JAIST Repository
https://dspace.jaist.ac.jp/

Title	区間二部グラフの効率の良い認識に関する研究
Author(s)	栗林，康之
Citation	
Issue Date	2010-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/8964
Rights	
Description	Supervisor:上原隆平，情報科学研究科，修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修 士 論 文

区間二部グラフの効率の良い認識に関する研究

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

栗林 康之

2010 年 3 月

修 士 論 文

区間二部グラフの効率の良い認識に関する研究

指導教官　上原 隆平 准教授

審査委員主査　上原 隆平 准教授
審査委員　浅野 哲夫 教授
審査委員　平石 邦彦 教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

0710025 栗林 康之

提出年月: 2010 年 2 月

Copyright © 2010 by Yasuyuki Kuribayashi

概 要

区間二部グラフとは，２種類の（数直線上の）区間の集合上で定義される交差グラフで，頂点が隣接するための必要十分条件は対応する２つの区間がそれぞれ異なる集合に属し，かつそれらが重なりを持つときである．現在，区間二部グラフを多項式時間で認識するアルゴリズムが知られているが，その実行時間は$O(n^5m^6\log n)$であり，実行効率は良いとは言えない．しかし，その後区間二部グラフの非常に単純な特徴づけがHell，Huangによって与えられた．その特徴づけとは，あるグラフが区間二部グラフであれば，そのグラフの補グラフが円弧グラフと呼ばれる違うグラフクラスに属している，というものである．

本研究では，区間二部グラフの認識アルゴリズムへの足がかりとして，真区間二部グラフの認識を行う．真区間二部グラフは区間二部グラフの部分クラスであり，区間二部グラフよりも良い特徴を持っている．そのため，区間二部グラフより真区間二部グラフのほうがアルゴリズムの開発が容易であることが予想される．Hell，Huangの論文では，真区間二部グラフに対しても，補グラフによる特徴づけが存在することが示された．本研究では，この補グラフによる特徴づけを用いて，真区間二部グラフの認識をする多項式時間アルゴリズムを提案する．

目 次

第 1 章　序論 **1**
1.1　背景と目的 1
1.2　本論文の構成 2

第 2 章　準備 **3**
2.1　基本用語 3
2.1.1　グラフとは 3
2.1.2　補グラフ 5
2.1.3　二部グラフ 5
2.1.4　誘導部分グラフ 6
2.1.5　クリーク 6
2.1.6　クリークカバーとクリークカバー数 7
2.1.7　隣接点集合 7
2.1.8　twin 8
2.1.9　縮約 8
2.1.10　prefix tree 9

第 3 章　グラフクラス **10**
3.1　区間二部グラフと真区間二部グラフ 10
3.1.1　区間二部グラフ 10
3.1.2　真区間二部グラフ 11
3.1.3　チェーングラフ 11
3.2　円弧グラフと真円弧グラフ 12
3.2.1　円弧グラフ 12
3.2.2　真円弧グラフ 13

第 4 章　新しい特徴づけ **14**
4.1　区間二部グラフの特徴づけ 14
4.2　真区間二部グラフの特徴づけ 15

第 5 章　真区間二部グラフの認識アルゴリズム **17**
5.1　問題の定義 17

5.2　認識アルゴリズム 17

第 6 章　まとめ **23**

第1章　序論

1.1　背景と目的

計算機で扱う多くの問題は，グラフ構造でモデル化することができる．こうした問題を効率よく解くには，グラフ理論と，アルゴリズム理論がともに重要な役割を果たす．グラフの認識問題は，アルゴリズム理論とグラフ理論に深く関係する問題の中でも，基本的な問題の一つである．

グラフクラス C の認識問題とは，あるグラフが与えられた時にそのグラフがグラフクラス C に属するかどうかを判定する問題である．グラフの認識問題の困難さは，グラフクラスの包含関係と無関係である．これまでに区間グラフや弦グラフなどのグラフクラスについて認識問題を解く高速なアルゴリズムが開発された [1]．本研究では，区間二部グラフの認識問題について扱う．

区間二部グラフとは，2 種類の (数直線上の) 区間の集合上で定義される交差グラフで，頂点が隣接するための必要十分条件は対応する区間が異なる集合に属し，かつそれらが重なりを持つときである．区間二部グラフは 1980 年代初頭に Harary, Kabell, McMorris によって導入された [2]．しかし，1997 年に元の論文の特徴づけに間違いがあることが指摘され，さらに区間二部グラフが多項式時間で認識できることが示された [3]．この認識アルゴリズムの計算時間は $O(nm^6(n+m)\log n)$ であった．しかしこの論文にも間違いがあることがわかり，修正版が Web 上で公開されている [4]．その Web 上で公開されている正しい認識アルゴリズムの計算時間は $O(n^5m^6\log n)$ である．

近年，区間二部グラフの非常に単純な特徴づけが与えられた [5]．その特徴づけとは, あるグラフが区間二部グラフであることの必要十分条件は，そのグラフの補グラフが，円弧グラフと呼ばれる違うグラフクラスに属していることである，というものである．この補グラフによる特徴づけは非常に優れたアイデアであった．しかし，これはグラフ理論的な結果であり，これに基づいたアルゴリズムは知られていない．区間二部グラフは，自然なグラフのモデルであるが，グラフアルゴリズムの観点からはあまり研究されているとはいえない．この [5] の特徴づけを利用すれば，区間二部グラフの既存の認識アルゴリズムを改善できると予想できる．

本研究では，区間二部グラフの認識アルゴリズムへの足がかりとして，真区間二部グラフの認識を行う．真区間二部グラフは区間二部グラフの部分クラスで，区間二部グラフよ

りも良い特徴を持っている．そのため，区間二部グラフより真区間二部グラフのほうがアルゴリズムの開発が容易であることが予想される．

真区間二部グラフの認識アルゴリズムについては別の特徴づけに基づく線形時間アルゴリズムが存在する [6]．しかし，それらのアルゴリズムを一般の区間二部グラフに拡張することは難しい．[5] の結果によると，真区間二部グラフに対しても, 補グラフによる特徴づけが存在する．本研究では, この補グラフによる特徴づけを用いて，真区間二部グラフの認識をする．

1.2 本論文の構成

本稿では，第 2 章をグラフの基本的な用語の定義，第 3 章をグラフクラスの定義，第 4 章を新しい特徴づけ，5 章を真区間二部グラフの認識アルゴリズムとし，第 6 章をまとめとする．

第2章　準備

本章では，グラフに関する基本的な用語と定義の説明をする．

2.1　基本用語

2.1.1　グラフとは

グラフ $G = (V, E)$ は**頂点**の集合 V と**辺**の集合 E からなる．E の要素は V の要素の 2 つ組である．辺のそれぞれの要素を**端点**と呼ぶ．頂点の数を $n = |V|$ とし，辺の数を $m = |E|$ とする．辺 $e \in E$ の端点が頂点 $v_i, v_j \in V$ であるとき，$e = \{v_i, v_j\}$ または $e = \{v_j, v_i\}$ と書く．このとき，v_i と v_j は**隣接**しているという．例えば，図 2.1 のグラフ G は 6 組の頂点 $v_1, \ldots, v_6$ から構成される．辺集合 $E = \{e_1, \ldots, e_8\}$ であり，$e_1 = \{v_4, v_5\}$,$e_2 = \{v_3, v_4\}$ である．

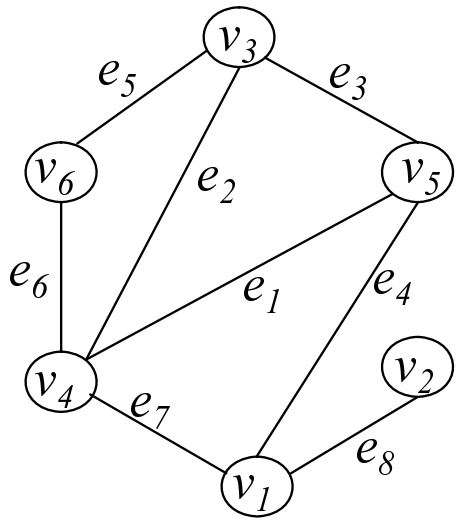

図 2.1: グラフ

グラフ $G = (V, E)$ において，$i = 1, 2, \ldots, l$ に対して $\{v_{i-1}, v_i\} \in E$ であるとき，頂点の列 $[v_0, v_1, v_2, \ldots, v_l]$ は v_0 から v_l への長さ l の**パス**であるという．$i = 1, 2, \ldots, l$ に対して $\{v_{i-1}, v_i\} \in E$ で，かつ $\{v_l, v_0\} \in E$ であるとき，頂点の列 $[v_0, v_1, v_2, \ldots, v_l, v_0]$ を長さ $l+1$ の**閉路**と呼ぶ．また，$i, j\{i \neq j\}$ に対して $v_i \neq v_j$ であるとき，閉路 $[v_0, v_1, v_2, \ldots, v_l, v_0]$ は**単純**であるという．図 2.2 において，頂点の列 $[v_1, v_3, v_5, v_6, v_4, v_3, v_5 v_7]$ は長さ 7 のパスである．また，$[v_1, v_3, v_5, v_2, v_1]$ は長さ 4 の単純閉路である．

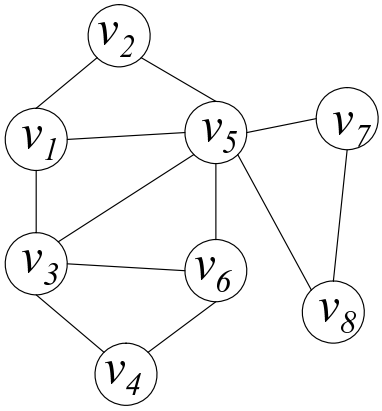

図 2.2: グラフ G

グラフ $G = (V, E)$ 上の任意の相異なる 2 頂点間にパスが存在するとき，そのグラフは**連結**であるという．図 2.1，図 2.2 は連結なグラフである．

グラフ $T = (V, E)$ が連結で，かつ単純閉路を持たないとき，T は**木**であるいう．図 2.3 は木の例である．

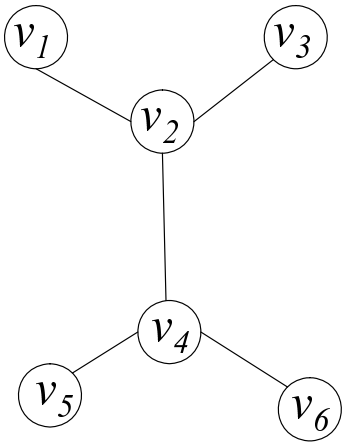

図 2.3: 木の例

2.1.2 補グラフ

グラフ $G = (V, E)$ の**補グラフ**$\bar{G} = (V, \bar{E})$ は，辺集合が $\bar{E} = \{\{x, y\} : x, y \in V, x \neq y, \{x, y\} \notin E\}$ で定義されるグラフである．図 2.4，図 2.5 を参照.

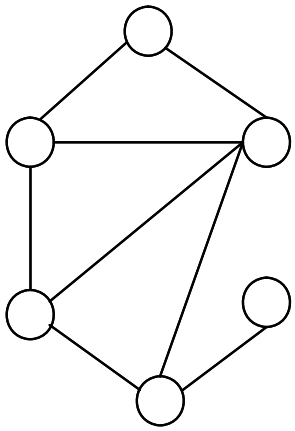

図 2.4: グラフ G

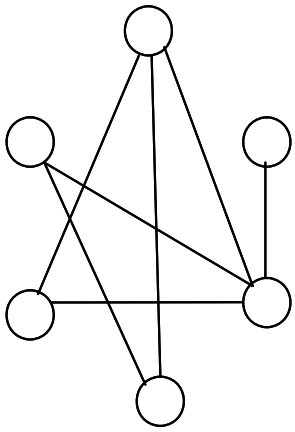

図 2.5: グラフ G の補グラフ

2.1.3 二部グラフ

グラフ G の頂点集合を 2 つの互いに素な集合 X,Y に分割し，G のすべての辺が X の頂点と Y の頂点を結ぶようにできるとき，$G = (V, E)$ は**二部グラフ**という．図 2.6 を参照.

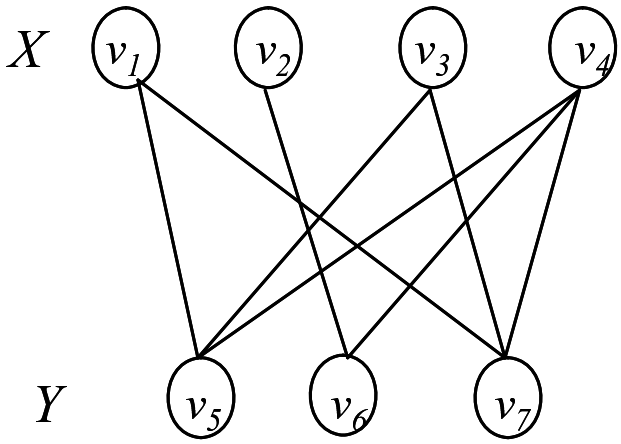

図 2.6: 二部グラフ

2.1.4 誘導部分グラフ

グラフ $G = (V, E)$ が与えられたとき，V の部分集合を U とし，E の部分集合を $E' = \{\{u, v\} \in E \mid u \in U$ かつ $v \in U\}$ とする．このとき，グラフ $G[U] = (U, E')$ を U による G の**誘導部分グラフ**という．図 2.7，図 2.8 を参照.

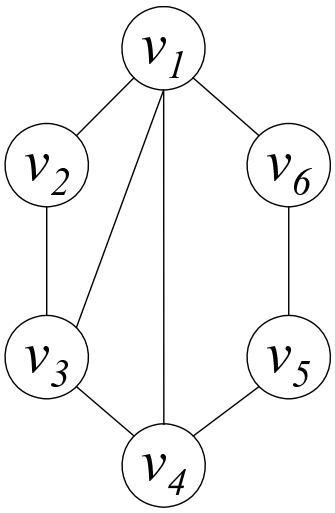

図 2.7: グラフ G

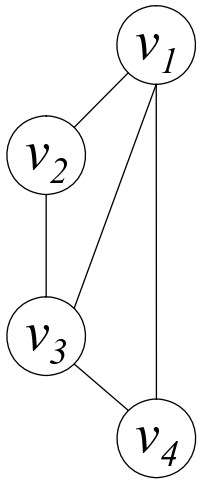

図 2.8: $\{v_1, v_2, v_3, v_4\}$ によって誘導された誘導部分グラフ

2.1.5 クリーク

グラフ $G = (V, E)$ において，頂点集合 $W \subset V$ に属する任意の 2 つの頂点が隣接するとき，W は**クリーク**であるという．図 2.9 を参照.

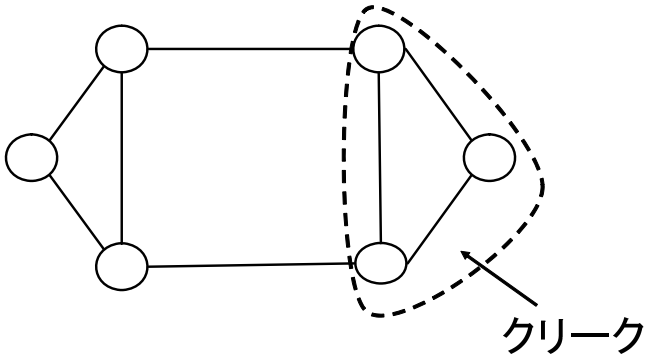

図 2.9: クリーク

2.1.6 クリークカバーとクリークカバー数

グラフ G の k-**クリークカバー**とは，G の頂点集合 V の k 個の分割 $V = [V_1 \cup V_2 \cup V_3 \cup \cdots \cup V_k], V_i \cap V_j = \phi (i \neq j)$ である．ただし，$V_i (i = 1, \ldots, k)$ はクリークである．また，グラフ G の $k-$ クリークカバーを与えるような k の最小値のことを**クリークカバー数**という．特にクリークカバー数 2 のグラフを **2-クリーク**という．

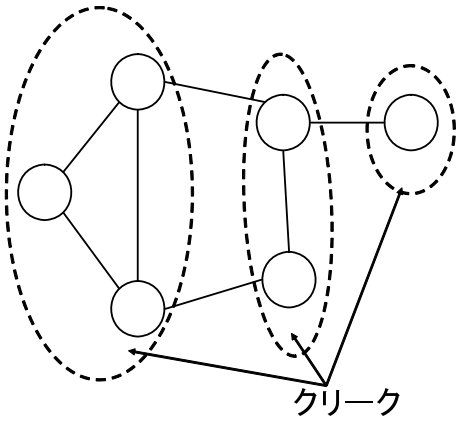

図 2.10: クリークカバー数が 3 の例

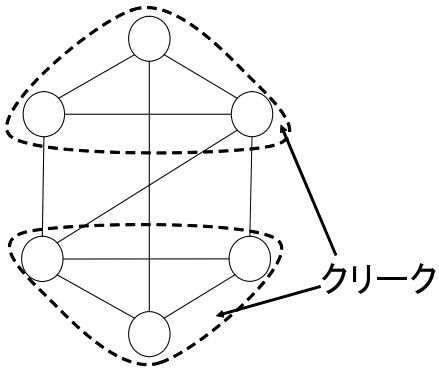

図 2.11: 2-クリーク

2.1.7 隣接点集合

グラフ $G = (V, E)$ における頂点 $v \in V$ の**隣接点集合**は $N(v) = \{u \in V \mid \{u, v\} \in E\}$ とする．また，頂点集合 $V_1 \subset V$ に対応する隣接点集合は $N(V_1) = \{v \in V \mid \exists u \in V_1, \{u, v\} \in E\}$ とする．例として，下図において頂点 v_1 の隣接点集合は $N(v_1) = \{v_2, v_3, v_4, v_5\}$，頂点集合 $\{v_1, v_6\}$ の隣接点集合は $N(\{v_1, v_6\}) = \{v_2, v_3, v_4, v_5, v_7, v_8\}$ である．

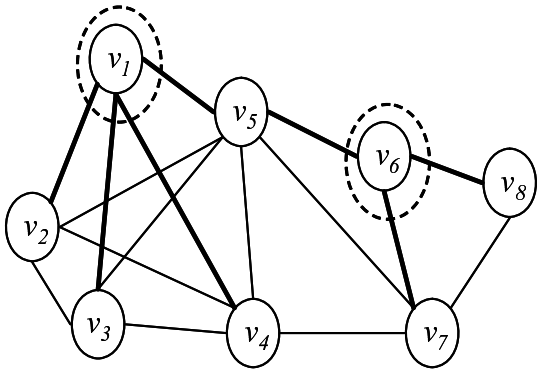

図 2.12: 隣接点集合の例

2.1.8 twin

グラフ $G = (V, E)$ のある頂点 $u, v \in V$ が **twin** であるとは，$N(u)$ と $N(v)$ が等しいことである．図 2.13 の v_1 と v_3 は twin である．

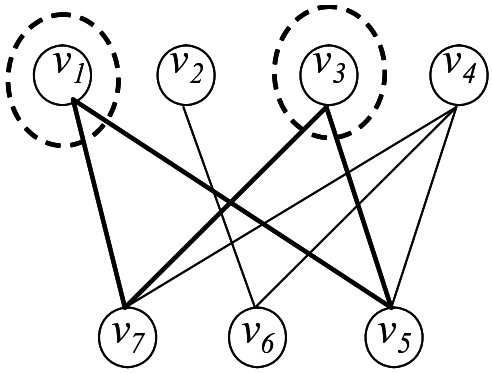

図 2.13: twin

2.1.9 縮約

x, y をグラフ $G = (V, E)$ の相異なる 2 頂点とする．グラフ G の x, y に関する**縮約**とは，頂点集合 $V' = (V \backslash \{x, y\}) \cup \{z\}$(ただし，$z \notin V$ である) と次の辺集合 E' を持つグラフ (V', E') である．

$E' = \{\{v, w\} \in E \mid \{v, w\} \cap \{x, y\} = \phi\} \cup \{\{z, w\} \mid \{x, w\} \in E \backslash \{x, y\}$ または $\{y, w\} \in E \backslash \{x, y\}\}$

図 2.14 に縮約の例を示す．

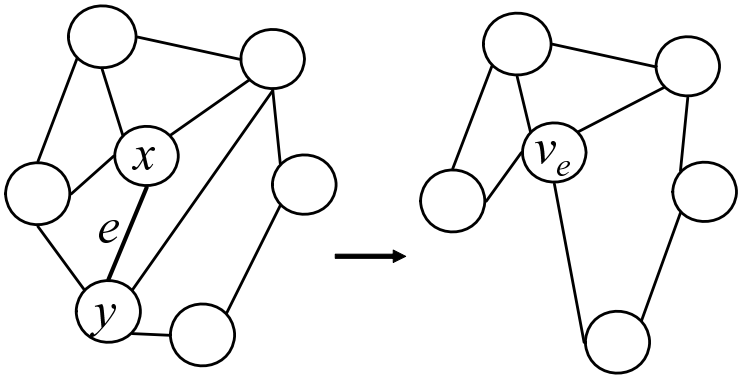

図 2.14: 辺 $e = \{x, y\}$ の縮約

2.1.10 prefix tree

prefix tree とは，順序付き木構造 (データ構造) の一種である．キーが文字列である連想配列の実装構造として使われる．各頂点に個々のキーが格納されるのではなく，木構造上の頂点の位置とキーが対応している．ある頂点の配下の全頂点は, 自身に対応する文字列に共通するプレフィックス（接頭部）があり，根には空の文字列が対応している．値は一般に全頂点に対応して存在するわけではなく，末端の頂点や一部の中間の頂点だけがキーに対応した値を格納している．

prefix tree にもとづいた [7] のアルゴリズムを用いると，twin を線形時間で見つけられることが知られている．

第3章　グラフクラス

3.1　区間二部グラフと真区間二部グラフ

3.1.1　区間二部グラフ

区間二部グラフは1980年代初頭にHarary, Kabell, McMorrisによって導入された[5]. 区間二部グラフの定義は以下の通りである.

定義 1. $B = (X, Y, E)$ を二部グラフとする．以下を満たすような区間集合 $\mathcal{I} = \{I_1, \ldots, I_r\}$ $(r = |X|)$ と $\mathcal{J} = \{J_1, \ldots, J_s\}$ $(s = |Y|)$ が存在するとき，B は**区間二部グラフ**という．B の頂点 $x_i \in X$ は区間 $\mathcal{I}$ に対応し，頂点 $y_j \in Y$ は区間 $\mathcal{J}$ に対応し，

$$\{x_i, y_j\} \in E \Leftrightarrow I_i \cap J_j \neq \phi \quad \text{ただし}, i \in \{1, \ldots, r\}, j \in \{1, \ldots, s\}$$

このような，区間の集合 $(\mathcal{I} \cup \mathcal{J})$ を B の**区間表現**という．図3.1は,$X = \{v_1, v_3, v_5, v_7, v_9\}$,$Y = \{v_2, v_4, v_6, v_8\}$ で, 図3.2の区間表現において区間 I_{v_4} と区間 I_{v_5} には重なりがあり，対応する区間二部グラフの頂点 v_4，頂点 v_5 の間には辺がある．また, 同じ頂点集合 X の二つの要素に対応する区間に重なりがあったとしても，対応する頂点間には辺がない．頂点集合 Y についても同様である．

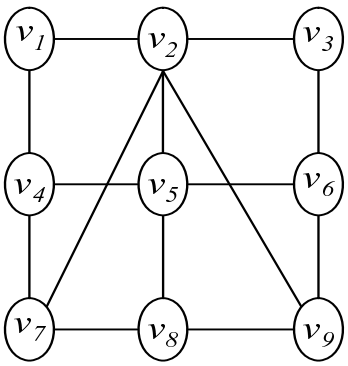

図 3.1: 区間二部グラフ

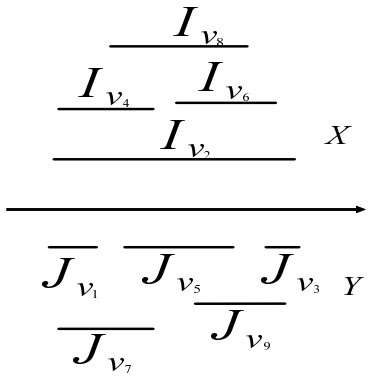

図 3.2: 区間表現

3.1.2 真区間二部グラフ

真区間二部グラフは区間二部グラフの部分クラスである．

真区間二部グラフは，どの区間も別の区間に真に含まれることはないという区間表現を持つ区間二部グラフのことである．

図 3.3, 図 3.4 は真区間二部グラフとそれに対応する区間表現である．図 3.4 の区間表現において，それぞれの区間は別の区間を真に含んではいない．

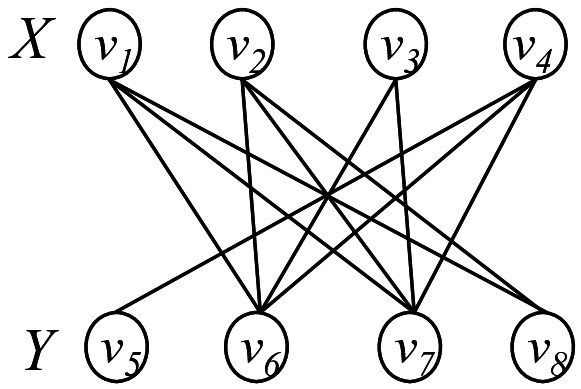

図 3.3: 真区間二部グラフ

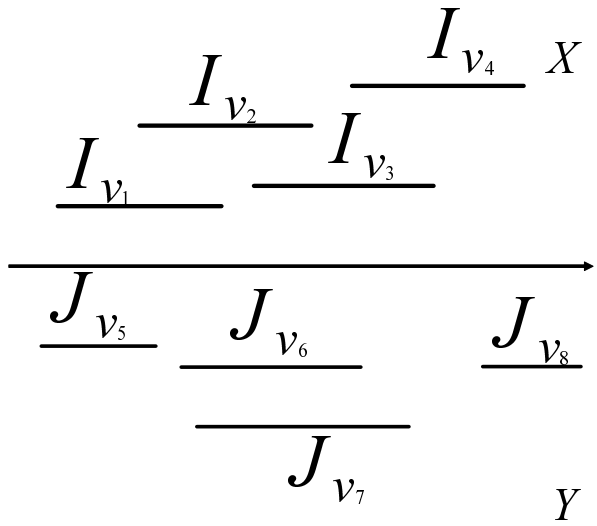

図 3.4: 区間表現

3.1.3 チェーングラフ

X の順序が **adjacency property** を持つことの必要十分条件は，任意の $y \in Y$ について $N(y)$ が X の順序上で連続していることである．二部グラフ $G = (X, Y, E)$ が**チェーングラフ**であることの必要十分条件は，adjacency property を満たす X と Y の順序が存在し，かつ X の順序が $N(x_n) \subseteq N(x_{n-1}) \subseteq \cdots \subseteq N(x_2) \subseteq N(x_1)$ を満たすことである．図 3.5 はチェーングラフの例である．

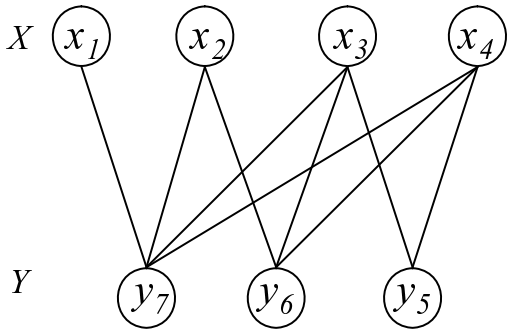

図 3.5: チェーングラフ

3.2 円弧グラフと真円弧グラフ

3.2.1 円弧グラフ

円弧グラフは以下の定義で与えられる．

定義 2. 円弧グラフとは，各頂点が円周上の互いに異なる円弧に対応し，2 つの円弧が重なっているとき，またそのときに限り，対応する 2 つの頂点は辺で結ばれるという弧の集合を持つグラフである．

弧の集合を**円弧表現**と呼ぶ．図 3.6，図 3.7 は円弧グラフと対応する円弧表現である．図 3.7 の円弧表現において弧 A_3 と弧 A_4 には重なりがあり，対応する円弧グラフの頂点 3，頂点 4 の間には辺がある．また，弧 A_1 と弧 A_5 には重なりがなく，対応する円弧グラフの頂点 1，頂点 5 には辺がない．

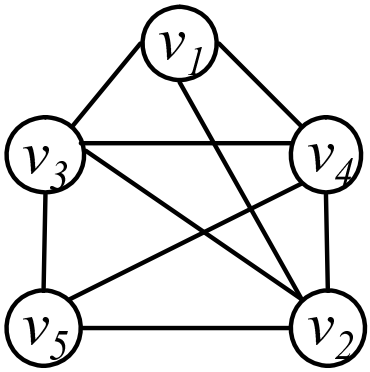

図 3.6: 円弧グラフ

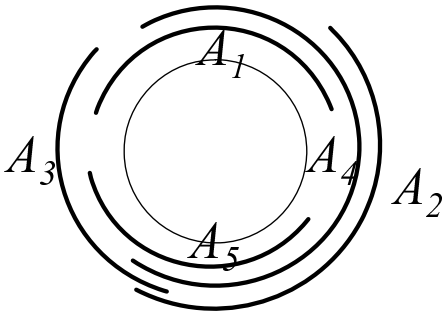

図 3.7: 円弧表現

3.2.2 真円弧グラフ

真円弧グラフは，どの弧も別の弧に真に含まれることはないという円弧表現を持つ円弧グラフのことである．

図 3.8，3.9 は真円弧グラフとそれに対応する円弧表現である．図 3.9 の円弧表現において，それぞれの円弧は別の円弧を真に含んではいない．

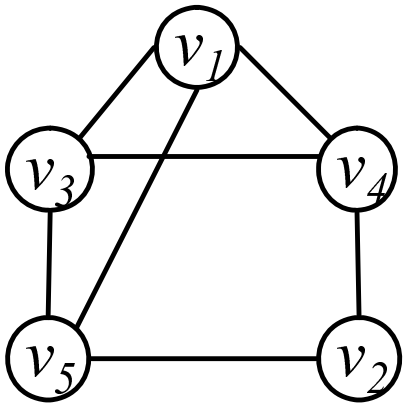

図 3.8: 真円弧グラフ

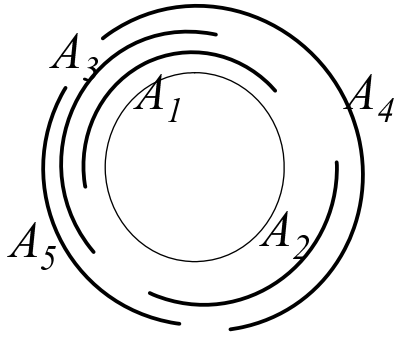

図 3.9: 円弧表現

第4章　新しい特徴づけ

本章の最初に，二部グラフの補グラフについて述べる．

二部グラフ $G = (X, Y, E)$ の頂点集合 X,Y は，どの二頂点間にも辺がない．よって G の補グラフ $\bar{G}$ の頂点集合 X,Y はそれぞれクリークである．つまり，$\bar{G}$ は 2-クリークグラフである．図 4.1 に例を示す．図 4.1 の左のグラフは二部グラフ G の例である．図 4.1 の右のグラフは $\bar{G}$ の例である．

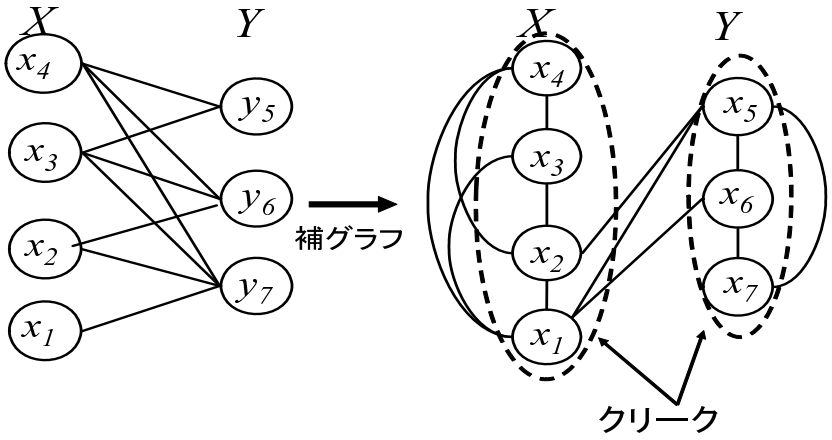

図 4.1: 二部グラフの補グラフの性質についての例

4.1　区間二部グラフの特徴づけ

区間二部グラフに対する [5] の特徴づけとは以下である．

補題 1. 二部グラフ $G = (X, Y, E)$ が区間二部グラフであることの必要十分条件は $\bar{G}$ が以下を満たす円弧表現を持つ円弧グラフであること．

1. どの二つの弧も円全体をカバーしない．

2. X の要素に対応する弧がすべて通り，
   Y の要素に対応する弧が１つも通らない点 p と，
   Y の要素に対応する弧がすべて通り，
   X の要素に対応する弧が１つも通らない点 q がある．

図 4.2，図 4.3 はそれぞれ，区間二部グラフ G と $\bar{G}$ に対応する円弧表現である．図 4.3 の区間表現の例では，弧集合 $X = \{x_1, x_2, x_3\}$ は点 p を含み，点 q を含まず，弧集合 $Y = \{y_4, y_5, y_6\}$ は点 q を含み，点 p を含んでいない．また，どの二つの弧を見ても円全体をカバーしていない．

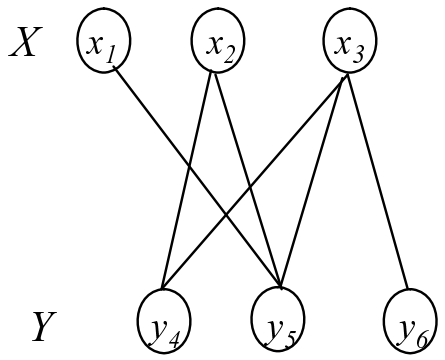

図 4.2: 区間二部グラフ G

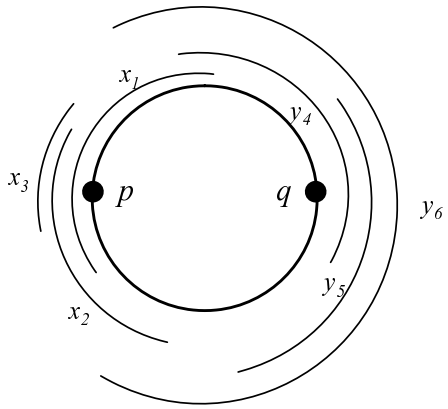

図 4.3: 補グラフ $\bar{G}$ の円弧表現

4.2 真区間二部グラフの特徴づけ

真区間二部グラフに対する [5] の特徴づけとは以下である．

補題 2. 二部グラフ $G = (X, Y, E)$ が真区間二部グラフであることの必要十分条件は $\bar{G}$ が以下を満たす真円弧表現を持つ円弧グラフであること．

1. どの二つの弧も円全体をカバーしない．

2. X の要素に対応する弧がすべて通り,
   Y の要素に対応する弧が１つも通らない点 p と,
   Y の要素に対応する弧がすべて通り,
   X の要素に対応する弧が１つも通らない点 q がある.
3. 各円弧が互いを真に含むことはない.

図 4.4, 図 4.5 はそれぞれ, 真区間二部グラフ G と $\bar{G}$ に対応する円弧表現である.

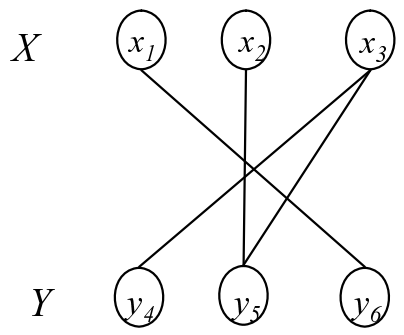

図 4.4: 真区間二部グラフ G

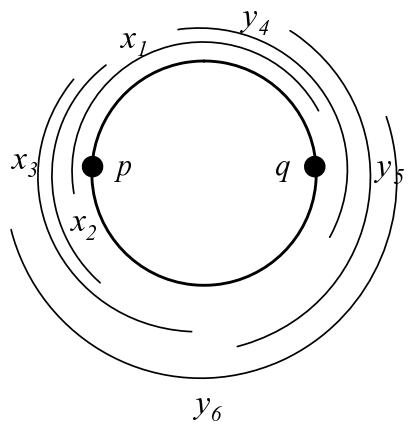

図 4.5: 補グラフ $\bar{G}$ の円弧表現

第5章　真区間二部グラフの認識アルゴリズム

5.1　問題の定義

真区間二部グラフの認識問題を以下のように定義する．

定義 3. 真区間二部グラフの認識問題
入力：二部グラフ $G = (X, Y, E)$
質問：グラフ G が真区間二部グラフかどうか？

5.2　認識アルゴリズム

提案アルゴリズムは，補題 4.1 の真区間二部グラフの補グラフによる特徴づけを用いる．つまり，与えられた二部グラフ G の補グラフを構成し，そのグラフ真円弧グラフかどうかを真円弧表現が作れるかどうかで，判定する．最初に，区間二部グラフの認識アルゴリズムの概要を示す．

1. 二部グラフ G の twin の関係にある頂点集合を 1 頂点に縮約したグラフを G' とする．
2. G' がチェーングラフであるかどうかを線形時間でチェックする．
3. G' の補グラフ $\bar{G}$ を構成する．
4. $\bar{G}$ の真円弧表現を作る．

各ステップについて述べる．1. について，今回提案するアルゴリズムはアルゴリズムを簡単にするために，twin の頂点を一つに縮約する．縮約された頂点は真円弧表現において，それらを同一の弧として持つ真円弧表現が存在するので twin の頂点集合を一頂点として考えることができる．twin を見つけるのに prefix tree を使うと，線形時間で見つけることができる．図 5.1 は twin の関係にある頂点集合を一頂点に縮約する例である．

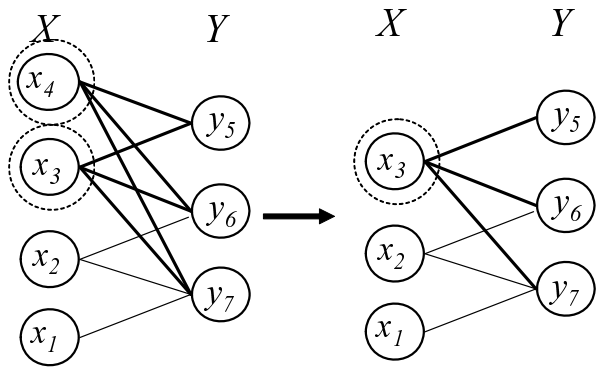

図 5.1: twin の関係にある頂点集合を一頂点に縮約

2. について述べる．グラフ G がチェーングラフであることと，これが [7] において提案された levelwise laminar ordering で 2 レベルしか持たないことは同値である．したがって，[7] の levelwise laminar ordering をチェックするアルゴリズムを使えば G' がチェーングラフであるかどうかを線形時間で判定できる．1.，2. より，3. 以降では，G は twin を持たず，チェーングラフでもないと仮定できる．

4. では，G' の真円弧表現を作る．4. のアルゴリズムを以下に示す．

4-(1) 最初に次の条件を満たす $x_{i_1}, x_{j_1} \in X$ を選ぶ．

$$N(x_{i_1}) \backslash N(x_{j_1}) \neq \phi \text{ かつ } N(x_{j_1}) \backslash N(x_{i_1}) \neq \phi$$

アルゴリズムの 1. より，グラフ $\bar{G} = (V', \bar{E})$ に対して，任意の 2 頂点 $v_{i_1}, v_{j_1} \in V'$ は，$N(v_{i_1}) \neq N(v_{j_1})$ である．また，G' はチェーングラフでないことから，こうした x_{i_1},x_{j_1} は存在する．$Y_{t_1} = N(x_{i_1}) \backslash N(x_{j_1}), Y_{t_1} = N(x_{j_1}) \backslash N(x_{i_1})$ とする．

真円弧表現について考える．補題 4.1 より真円弧表現において，頂点 $x \in X$ は点 p を通り，q を通らず，頂点 $y \in Y$ は点 q を通り，p を通らないという，2 点 p, q が円周上に存在する．p から q への時計周りの半弧を *top* とし，q から p への時計周りの半弧を *bottom* とする．ここで，円弧表現において，x_{i_1} と Y_{t_1} は *top* で重なるとする．このとき，Y_{b_1} は x_{i_1} と重なりを持たず，また x_{j_1} と Y_{t_1} は *top* で重なりを持たないので，x_{j_1} と Y_{b_1} は *bottom* で重なりを持つ．図 5.2 に例を示す．

4-(2) 適当な $y_{i_1} \in Y_{t_1}, y_{j_1} \in Y_{b_1}$ を選ぶ．

ここで，$X_{t_1}, X_{b_1} \subset X$ を $X_{t_1} = N(y_{i_1}) \backslash N(y_{j_1}), X_{b_1} = N(y_{j_1}) \backslash N(y_{i_1})$ とする．このとき，y_{i_1} と X_{b_1} は重なりを持たない．また，y_{j_1} と X_{t_1} は重なりを持たない．そのため，円弧表現において，X_{t_1} は y_{i_1} と *top* で重なりを持ち，X_{b_1} は y_{j_1} と *bottom* で重なりを持つ (図 5.3 参照)．ここで $x_{i_1} \in X_{t_1}, x_{i_2} \in X_{b_1}$ に注意する．

4-(3) *top* に現れる X_{t_1} の端点の位置を決める．

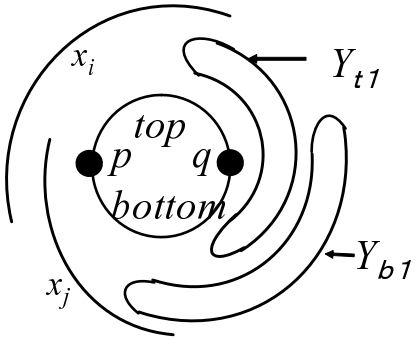

図 5.2: 円弧表現

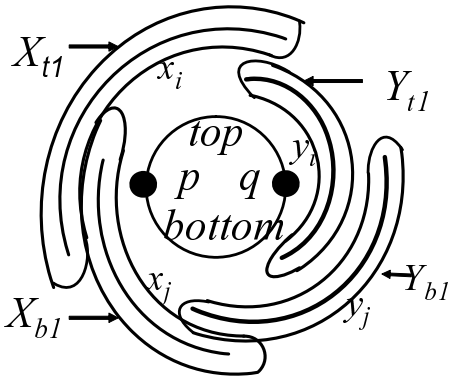

図 5.3: 円弧表現

任意の 2 頂点 $x_{t_1}, x_{t_2} \in X_{t_1}$ に対し，次の 3 つの場合分けによって，x_{t_1} と x_{t_2} のどちらの端点が q に近いかを決める．

case(a) $|N(x_{t_1}) \cap Y_{t_1}| \neq |N(x_{t_2}) \cap Y_{t_1}|$

$|N(x_{t_1}) \cap Y_{t_1}|$ と $|N(x_{t_2}) \cap Y_{t_1}|$ を比べる．

頂点集合 $X_{t_1} = N(y_{i_1}) \backslash N(y_{j_1})$ なので，y_{j_1} と X_{t_1} の各頂点は隣接していない．X_{t_1} の頂点が Y_{t_1} の頂点と隣接しているときは, 真円弧表現において対応する弧は *top* で重なりがある．そのため，$|N(x_{t_1}) \cap Y_{t_1}|$ が大きいとき，x_{t_1} の *top* の弧の端点は x_{t_2} の弧のよりも, 点 q に近づける．同様に，$|N(x_{t_2}) \cap Y_{t_1}|$ が大きいとき，x_{t_2} の *top* の弧の端点は x_{t_1} の弧のよりも, 点 q に近づける．

case(b) $|N(x_{t_1}) \cap Y_{t_1}| = |N(x_{t_2}) \cap Y_{t_1}|$ かつ
$|N(x_{t_1}) \backslash (Y_{t_1} \cup Y_{b_1} \cup N(X_{b_1}))| \neq |N(x_{t_2}) \backslash (Y_{t_1} \cup Y_{b_1} \cup N(X_{b_1}))|$

$N(x_{t_1}) \backslash (Y_{t_1} \cup Y_{b_1} \cup N(X_{b_1}))$ と $N(x_{t_2}) \backslash (Y_{t_1} \cup Y_{b_1} \cup N(X_{b_1}))$ の頂点集合の大きさを比べる．

今，$Y_{r_1} = N(x_{t_1}) \backslash (Y_{t_1} \cup Y_{b_1} \cup N(X_{b_1}))$, $Y_{r_2} = N(x_{t_2}) \backslash (Y_{t_1} \cup Y_{b_1} \cup N(X_{b_1}))$ とする．頂点集合 Y_{r_1}, Y_{r_2} の各頂点は X_{b_1} とは隣接しないので，x_{t_i} や x_{t_2} に対応する弧と *bottom* で重なりを持たず，*top* で重なりを持つ．よって，$|N(x_{t_1}) \backslash (Y_{t_1} \cup Y_{b_1} \cup N(X_{b_1}))|$ が大きいとき，x_{t_1} の *top* の弧の端点は x_{t_2} の *top* の弧の端点よりも，点 q に近づける．同様に，$|N(x_{t_2}) \backslash (Y_{t_1} \cup Y_{b_1} \cup N(X_{b_1}))|$ が大きいとき，x_{t_2} の *top* の弧の端点は x_{t_1} の弧の端点よりも，点 q に近づける．

case(c) $|N(x_{t_1}) \cap Y_{t_1}| = |N(x_{t_2}) \cap Y_{t_1}|$ かつ
$|N(x_{t_1}) \backslash (Y_{t_1} \cup Y_{b_1} \cup N(X_{b_1}))| = |N(x_{t_2}) \backslash (Y_{t_1} \cup Y_{b_1} \cup N(X_{b_1}))|$

$N(x_{t_1}) \cap N(X_{b_1})$ と $N(x_{t_2}) \cap N(X_{b_1})$ の頂点集合の大きさを比べる．

case(c) を選ぶ場合，case(a)，case(b) によって端点の位置を決めることができない．つまり，*top* での重なりによって，弧の端点の位置が決まっていない．また，1. より，隣接関係の等しい頂点はない．そのため，*bottom* で弧の位置を決めることができる．$|N(x_{t_1}) \cap N(X_{b_1})|$ と $|N(x_{t_2}) \cap N(X_{b_1})|$ を比べる．$|N(x_{t_1}) \cap N(X_{b_1})|$ が大きいとき，x_{t_1} の *bottom* の弧の端点は，x_{t_2} の弧の端点よりも，q に近づける．$|N(x_{t_2}) \cap N(X_{b_1})|$ が大きいとき，x_{t_2} の *bottom* の弧の端点は，x_{t_1} の弧の端点よりも，q に近づける．これで，X_{t_1} の弧の位置が決まる．

頂点集合 X_{b_1},Y_{t_1},Y_{b_1} についても，同様である．この時の真円弧表現を $C_{i_1 j_1}$ とする．

4-(4) 頂点集合 $X_c = X \backslash (X_{t_1} \cup X_{b_1})$ について，頂点が選べなくなるまで 4-(1) から 4-(3) を繰り返す．これにより，真円弧表現 $C_{i_1 j_1}$，$C_{i_2 j_2}$，$\ldots$ ができる．

4-(5) 真円弧表現 $C_{i_1j_1}$，$C_{i_2j_2}$，$\ldots$ を組み合わせて $\bar{G}$ の真円弧表現を作る．

各 $r = 1, 2, \ldots$ に対して，$C_{i_rj_r}$ において，4-(1) から 4-(2) の操作で 4 つの集合に分けた頂点集合を $X_{i_r}, X_{j_r}, Y_{i_r}, Y_{i_r}$ とする．このとき，頂点集合 X_{i_r} と Y_{i_r} に対応する弧は，*top* で重なりを持つとする．頂点集合 X_{j_r} と Y_{j_r} に対応する弧は，*bottom* で重なりを持つ．それぞれの $X_{i_r}, X_{j_r}, Y_{i_r}, Y_{j_r}$ に対して，*top* で最も点 q 側にある弧の端点に対応する頂点 $x_{q_{i_r}} \in X_{i_r}, x_{q_{j_r}} \in X_{j_r}, y_{q_{i_r}} \in Y_{i_r}, y_{q_{j_r}} \in Y_{j_r}$ と，最も p 側にある弧の端点に対応する頂点 $x_{p_{i_r}} \in X_{i_r}, x_{p_{j_r}} \in X_{j_r}, y_{p_{i_r}} \in Y_{i_r}, y_{p_{j_r}} \in Y_{j_r}$ を選ぶ．

$C_{i_rj_r}$ と $C_{i_{r'}j_{r'}}$ の真円弧表現を組み合わせる．

$C_{i_rj_r}$ で選んだ頂点集合 $\{x_{q_{i_r}}, x_{q_{j_r}}, y_{q_{i_r}}, y_{q_{j_r}}, x_{p_{i_r}}, x_{p_{j_r}}, y_{p_{i_r}}, y_{p_{j_r}}\}$ に対して，次の 2 つの場合わけを用いる．

case(1) $C_{i_rj_r}$ で選んだ頂点集合は $C_{i_{r'}j_{r'}}$ で選んだ頂点すべてと隣接している．

この場合，$C_{i_rj_r}$ は最後に弧の端点の位置を決めることができる．よってこの時は保留し，一番最後に，今までにできた真円弧表現と組み合わせる．

case(2) $C_{i_rj_r}$ で選んだ頂点集合のある頂点と，$C_{i_{r'}j_{r'}}$ で選んだ頂点集合のある頂点が隣接していない．

この場合，$C_{i_rj_r}$ の頂点集合 $\{x_{q_{i_r}}, x_{q_{j_r}}, y_{q_{i_r}}, y_{q_{j_r}}, x_{p_{i_r}}, x_{p_{j_r}}, y_{p_{i_r}}, y_{p_{j_r}}\}$ に対して，$C_{i_{r'}j_{r'}}$ の頂点集合 $\{x_{q_{i_{r'}}}, x_{q_{j_{r'}}}, y_{q_{i_{r'}}}, y_{q_{j_{r'}}}, x_{p_{i_{r'}}}, x_{p_{j_{r'}}}, y_{p_{i_{r'}}}, y_{p_{j_{r'}}}\}$ を組み合わせることができる．これは，頂点集合 $\{x_{q_{i_r}}, x_{q_{j_r}}, y_{q_{i_r}}, y_{q_{j_r}}, x_{p_{i_r}}, x_{p_{j_r}}, y_{p_{i_r}}, y_{p_{j_r}}\}$ と頂点集合 $\{x_{q_{i_{r'}}}, x_{q_{j_{r'}}}, y_{q_{i_{r'}}}, y_{q_{j_{r'}}}, x_{p_{i_{r'}}}, x_{p_{j_{r'}}}, y_{p_{i_{r'}}}, y_{p_{j_{r'}}}\}$ について 4-(1) から 4-(3) を適用することで得られる．ここから，$C_{i_rj_r}$ と $C_{i_{r'}j_{r'}}$ の組み合わせを得ることができる．得られた真円弧表現を $C_{i_rj_r}$ とする．

上記の議論により，以下の定理が与えられる．

定理 1. 与えられたグラフ G が真区間二部グラフならば，提案アルゴリズムは $\bar{G}$ の真円弧表現を $O(n^2)$ 時間で作る．

証明． ここでは，定理 1 で作った提案アルゴリズムの計算時間を評価する．まず，1. の twin の頂点を一つに縮約するための計算時間はすでに述べた通り，$O(n+m)$ 時間である．2. のチェーングラフであるかどうかをチェックする計算時間についても同様に，$O(n+m)$ 時間である．3. の補グラフを構成するときは，すべての 2 頂点間に対して，辺が存在するかどうかを調べる必要がある．よって補グラフを構成する計算時間は $O(n^2)$ である．1. と 2. および 3. のステップは独立なのでこれらのステップの計算時間は，$O(n^2)$ である．次にステップ 4 の計算時間を考える．ステップ $4-(1)$ は x_i, x_j を選ぶとき，各頂点 $x_i (1 \leq i \leq |X|)$ に対して，x_j の候補が $|X|$ 通りあるので，$O(n^2)$ 通り考えればよい．よって，x_i, x_j を選ぶ計算時間は $O(n^2)$ 時間である．

ステップ $4-(2)$ においては，まず y_i, y_j は適当に選ぶことができる．したがって計算時間は，$O(1)$ 時間である．次に，X_{t_1} と X_{b_1} を選ぶときの計算時間は $4-(1)$ と同様に計算

できる．よって，X_{t_1} と X_{b_1} を選ぶ計算時間は $O(n^2)$ 時間である．ステップ $4-(2)$ での計算時間は $O(n^2)$ 時間である．

ステップ $4-(3)$ において，まず $case(a)$ の計算時間を考える．$|N(x_{t_1})\cap Y_{t_1}|$ と $|N(x_{t_2})\cap Y_{t_1}|$ を比較し，次数の大きい頂点の順に弧の端点を並べることができる．よって，この計算時間は $O(n)$ 時間である．

次に，$case(b)$ の計算時間を考える．$|N(x_{t_1})\backslash(Y_{t_1}\cup Y_{b_1}\cup N(X_{b_1}))|$ と $|N(x_{t_2})\backslash(Y_{t_1}\cup Y_{b_1}\cup N(X_{b_1}))|$ を比較し，次数の大きい頂点の順に弧の端点を並べることができる．よって，この計算時間は $O(n)$ 時間である．$case(c)$ の計算時間を考える．$|N(x_{t_1})\cap N(X_{b_1})|$ と $|N(x_{t_2})\cap N(X_{b_1})|$ を比較し，次数の大きい頂点の順に弧の端点を並べることができるから，この計算時間も $O(n)$ 時間である．

X_{b_1},Y_{t_1},Y_{b_1} についても同様である．したがって，ステップ $4-(3)$ の計算時間は $O(n)$ 時間である．

ステップ $4-(4)$ に関しては，この時点で選ばれていない頂点集合に対して，$4-(1),(2),(3)$ の操作を適用するので，計算時間は $O(n^2)$ 時間である．

ステップ $4-(5)$ の計算時間は $O(n^2)$ 時間である．まず，それぞれの頂点集合 X_{i_r}, X_{j_r}, Y_{i_r}, Y_{i_r} から top で最も点 q 側にある弧の端点に対応する頂点と，最も p 側にある弧の端点に対応する頂点を選ぶときの計算時間を考える．この計算時間は $O(n)$ である．なぜなら，すでに順序づけられた真円弧表現について弧の端点を順に調べればよいからである．$case(1)$ の計算時間を考える．この計算時間は $O(n)$ 時間である．なぜならば，各 $C_{i_1j_1}$，$C_{i_2j_2}$，$\ldots$ に対して，top で最も点 q 側にある弧の端点に対応する頂点と，最も p 側にある弧の端点に対応する頂点を選ぶのに必要な計算時間は $O(n)$ 時間であり，すべての真円弧表現を組み合わせるのに必要な計算時間は $O(n)$ 時間である．$case(2)$ の計算時間を考える．この計算時間は $O(n^2)$ 時間である．なぜなら，$4-(1)$ から $4-(3)$ を繰り返すので，計算時間は $O(n^2)$ 時間である．

各ステップの計算は独立なので，このアルゴリズムの全体の計算時間は，$O(n^2)$ 時間である． □

第6章　まとめ

本論文では，真区間二部グラフに対して，[5] の特徴づけによる $O(n^2)$ 時間の認識アルゴリズムを提案した．

今後の課題は，[5] の補グラフによる特徴づけを用いて，既存の区間二部グラフに対する認識アルゴリズムよりも，高速な認識アルゴリズムの開発があげられる．

謝辞

本研究を行うにあたり，日頃より懇切丁寧な指導を賜りました上原隆平准教授に心より感謝いたします．浅野哲夫教授，金沢工業高等専門学校元木光雄准教授をはじめとする浅野研究室，上原研究室の学生の皆様には，数多くの助言やご支援をいただき，厚くお礼申し上げます．特に博士後期課程の斎藤寿樹氏，清見礼助教には研究にさいし，言葉に尽くせぬどの面倒を見ていただいた事に，深く感謝致します．最後に大学院での研究を支えてくれた両親に感謝します．

参考文献

[1] A. Brandstädt, V.B. Le, and J.P. Spinrad. Graph Classes: A Survey. SIAM, 1999.

[2] F. Harary, J.A Kabell, and F.R. McMorris. Bipartite intersection graphs. Comment. Math. Univ. Carolin., 23:739-745,1982.

[3] H. Müller. Recognizing Interval Digraphs and Interval Bigraphs in Polynomial Time. Disc. Appl. Math., 78:189–205, 1997.

[4] http://www.comp.leeds.ac.uk/hm/pub/ node1.html.

[5] P. Hell and J. Huang. Interval Bigraphs and Circular Arc Graphs. J. of Graph Theory. http://www.cs.sfu.ca/˜pavol/intBig.ps, 2004.

[6] P.Hell and J.Huang,Certifying LexBFS recognition algorithms for proper interval graph, and proper interval bigraphs, manuscript 2003.

[7] S.-i. Nakano, R. Uehara, and T. Uno. A New Approach to Graph Recognition and Applications to Distance-Hereditary Graphs. Journal of Computer Science and Technology 24(3): 517-533, 2009.