

## ④ Answering Linear Recursive Queries in Cyclic Databases

C.S.Wu, L.J.Henschen (Veterans General Hospital, 中国)

### 発表要旨

演繹データベースにおいては、IDBが線型再帰ルールによって記述された場合にどのように効率的に問い合わせの処理を行うかが、ひとつの問題となっている。counting methodはEDBの構造がasynchronousな場合は有効な方法であるが、cyclicな場合には効率的でない。これは問合せに対する解とはならない大量の中間リレーションが生成されるからである。本発表ではrecurrence sequences (RSs) と呼ぶlevel informationに基づいた新しい処理の方法が示された。その方法では、最初にグラフ探索によりデータベースのサイクル情報を集めた後、データ値とサイクル情報を関係づけるために各分岐点に関連したcycle treeを構成する。この情報から分岐点ごとにRSsが割り付けられる。RSsは、その分岐点のレベルの出現可能性のあるQuery Constantを代表する。最後に、各々のサイクルを連結点となる分岐点のRSsを用いて、問い合わせに対する最終結果が得られる。

### 質疑応答

質問：あなたのアルゴリズムのcomplexityの概略をお話してください。

回答：最初のvirtual cycle計算のステップg1は $O(n+e)$   $n$ はノード数、 $e$ はエッジ数です。またg2, g3ステップの $O(n+e)$ です。g3では $O(n+e)$ のdepth firstの探索アルゴリズムを用いています。g4の実行には少し時間がかかります、というのは、ここではサイクルの数え上げをしなくてはならないからです。しかし我々は、全てのサイクルの数え上げを避ける方法を考案したので十分扱えるcomplexityに収まっています。(Proceedingによると $O(n \cdot n)$ :書記注)次のg5は、RS(a)とRS(b)を計算するステップで、木の併合を行いますから探索降下に過ぎません。cycle treeの空間は分岐点の数に比べると定数とみなせるので、併合も定数時間で行えます。また実はcの部分も計算する必要はありません。従ってこのステップは $O(n+e)$ です。(Proceedingによると $O(n \cdot e)$ )一部分にのみ注目すればより効率のよいアルゴリズムが存在する。その時重要になるのは終了条件ではなく、数え上げをする条件であり、これについては12月16日に台北で行われる国際コンピュータシンポジウムで報告します。