ELSEVIER
Discrete Applied Mathematics 116 (2002) 55–71
DISCRETE APPLIED MATHEMATICS

A constructive enumeration of nanotube caps

G. Brinkmann[a,*], U.v. Nathusius[a], A.H.R. Palser[b]

[a] *Fakultät für Mathematik, Universität Bielefeld, Postfach 100131, D 33501 Bielefeld, Germany*
[b] *Physical and Theoretical, Chemistry Laboratory, University of Oxford, South Parks Road, Oxford OX1 3QZ, UK*

Received 7 January 2000; revised 31 August 2000; accepted 11 September 2000

Abstract

In this article, we describe a method of constructing all non-isomorphic caps for a nanotube with given parameters l, m and present results of a computer program based on this algorithm.

Keywords: Planar graph; Fullerene; Nanotube; Construction; Isomorphism

1. Introduction

The icosahedral C_{60} molecule discovered by Kroto et al. in 1985 [7] was the first known example of a fullerene: a molecular cage of carbon consisting of fused hexagon and pentagon rings. The interest generated by this discovery led to the identification of many other fullerenes and related structures [11], and in particular to the observation of carbon nanotubes.

Carbon nanotubes were first discovered by Sumio Iijima in 1991 [6] in the deposit formed by the method used for bulk fullerene generation. Each tube has the appearance of a rolled-up strip of graphite capped with a half fullerene at each end, and they have been observed both individually (as single-walled nanotubes) and nested within one another (as multi-walled nanotubes). Nanotubes have been seen with diameters between 7 Å (the diameter of C_{60}) and approximately 500 Å . They can grow up to 5–10 μm in length. Since their discovery, they have been the subject of many experimental and theoretical studies owing to their remarkable structural and electronic properties [10] which promise a variety of potential applications. For an informal survey, see e.g. [8,9].

The body of any single-walled carbon nanotube can be specified by two integers $l \geqslant m$ and $m \geqslant 0$ which define the circumference vector $\boldsymbol{c} = l\boldsymbol{e}_1 + m\boldsymbol{e}_2$ of the tube when

* Corresponding author.
E-mail addresses: gunnar@mathematik.uni-bielefeld.de (G. Brinkmann), unathusi@mathematik.uni-bielefeld.de (U.v. Nathusius), adam.palser@balliol.oxford.ac.uk (A.H.R. Palser).

0166-218X/02/$ - see front matter
PII: S0166-218X(00)00328-0

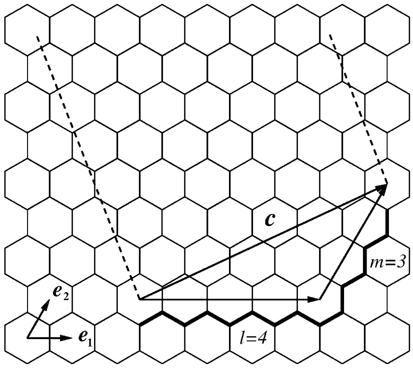

Fig. 1.

it is projected onto a graphitic (hexagonal) lattice of carbon atoms [5] (see Fig. 1). The nanotube is then constructed theoretically by cutting the graphite sheet perpendicularly to this circumference vector and rolling the edges of the resultant strip together.

The aim of this study is to present an efficient algorithm for enumerating all caps which are consistent with an arbitrary nanotube body. Although several approaches have been proposed for dealing with this problem, most of them are either inaccurate, unable to enumerate the caps of large diameter nanotubes in a reasonable time, or both [3,1]. The first algorithm to produce reliable results is the method presented in Ref. [2], and the aim of the present work is to give an independent check of the published results, and also to improve on the efficiency of earlier algorithms.

2. Definitions

In mathematical language, a *fullerene* is a cubic planar graph embedded in the sphere so that all the faces are hexagons and pentagons. Owing to the Euler formula there are exactly 12 pentagons.

A *half-tube* comprises a cap joined to a nanotube body which extends infinitely in one direction. Therefore, a half-tube is an infinite cubic tiling of the plane with hexagons and exactly six pentagons. A *cap* is a simply connected part of this tiling containing the six pentagons, and the *tube body* is the remaining infinite region. If the boundary between the cap and the tube is chosen appropriately, the tube body can be imagined as a rolled strip of graphite without any pentagons and can be characterised by a vector (l,m) as shown in Fig. 1. We can always assume $l \geqslant m$ since we do not want to distinguish between mirror–images. We also do not want to distinguish between caps that lead to isomorphic half-tubes.

Since there is no *natural* boundary that separates the cap from the tube, the boundary of the two parts can be chosen to ensure that these caps can be enumerated as efficiently as possible.

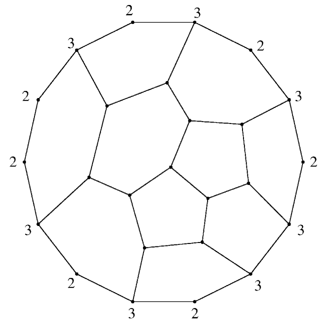

Fig. 2.

First, we introduce some terminology. A *patch* is a simply connected set of hexagons and pentagons in the (oriented) plane with all interior vertices having valency 3 and all boundary vertices having valency 2 or 3. A vertex with valency 2 on the boundary of a patch is called a 2-*vertex*, and one with valency 3 is called a 3-*vertex*. An edge is *convex* if its endpoints are 2-vertices, and *concave* if its endpoints are 3-vertices. A patch without concave edges on the boundary is called *pseudoconvex*, and if there is exactly one concave edge it will be called *almost pseudoconvex*. We will not consider other patches.

The boundary between the tube and the cap can be defined to fulfil the special requirements shown in Fig. 2: labelling the 2-vertices of the boundary "2" and the 3-vertices "3", one can choose a boundary of an (l,m) nanotube of the form $(23)^l(32)^m$, where $m=0$ in the case of a zig-zag boundary. We will construct caps with such a boundary as representatives of the equivalence class of all caps giving the same half-tube.

The existence of such a boundary can be seen by following the edges in the tiling neighbouring the vectors $l\boldsymbol{e}_1$ and $m\boldsymbol{e}_2$ in Fig. 1, and an example of a cap with this boundary is shown in Fig. 2. In both figures, the parameters are $l=4$ and $m=3$. The distances in both directions between the convex and the concave edge on the boundary of the tube body correspond to the vector (l,m).

If a patch has such a boundary, an arbitrary number of hexagon layers may be added without changing the shape of the boundary or the corresponding half-tube. Therefore, we may restrict our attention to those caps having at least one pentagon in the boundary. In fact, the following stronger result also holds:

Lemma 2.1. *Among all caps with a* $(23)^l(32)^m$ *boundary in a given half-tube, there exists at least one cap with a pentagon in both parts of the boundary between the concave and convex edges. A pentagon carrying the convex edge is considered as belonging to both parts.*

This can be proven by removing layers or rows of hexagons from the boundary until the pentagons occur. Details of the proof are similar to those in the proof of Proposition 3.3, so we will omit them here.

The boundary of an arbitrary pseudoconvex patch can be described by a cyclic boundary sequence. This sequence consists of the number of 2-vertices between convex edges in a clockwise direction. In the absence of any convex edges but at least one pentagon in the boundary, the cyclic boundary sequence consists of the numbers of 2-vertices between pentagons. These sequences are defined to be canonical if they are lexicographically minimal among all cyclic permutations.

The boundary of an almost pseudoconvex patch is described in a similar way. The canonical sequence always starts at the first convex edge following the unique concave edge in clockwise direction and records the distance between the concave edge and its neighbouring convex edges in addition to the distances between the convex edges.

3. Construction

Applying the considerations above, we can describe our first task as that of constructing non-isomorphic, simply connected patches of hexagons and six pentagons with at least one pentagon in a given boundary of the form $(23)^l(32)^m$.

Using the following result obtained in [4] we can compute an upper bound for the number of vertices $\boldsymbol{V(n)}$ in such a patch for any given boundary length $n{:=}l+m$ and therefore prove the finiteness of the class of all nonequivalent caps for given parameters l, m:

Theorem 3.1. *Given $n,k \in \mathbb{N}$, $k < 6$, we get a patch with the maximum possible number of hexagons among all patches with boundary length at most n and k pentagons by arranging the faces in a spiral beginning with the pentagons and adding hexagons until the boundary would be too long.*

For a proof see [4].

The maximum number of vertices for this patch will be denoted $V(n,k)$. If we apply this result to our case we only obtain a statement about patches with five pentagons. But since we are interested in patches with at least one pentagon in the boundary we can substitute one boundary pentagon by a hexagon. This operation increases the vertex number by one. Therefore, an upper bound $V(n)$ for the number of vertices of a possible cap with boundary length n is $V(n) = V(n+1,5) - 1$.

The class of patches with boundary length at most n, at most six pentagons, and at least one pentagon in the boundary when the number of pentagons is 6 will be denoted by $\mathscr{P}_{(\boldsymbol{n})}$.

New patches are constructed by adding faces to an existing patch. This must be done in a systematic way in order to avoid either missing a patch or constructing the same patch more than once. A powerful strategy to achieve this is the construction of “marked patches” – i.e. patches carrying a mark on one of the convex boundary edges resp. on a boundary edge of a pentagon if there is no convex edge. If the mark is at a position so that evaluating the boundary description at this point gives a canonical sequence, the mark is called canonical. The class of all canonically marked patches with corresponding unmarked patches in $\mathscr{P}_{(n)}$ will be denoted by $\bar{\mathscr{P}}_{(n)}$. Despite the fact that different marked patches may occur during the construction which are in fact isomorphic as unmarked patches, this will not concern the resulting caps with $m \neq 0$, since they have exactly one convex edge in the boundary, so that the position for the mark is unique and there is a one-to-one correspondence between marked and unmarked patches. Nevertheless, one should keep in mind that the orientation of patches is fixed, and therefore in the case $m = n$ two different patches might be mirror images of each other. For the case $m = 0$ there might be several pentagons in the boundary and so no one-to-one correspondence between marked and unmarked patches exists.

The enlarging operations can be best described as the inverse of reduction operations. The reduction operations we use are defined as follows:

Delete a row of faces in a clockwise direction, beginning with the marked face (i.e. the face with the marked edge), until either

- a removed face is a pentagon,
- a face carrying a second convex edge is removed (so stop at once if the first face already carries two convex edges),
- the next face carries a concave edge sharing a vertex with the actual face or,
- the face which would be next in the patch without the faces deleted has already been deleted.

The mark is set on the next canonical possibility in a counterclockwise direction.

It can be easily seen that this reduction is well defined for every almost pseudoconvex patch and that after deleting the row of faces no second concave edge will be present. Furthermore, all faces that are deleted belong to the boundary of the patch.

Theorem 3.2. *For given $l, m \in \mathbb{N}$, $l + m \geqslant 6$, every patch in $\bar{\mathscr{P}}_{(l+m)}$ can be recursively constructed starting with a single pentagon and a single hexagon.*

We can prove this by showing that every patch in $\bar{\mathscr{P}}_{(l+m)}$ except the pentagon and the hexagon can be reduced to a smaller member of $\bar{\mathscr{P}}_{(l+m)}$ using the reduction operations defined above.

The construction makes use of the fact that the reduction defined above leads to a *unique predecessor* of the original patch. This approach would fail if the resulting patch did not belong to the class of patches we want to generate.

Proposition 3.3. *$\bar{\mathscr{P}}_{(l+m)}$ is closed under the reduction rules.*

This statement depends on a patch not containing more than one concave edge or faces larger than a hexagon. Otherwise, the patch might be disconnected after applying the reduction rules.

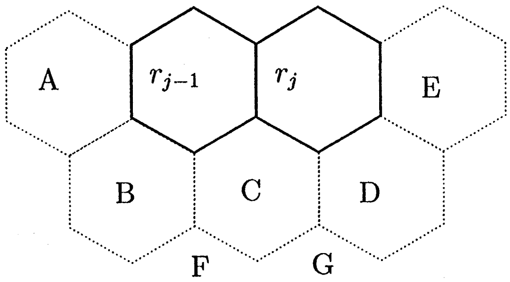

Fig. 3.

First, we will prove the following lemma:

Lemma 3.4. *Let P be a patch, R the set of faces to be deleted according to the reduction rule, and I a component of $P \setminus R$ so that there are faces in R not neighbouring I. Faces of R neighbouring I will be denoted by R_I.*

Then there exists a concave edge, carried by a face either of R_I or of I (i.e. both vertices of the edge belong to R_I or I).

Proof. There is a canonical numbering $r_1, \ldots, r_n$ of the faces in R, beginning with the marked face and proceeding in the order in which the faces are deleted. Owing to the reduction rules all "inner faces" of R – that is $r_2, \ldots, r_{n-1}$ – have at least one 2-vertex in the boundary and two neighbours in R. Therefore, they have at most two neighbours outside R.

From the connectedness of P it can be easily deduced that R_I is nonempty, so choose the face $r_j \notin R_I$ with the minimum label j so that r_{j+1} or r_{j-1} is an element of R_I.

We will only discuss the case when $r_{j-1} \in R_I$ and $r_j \notin R_I$. The case $r_{j+1} \in R_I$ can be done analogously. Furthermore, we will restrict ourselves to the case where all faces involved are hexagons. The cases with pentagons involved follow by the same argument.

We distinguish how the different positions shown in Fig. 3 could be occupied (or not). We will use the same name for the position as well as for the face occupying it. Since r_j is not neighbouring I we have that $C, D, E \notin I$ whenever there is a face on this position. At least one of A or B has to be in I. If there is a face on position A this has to be in R, so B must be in I in any case.

If position C was empty this would induce a concave edge on r_{j-1}, proving the lemma. So assume that $C \in P \setminus I$ which implies – since C is neighbouring I – that $C \in R_I$. Note that C cannot be the direct successor or predecessor in R of r_j or r_{j-1}, owing to the reduction rules.

Now, we distinguish four cases:
1. the positions F and G are both occupied,
2. only G is occupied,

3. only F is occupied,
4. both are empty.

Case 1: C would either be an inner face (D occupied) – contradicting $C \in R$ – or carry a concave edge, thus proving our lemma.

Case 2: Again there would be a concave edge on C.

Case 3: A face on position D would already give a concave edge on C, so assume that D is empty. Therefore, C can only have F as its sole ancestor or descendant in R, i.e., C must be the first or last face of R. Since C has no convex edge (and is not a pentagon in the absence of convex edges) it can only be the last face r_n. However, the cycle in the inner dual, induced by the faces $r_{j-1}, r_j, r_{j+1}, \ldots, F, C$ is a Jordan curve in P with the empty space D in its interior – contradicting the simple connectedness of P. (For this case, the argument is different when $r_{j+1} \in R_I$ and $r_j \notin R_I$. It can be shown that r_1 would have at least two convex edges or be a pentagon and therefore would have been deleted alone.)

Case 4: As the only ancestor or descendant of C in R, D must be occupied and belong to R. Again we get $C = r_n$ and there must be a concave edge at B in order to stop the removal of faces.

The following lemma will also be a useful tool to prove the proposition:

Lemma 3.5. *In the case of a patch being disconnected after applying the reduction rules, there is no face in the removed row R neighbouring two different components.*

Proof. Assume there are such faces. Among all faces with this property choose the one with the minimal label, say r_j. Again, we will restrict our considerations to the case of r_j being a hexagon and leave the pentagon case to the reader. There are two possible configurations for r_j depending on whether the smallest number of edges in the boundary of r_j separating the faces in I from those in J is 1 or 2. They are shown in Figs. 4 and 5.

In Fig. 4, r_j cannot have ancestors or successors in R, since their existence would imply a concave edge on r_j, and so r_j would not have been deleted. Thus, r_j must be the single element in R. However, this is not possible since it does not carry a convex edge and cannot therefore be the start.

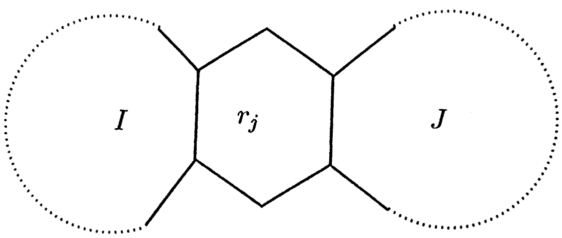

Fig. 4.

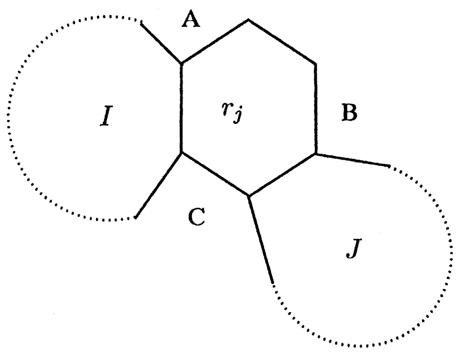

Fig. 5.

Hence, we may assume that we have the configuration shown in Fig. 5.

Using similar arguments as for Fig. 4 it can be seen that A, B and C cannot belong to I or J. The case $j = 1$ is not possible because a possible successor r_{j+1} would be on position B so that r_j would not carry a convex edge. If r_1 is the only element of R (or in general $r_j = r_n$), there would have to be a concave edge at the intersection of the (empty) position B with J to stop the removal of further faces. Since there may be only one concave edge, this would force another face at position C which was an element of R with a smaller index, contradicting the choice of r_j.

Therefore, we have $1 < j < n$. This would force elements of R onto positions A and B, introducing a concave edge between them and contradicting the rule that the deletion stops before a concave edge. □

Proof of Proposition 3.3. That at most one concave edge is present in the reduced patch, that only pentagons and hexagons occur, that the boundary length is at most $l+m$ and that the components of the reduced patch are all simply connected is obvious. What remains to be shown is that the reduced patch is connected.

Assume that there is a patch P which is disconnected after the deletion of a layer R so there are at least two components I and J. Owing to the connectedness of P there are faces neighbouring I and faces neighbouring J, so Lemma 3.5 implies that there are elements in R *not* neighbouring I and elements not neighbouring J.

Therefore, we may apply Lemma 3.4 for both components I and J and get the existence of one concave edge for each of the parts which (owing to Lemma 3.5) cannot be identical. Hence, we have at least two concave edges, which contradicts the definition of $\bar{\mathscr{P}}_{(l+m)}$. □

The set of enlarging operations is defined to be the "inverse" of all possible reductions. Having applied one of these operations we put the new mark on the first newly added face and accept the patch for further operations if and only if the mark is at a *canonical* position. This guarantees a *correct* operation, i.e., a construction operation that is the inverse of a reduction operation of a canonically marked patch.

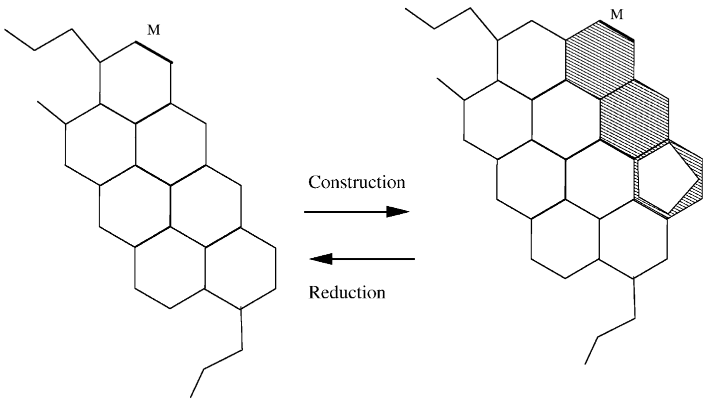

Fig. 6.

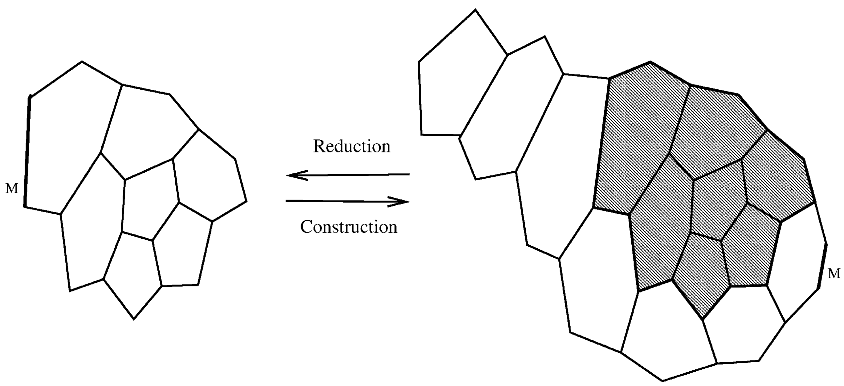

Fig. 7.

The unique rule for the reduction of a patch induces a set of several enlarging operations, since the structure and position of the row R depend on the patch that is reduced. We distinguish three classes of operations depending on the boundary characteristics both of the starting and of the resulting patch. A pseudoconvex patch may be reduced to another pseudoconvex patch, and an almost pseudoconvex patch may be reduced to either a pseudoconvex or an almost pseudoconvex patch. Each of these classes has to be considered separately in order to obtain the complete set of enlarging operations.

As an example, the *standard operation* is shown in Fig. 6. A layer of several hexagons is added to a patch, where the last face may also be a pentagon. The new row starts and ends at faces of the original patch which carry convex edges. This rule may be applied to pseudoconvex as well as to almost pseudoconvex patches. The new patch will remain in the same class.

A more tricky operation is shown in Fig. 7. It can only be applied to a pseudoconvex patch with exactly three convex edges, two of them adjacent. Starting at the single

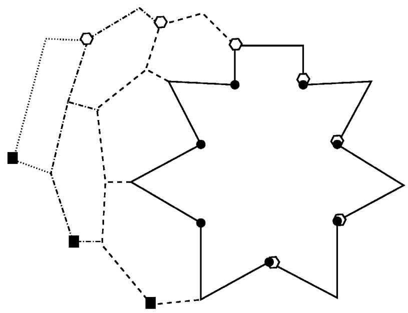

Fig. 8.

convex edge, hexagons are added until they cover the next convex edge, followed by an arbitrary number of hexagons which will be added to the last new face. The consequent "tail" can be finished by a pentagon or a hexagon.

4. Optimising the construction algorithm

In order to speed up the generation process it is necessary to detect patches which do not lead to caps with the desired boundary structure at the earliest opportunity. To this end various criteria for early pruning of the search tree are included in the program. This section gives two such criteria as examples.

The easiest bounding criterion comes from the boundary length b, defined as the number of 2-vertices in the boundary. Since it increases monotonically during the construction process, we can stop as soon as $x > l + m$.

An example of a more complex criterion is as follows: if there is exactly one convex edge (resp. two convex edges) and one concave edge, the remaining steps are uniquely determined, since there are already five pentagons in the patch. We will illustrate this for the case of only one convex edge (see Fig. 8): let $P \in \bar{\mathscr{P}}_{(n)}$ be a patch with exactly one convex edge and x 3-vertices on the boundary and suppose we want to generate patches with an (l,m) boundary.

An operation which could be applied in this case is the addition of an incomplete layer of hexagons (i.e., a layer of at most $x-1$ hexagons). The operation which adds x hexagons also leads to an almost pseudoconvex patch, but it will not be discussed here. The layer starts somewhere in the middle of the row, forms a concave edge at its beginning, and ends at the convex edge. First, we observe that in order to construct a patch with an (l,m) boundary the parameter l must equal x. This can be easily seen by observing that the number of 3-vertices in each newly added row decreases by one in each step while the distance between the end of the row and the concave edge (in a clockwise direction) increases by 1 in each step, so that the total number of 3-vertices between the beginning of the row and the concave edge remains constant.

Furthermore, we can use the fact that there has to be a pentagon in each boundary part. We know that there are only hexagons between the concave edge and the following convex edge, and thus there is only one possibility for having a pentagon in each boundary part: the pentagon must belong to both parts. Therefore, it must be added on its own as the final layer.

The parameter m equals the number of rows added, and so the maximum value of m is $x-1$. First, we check that $x=l$ and $m<x$. Then we know exactly how many layers must be added and that the first row must have length m. In this way, we can complete the patch to form a cap with the required parameters without further branching.

5. Isomorphism rejection

The generation algorithm described in Section 3 enumerates all possible in-fillings of a given boundary path to produce a set of marked patches. However, two or more marked patches can lead to isomorphic half-tubes when they are glued to the body of a nanotube. In order to obtain only one patch for every possible half-tube, it is therefore necessary to define a set of canonicity requirements (see, for example, [2]).

First (as discussed in Section 2) note that in the case $m \neq 0$ if a patch does not contain a pentagon in one of the boundary paths joining the convex and concave edges, another properly formed cap can be obtained by removing all hexagons adjacent to this path. Patches which satisfy the criterion of having at least one pentagon in both boundary paths are called minimal patches. A pentagon bordering the convex edge is considered to lie in both parts of the boundary, whereas a pentagon bordering the concave edge does not lie in either part.

In the case $m=0$, we can require at least one pentagon to be in the boundary. Patches with this property are again called minimal. The following theorem gives some information about the relation between $m=0$ minimal patches that lead to isomorphic half-tubes:

Theorem 5.1. *When $m=0$, two or more minimal marked caps can lead to isomorphic half-tubes if and only if they are themselves isomorphic as unmarked patches.*

Proof. First note that owing to the requirement of a canonical cap having at least one pentagon in the boundary, no two caps can be properly contained within one another. If the boundaries a and b of two patches A and B intersect, then according to the Jordan curve theorem they must do so at least twice. A boundary segment b_1 of patch B which is outside A and neighbours the unbounded region therefore has two endpoints which split the boundary of A into two parts a_1 and a_2. Both $b_1 \cup a_1$ and $b_1 \cup a_2$ are Jordan curves, and the interior of one of them is disjoint of the interior of A. This region is defined as C, and only consists of hexagons.

Owing to the Euler formula, a connected patch of hexagons must possess at least six convex edges. Each of the two intersections of a and b can give rise to at most two

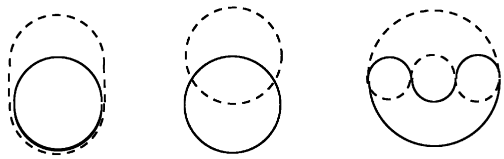

Fig. 9.

convex edges on the boundary of C, but there can be no other convex edges. Hence, region C cannot exist, and the only possibility for isomorphism in the case $m = 0$ is if marked caps are isomorphic as unmarked patches. □

Lemma 5.2. *For the case $m = 0$ there is a canonicity check with complexity* $O(v)$, *with v the number of vertices in the cap.*

Proof. In this instance, the canonically marked representative is chosen in the following manner: for a given patch, label the marked vertex 1, and then label the connected vertices 2 and 3 in a clockwise direction such that the directed edge (1,2) has the marked pentagon on the right. Having labelled all the vertices around a vertex i, go to vertex $i+1$ and inspect each neighbour in a clockwise direction, starting with the vertex that is labelled with the smallest number. An unlabelled neighbour is assigned the next unused number, and this process is repeated until all the vertices have been labelled. The code corresponding to the mark on vertex 1 is then generated by concatenating the list of neighbours for each vertex, and separating each list by a zero. The codes are computed and compared for every possible mark and also for the possible marks in the mirror image of the patch, and only that marked patch which corresponds to the lexicographically minimal code is accepted.

Since the labelling can clearly be done in linear time and since there are at most six pentagons in the boundary at which to start, the total complexity is $O(v)$. □

Lemma 5.3. *For the case $m \neq 0$ there is a canonicity check with complexity* $O(l^2+v)$, *with v the number of vertices in the cap.*

Proof. When $m \neq 0$, two patches are considered to be isomorphic if they can be cut out of the same half-tube. To determine canonicity, a minimal patch is glued to a tube body large enough to contain all other minimal patches. This tube can be constructed in time $O((l+m)^2) = O(l^2)$. Boundary paths of the form $(23)^l(32)^m$ which enclose other such minimal patches can intersect in three possible ways, as illustrated in Fig. 9. Since minimal patches have at least one pentagon in each section of their boundary, all other isomorphic patches can be found by searching from every pentagon edge in turn. A necessary criterion for a patch to be canonical is to have the shortest distance between the convex edge and a pentagon in the m part of the boundary of all the possible caps. Any remaining ambiguity is removed by computing a code in the same way as described for the $m = 0$ case. Once again, the representative corresponding to the lexicographically minimal code is accepted and in the case of $l=m$ mirror images also have to be taken into account.

An upper bound on the complexity of this isomorphism procedure can be obtained by considering each of the two parts. First, the maximum number of search paths that must be constructed is $m * n_s$, with the number of possible starting edges n_s equal to 30 – the number of pentagon edges. Since each path has a maximum length of $l+m$, the complexity scales as $m*(l+m)$, which is bounded by $O(l^2)$. The second stage of the isomorphism algorithm involves calculating a code for the original patch and checking it against the codes of at most n_s other patches. Calculating the code of the original patch scales linearly with the number of vertices v. The codes of every other patch can then be calculated and checked vertex-by-vertex to ascertain which is lexicographically shorter. Hence, only a maximum of v vertices must be examined in each case and the complexity of computing and checking codes for isomorphic patches scales as $O(v)$. In case $l=m$ all this has to be done in the mirror image too, giving another constant factor of 2. An upper bound on the overall complexity is therefore $O(l^2+v)$. □

Furthermore, this procedure can be optimised for $m \neq 0$ patches by making use of the following theorem:

Theorem 5.4. *A minimal patch with a pentagon edge as the convex edge is the unique canonical patch.*

Proof. From the proof of Theorem 5.1, it is clear that if two patches A and B intersect, then there exists a region C bounded by one boundary segment of A and one of B which consists solely of hexagons. This connected patch of hexagons must possess at least six convex edges.

First, note that if the concave edge of patch A neighbours patch C, then it will be a convex edge on the boundary of C. Next, note that each of the two intersections of a and b can give rise to at most two convex edges on the boundary of C that were not originally concave or convex edges of a and b. The remaining convex edge of patch C must therefore be the convex edge of patch B. However, if the convex edges of A and B are also pentagon edges, then the convex edge of B is in the region $A \cap B$, and cannot be on the boundary of C. This forces a contradiction. Region C cannot exist and hence there exists only one minimal patch with a pentagon edge as the convex edge. □

6. Results

The algorithm enables the computation of half-tubes with quite a large boundary length. In Tables 1–4, we present the most chemically interesting examples of $m=0$

Table 1
Number of half-tubes with isolated pentagons with parameters (l, l)

l	No. of caps	Computation time (s)	Structures per second
5	1	—	—
6	18	—	—
7	145	—	—
8	805	1.9	423.7
9	3047	8.8	346.2
10	9342	32.3	289.2
11	24 195	104.8	230.9
12	56 118	283.1	198.2
13	118 429	711.7	166.4
14	233 409	1777.0	131.4
15	433 119	3852.0	112.4
16	766 799	7447.2	103.0
17	1 301 531	15 120.4	86.1
18	2 134 521	27 492.2	77.6
19	3 392 685	50 940.5	66.6
20	5 252 207	93 551.0	56.1
21	7 936 157	158 394.5	50.1
22	11 743 417	266 905.1	44.0
23	17 043 573	435 610.7	39.1
24	24 318 288	790 184.9	30.8
25	34 151 231	1 069 405.8	31.9

Table 2
Number of half-tubes with parameters (l, l)

l	No. of caps	Computation time (s)	Structures per second
3	1	—	—
4	12	—	—
5	73	—	—
6	348	—	—
7	1223	1.6	764.4
8	3731	6.7	556.9
9	9787	22.9	427.4
10	23 316	69.4	336.0
11	50 702	188.5	269.0
12	103 284	474.6	217.6
13	197 823	1094.7	180.7
14	361 440	2348.0	153.9
15	631 892	5210.6	121.3
16	1 066 023	10 190.4	104.6
17	1 739 664	20 551.7	84.6
18	2 761 278	38 633.0	71.5
19	4 270 494	72 222.6	59.1
20	6 459 406	130480.3	49.5

Table 3
Number of half-tubes with isolated pentagons with parameters $(l, 0)$

l	No. of caps	Computation time (s)	Structures per second
5	0	—	—
6	0	—	—
7	0	—	—
8	0	—	—
9	1	—	—
10	7	—	—
11	31	—	—
12	124	—	—
13	347	—	—
14	889	—	—
15	1963	—	—
16	4032	1.4	2880.0
17	7617	2.9	2626.6
18	13 754	6.0	2292.3
19	23 473	10.9	2153.5
20	38 777	17.1	2267.7
21	61 639	33.4	1845.5
22	95 515	58.2	1641.2
23	143 861	86.9	1655.5
24	212 433	145.2	1463.0
25	306 805	231.1	1327.6
26	436 049	369.5	1180.1
27	608 953	535.8	1136.5
28	839 083	661.9	1267.7
29	1 139 711	1099.7	1036.4
30	1 530 710	1240.8	1233.6
31	2 031 275	2209.5	919.3
32	2 669 656	3099.2	861.4
33	3 473 175	4128.4	841.3
34	4 480 775	5569.8	804.5
35	5 730 375	7538.6	760.1
36	7 274 912	10 100.6	720.2
37	9 165 629	13 395.1	684.3
38	11 473 127	17 568.9	653.0
39	14 265 835	22 899.1	623.0
40	17 636 042	29 508.5	597.7
41	21 673 607	34 571.2	626.9
42	26 497 932	42 027.7	630.5
43	32 224 919	52 239.5	616.9
44	39 007 003	66 214.7	589.1
45	46 992 323	83 153.3	565.1
46	56 372 753	105 764.4	533.0
47	67 335 375	130 564.2	515.7
48	80 119 801	163 690.0	489.5
49	94 959 257	202 769.6	468.3
50	112 150 037	250 264.8	448.1

Table 4
Number of half-tubes with parameters $(l, 0)$

l	No. of caps	Computation time (s)	Structures per second
5	1	—	—
6	5	—	—
7	13	—	—
8	42	—	—
9	106	—	—
10	258	—	—
11	552	—	—
12	1153	—	—
13	2199	—	—
14	4083	1.1	3711.8
15	7157	2.1	3408.1
16	12 208	4.0	3052.0
17	19 984	7.1	2814.6
18	31 998	12.2	2622.8
19	49 632	21.2	2341.1
20	75 558	36.1	2093.0
21	112 251	51.0	2201.0
22	164 087	79.7	2058.8
23	235 213	121.7	1932.7
24	332 585	186.9	1779.5
25	462 726	288.5	1603.9
26	636 240	437.2	1455.3
27	863 140	588.4	1466.9
28	1 158 950	846.1	1369.8
29	1 538 495	1212.9	1268.4
30	2 024 051	1666.8	1214.3
31	2 636 801	2434.7	1083.0
32	3 407 908	3182.2	1070.9
33	4 367 122	4483.1	974.1
34	5 556 944	6190.3	897.7
35	7 018 226	7612.1	922.0
36	8 808 111	10 415.4	845.7
37	10 981 431	13 511.1	812.8
38	13 613 719	18 219.0	747.2
39	16 777 635	23 027.5	728.6
40	20 571 308	29 236.1	703.6

and $l = m$ nanotubes up to a length of $l + m = 50$ for the case of isolated pentagons and $l + m = 40$ for the general case.

The computation times given were on Linux machines with a Pentium II, 350 MHz processor.

The results were checked as far as possible against independent results in [2] and were found to be in complete agreement.

The generation program based on this algorithm is free for scientific use and can be obtained from any of the authors.

References

[1] T.Y. Astakhova, G.A. Vinogradov, E. Osawa, Numerical generation of nanotube caps (ii), Fullerene Sci. Technol. 7 (5) (1999) 769–779.

[2] G. Brinkmann, P.W. Fowler, D.E. Manolopoulos, A.H.R. Palser, A census of nanotube caps, Chem. Phys. Lett., to appear.

[3] M.S. Dresselhaus, G. Dresselhaus, P.C. Eklund, Science of Fullerenes and Carbon Nanotubes, Academic Press, New York, 1996, ISBN 0-12-221820-5.

[4] J. Greinus, Patches mit minimaler Randlänge, Diplomarbeit, Bielefeld, 2000.

[5] N. Hamada, S. Sawada, A. Oshiyama, New one-dimensional conductors – graphitic microtubules, Phys. Rev. Lett. 68 (1992) 1579–1581.

[6] S. Iijima, Helical microtubules of graphitic carbon, Nature 354 (1991) 56–58.

[7] H.W. Kroto, J.R. Heath, S.C. O'Brien, R.F. Curl, R.E. Smalley, C_{60}: Buckminsterfullerene, Nature 318 (1995) 162–163.

[8] R.F. Service, Mixing nanotube structures to make a tiny switch, Science 271 (1996) 1232.

[9] R.F. Service, Superstrong nanotubes, show they are smart, too, Science 281 (1998) 940–942.

[10] S.J. Tans, M.H. Devoret, H.J. Dai, A. Thess, R.E. Smalley, L.J. Geerligs, C. Dekker, Individual single-wall carbon nanotubes as quantum wires, Nature 386 (1997) 474–477.

[11] M. Terrones, W.K. Hsu, J.P. Hare, H.W. Kroto, H. Terrones, D.R. Walton, Graphitic structures: from planar to spheres, toroids and helices, Phil. Trans. R. Soc. Lond. A 354 (1996) 2025–2054.