Characterizations and Linear Time Recognition of Helly Circular-Arc Graphs

Min Chih Lin[1,⋆] and Jayme L. Szwarcfiter[2,⋆⋆]

[1] Universidad de Buenos Aires, Facultad de Ciencias Exactas y Naturales, Departamento de Computación, Buenos Aires, Argentina
oscarlin@dc.uba.ar

[2] Universidade Federal do Rio de Janeiro, Instituto de Matemática, NCE and COPPE, Caixa Postal 2324, 20001-970 Rio de Janeiro, RJ, Brasil
jayme@nce.ufrj.br

Abstract. A circular-arc model $(C, \mathcal{A})$ is a circle C together with a collection $\mathcal{A}$ of arcs of C. If $\mathcal{A}$ satisfies the Helly Property then $(C, \mathcal{A})$ is a Helly circular-arc model. A (Helly) circular-arc graph is the intersection graph of a (Helly) circular-arc model. Circular-arc graphs and their subclasses have been the object of a great deal of attention, in the literature. Linear time recognition algorithm have been described both for the general class and for some of its subclasses. However, for Helly circular-arc graphs, the best recognition algorithm is that by Gavril, whose complexity is $O(n^3)$. In this article, we describe different characterizations for Helly circular-arc graphs, including a characterization by forbidden induced subgraphs for the class. The characterizations lead to a linear time recognition algorithm for recognizing graphs of this class. The algorithm also produces certificates for a negative answer, by exhibiting a forbidden subgraph of it, within this same bound.

Keywords: algorithms, circular-arc graphs, forbidden subgraphs, Helly circular-arc graphs.

1 Introduction

Circular-arc graphs form a class of graphs which has attracted much interest, since its first characterization by Tucker, almost fourty years ago [9]. There is a particular interest in the study of subclasses of it. The most common of these subclasses are the proper circular-arc graphs, unit circular-arc graphs and Helly circular-arc graphs (Golumbic [3]). Linear time recognition and representation algorithms have been already formulated for general circular-arc graphs (McConnell [7], Kaplan and Nussbaum [5]), proper circular-arc graphs (Deng,

⋆ Partially supported by UBACyT Grants X184 and X212, PICT ANPCyT Grant 11-09112, CNPq under PROSUL project Proc. 490333/2004-4.

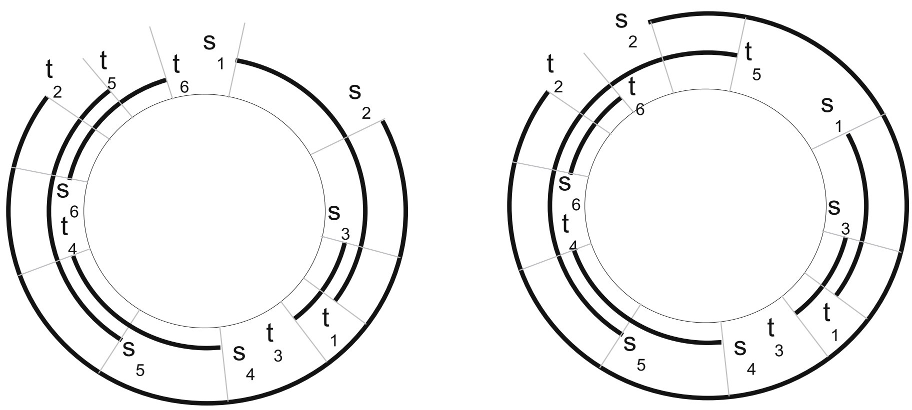
⋆⋆ Partially supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, and Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro, FAPERJ, Brasil.

D.Z. Chen and D.T. Lee (Eds.): COCOON 2006, LNCS 4112, pp. 73–82, 2006.

Hell and Huang [1]) and unit circular-arc arc graphs (Lin and Szwarcfiter [6]). For Helly circular-arc graphs, the best recognition algorithm is by Gavril [2], which requires $O(n^3)$ time. Such an algorithm is based on characterizing Helly circular-arc graphs, as being exactly those graphs whose clique matrices admit the circular 1's property on their columns [2]. The book by Spinrad [8] contains an appraisal of circular-arc graph algorithms.

In the present article, we propose new characterizations for Helly circular-arc graphs, including a characterization by forbidden induced subgraphs for the class. The characterizations lead to a linear time algorithm for recognizing graphs of the class and constructing the corresponding Helly circular-arc models. In case a graph does not belong to the class, the method exhibits a certificate, namely a forbbiden induced subgraph of it, also in linear time.

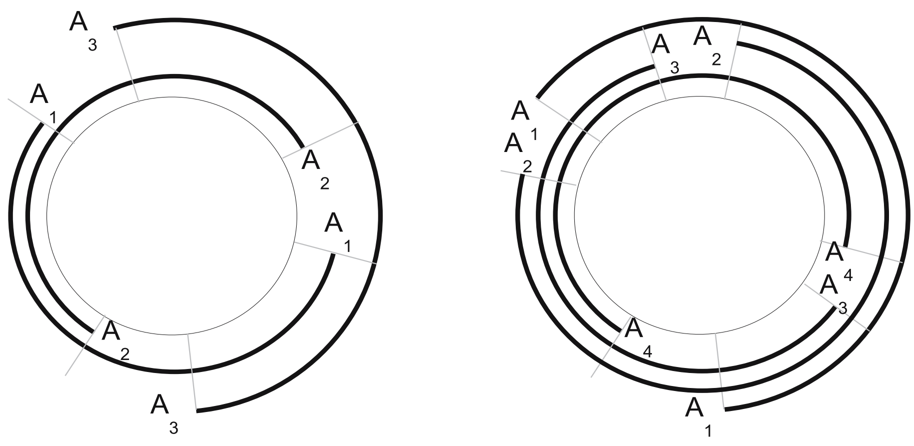
Let G be a graph, V_G, E_G its sets of vertices and edges, respectively, $|V_G| = n$ and $|E_G| = m$. Write $e = v_i v_j$, for an edge $e \in E_G$, incident to $v_i, v_j \in V_G$. A *clique* of G is a maximal subset of pairwise adjacent vertices. Denote $N(v_i) = \{v_j \in V_G | v_i v_j \in E_G\}$, call $v_j \in N(v_i)$ a *neighbour* of v_i and write and $d(v_i) = |N(v_i)|$.



(a) (b)

Fig. 1. Two circular-arc models

A *circular-arc (CA) model* $(C, \mathcal{A})$ is a circle C together with a collection $\mathcal{A}$ of arcs of C. Unless otherwise stated, we always traverse C in the clockwise direction. Each arc $A_i \in \mathcal{A}$ is written as $A_i = (s_i, t_i)$, where $s_i, t_i \in C$ are the *extreme points* of A_i, with s_i the *start point* and t_i the *end point* of the arc, respectively, in the clockwise direction. The *extremes* of $\mathcal{A}$ are those of all arcs $A_i \in \mathcal{A}$. As usual, we assume that no single arc of $\mathcal{A}$ covers C, that no two extremes of $\mathcal{A}$ coincide and that all arcs of $\mathcal{A}$ are open. When traversing C, we obtain a circular ordering of the extreme points of $\mathcal{A}$. Furthermore, we also consider a circular ordering $A_1, \ldots, A_n$ of the arcs of $\mathcal{A}$, defined by the corresponding circular ordering $s_1, \ldots, s_n$ of their respective start points. In general, when dealing with a sequence $x_1, \ldots, x_t$ of t objects circularly ordered, we assume that all the additions and subtractions of the indices i of the objects x_i are modulo t. Figure 1 illustrates two CA models, with the orderings of their arcs.



(a) (b)

Fig. 2. Two minimally non Helly models

In a model $(C, \mathcal{A})$, the *complement* of an arc $A_i = (s_i, t_i)$ is the arc $\overline{A_i} = (t_i, s_i)$. Complements of arcs have been employed before by McConnell [7], under the name *arc flippings*. The *complement* of $(C, \mathcal{A})$ is the model $(C, \overline{\mathcal{A}})$, where $\overline{\mathcal{A}} = \{\overline{A_i} | A_i \in \mathcal{A}\}$.

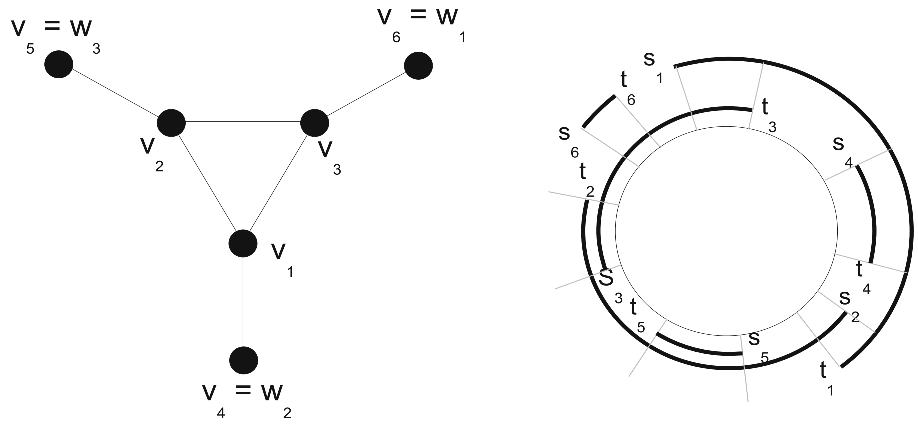
In the model $(C, \mathcal{A})$, a subfamily of arcs of $\mathcal{A}$ is *intersecting* when they pairwise intersect. Say that $\mathcal{A}$ is *Helly*, when every intersecting subfamily of it contains a common point of C. In this case, $(C, \mathcal{A})$ is a *Helly circular-arc (HCA) model*. When $\mathcal{A}$ is not Helly, it contains a minimal non Helly subfamily $\mathcal{A}'$, that is $\mathcal{A}'$ is not Helly, but $\mathcal{A}' \setminus A_i$ is so, for any $A_i \in \mathcal{A}'$. The model $(C, \mathcal{A}')$ is then *minimally non HCA*. Figure 2 depicts two minimally non Helly models.

A *circular-arc (CA) graph* G is the intersection graph of some CA model $(C, \mathcal{A})$. Denote by $v_i \in V_G$ the vertex of G corresponding to $A_i \in \mathcal{A}$. Similarly, a *Helly circular-arc (HCA) graph* is the intersection graph of some HCA model. In a HCA graph, each clique $Q \subseteq V_G$ can be represented by a point $q \in C$, which is common to all those arcs of $\mathcal{A}$, which correspond to the vertices of Q. Clearly, two distinct cliques must be represented by distinct points. Finally, two CA models are *equivalent* when they share the same intersection graph.

In the next section, we present the main basic concepts, in which the proposed characterizations are based. In Section 3, we characterize HCA models, while HCA graphs are characterized in Section 4. In Section 5, we describe the construction of a special CA model, which is employed in the recognition algorithm. Finally, Section 6 describes the recognition algorithm, together with its certificates. Withou loss of generality, we consider all given graphs to be connected.

2 Central Definitions

In this section, we describe usefull concepts for the proposed method. Let G be a graph and $(C, \mathcal{A})$ a CA model of it. First, define special sequences of extremes of the arcs of $\mathcal{A}$.



(a) (b)

Fig. 3. An obstacle and its non Helly stable model

An *s-sequence (t-sequence)* is a maximal sequence of start points (end points) of $\mathcal{A}$, in the circular ordering of C. Write *extreme sequence* to mean an s-sequence or t-sequence. The $2n$ start points and end points are then partitioned into s-sequences and t-sequences, which alternate in C. For an extreme sequence E, denote by $FIRST(E)$ the first element of E, while the notations $NEXT(E)$ and $NEXT^{-1}(E)$ represent the extreme sequences which succeeds and preceeds E in C, respectively. For an extreme point $p \in C$, denote $SEQUENCE(p)$ the extreme sequence which contains p, while $NEXT(p)$ means the sequence $NEXT(SEQUENCE(p))$. Through the paper, we employ operations on the CA models, which possibly modify them, while preserving equivalence. A simple example of such operations is to permute the extremes of the arcs, whitin a same extreme sequence.

Next, we define a special model of interest.

Definition 1. *Let s_i be a start point of $\mathcal{A}$ and $S = SEQUENCE(s_i)$. Say that s_i is stable when $i = j$ or $A_i \cap A_j = \emptyset$, for every $t_j \in NEXT^{-1}(S)$.*

Definition 2. *A model* $(C, \mathcal{A})$ *is* stable *when all its start points are stable.*

As examples, the models of Figures 1(a) and 1(b) are not stable, while that of Figure 3(b) is.

We will employ stable models in the recognition process of HCA graphs.

Finally, define a special family of graphs.

Definition 3. *An* obstacle *is a graph* H *containing a clique* $K_t \subseteq V_H$, $t \geq 3$, *whose vertices admit a circular ordering* $v_1, \ldots, v_t$, *such that each edge* $v_i v_{i+1}$, $i = 1, \ldots, t$, *satisfies:*

(i) $N(w_i) \cap K_t = K_t \setminus \{v_i, v_{i+1}\}$, *for some* $w_i \in V_H \setminus K_t$, *or*
(ii) $N(u_i) \cap K_t = K_t \setminus \{v_i\}$ *and* $N(z_i) \cap K_t = K_t \setminus \{v_{i+1}\}$, *for some adjacent vertices* $u_i, z_i \in V_H \setminus K_t$.

As example, the graph of Figure 3(a) is obstacle.

We will show that the obstacles form a family of forbidden induced subgraphs for a CA graph to be HCA.

3 Characterizing HCA Models

In this section, we describe a characterization and a recognition algorithm for HCA models. The characterization is as follows:

Theorem 1. *A CA model* $(C, \mathcal{A})$ *is HCA if and only if*

(i) if three arcs of $\mathcal{A}$ *cover* C *then two of these three arcs also cover it, and*
(ii) the intersection graph of $(C, \overline{\mathcal{A}})$ *is chordal.*

Proof. By hypothesis, $(C, \mathcal{A})$ is a HCA model. Condition (i) is clear, otherwise $(C, \mathcal{A})$ can not be HCA. Suppose Condition (ii) fails. Then the intersection graph G^c of $(C, \overline{\mathcal{A}})$ contains an induced cycle C^c, with length $k > 3$. Let $\overline{\mathcal{A}'} \subseteq \overline{\mathcal{A}}$ be the set of arcs of $\overline{\mathcal{A}}$, corresponding to the vertices of C^c, and $\mathcal{A}' \subseteq \mathcal{A}$ the sets of the complements of the arcs $\overline{A_i} \in \overline{\mathcal{A}'}$. First, observe that no two arcs of $\overline{\mathcal{A}'}$ cover the circle, otherwise C^c would contain a chord. Consequently, $\overline{\mathcal{A}'}$ consists of k arcs circularly ordered as $\overline{A_1}, \ldots, \overline{A_k}$ and satisfying: $\overline{A_i} \cap \overline{A_j} \neq \emptyset$ if and only if $\overline{A_i}, \overline{A_j}$ are consecutive in the circular ordering. In general, comparing a model $(C, \mathcal{A})$ to its complement model $(C, \overline{\mathcal{A}})$, we conclude that two arcs of $\mathcal{A}$ intersect if and only if their complements in $\overline{\mathcal{A}}$ are either disjoint or intersect without covering the circle. Consequently, $\mathcal{A}'$ must be an intersecting family. On the other hand, the arcs of $\mathcal{A}'$ can not have a common point $p \in C$. Because, otherwise $p \notin \overline{A_i}$, for all $\overline{A_i}$, meaning that the arcs of $\overline{\mathcal{A}'}$ do not cover the circle, contradicting C^c to be an induced cycle. The inexistence of a common point in $\mathcal{A}'$ implies that $\mathcal{A}$ is not a Helly family, a contradiction. Then (ii) holds. The converse is similar. $\triangle$

The following characterizes minimally non Hely models.

Corollary 1. *A model $(C, \mathcal{A})$ is minimaly non HCA if and only if*

(i) $\mathcal{A}$ is intersecting and covers C, and
(ii) two arcs of $\mathcal{A}$ cover C precisely when they are not consecutive in the circular ordering of $\mathcal{A}$.

Theorem 1 leads directly to a simple algorithm for recognizing Helly models, as follows. Given a model $(C, \mathcal{A})$ of some graph G, verify if $(C, \mathcal{A})$ satisfies Condition (i) and then if it satisfies Condition (ii). Clearly, $(C, \mathcal{A})$ is HCA if and only if both conditions are satisfied. Next, we describe methods for checking them.

For Condition (i), we seek directly for the existence of three arcs $A_i, A_j, A_k \in \mathcal{A}$ that cover C, two of them not covering it. Observe that there exist such arcs if and only if the circular ordering of their extremes is $s_i, t_k, s_j, t_i, s_k, t_j$. For each $A_i \in \mathcal{A}$, we repeat the following procedure, which looks for the other two arcs A_j, A_k whose extreme points satisfy this ordering. Let L_1 be the list of extreme points of the arcs contained in (s_i, t_i), in the ordering of C. First, remove from L_1 all pairs of extremes s_q, t_q of a same arc, which may possibly occur. Let L_2 be the list formed by the other extremes of the arcs represented in L_1. That is, $s_q \in L_1$ if and only if $t_q \in L_2$, and $t_q \in L_1$ if and only if $s_q \in L_2$, for any $A_q \in \mathcal{A}$. Clearly, the extremes points which form L_2 are all contained in (t_i, s_i), and we consider them in the circular ordering of C. Denote by $FIRST(L_1)$ and $LAST(L_2)$ the first and last extreme points of L_1 and L_2, in the considered orderings, respectively. Finally, iteratively perform the steps below, until either $L_1 = \emptyset$, or $FIRST(L_1) = t_k$ and $LAST(L_2) = t_j$, for some j, k.

if $FIRST(L_1)$ is a start point s_q then remove s_q from L_1 and t_q from L_2
if $LAST(L_2)$ is a start point s_q then remove s_q from L_2 and t_q from L_1

If the iterations terminate because $L_1 = \emptyset$ then there are no two arcs which together with A_i satisfy the above requirements, completing the computations relative to A_i. Otherwise, the arcs A_k and A_j, whose end points are $FIRST(L_1)$ and $LAST(L_2)$, form together with A_i a certificate for the failure of Condition (i). Each of the n lists L_2 needs to be sorted. There is no difficulty to sort them all together in time $O(m)$, at the beginning of the process. The computations relative to A_i require $O(d(v_i))$ steps. That is, the overall complexity of checking Condition (i) is $O(m)$.

For Condition (ii), the direct approach would be to construct the model $(C, \overline{\mathcal{A}})$, its intersection graph G^c and apply a chordal graph recognition algorithm to decide if G^c is chordal. However, the number of edges of G^c could be $O(n^2)$, breaking the linearity of the proposed method. Alternatively, we check whether the complement $\overline{G^c}$ of G^c is co-chordal. Observe that two vertices of $\overline{G^c}$ are adjacent if and only if their corresponding arcs in $\mathcal{A}$ cover the circle. Consequently, the number of edges of $\overline{G^c}$ is at most that of G, i.e. $\leq m$. Since co-chordal graphs can be recognized in linear time (Habib, McConnell, Paul and Viennot [4]), the complexity of the method for verifying Condition (ii) is $O(m)$.

Consequently, HCA models can be recognized in linear time.

4 Characterizing HCA Graphs

In this section, we describe the proposed characterizations for HCA graphs.

Theorem 2. *The following affirmative are equivalent for a CA graph G.*

(a) G is HCA.
(b) G does not contain obstacles, as induced subgraphs.
(c) All stable models of G are HCA.
(d) One stable model of G is HCA.

Proof. (a) $\Rightarrow$ (b): By hypothesis, G is HCA. Since HCA graphs are hereditary, it is sufficient to prove that no obstacle H is a HCA graph. By contrary, suppose H admits a HCA model $(C, \mathcal{A})$. Let K_t be the core of H. By Definition 3, there is a circular ordering $v_1, \ldots, v_t$ of the vertices of K_t which satisfies Conditions (i) or (ii) of it. Denote by $\mathcal{A}' = \{A_1, \ldots, A_t\} \subseteq \mathcal{A}$ the family of arcs corresponding to K_t. Define a clique C_i of H, for each $i = 1, \ldots, t$, as follows. If (i) of Definition 3 is satisfied then $C_i \supseteq \{w_i\} \cup K_t \setminus \{v_i, v_{i+1}\}$, otherwise (ii) is satisfied and $C_i \supseteq \{u_i, z_i\} \cup K_t\{v_i, v_{i+1}\}$. Clearly, all cliques $C_1, \ldots, C_t$ are distinct, because any two of them contain distinct subsets of K_t. Since H is HCA, there are distinct points $p_1, \ldots, p_t \in C$, representing $C_1, \ldots, C_t$, respectively. We know that $v_i \in C_j$ if and only if $i \neq j-1, j$. Consequently, $p_j \in A_i$ if and only if $i \neq j-1, j$. The latter implies that $p_1, \ldots, p_t$ are also in the circular ordering of C. On the other hand, because K_t is a clique distinct from any C_i, there is also a point $p \in C$ representing K_t. Try to locate p in C. Clearly, p lies between two consecutive points p_{i-1}, p_i. Examine the vertex $v_i \in K_t$ and its corresponding arc $A_i \in \mathcal{A}'$. We already know that $p \in A_i$, while $p_{i-1}, p_i \notin A_i$. Furthermore, because $t \geq 3$, there is $j \neq i-1, i$ such that $p_j \in A_i$. Such situation can not be realized by arc A_i. Then $(C, \mathcal{A})$ is not HCA, a contradiction.

(b) $\Rightarrow$ (c): By hypothesis, G does not contain obstacles. By contrary, suppose that there exists a stable model $(C, \mathcal{A})$ of G, which is not HCA. Let $\mathcal{A}' \subseteq \mathcal{A}$ be a minimally non Helly subfamily of $\mathcal{A}$. Denote by $A_1, \ldots, A_t$ the arcs of $\mathcal{A}'$ in the circular ordering. Their corresponding vertices in G are $v_1, \ldots, v_t$, forming a clique $K_t \subseteq V_G$. Let A_i, A_{i+1} be two consecutive arcs of $\mathcal{A}'$, in the circular ordering. By Corollary 1, A_i, A_{i+1} do not cover C. Denote $T = SEQUENCE(t_{i+1})$ and $S = SEQUENCE(s_i)$. Because $(C, \mathcal{A})$ is stable, $S \neq NEXT(T)$. Let $S' = NEXT(T)$ and $T' = NEXT^{-1}(S)$. Choose $s_z \in S$ and $t_u \in T'$. We know that A_z does not intersect A_{i+1}, nor does A_u intersect A_i, again because the model is stable. Since $s_z, t_u \in (t_{i+1}, s_i)$, Corollary 1 implies that $s_z, t_u \in A_j$, for any $A_j \in \mathcal{A}'$, $A_j \neq A_i, A_{i+1}$. Denote by z_i and u_i the vertices of G corresponding to A_z and A_u, respectively. Examine the following alternatives.

If z_i and v_i are not adjacent, rename z_i as w_i. Similarly, if u_i and v_{+1} are not adjacent, let w_i be the vertex u_i. In any of these two alternatives, it follows that $N(w_i) \cap K_t = K_t \setminus \{v_i, v_{i+1}\}$. The latter means that Condition (i) of Definition 3 holds. When none of the above alternatives occurs, the arcs A_z and A_u intersect, because s_z preceeds t_u in (t_{i+1}, s_i). That is, z_i and w_i are adjacent vertices satisfying $N(z_i) \cap K_t = K_t \setminus \{v_{i+1}\}$ and $N(u_i) \cap K_t = K_t \setminus \{v_i\}$. This corresponds to Condition (ii) of Definition 3. Consequently, for any pair of vertices $v_i, v_{i+1} \in K_t$ it is always possible to select a vertex $w_i \notin K_t$, or a pair of vertices $z_i, u_i \notin K_t$, so that Definition 3 is satisfied. That is, G contains an obstacle as an induced subgraph. This contradiction means all stable models of G are HCA.

The implications (c) $\Rightarrow$ (d) and (d) $\Rightarrow$ (a) are trivial, meaning that the proof is complete. $\triangle$

5 Constructing Stable Models

Motivated by the characterizations of HCA graphs in terms of stable models, described in the previous section, we present below an algorithm for constructing a stable model of a CA graph. Let $(C, \mathcal{A})$ be a CA model of some graph G, and $A_1, \ldots, A_n$ the circular ordering of the arcs of $\mathcal{A}$. Define the following expansion operations on the end points t_j and start points s_i of $\mathcal{A}$.

$expansion(t_j)$:
Examine the extremes points of $\mathcal{A}$, starting from t_j, in the clockwise direction, and choosing the closest start point s_i satisfying $i = j$ or $A_i \cap A_j = \emptyset$. Then move t_j so as to become the extreme point preceeding s_i in the model.

$expansion(s_i)$:
First, examine the extreme points of $\mathcal{A}$, starting from s_i, in the counterclockwise direction, and choosing the closest end point t_j satisfying $i = j$ or $A_i \cap A_j = \emptyset$. Let $T = SEQUENCE(t_j)$. Then move s_i counterclocwise towards T, transforming T into the sequences $T' s_i T''$, where $T' = \{t_j \in T | i = j$ or $A_i \cap A_j = \emptyset\}$ and $T'' = T \setminus T'$.

The following lemma asserts that the intersections of the arcs are preserved by these operations.

Lemma 1. *The operations $expansion(t_j)$ or $expansion(s_i)$ applied to a model $(C, \mathcal{A})$ construct models equivalent to $(C, \mathcal{A})$.*

We describe the following algorithm for finding a stable model of a given CA model, with end points t_j and start points s_i:

1. Perform $expansion(t_j)$, for $j = 1, \ldots, n$.
2. Perform $expansion(s_i)$, for $i = 1, \ldots, n$.

The correctness of this algorithm then follows from Lemma 1 and from the following theorem.

Theorem 3. *The model constructed by the above algorithm is stable.*

Proof. Let $(C, \mathcal{A})$ be a given CA model, input to the algorithm. We show that all its start points are stable, at the end of the process. After the completion of Step 1, we know that $s_i = FIRST(S)$ is already stable, for any s-sequence S. Otherwise, there would exist some end point $t_j \in NEXT^{-1}(S)$ satisfying

$i \neq j$ and $A_i \cap A_j \neq \emptyset$, meaning that t_j would have been moved after s_i in the clockwise direction, by $expansion(t_j)$.

Next, examine Step 2. Choose a start point s_i and follow the operation $expansion(s_i)$. If s_i is already stable, the algorithm does nothing. Suppose s_i is not stable. Let $S^* = SEQUENCE(s_i)$ and S the s-sequence closest to S^* in the counterclockwise direction, where $T = NEXT^{-1}(S)$ contains some t_j satisfying $i = j$ or $A_i \cap A_j = \emptyset$. Then $expansion(s_i)$ transforms T into the sequences $T's_iT''$, where $T' = \{t_j \in T | i = j \text{ or } A_i \cap A_j = \emptyset\}$ and $T'' = T \setminus T'$. Analyze the new sequences that have been formed. Clearly, $T' \neq \emptyset$, otherwise s_i would have been moved further from S^*. On the other hand, T'' could possibly be empty. However, the latter would only imply that T remains unchanged and that s_i has been incorporated to S. In any case, T' is the t-sequence which preceeds s_i. By the construction of T', it follows that s_i is now stable. In addition, previously stable start points of S remain so, because $T'' \subset T$. Furthermore, observe that $s_i \neq FIRST(S^*)$, because $FIRST(S^*)$ was before stable, whereas s_i was not. Consequently, S^* does not become empty by moving s_i out of it, implying that no parts of distinct t-sequences can be merged during the process. The latter assertion preserves the stability of the stable vertices belonging to the s-sequence which follows S^* in C. The remaining start points are not affected by $expansion(s_i)$. Consequently, s_i becomes now stable and all previousloy stable start points remain so. The algorithm is correct. $\triangle$

Corollary 2. *Every CA model admits an equivalent stable model.*

Next, we discuss the complexity of the algorithm. The number of extreme points examined during the operation $expansion(t_j)$ is at most $d(v_j) + 1$, since the operation stops at the first extreme t_i, such that either $i = j$ or $A_i \cap A_j = \emptyset$. Consequently, Step 1 requires $O(m)$ time. As for the operation $expansion(s_i)$, we divide it into two parts. First, for finding the required s-sequence S, the above argument applies, that is, $O(m)$ time suffices for all s_i. As for the determination of the sequences T' and T'', a straightforward implementation of it would consist of examining the entire t-sequence $T = T' \cup T''$, for each corresponding s_i, meaning $O(n^2)$ time, overall. However, a more elaborate implementation is possible, as follows.

To start, after the completion of Step 1, order the end points of each t-sequence T, in reverse ordering of their corresponding start points. That is, if $t_j, t_k \in T$ then in the clockwise direction, the extreme points of A_j and A_k appear as $\dots t_j \dots t_k \dots s_k \dots s_j \dots$. Such an ordering can be obtained in overall $O(n)$ time. With the end points so ordered, when traversing $T = NEXT^{-1}(S)$, for completing the operation $expansion(s_i)$, we can stop at the first $t_j \in T$ satisfying $i = j$ or $A_i \cap A_j = \emptyset$. In case the condition $i = j$ holds, we exchange in T, the positions of t_j and $FIRST(T)$. Afterwards, in any of the two alternatives, move s_i to the position just before t_j in the counterclockwise direction. We would need no more than additional $d(v_i) + 1$ steps for it, in the worst case. Consequently, $expansion(s_i)$ can be completed in $O(m)$ time, for all start points. Therefore the complexity of the algorithms is $O(m)$.

6 Recognition Algorithm for HCA Graphs

We are now ready to formulate the algorithm for recognizing HCA graphs. Let G be a graph.

1. Apply the algorithm [7] to recognize whether G is a CA graph. In the affirmative case, let $(C, \mathcal{A})$ be the model constructed by [7]. Otherwise terminate the algorithm (G is not HCA).
2. Transform $(C, \mathcal{A})$ into a stable model, applying the algorithm of Section 5.
3. Verify if $(C, \mathcal{A})$ is a HCA model, applying the algorithm of Section 3. Then terminate the algorithm (G is HCA if $(C, \mathcal{A})$ is HCA, and otherwise G is not HCA).

The correctness of the algorithm follows directly from Theorems 1, 2 and 3.

Each of the above steps can be implemented in $O(m)$ time. The complexity of the algorithm is $O(m)$.

The algorithm constructs a HCA model of the input graph G, in case G is HCA. If G is CA but not HCA, we can exhibit a certificate of this fact, by showing a forbidden subgraph of G, that is, an obstacle. In order to construct the obstacle, we may need certificates of non co-chordality. There is no difficulty to modify the algorithm [4] so as to produce such certificates. The entire process can also be implemented in linear time.

References

1. X. Deng and P. Hell and J. Huang, Linear time representation algorithms for proper circular-arc graphs and proper interval graphs, *SIAM J. Computing*, **25** (1996), pp. 390-403.
2. F. Gavril, Algorithms on circular-arc graphs, *Networks* **4** (1974), pp. 357-369.
3. M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, 1980, 2nd ed. 2004.
4. M. Habib, R. M. McConnell, C. Paul, and L. Viennot, Lex-bfs and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing, *Theoretical Computer Science*, **234** (2000), pp. 59-84.
5. H. Kaplan and Y. Nussbaum, A Simpler Linear-Time Recognition of Circular-Arc Graphs, accepted for publication in *10th Scandinavian Workshop on Algorithm Theory* (2006).
6. M. C. Lin and J. L. Szwarcfiter, Efficient Construction of Unit Circular-Arc Models, *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms* (2006), pp. 309-315.
7. R. M. McConnell, Linear-time recognition of circular-arc graphs, *Algorithmica* **37 (2)** (2003), pp. 93-147.
8. J. Spinrad, Efficient Graph Representations, *American Mathematical Society* (2003).
9. A. Tucker, Characterizing circular-arc graphs, *Bull. American Mathematical Society* **76** (1970), pp. 1257-1260.