Query Inference Problem Using Steiner Trees

Venkata Subba Reddy Poli

April 6, 2023

Query Inference Problem Using Steiner Trees

Poli Venkata subba Reddy

Abstract: Logical and Physical independence is necessary for Information retrieval. If logical query is ambiguous then physical query is ambiguous. Logical independence is achieved through the unambiguous query . Unambiguous query designed through Steiner tree and the cost function.. Different queries are occurred for the same information. The best logical query is achieved through the cost function. In this paper unambiguous logical queries are designed through the Steiner tress and cost functions and Blockchain system is discusses using Steiner trees for physical query independence. Some examples are given for these problems.

Keywords- Query Language, Conceptual design, Physical Design, Steiner trees

1. Introduction

Various languages are developed for querying relational databases. These are based on the relational calculus and relational algebra. Higher level languages such as SQL[1], QUEL[8] and QBE[10] provided physical data independence. The logical structure do not provide the logical data independence. The query languages combine with logical structure to provide the logical data independence.

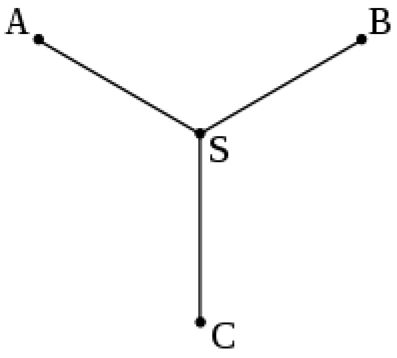
The query inference problem is to translate a sentence of a query language (QL) into an unambiguous representation of a query. We construct data base graph using entity relationship for data base[9]. We translate query into query graph. This query graph is a sub graph of data base graph. Query graph which does not contain any cycles is a query tree. The query tree which is the internal representation of query, can be executed directly or translated into logical expressions such as relational calculus or relational algebra[4]. A measure of logical data independence can be achieved in QL. The relational QL without relation[6] is an example of such a language.

Let A be set of attributes, R be set of relations and S be set of attributes. In a query the set of relations R and set of entities S projects into sub-sets of attribute set A.

The target part of the sentence of the query can be represented as target graph. It is sub-graph of the data base graph. If the target part of the simple sentence is contained in more than one query trees, then we get ambiguity. To solve this ambiguity, we find that minimal query which contains the target part and has minimum cost function derived through directed steiner trees path. This method is referred to as minimum directed cost steiner tree (MDCST). Both target part and qualification part of a query can be represented by a sentence graph. If the sentence graph contains more than one minimal query tree then we get ambiguity. We resolve this ambiguity by using MDCST problem.

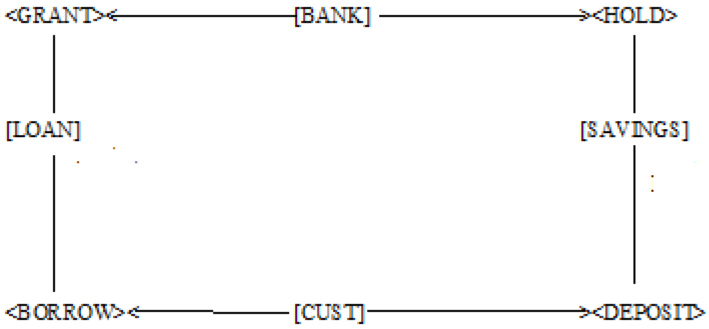
2. Steiner Trees for Database Systems

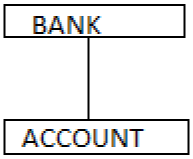
A Steiner tree is tree with Steiner node (introductory node). A Steiner graph is a graph with Steiner nodes.



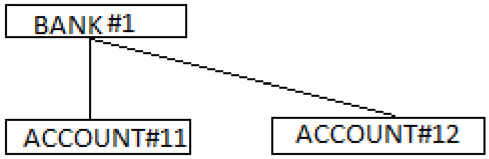
Here S is Steiner nod.

In DBTG Model, sometimes intermediate nodes has to be introduced, database task group(DBTG) published a report in 1971 which contains definitions and guidelines for the construction of database. The DBTG view is related to DBTG set structure. The DBTG set structure is hierarchical data structure consisting of two levels. DBYG set occurrence is a collection of data sets one of which is owner and others are members. The occurrences of dataset are of same type.



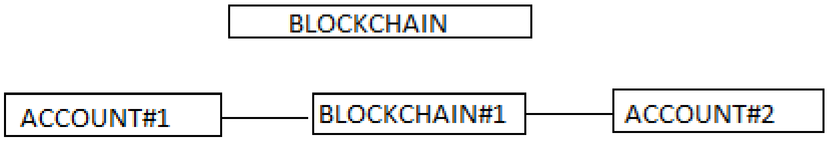


a. All owner records of occurrences of the same data set type. The members are different data set type.
b. All owner data sets are different data sets of member type.
c. A member data item is allowed only once.



Suppose a transaction can’t be made unless introduce intermediate node is called Steiner node.

A transaction is made between two accounted is called blockchain.



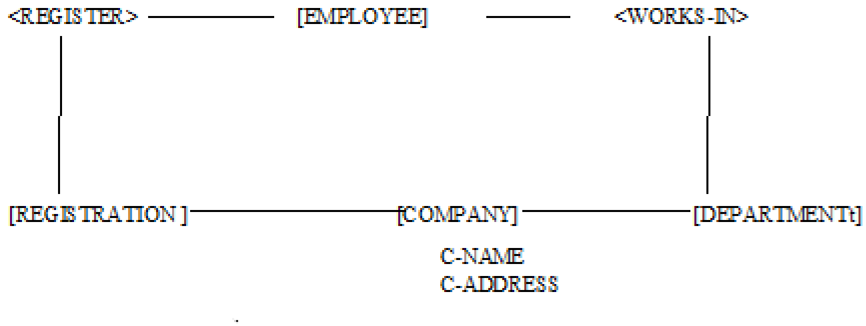
**

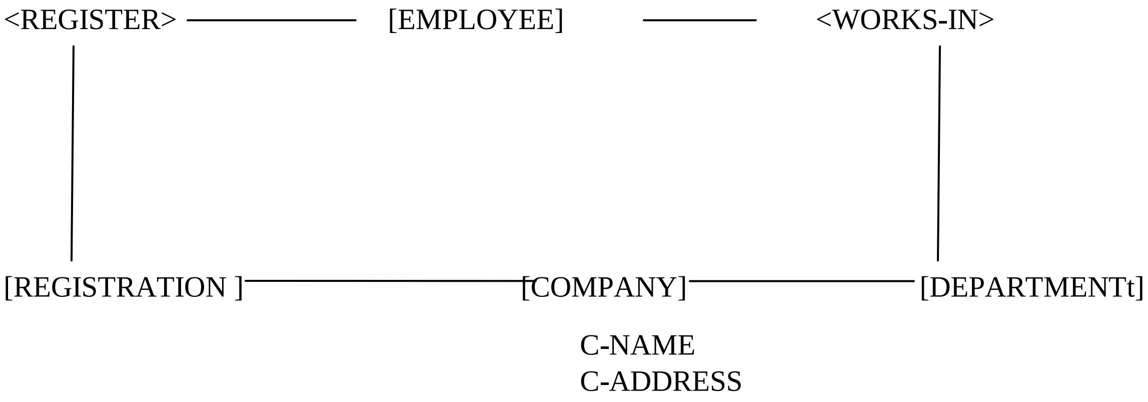
Blockchain using Steiner trees for normalization

2.1 Steiner trees for query inference

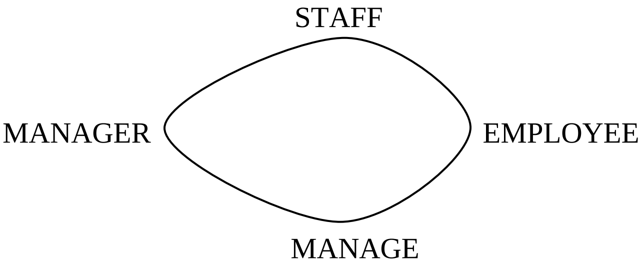
We show the representation of data base schema by graph. The entity relationship (ER) model is of the view that the real world consists of entities and relationships. The ER diagram is generally used for describing an ER data base schema. Consider entity relationship diagram for the base ball data base[9]. We shall show how it represents data base graph using ER diagram.

Let D be the set of attributes, S be the set of entities and R be the set of relations. The data base graph of an ER diagram schema is a graph <V, E> where V = D ∪ S ∪ R, W = S ∪ R and E ⊆ W X V. If w ∈ W and d is an attribute of W, then the edge {w, d} ∈ E. If entity S is involved in the relation Y, then the edge {S, Y} ∈ E.Consider an example:



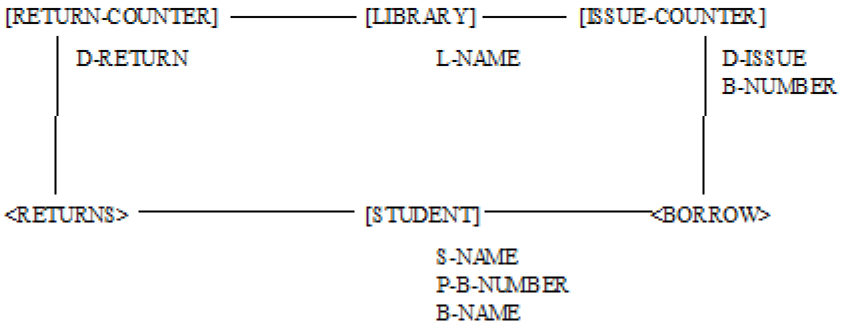


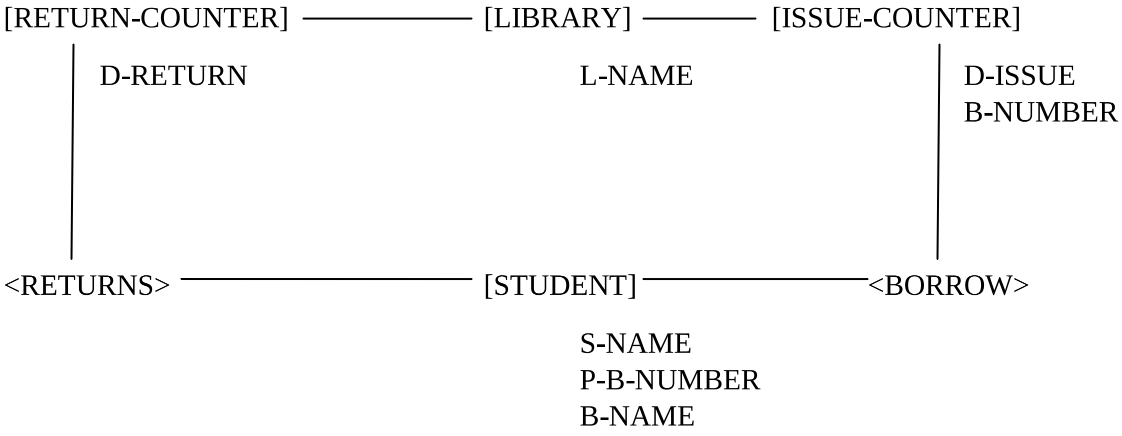
Multi graph is a graph which has multiple edges or arcs connecting two specific nodes. A data base graph is a multi graph in case of recursive relationships. Consider data base schema in which EMPLOYEE is the only entity and MANAGE is a relationship. One EMPLOYEE plays the role of the MANAGER, other EMPLOYEE plays the role of the STAFF. This is represented in the data base graph by two edges between EMPLOYEE and MANAGE. One edge is labeled as STAFF while the other is labeled as MANAGER.



2.2 QUERIES AND QUERY TREES:

We discuss queries and query trees using entity relationship diagram by considering library data base. A query Q is a relational expression which yields a relation REL [Q] as a result when applied to a data base.





We show how the queries can be represented by sub trees of a data base. We consider here cyclic data base for the library.

Consider the following description of library.

Library has a name.

Each student has name and pass book number.

Issuing counter has date of issue and book number.

Return counter has date of return.

The student borrows the book from the library through issue counter and return to the library through return counter.

Consider the following language queries for library.

Q_1 : For each student borrowing from the library display student name, book name and date of issue.

Q_1 can be represented as the STUDENT entity, BORROW relationship and ISSUE-COUNTER projected onto S-NAME, B-NAME and D-ISSUE. Then the edges of query graph are [STUDENT, S-NAME], [STUDENT, B-NAME], [STUDENT, BORROW], [BORROW, ISSUE-

COUNTER], [ISSUE-COUNTER, D-ISSUE]. The query Q_1 gives the query graph and it is denoted by

QGRAPH [Q_1] = [STUDENT, S-NAME, B-NAME, BORROW, ISSUE-COUNTER, D-ISSUE].

Q_2 : For each student returning to the library display student name, book name and date of return. This query gives the query graph and is denoted by

QGRAPH [Q_2] = [STUDENT, S-NAME, B-NAME, RETURN, RETURN-COUNTER, D-RETURN]

Q_3 : For each student in the library display student name, book name, date of issue, and date of return if student borrows from the issue counter, or student returns to the return counter, the query Q_3 gives the query graph and is denoted by

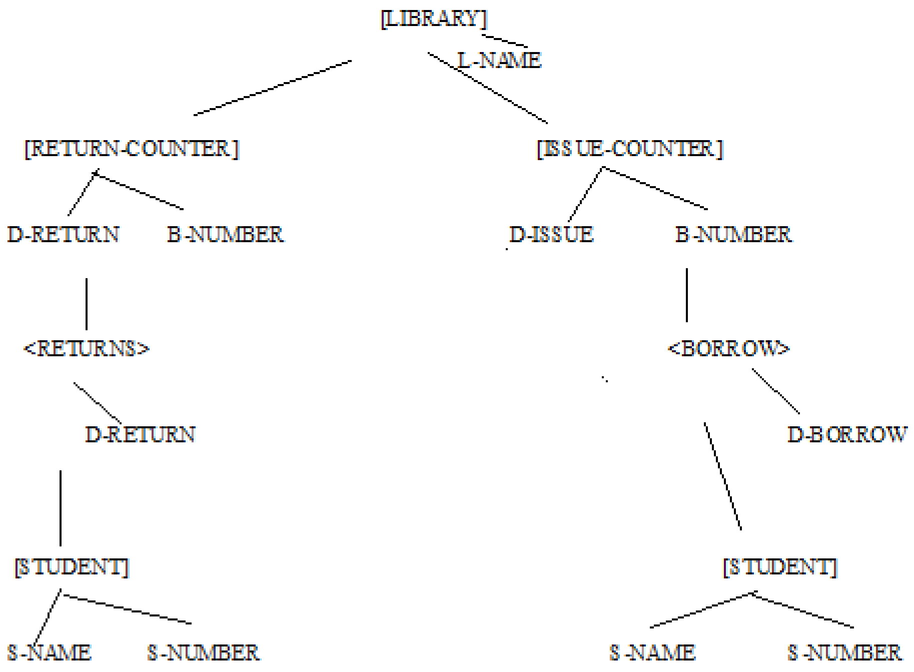
QGRAPH[Q_3] = {(STUDENT, S-NAME, B-NAME, D-ISSUE, D-RETURN, BORROW, ISSUE-COUNTER), (STUDENT, RETURN, RETURN-COUNTER)}

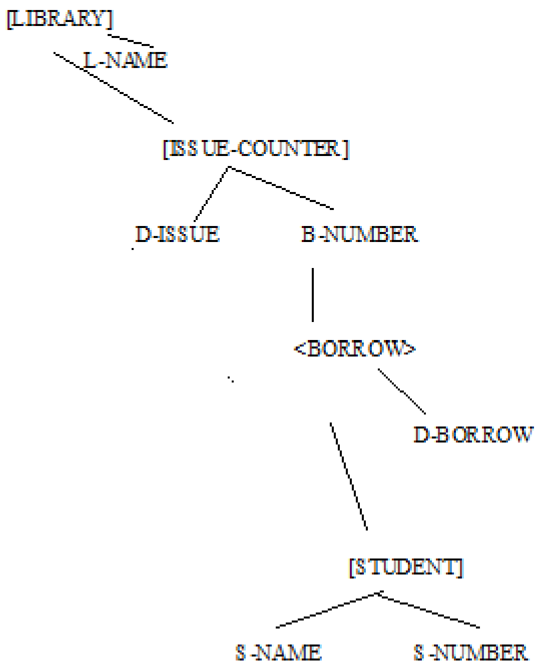
Q_4 : For each student in library display student name, book name if student borrows from the issuing counter. It gives the query graph and is denoted by

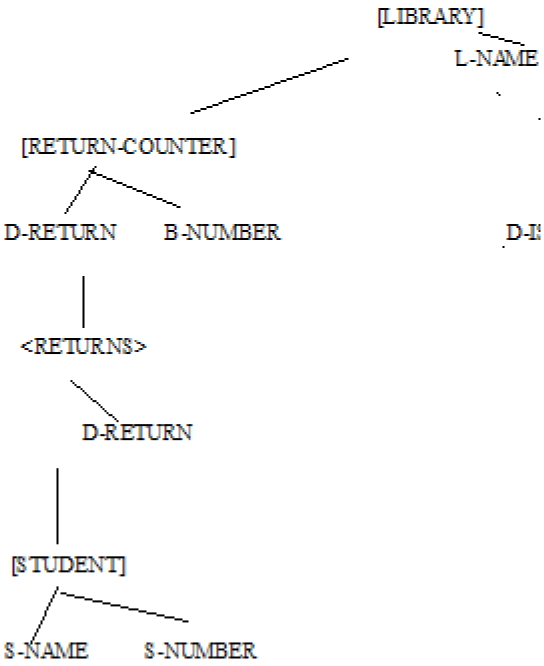
QGRAPH[Q_4] = {(STUDENT, S-NAME, B-NAME),
(STUDENT, BORROW, ISSUE-COUNTER)}

Q_5 : For each student in library display student name, date of issue, book name and date of return if student borrows from the issuing counter and student returns through the return counter.

A query graph does not contain any cycle, it is called query tree. The query Q_1 is a query tree because it does not contain any cycle.


[LIBRARY]
L-NAME
[RETURN-COUNTER]
[ISSUE-COUNTER]
D-RETURN
B-NUMBER
D-ISSUE
B-NUMBER
<RETURNS>
<BORROW>
D-RETURN
D-BORROW
[STUDENT]
[STUDENT]
S-NAME
S-NUMBER
S-NAME
S-NUMBER


[LIBRARY]
L-NAME
[ISSUE-COUNTER]
D-ISSUE
B-NUMBER
<BORROW>
D-BORROW
[STUDENT]
S-NAME
S-NUMBER



[LIBRARY]

L-NAME

[RETURN-COUNTER] [ISSUE-COUNTER]

D-RETURN B-NUMBER D-ISSUE B-NUMBER

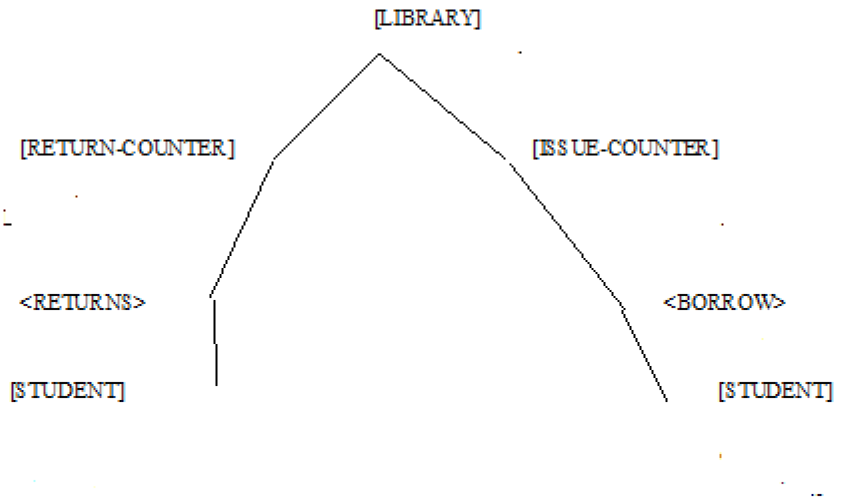
<RETURNS> <BORROW>

D-RETURN D-BORROW

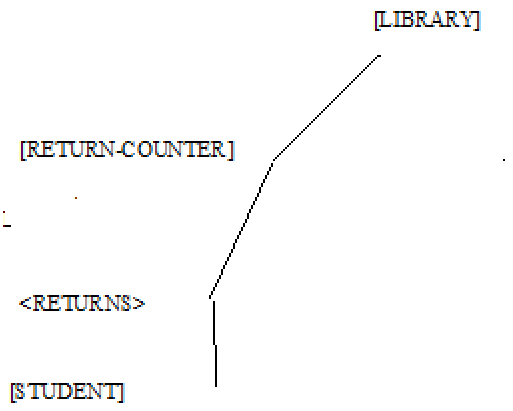
[STUDENT] [STUDENT]

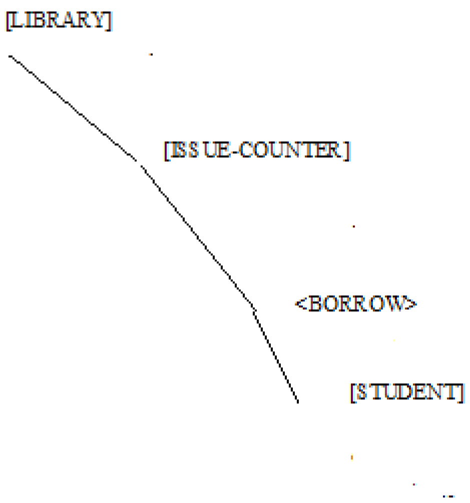
S-NAME S-NUMBER S-NAME S-NUMBER

Database Scheme Tree

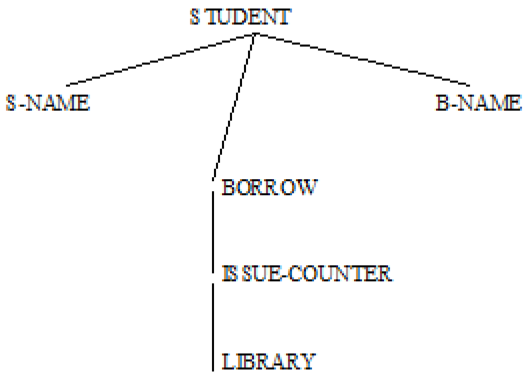


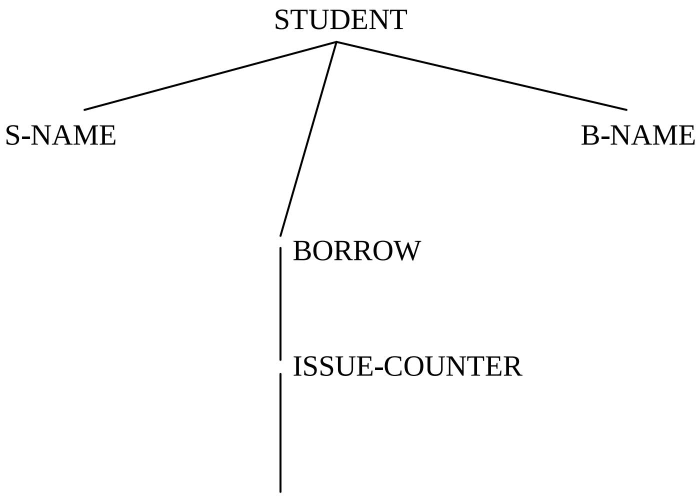
Database Scheme Sub-trees




[LIBRARY]
[ISSUE-COUNTER]
<BORROW>
[STUDENT]

Query Tree


STUDENT
S-NAME
B-NAME
BORROW
ISSUE-COUNTER
LIBRARY


STUDENT
S-NAME
B-NAME
BORROW
ISSUE-COUNTER

LIBRARY

Sub-Tree

The nodes without any further connection are called leaf nodes. Here the leaf nodes of the query tree are S-NAME, B-NAME, D-ISSUE. Query graph Q_2 also gives query tree. The query graph Q_3 is not query tree because it is cyclic. The query graph Q_3 contain cycle with an edge [BORROW, S-NAME] in query graph Q_3. A cyclic query can be looked upon as a relational expression in terms of tree queries. For example

$$\text{REL } [Q_5] = \text{REL } [Q_1] \cap \text{REL } [Q_2]$$

A query is a simple query if it can be expressed by a single query graph. The information retrieval of the query Q_3 is equal to the both information retrieval of the query Q_1 and query Q_2.

$$\text{REL } [Q_3] = \text{REL } [Q_1] \cup \text{REL } [Q_2]$$

The query graphs Q_1 and Q_2 are simple queries.

A query is a compound query if it cannot be represented by more than one query graph. The query graph Q_3 is compound because Q_3 represents union of simple queries Q_1 and Q_2.

2.3 TARGET AND SENTENCE GRAPHS OF SENTENCES:

A query has two basic parts, one is called the qualification part and the other is target part. For example, consider the query 'give the names of the students with grade B'. Here names of the students is target part and grade B is the qualification part. The query is stated in a language called query language. The sentence of the QL can be represented by a sub-graph of the data base graph. The QL based on relation manipulation may be divided into two types. Those based on relational algebra and those based on relational calculus. We discuss QL based on relational calculus. Let us consider sentence of the form

List <target part> for <qualification part>, where < target part > is a list of tuple attributes and <qualification part> is predicate. Consider the following sentences related to figure 1:

S_1 : List NAME, HOME for CITY = 'DELHI'

S_2 : List NAME, HOME for HOME = 'SBI'

S_3 : List NAME, HOME for CITY = 'DELHI' & HOME = 'SBI'

Sentence S_1 display NAME, HOME of the teams of the city equal to Delhi. Sentence S_2 displays NAME, HOME of the teams for the Home equal to SBI. S_3 is an intersection of S_1 and S_2.

A sentence of a QL is called a simple sentence if and only if expresses one derived relation. That is, a sentence is simple if it contains exactly one tuple variable and does not contain any relational algebraic operators:

The following examples are simple sentences.

S_4 : List NAME, HOME for PLAYS

S_5 : List POS-NAME, POS-NUMBER, NAME for SEASON.

A sentence is said to be compound if it is a relational expression with more than one derived relation. That is, it contains more than one tuple variable or contains relational algebraic operator.

For example :

S_6 : List POS-NAME, POS-NUMBER for NAME = ‘BENERGY’

S_7 : List POS-NAME, POS-NUMBER for HOME = ‘CALCUTTA’

S_8 : List POS-NAME, POS-NUMBER for NAME = ‘BENERGY’
& HOME = ‘CALCUTTA’

The target graph is a sub graph of data base graph, consider the following examples S_4 and S_5.

TARGET [S_4] = [NAME, HOME, PLAYER]

TARGET [S_5] = [POS-NAME, POS-NUMBER, NAME, SEASON]

The target graph of the sentences S_1, S_2 and S_3 are different from the target graph of the simple sentences S_4 and S_5. The target graph of the sentences S_1, S_2 and S_3 are same. It gives

TARGET [S_1] = [NAME, HOME]

The target graph may be cyclic graph for simple sentence. For example

S_9 : List S-NAME, D-ISSUE, D-RETURN for LIBRARY

TARGET [S_9] = [S-NAME, D-ISSUE, D-RETURN, LIBRARY] is cyclic target graph.

We define a sentence graph of a compound sentence as the union of the target and qualification graphs. For example, for sentence S_8, we have

TARGET [S_8] = [POS-NAME, POS-NUMBER]

QUALIFICATION [S_8] = [NAME, HOME]

SENTENCE [S_8] = [POS-NAME, POS-NUMBER, NAME, HOME]

= TARGET [S_8] ∪ QUALIFICATION [S_8]

2.4 QUERY INFERENCE PROBLEM:

The query inference problem is to determine the best query tree containing either the target graph of a simple sentence or containing the sentence graph of a compound sentence. Consider the ER diagram for the University example [4]

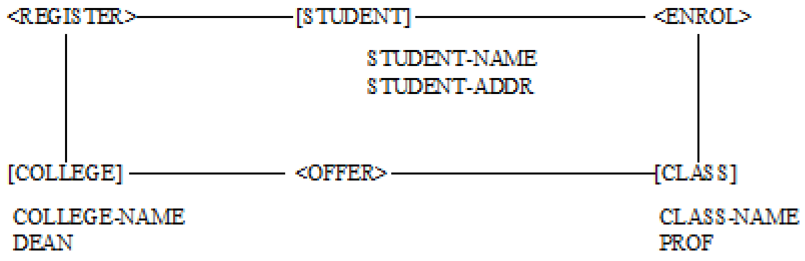
If a data base graph is acyclic, then it is tree. Hence the unique minimal query tree can be obtained by successively deleting all leaf nodes which are not in the target graph. In this way it follows that every sentence is unambiguous with respect to acyclic data base graph. This approach was used by Lien [5] for query interpretation when the ER diagram is a tree.

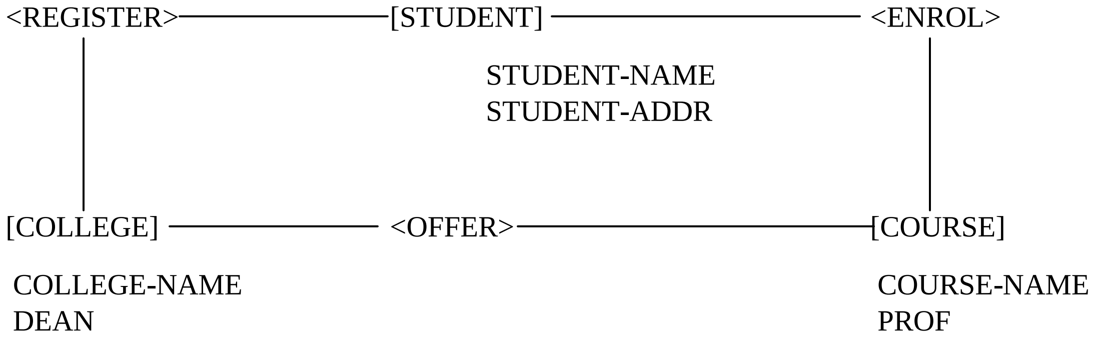
If a data base graph is cyclic, a sentence may be ambiguous, such as the sentence S. We feel intuitively that Q_5 expresses more direct relationship between STUDENT-NAME and COLLEGE-NAME; thus qualifying itself as better interpreter of the sentence S. This ambiguity created by cyclic data base graphs is resolved by considering the directed cost function [4].

The directed cost function DC : E → N is a function from the edge E of the data base graph to the positive integer N. If {w, d} is an edge in E where d is an attribute of w, then DC(w, d) = 1, DC(d, w) = 1. If {s, r} is an edge in E where s is an entity involved in a relationship r, then DC(r, s) = 1, DC(s, r) = 1 if the entity s can be involved in at most one instance of relationship r ; otherwise DC(s, r) = μ (a sufficiently large number such as the cardinality of V). This means that the cost of traversing in one direction is cheap, while in the opposite direction it is expensive. cyclic graph is expensive.

A simple sentence in a QL is said to resolvable if the cheapest query tree containing the target graph is unique ; otherwise, the sentence is irresolvable. The best interpretation of a resolvable sentence is the query with cheapest query tree containing the target graph. Using the cost function the target graph is made complete into the minimum cost query tree. For this steiner trees are introduced.

Let G = <V, E> be a connected graph, let TG = <TV, TE> be the target graph where TV ⊆ V, TE ⊆ E and TG is acyclic. A steiner tree is a graph SG = <SV, SE> such that SG is a sub tree of G and TG is a sub graph of SG. Consider a cost function COST from the edges E to the positive numbers N defined as COST : E → N. The sum of COST (e) for edges e in SE are minimized by the minimum cost steiner tree (MCST). The MCST is called the minimum cost path between two nodes x and y if TG =<{x, y}, 0>. Similarly the MCST is known as the minimum cost spanning tree constrained to use edge {x, y} if TG = <v, {x, y}>. Choose a root in an undirected tree that defines a root directed tree in which all edges are directed away from the root. Let DC : E → N be a directed cost function. The minimum directed cost steiner tree (MDCST) is a root directed steiner tree TG = <TV, TE> that minimizes the sum of DC(e) for directed edges e in TE.





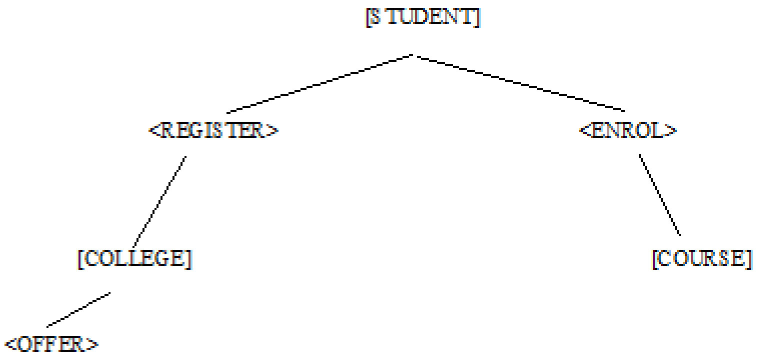


FIGURE-3

Consider the three natural language queries of the above example.

Q_5 : For each student registered in a college, display the name of the student and the name of the college in which the student is registered.

Q_6 : For each enrolment of a student in a course, display the name of the student and the name of the college which offers the course.

Q_7 : For each student registered in a college, display the student name and college name if the student is enrolled in any course.

The above queries give the query trees using previous discussions.

QTREE [Q_5] = [(STUDENT-NAME, STUDENT, REGISTER, COLLEGE, COLLEGE-NAME)]

QTREE [Q_6] = [(STUDENT-NAME, STUDENT, ENROLL, COURSE, OFFER, COLLEGE, COLLEGE-NAME)]

QTREE [Q_7] = [(STUDENT-NAME, STUDENT, REGISTER, COLLEGE-NAME), (STUDENT, ENROLL)]

Consider the sentence S = ‘List STUDENT-NAME, COLLEGE-NAME’

The target graph of this sentence is

TARGET [S] = [STUDENT-NAME, COLLEGE-NAME]

This TARGET graph is contained in QTREE [Q_5], QTREE [Q_6] and QTREE [Q_7]. The queries Q_5, Q_6 and Q_7 may be taken as possible interpretations of sentence S. A query tree is minimal for simple sentence if each leaf node is contained in the target graph. QTREE [Q_5] and QTREE [Q_6] are minimal because the leaf nodes of QTREE [Q_5] and QTREE [Q_6] are contained in TARGET [S]. The QTREE[Q_7] is not minimal because it contains leaf node ENROL which is not contained in TARGET [S]. A minimal query tree is known as the best query tree. If the tabrget graph of a simple sentence belongs to more than one minimal tree, then it is known as ambiguous; otherwise, it is unambiguous. The sentence S is ambiguous, because its target graph is contained in two minimal query trees QTREE [Q_5] and QTREE [Q_6].

Example 2.4.1 :

Consider triangle for directed tree having vertices A, B, C and the directed edges (A, B), (B, A), (B, C), (C, B), (C, A), (A, C). Let the directed costs of the edges be

DC (A, B) = 2	DC (B, A) = 4
DC (C, A) = 5	DC (A, C) = 4
DC (B, C) = 1	DC (C, B) = 2

The directed trees passing through all points are given below:

Cost {(A, B), (A, C)} = 6

Cost {(B, A), (B, C)} = 5

Cost {(C, A), (C, B)} = 7

Cost {(A, B), (B, C)} = 3

Cost {(B, A), (A, C)} = 8

Cost {(C, A), (A, B)} = 7

The MDCST of the three nodes is {(A, B), (B, C)}.

Example 2.4.2 :

We resolve the ambiguity for simple sentences using MDCST problem. Consider the figure 3.

Consider the simple sentence S = ‘list STUDENT-NAME, COLLEGE-NAME’.

The sentence S contains two minimal query trees Q_5 and Q_6.

QTREE [Q_5] = [(STUDENT-NAME, STUDENT, REGISTER, COLLEGE, COLLEGE-NAME)]

QTREE [Q_6] = [(STUDENT-NAME, STUDENT, ENROLL, COURSE, OFFER, COLLEGE, COLLEGE-NAME)]

The steiner tree is a sub tree of the query tree. The steiner trees with respect to sentence S are

QTREE [S-Q_5] = {(STUDENT, REGISTER, COLLEGE)}

QTREE [S-Q_6] = {(STUDENT, ENROLL, COURSE, OFFER, COLLEGE)}

We apply cost function to steiner trees QTREE [S-Q_5] and QTREE [S-Q_6]

DC (STUDENT, REGISTER) = 1

DC (REGISTER, COLLEGE) = 1

The total cost for QTREE [S-Q_5] is 2

DC (STUDENT, ENROLL) = µ DC (ENROLL, COURSE) = 1

DC (STUDENT, REGISTER) = 1 DC (COURSE, OFFER) = μ

DC (COLLEGE, OFFER) = 1

The total cost for QTREE [S-Q_6] is $2\mu +3$. The minimum directed cost for the two steiner trees, i.e., MDCST is min $(2, 2\mu +3) = 2$. This shows Q_5 is the best interpreter for the sentence S.

...

REFERENCES

1. CHAMBERLIN, D.D. ASTRAHAN, M.M ESWARAN, K.P., GRIFFITHS, P.P., LORIE, R.A., MOHL., J.W.PEISNER, P AND WADE, B.W. SEQUEL : A unified approach to data definition, manipulation and control, IBM J. Res. Dev. 20 (Nov. 1976), pp 560-575.

2. FAGIN, R., MENDELZON, N.O., AND ULLMAN, J.D. : A specified universal relation assumption and its properties ACM Trans. Database syst. 7, 3 (Sept.1982), pp 343-360.

3. HARARY, F. : Graph Theory. Addison- Wesley, Reading, Mass, 1969.

4. JOSEPH A.WALD, AND PAUL G. SORENSON. : Resolving the Query inference problem using Steiner trees. ACM Trans. Database Systems 9, 3 (Sept. 1984) pp 348-368.

5. LIEN, Y.E : On the semantics of the entity relationship data model. In Entity-Relationship Approach to Systems Analysis and Design PP. CHEN ED. North-Holland, Amsterdam, 1980, pp 155-167.

6. MAIER, D., ROSEN SHTEIN, D., SALVETER, S., STEIN, J AND WARREN, D.S : Towards logical data independence: A relational query language without relations. In proceedings 1982 Int. Conf. on Management of Data (Orlando, Fla., June, 1982) ACM, New York, pp 51-60.

7. MAIER, D., AND ULLMAN, J.D. : Maximal Objects and the semantics of universal relation databases. ACM Trans. Database Syst. 8,1 (Mar,1983) pp 1-14.

8. STONEBRAKER, M., HONG., E., KEEPS, P., AND HELD, G. : The design and implementation of INGRES, ACM Trans. Database Syst. 1, 3 (Sep. 1976) pp 189-222.

9. ULLMAN, J.D. : Principles of Database Systems, 1985,
2nd Edn., Galgotia, publ. New Delhi.

10. ZLOOF, M.M : Query by example, In proceedings
National Computer Conference May,
1975 AFIPS Press, Arlington, Va.,
pp 431-437.