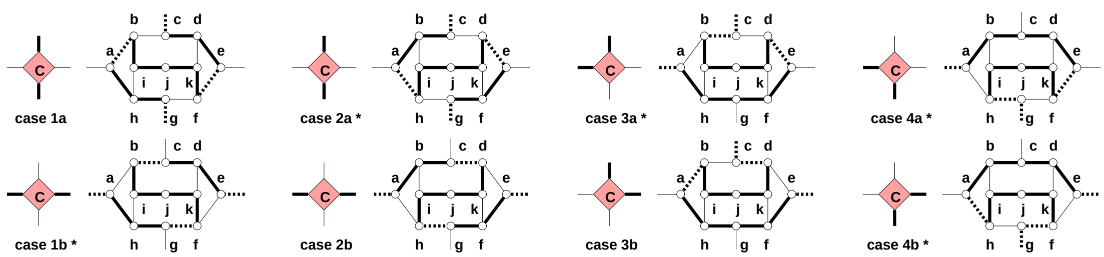


Figure 30: Hamiltonian configurations for the CENTRAL gadget (*: resp. the EAST gadget)

Proof: Let H be a Hamiltonian configuration in the CENTRAL gadget. Notice that both (i, j) and (j, k) must be bold because of the control-vertex j. Now there are 4 main cases:

1. Suppose that (b, i) and (k, f) are bold. This leaves only edges (a, h), (h, g) to capture vertex h, and (c, d), (d, e) to capture vertex d. Thus, all these edges must be bold. We now have two subcases:
   a. Either (a, b) is bold. Then (g, f) must be plain, since otherwise a cycle would occur. This implies the boldness of (f, e) and the pending edge at g. All the vertices are captured, and it follows that the pending edge at c must be bold. See Figure 30(1a).

b. Either (a, b) is plain. Then both (b, c) and the pending edge at a are bold. Hence, (f, e) must be plain because otherwise a cycle would occur. This immediately implies that (g, f) and the pending edge at e are bold. See Figure 30(1b).

2. Suppose that (i, h) and (d, k) are bold. Then, since it is the symmetrical case of 1, this leads to the two Hamiltonian configurations shown in Figure 30(2a,2b), by vertical symmetry of the gadget. This case is the source of the non-parsimony of the gadget.

3. Suppose that (b, i) and (d, k) are bold. This leaves only edges (a, h), (h, g) to capture vertex h, and (g, f), (f, e) to capture vertex e. Hence, all these edges must be bold. We now have two subcases:
   a. Either (b, c) is bold. Then (c, d) must be plain, otherwise a cycle would occur. This implies that (d, e) and the pending edge at c must be bold. All the vertices are now captured and it follows that the pending edge at a must be bold. See Figure 30(3a).
   b. Either (b, c) is plain, that is, (c, d) is bold in order to capture c. This is the symmetrical case of 3a. Hence, this leads to the Hamiltonian configuration shown in Figure 30(3b), by vertical symmetry of the gadget.

4. Suppose that (i, h) and (k, f) are bold. Then, since it is a symmetrical case of 3, this leads to the two Hamiltonian configurations of Figure 30(4a,4b), by horizontal symmetry of the gadget. ■

Remark 10.3 *As far as the EAST gadget is concerned, it is easy to notice that it allows only the Hamiltonian configurations of cases 1b, 2a, 3a, 4a and 4b (marked by stars) in Figure 30, due to the control vertices X and Y set on edges (b, c) and (d, e). Also, note that it removes the duplicates for the straight vertical and horizontal patterns, and thus the EAST gadget is parsimonious.*

Property 10.4 *The CENTRAL and EAST gadgets are suitable to replace respectively the central and east vertices of the AND gadget so that its behaviour remains unchanged and parsimonious.*

Proof: Notice on Figures 10 and 11 of the AND gadget that the only patterns needed to simulate the central vertex are the cases 3a, 3b, 4a and 4c of Figure 30. Thus, the straight patterns of cases 1a, 1b, 2a and 2b will remain unused, and the duplicates won't be a source of non-parsimony for the AND gadget. Now, this is not the case for the east vertex since one of the straight pattern must be used. However, notice that the patterns needed to simulate the east vertex are the cases 2a, 3a, 4a, 4b and they are still allowed by the EAST gadget. The case 1b is also allowed but will remain unused. Since the EAST gadget is parsimonious, we may use it in place of the east-vertex inside the AND gadget and keep it parsimonious. ■

The result above is provably optimal for all the HAM variants except UHAM-PATH(x, y), in the sense that the degree-2 vertices cannot be eliminated without losing parsimony (in other words, the graphs cannot be required to be cubic).

Property 10.5 *There is no parsimonious reduction from our HAM variants (which all can have a unique solution) to cubic UHAM-CYCLE.*

Proof: Any cubic graph has a even number of Hamiltonian cycles using an given edge (see [Baz98] or [Pap94] for a proof). Hence, a cubic graph with one Hamiltonian cycle has necessarily another one, and there is no reduction to cubic UHAM-CYCLE preserving the unicity of the solutions. ■

Property 10.6 *There is no parsimonious reduction from our HAM variants to cubic UHAM-PATH and to cubic UHAM-PATH(x).*

Proof: If we have a solution path P_1 to one of these problems, then P_1 can be easily transformed into a distinct solution P_2: Let $y \neq x$ be the other end of P_1. Let (y, b) be one of the two unused edges in P_1, and a the vertex preceding b in P_1, starting from a. Just make (a, b) plain and (y, b) bold. We have a new Hamiltonian path P_2 joining a to x. Hence, there is no reduction to cubic UHAM-PATH and UHAM-PATH(x) preserving the unicity of the solutions. ■

We leave open whether cubic UHAM-PATH(x, y) may be included in the class of Figure 2 or not. We believe that it cannot.

11 The PLAN-SAT problem

Thanks to their modularity, the gadgets developed in sections 4, 5, 6, 7 are highly reusable. As an example of a by-product of these gadgets, we show in this section that PLAN-UHAM-CYCLE expresses PLAN-SAT. An instance of the clause-satisfiability problem is said to be planar if its underlying bipartite graph G has a planar embedding. G is defined as follows: for each variable x create a vertex $v(x)$, for each clause c create a vertex $v(c)$, and everywhere a variable x occurs as a literal x or $\neg x$ in a clause c, create an edge $(v(x), v(c))$.

In [HMRS98], Hunt et al. showed that problem UNIQUE-PLAN-SAT, which asks whether there exists a unique satisfying assignment for a PLAN-SAT instance, is DP-complete under randomized reductions. Since the parsimony preserves the unicity of solutions, and all the studied variants of PLAN-HAM have been proved to be equivalent under (linear) parsimonious reductions, our (linear) parsimonious reduction PLAN-SAT $\leq$ PLAN-UHAM-CYCLE yields the following corollary: All our variants of UNIQUE-PLAN-HAM are DP-complete under randomized reductions.

The involved reduction associates a gadget to each clause and each variable, and embed them in the same way as in the PLAN-SAT instance. The variable-gadget will be simply made of a wire enforced to carry a bit, and the clause-gadget for a clause c of length k will have k wires, one for each literal in c and embedded on the outer-face of the gadget, with their clockwise order following the order of the literals in clause c. Given a Hamiltonian path in the reduced graph, the bit carried by the wire of the variable or a literal in a clause will represent its assignment. The clause-gadget will be designed so that it allows only Hamiltonian configurations that assign at least one literal to true. Also, the wires of a variable v and corresponding literals in clause-gadgets will be linked by XOR-EDGE so that the values carried in their wires are consistent. Moreover, all these components will be connected together following an arbitrary spanning tree in the dual graph of the PLAN-SAT instance. Using a tree spanning the faces as a backbone to fetch all the gadgets has the advantage to involve only a linear number of XOR-EDGES crossover-boxes (see Section 5) when crossing face boundaries formed by XOR-EDGES relations.

Gadget 11.1 (CLAUSE gadget) *The* CLAUSE gadget *of length k is the planar Hamiltonian gadget shown in Figure 31. It has two pending edges and k edges, called* bulbs*, lying on the boundary of its outer-face and ordered in clockwise order between the two pending edges. The bulbs represent literals (ordered the same way) in a planar clause, and the bold bulbs in a Hamiltonian configuration H are that ones whose associated literals are satisfied by the partial assignment encoded by H.*

Property 11.2 *Let S be the set of bold bulbs of an Hamiltonian configuration H in a CLAUSE gadget of length k. Then $|S| \geq 1$ (meaning that the disjunction of literals is satisfied) and H consists of one path using both pending edges and uniquely defined by S.*

Proof: The $k-1$ wires of the lower stage are accumulators whose bits hold the value of the sequential disjunction between the k wires of the upper-stage whose 1-channels are the k bulbs of the CLAUSE gadget. Thus the last wire of the lower stage holds a bit equal to the result of the disjunction of the bits of all the wires of the upper stage. Yet, its value is enforced to be 1 by a control-vertex c set on its 1-channel. Thus the disjunction must be satisfied, that is, at least one bulb must be bold. Since all wires are chained, H consists of only one path entering and leaving the gadget at the two pending edges. The parsimony is inherited by the parsimony of the OR gadget. Moreover, notice that only $O(k)$ crossover-boxes are used. ■

The global reduction is as follows: Let G be the undirected bipartite labeled graph, input of the PLAN-SAT problem, where V is the class of the variable-vertices, C is the class of the clause-vertices, and P (resp. N) is the class of edges $(v, c) \in V \times C$ labeled "positive" (resp. "negative"), meaning that the variable v occurs

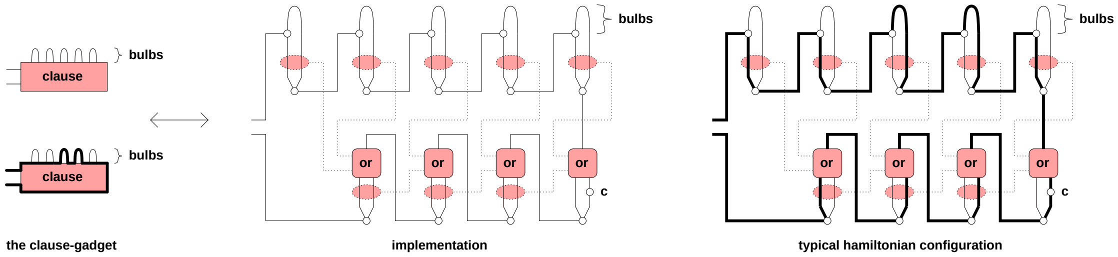


Figure 31: A CLAUSE gadget of length $k = 5$

positively (resp. negatively) in the clause c. We build G' as follows: for each clause-vertex c of length k in C, create a CLAUSE gadget $g(c)$ of length k. For each variable-vertex v in V, create an edge $g(v)$ which splits into a wire. For each edge $(v, c) \in P$, link with a XOR-EDGE the 0-channel of $g(v)$ and the bulb of $g(c)$ associated to the literal of v, and symmetrically, for each edge $(v, c) \in N$, link with a XOR-EDGE the 1-channel of $g(v)$ and the bulb of $g(c)$ associated to the literal of v, following the same embedding as in G. The partial graph computed at this point is called *the skeleton* of G'. An example of a skeleton is shown in Figure 32.

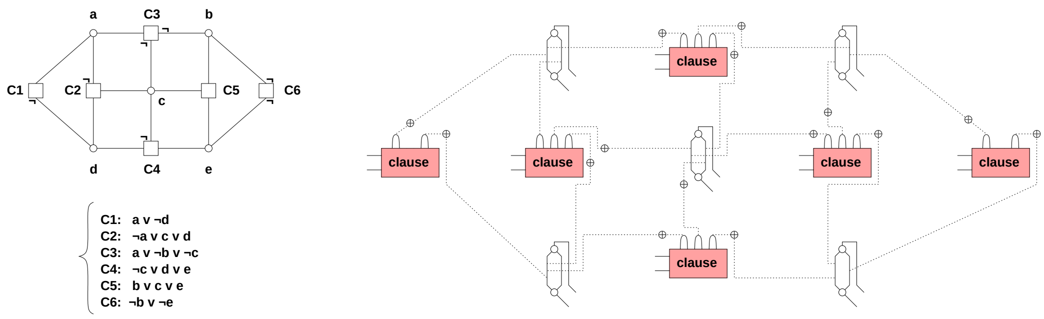


Figure 32: Skeleton of the reduction PLAN-SAT $\leq$ PLAN-UHAM-CYCLE

Since all the components are only connected by XOR-EDGES, there is still no Hamiltonian cycle in this skeleton. We need some way to connect all the pending edges of the gadgets, which are somewhat prisoner inside the faces of the skeleton corresponding to the faces of G. Fortunately, since the boundaries of these faces merely consist of XOR-EDGES, and since we have a parsimonious crossover-box for the XOR-EDGES, we may go through them easily. To obtain the linearity, we choose to visit these faces following an arbitrary spanning tree T in the dual graph of G The dual graph and its spanning tree are easily computed in linear time on a RAM. If T is transposed in the skeleton, we get a unique way to connect the gadgets by turning around the tree T (see Figure 34).

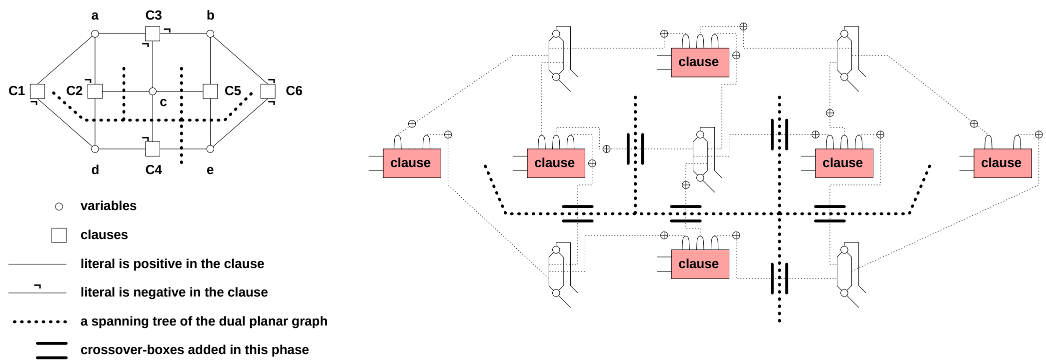


Figure 33: Addition of the crossover-boxes along the spanning tree

The edges added to the skeleton form what we call *the backbone* of G'. It is built as follows: for each edge $t = (f_1, f_2) \in T$, where f_1, f_2 are some faces of G, let F_1,F_2 be the corresponding faces in the skeleton, and let e be the corresponding primal edge in G, and let x be the XOR-EDGE associated to e in the skeleton; create a crossover-box on x for two normal edges with the two pending edges lying inside F_1 and the two others inside F_2 (see Figure 33). For each face F of the skeleton associated to a face in G, scan in clockwise order each gadget g (possibly a clause gadget, a variable gadget or a crossover-box) whose two pending edges lie inside F, and connect the last pending edge of g to the first pending edge of its successor for the clockwise order (see Figure 34).

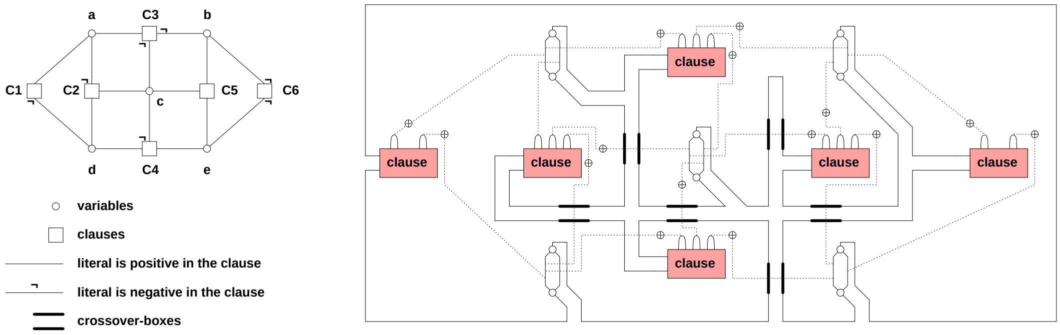


Figure 34: Backbone added to the skeleton and connecting the components

Property 11.3 *The linear planarity-preserving construction described above is a parsimonious reduction PLAN-SAT $\leq$ PLAN-UHAM-CYCLE.*

Proof: Let H be a Hamiltonian cycle in G'. All gadgets in G' must be in some Hamiltonian configuration. Thus, for any variable-gadget $g(v)$, its pending edges must be bold, and then exactly one of the two channels of its wire must be bold. Therefore, the wire in $g(v)$ carries a bit. The values of the bits carried by the wires of the variable-gadgets uniquely define a valuation in G and determine whether the bulbs they are connected to via XOR-EDGES are bold or plain. So, the set of bold bulbs in all CLAUSE gadgets uniquely defines the way the valuation attempts to satisfy the clauses. Since every CLAUSE gadget must be in a Hamiltonian configuration, at least one bulb per gadget must be bold, that is, at least one literal in the associated clause must be satisfied by the valuation. Hence, the valuation must be satisfying. Now all gadgets of the skeleton are only connected to each other by the edges of the backbone in a circular fashion, which implies that the whole backbone is bold. This defines the one-to-one correspondence between the Hamiltonian cycles in G' and the satisfying solutions in G. Figure 35 shows an example to make things clear. ∎

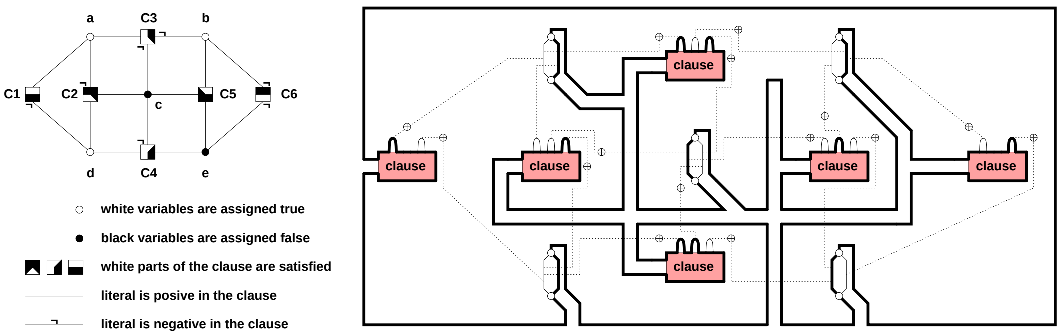


Figure 35: The unique Hamiltonian cycle corresponding to the satisfying assignment $a, b, d = 1$; $c, e = 0$

12 Conclusion

We have proved the following results:

- All the non-planar variants of HAM in Figure 2 are equivalent under exact reductions. In particular, there is an exact reduction from any HAM variant with unbounded vertex-degree to any HAM variant whose vertex-degree is bounded to 3.
- All the variants of PLAN-HAM in Figure 2 are equivalent under exact reductions. In particular, there is an exact reduction from any directed planar variant to any undirected planar variant, and from any planar path variant to any planar cycle variant (even if the ends of the path do not lie on the boundary of the same face), and from any PLAN-HAM variant with unbounded vertex-degree to any PLAN-HAM variant whose vertex-degree is bounded to 3.
- As far as the vertex-degree bound is concerned, our results are somewhat optimal, since one easily proves that the studied variants of HAM cannot be reduced parsimoniously to the following cubic HAM variants (where all vertices have exactly degree 3): cubic Hamiltonian cycle, cubic Hamiltonian path with one or both unspecified ends. We believe that it is also the case for the cubic Hamiltonian path with both ends specified.
- PLAN-SAT is exactly reducible to all the PLAN-HAM variants, and hence all the associated UNIQUE-PLAN-HAM variants are DP-complete under randomized reductions.

For our reductions, we used parsimonious planarity-preserving gadgets making logical and arithmetical operations easy in an Hamiltonian context. This family of gadgets may be easily extended to support subtractions, equality tests, order tests, etc. in the same modular and incremental fashion. With this extended family, we have recently shown in a companion paper (see [Bar]) that some problems are exactly reducible to the non-planar HAM problem. In particular:

- The problem of k disjoint paths (ref. ND40 in [GJ79]) has an exact reduction to HAM with complexity $O(m \log k)$ for any constant k;
- The traveling salesman problem (TSP) in a graph with integer edge lengths bounded by some constant Δ (ref. ND22 in [GJ79]) has an exact reduction to HAM with complexity $O(m \log \Delta)$, thus HAM and TSP with bounded integer edge lengths have the same expressiveness;
- The problem of the longest path (or cycle) in a graph with integer edge lengths bounded by some constant Δ (ref. ND28,29 in [GJ79]) has an exact reduction to HAM, so both problems have the same expressiveness;
- Some problems about the existence of spanning trees with special properties (such as a bound on the total edge capacity, on the maximal vertex degree, or on the number of leaves, refs. ND1,2,5 in [GJ79]) also have an exact reduction to HAM if the edge weights are also bounded.

We leave the following problems open: Is the planar undirected cubic Hamiltonian path problem with both ends specified belongs to the class of Figure 2? Also, our proof that the planar undirected Hamiltonian variants with degree bounded to 3 belong to this class immediately implies that the planar directed variants with in-degree and out-degree bounded to 3 (that is global degree 6) also belong to this class. Is it possible to obtain the same results for the directed variants whose global degree is bounded to 3 by adapting the tools developed by Plesnik in [Ple79] who gives polynomial but non-linear and non-parsimonious reductions from these variants to SAT?

Finally, the class of problems exactly reducible to HAM (denoted the HAM class) seems to be quite large and a natural question arises: does this class have an intrinsic (i.e. machine-independent) characterization? Does it have a logical characterization which could be comparable to Monadic NLIN, a logical class of problems defined by [LW99] who prove that it is equal to the SAT class, that is the class of problems linearly reducible to SAT via some logical reduction? Our hope is motivated by the robustness of the HAM class justified by the results of [Bar] presented above. We are convinced that the HAM class is at least as robust as the SAT class and so should have a machine-independent characterization. We also conjecture that the HAM class is strictly larger than the SAT class. A formal proof of such a conjecture would require an even more restrictive reduction than our exact reduction, and should take in account the locality of the reductions designed in this paper (they essentially consist of local substitutions by gadgets).

References

[Bar] R. Barbanchon, Problems equivalent to Sat and Hamilton under linear parsimonious reductions. *in preparation.*

[Baz98] C. Bazgan, Approximation de problèmes d'optimisation et de fonctions totales de NP. *Ph.D Thesis, University of Paris XI Orsay*, (1998).

[BDG90] J.L. Balcazar, J. Diaz, J. Gabarro, Structural Complexity, *Springer-Verlag*, vols. I and II (1990).

[Coo73] S.A. Cook, A hierarchy for non-deterministic time complexity, JCSS, 7 (1973), 343-353.

[Cre95] N. Creignou, The class of problems that are linearly equivalent to Satisfiability or a uniform method for proving NP-completeness. *Theor. Comput. Sci.*, 145 (1995), 111-145.

[GJ79] M.R. Garey, D.S. Johnson, Computers And Intractability, a guide to NP-completeness. *Freeman*, (1979).

[GJT76] M.R. Garey, D.S. Johnson, R.E. Tarjan, The planar Hamiltonian circuit problem is NP-complete. *SIAM J. Comput.*, 5 (1976), 704-714.

[GO98] E. Grandjean, F. Olive, Monadic Logical Definability of Nondeterministic Linear Time. *Comput. Complex.*, 7 (1998), 54-97.

[Gra90] E. Grandjean, A Nontrivial Lower Bound for an NP Problem on Automata. *SIAM J. Comput.* 19 (1990), 438-451.

[Gra94] E. Grandjean, Invariance Properties of RAMS and Linear Time. *Comput. Complex.*, 4 (1994), 62-106.

[Gra96] E. Grandjean, Sorting, Linear Time and the Satisfiability Problem. *Annals of Mathematics and Artificial Intelligence*, 16 (1996), 183-236.

[GS89] Y. Gurevich, S. Shelah, Nearly linear time. *Proc. Symposium on logical foundations of computer science, Lect. Notes Comput. Sci.*, 363 (1989), 108- 118.

[GS99] E. Grandjean, T. Schwentick, Machine-independent characterizations and complete problems for deterministic linear time. *submitted to SIAM. J. Comput.*, (1999).

[HMRS98] H.B. Hunt III, M.V. Marathe, V. Radhakrishnan, R.E. Stearns, The Complexity of Planar Counting Problems. *SIAM J. Comput.*, 27 (1998), 1142-1167.

[JP85] D.S. Johnson, C.H. Papadimitriou, Computational Complexity, in The Traveling Salesman Problem. *Lawler, Lenstra editors*, (1985) 37-85.

[Lic82] D. Lichtenstein, Planar Formulae and their uses. *SIAM J. Comput.*, 11 (1982), 329-343.

[LW99] C. Lautemann, B. Weinzinger, MonadicNLIN and Quantifier-Free Reductions. *CSL, 8th annual conference of the EACSL, Lect. Notes Comput. Sci.*, 1683 (1999), 322-337.

[Pap94] C.H. Papadimitriou, Computational Complexity. *Addison-Wesley*, (1994).

[Ple79] J. Plesnik, The NP-completeness of the Hamiltonian Cycle Problem in planar digraphs with degree bound 2. *IPL*, 8 (1979), 199-201.

[PPST83] W.J. Paul, N. Pippenger, E. Szemeredi, W.T. Trotter, On Determinism versus Non-Determinism and Related Problems. *FOCS*, (1983) 429-438.

[Sch78] C.P. Schnorr, Satisfiability is Quasilinear Complete in NQL. *J. of the ACM*, 25 (1978), 136-145.

[Sch97] T. Schwentick, Algebraic and logical characterization of deterministic linear time, *STACS*, (1997), 463-474.

[SFM78] J.I. Seiferas, M.J. Fischer, A.R. Meyer, Separating Nondeterministic Time Complexity Classes. *JACM*, 25 (1978), 146-167.

[VV86] L.G.Valiant, V.V. Vazinari, NP is as easy as detecting unique solutions. *Proc. 17th ACM Symp. in Theory of Computing*, (1994), 458-463.