制約プログラミングによる木の美的描画

5N－4

今木 孝哲†，　安達 由洋†，　土田 賢省†，　夜久 竹夫††
†東洋大学，　††日本大学

1．はじめに

我々は，制約解消の技法とその実用性について研究している．制約は様々な分野で活用することができる．例えば，電子回路のレイアウトやスケジューリング問題などは複雑であり，アルゴリズムで解決しようとするとプログラミングが非常に難しい．

本発表では，与えられた木構造に対して美的描画条件に対応する制約プログラムをPrologのメタプログラミング技法を用いて自動的に生成し，制約解消により描画を行うシステムについて報告する．本システムは，NP完全な条件の下で，ノード数約120，領域250の木構造図の問題を約1秒で解ける．また，解決した問題の木構造図をX Window上にOSF/Motifを用いて，ビューアを自動的に表示させる．

2．描画問題の分類[4]

木構造図描画の基本条件は、木構造図の配置と関係する．今後，T=(V, E, Width, Depth)と木構造図D=(T, π)として表現された木構造図をとる．木構造図Tによって、Tの配置に関係する条件を示す．

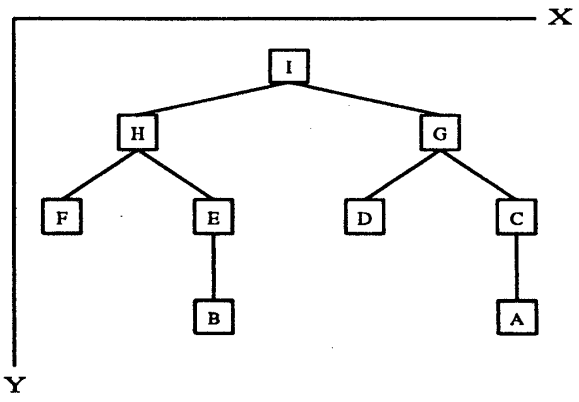
B0 :edgeが直線で描かれるとき，他のedgeやcellと交差しない．
B1a:cellのX-座標は，祖先のcellの深さの総和により定められる．
B1b:同じレベルであるすべてのセルは，同じX-座標である．
B2 :親のcellは，子のcell中央に配置される．
B3 :兄弟のcellは，順番に頂点から底へ配置される．
B4 :どのedgeも他のedgeと交差しない．
B5 :同じ構造の部分木は，平行移動をすると重なる．
B6 :どのcellも他のcellと重複しない．
B7 :X-座標上で部分木の重複を許す．
B8 :部分木の重複は，他の部分木の中央まで許す．

本研究では、この描画問題に基づいて次の制約条件を実現した．

制約1：B1∧B2∧B3∧B4
制約2：B1∧B2∧B3∧B4∧B5
制約3：all adjacent leaves have the same gap

3．制約式の生成

上の制約2は，同形性を調べて制約式を生成している．次の木構造図を用いて，具体的に説明する．



各ノードをルートとする部分木の構造をリストで表現し，そのリストを一つ一つPrologによるユニフィケーションを用いて同形であるか調べる．そして，同形であるとき次のような制約式を生成する．

E-C ?= B-A

次に，隣り合うすべてのノードの間隔を一定にする制約式は，

A ?>= B+2

となり，間隔の幅を変えることも可能である．前述の制約式に加えて，すべての親は，子の幅の中心に配置する制約式は，

2*G ?>= D+C, 2*G ?=< D+C+1

の二つが生成され，この式が成り立つように親ノードの位置が求まる．

以上の制約式の生成により制約条件に基づいたメ

Tree-Structured Diagrams Drawing by Using Constraints Programming
Takanori Imaki†, Yoshihiro Adachi†, Kensei Tutida†
Takeo Yaku††
†Faculty of Engineering, Toyo University
††College of Humanities and Sciences, Nihon University

タプログラミング技法で生成された制約式のゴールをPrologの制約パッケージに渡すことで，NP-完全な条件の下でも問題が高速に解決される．実際に生成されたゴールを次に示す．

```
solve_constraints([(8,6,A),(7,6,B),(6,4,C),(5,4,D),
   (4,4,E),(3,4,F),(2,2,G),(1,2,H),(0,0,I)]) :-
      [A,B,C,D,E,F,G,H,I] in 0..17,
/* すべてのノードに一定の間隔あける．  */
      A?>=B+2,  C?>=D+2,  C?=A,  D?>=E+2,  E?>=F+2,
      E?=B,  G?>=H+2,
/* すべての親は，子の幅の中心に配置する．  */
      2*G?>=D+C,  2*G?=<D+C+1,  2*H?>=F+E,
      2*H?=<F+E+1,  2*I?>=H+G,  2*I?=<H+G+1,
/* 同形のチェック */
      E-C?=B-A,  E-C?=B-A,  H-G?=E-C,  H-G?=F-D,
      minimize_maximum(label([A,B,C,D,E,F,G,H,I]),
         [A,B,C,D,E,F,G,H,I]).
```

4．制約解消システム

本システムは、次の木構造図データを入力としている．

```
node(ID,Width,Height) :ノードのデータ
edge(Parent,Children) :ノード間の関係
   ID        :ノードの識別番号
   Width     :ノードの幅
   Height    :ノードの高さ
   Parent    :ノードの親
   Children  :ノードの子
```

この入力から，メタプログラミング技法により，制約式が次々と自動的に生成され，一つのこのゴールとなり，それをPrologの制約パッケージに渡すことで、問題が高速かつ正確に解決される．その結果から対応する木構造図を出力する．本システム全体の動作の流れを図4.1に示す．

先に述べた制約生成と問題解決の動作時間と木構造図のノード数との関係を図4.2に示す．

5．おわりに

我々は，制約プログラミングによる木構造図の美的描画システムを実現し，その性能について実験的に検討した．なお，本システムは約2500行で実現されており，IF/Prolog と制約パッケージにより記述されSUNOS 4.1.3及びSolaris 2.4上で稼働している．

本論文で提案した制約プログラミングの技法は，Hichart[5]などの木構造型プログラム図言語の美的描画問題だけでなく，電子回路のレイアウトやスケジューリング問題などへも応用が可能である．

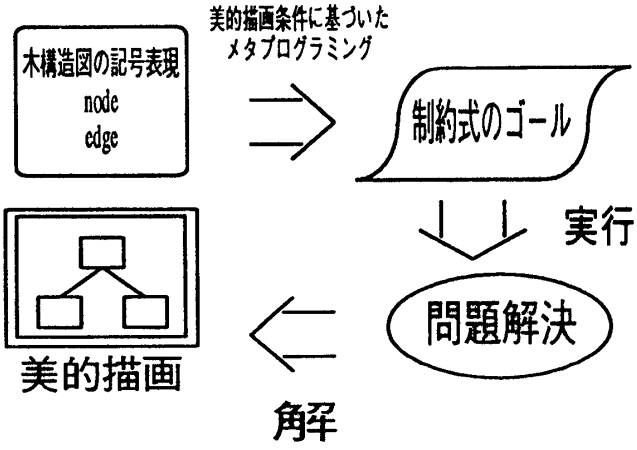


図4.1　動作の流れ

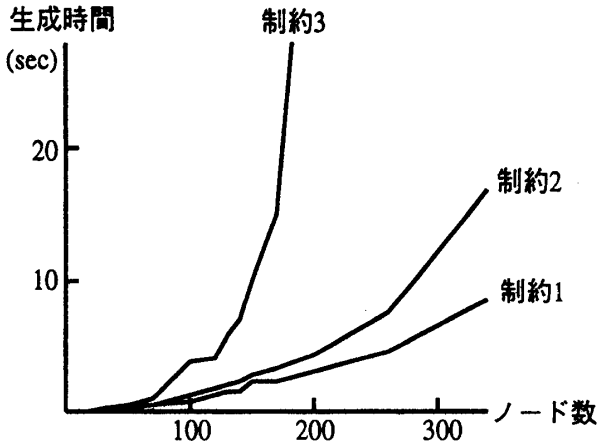


図4.2　時間とノード数の関係

参考文献

[1]Tsuchida, K: The complexity of tidy drawings of trees, Topology and Computer Science(S. Suzuki ed.), Kinokuniya, Tokyo, pp. 487-520(1987)

[2]Tsuchida, K: O(n) and O(n 2) time algorithms for the drawing problems of trees, Trans. IEICE, vol. J76-D-I, no. 6, pp. 237-246(1993)

[3]Tsuchida, K: The Complexity of Drawing Tree-Structured Diagrams, Trans. IEICE, vol. E78-D, no. 7, pp. 901-908(1995)

[4]Tsuchida, K: Constriants and Algorithms for Drawing Tree-Structured Diagrams, Department of Computer Science, pp. 87-101(1995)

[5]夜久，他： 階層的流れ図言語Hichartの情報処理記号，早稲田大学情報科学研究教育センター紀要，Vol. 3, pp. 92-107(1986)