Graph Query: A Tool To Detect Patterns of Abusive Tax Transactions

Rahul Tikekar, Kay Wolman, and Larry May, Internal Revenue Service

---

The U.S. Internal Revenue Service (IRS) is charged by the U.S. Congress to collect taxes from individuals and businesses and to enforce the tax laws. Every year, the IRS receives and processes about 200 million tax returns. Each return is filed by an "entity." Example entities include an individual (e.g., John Doe), a married couple (e.g., John and Betty Doe), a corporation, a partnership, an S corporation, and a Limited Liability Corporation (LLC).

Many entities associate with other entities. For example, an individual may work for a corporation. As another example, two individuals can form a partnership. Similarly, an individual and an LLC can form a partnership, which in turn can form a new partnership with yet another partnership, and so on. There is no limit to the complexity of associations among entities.

There exist special types of entities called flowthrough or passthrough entities. These are legal entities that are formed by one or more entities—known as shareholders or owners. The term "flowthrough" is used to describe the flow of income and losses to the shareholders or owners. The flowthrough entity is not subject to income tax at the entity level. The income generated in the business will flow through to the shareholders or the owners of the business, and the owners have to pay taxes on that income. Examples of flowthrough entities include S corporations, LLCs, and partnerships.

A tax shelter is any method of reducing taxable income that results in reduced tax. There are many tax shelters that are legal. Investing in a company-sponsored retirement plan is a common method to reduce taxable income. The objective of this paper is to describe work being performed to identify illegal tax shelters—associations among entities that are formed solely for the purpose of abusing tax laws, so as to avoid paying taxes—also termed Abusive Tax Avoidance Transactions (ATATs).

Abusive Tax Avoidance Transactions

Although there is no all-inclusive definition of an ATAT, the term generally includes any partnership, trust, investment plan, or any other entity

or association designed or structured to obtain tax benefits not allowed by law. Promoters are aggressively marketing ATAT schemes that undermine the U.S. voluntary tax system. The business of promoting ATAT schemes has expanded in recent years to encompass all socioeconomic levels. In response to the explosion of abusive tax strategies offered to the general public, the IRS Commissioner has designated investigations of these promotions as a key compliance strategy for the IRS.

While IRS enforcement personnel attempt to be versed in all areas of Federal taxation, they tend to focus or specialize in one or two domains, as well as one or two non-Federal jurisdictions. ATATs are frequently structured to shroud the facts through a fabricated complex situation. From a tax perspective, this obfuscation occurs along three general lines:

1. Increased complexity through dispersed geographic locations and multiple jurisdictions (both State and international).
2. Increased complexity by exploiting the organizational structure of the IRS. A transaction may involve multiple operating divisions and multiple tax specialties.
3. Increased complexity by intermingling and manipulating various aspects of tax law to obtain unintended consequences. ATATs include schemes that rely on:
    - The misuse of disparate sections of the Internal Revenue Code (IRC) to produce clearly unintended results.
    - The intentional manipulation of potential ambiguities of the tax laws in order to claim tax benefits improperly.
    - Sham arrangements having no economic significance other than tax reduction.
    - Gross valuation overstatements that ascribe a value to an asset or service that is more than the asset's correct value, and the overvaluation results in a tax reduction.
    - False statements about the allowability of tax benefits to participants, which are contrary to clearly established law.

A very simple example of an ATAT follows. IRS regulations require that partners in a partnership pay a self-employment tax on income received from the partnership. Shareholders in an S corporation, however, are not required to pay a self-employment tax on the flowthrough income distributed from the S corporation. By creating an S corporation—as one of the partners—to receive income from a partnership that is then distributed to the

individual, it is possible for an entity to avoid paying the self-employment tax (Figure 1). This is an example of an ATAT because the S corporation is created solely with the objective of avoiding the self-employment tax.

Some ATATs are designed to appear, and often are, quite complex. They can involve various financial products, as well as numerous entities, including partnerships, corporations, LLCs, and offshore entities. This, by design, is an effort to make it difficult to track and follow a transaction and,

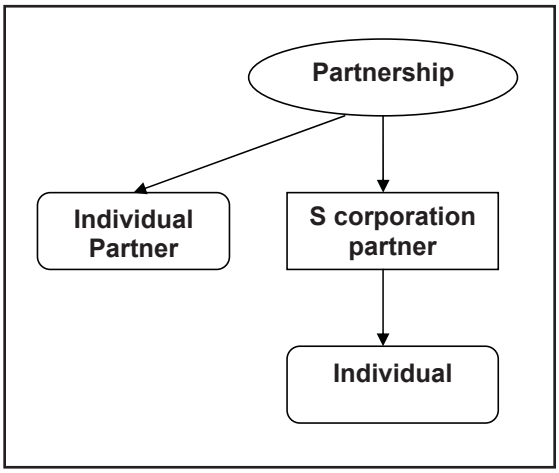

Figure 1

hence, difficult to determine the abusive nature of that transaction. Through the use of flowthrough entities, such as LLCs and partnerships, it is also difficult to identify the tax benefits claimed on a participant's tax return.

Once an ATAT promoter has devised a particular scheme, it may be replicated multiple times for the benefit of many clients.

An IRS agent may discover an abusive transaction through a routine examination and draw on the expertise of many specialists to fully develop the issue. After detecting the abusive transaction of one taxpayer, it is natural to wonder if other taxpayers are involved in similar schemes.

Currently, the selection of returns for examination of improper claims is based on that return alone. It would be useful to get a more complete picture of an entity by piecing together various associations of each entity. But this exercise is difficult and time-consuming. In the example given above, the individual's return would show an income from the S corporation and

may not seem suspicious. A more complete picture of the individual's relationship, as in Figure 1, would be beneficial in performing the function of identifying compliance risk. The challenge lies in providing such a picture of an entity quickly using data from actual tax returns.

Link Analysis

IRS began to transcribe Schedule K1 for the first time for Tax Year 2000. Schedule K1 is used by flowthrough entities to report to shareholders/owners (and to the IRS) how much income, etc., is flowing through to them (the shareholders/owners). The Market Review and Technology Assessment committee proposed that IRS use link analysis technology to make use of the newly available K1 data. Link analysis technology uses the concept of relationships (or links) between entities to present to a user the associations, or links, in which a given entity participates. As a result of the recommendation to use link analysis technology, in August 2002, the IRS Office of Research contracted with MITRE Corporation to build a prototype link analysis tool, as a proof of concept, to demonstrate the value of link analysis. This prototype was completed fairly quickly in May 2003.

Link analysis of an entity begins with the user specifying the taxpayer identification number (TIN), which could be a Social Security number or an employer identification number, of an entity of interest. The tool then searches a database and provides the associations of the entity with those that are documented on Schedule K1. Usually these associations are presented in the form of a diagram (or graph) that shows the entity in question connected (or linked) to other entities to which it is related (see Figure 2)—links connecting two entities show the flow of money between them. In addition, the associations of the other entities involved may also be shown. Such a graph provides a "big picture" that is often very useful in making decisions. The tool shows the entities involved even in the most complex arrangement. Such a tool can help auditors and researchers identify questionable transactions, some of which may turn out to be ATATs.

During the use of a link analysis tool, an analyst may discover an abusive pattern—or structure—that appears frequently enough so as not to consider it a coincidence. The analyst may want to know how many, and which, other entities participate in a similar structure. One option the analyst may use would be to continue to use the link analysis tool to specify many different TINs and look for that pattern in the resulting graph. Not only is such a technique inefficient but, quite possibly, infeasible as well. A tool that can provide the answer to the analyst's question would prove very useful in identifying entities involved in ATAT schemes.

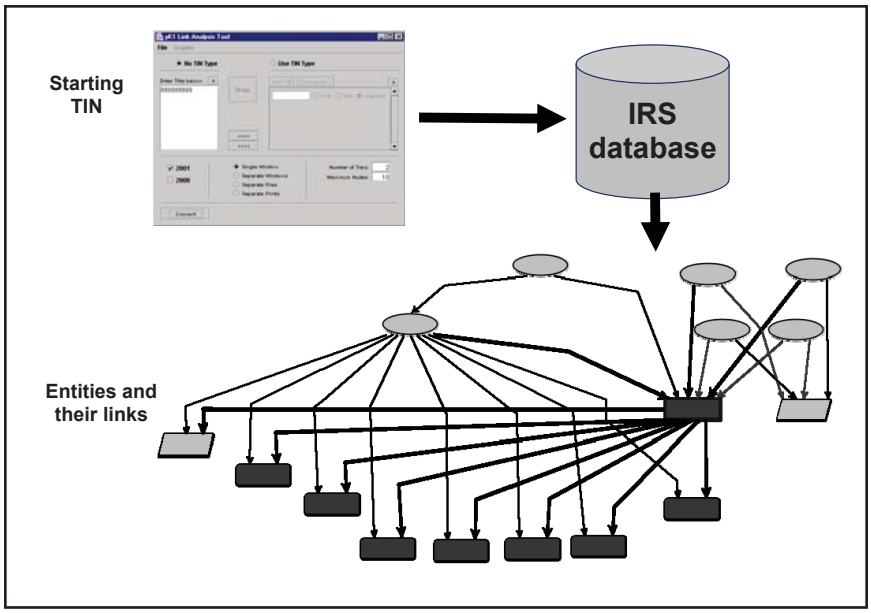

Figure 2

Problem Statement

This paper describes a tool that will provide a solution to the following problem: Given a structure or pattern of entities and their relationships, find other entities in the database that participate with yet other entities in a relationship similar to the provided pattern. The input to the tool will be the pattern in question, specified in the form of a graph, while the output will be a list of entity TINs. In the computer science domain, such a problem of finding matching graphs is called the graph isomorphism problem. Finding solutions to the graph isomorphism problem usually takes a great deal of computing power and time.

Modeling ATATs as Graphs

Because ATATs can be conceptualized as associations among entities, they can be modeled as graphs involving nodes (vertices) and edges (links). Conditions can be imposed on nodes and edges, thereby creating, what is termed, a labeled graph. The graph then becomes the starting point for further explorations. This is in contrast to a link analysis tool when the starting point is

usually an entity TIN. Thus, a tool to look for graphs complements a link analysis tool.

A graph is a collection of nodes and edges, where nodes are usually connected by edges. Figure 3 shows an example of a graph involving two nodes. One node represents a trust entity, while the other represents an address entity. The link between the two nodes represents a flow of money. Thus, the graph models schemes where money from a trust goes into an entity that is based outside the U.S.

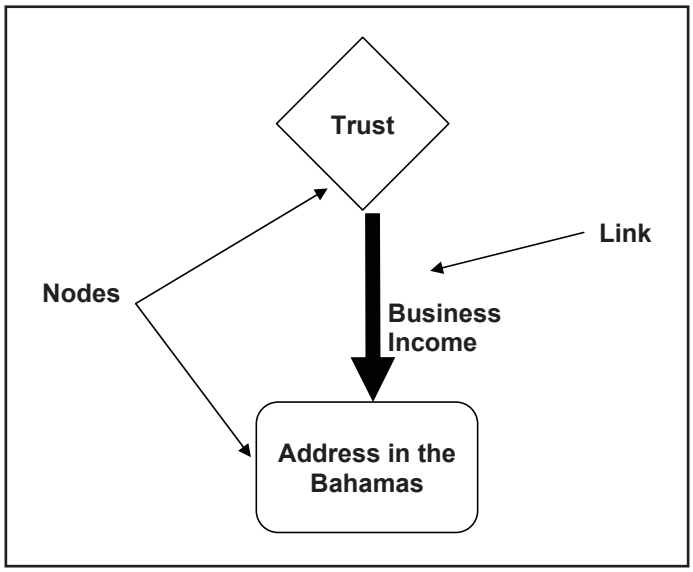

Figure 3

A more complicated, but still oversimplified, version of an ATAT can be demonstrated in the following scenario:

- Suppose that Entity A makes a significant gain (say $100M) on the sale of a business and does not wish to pay the tax on the gain.
- A creates a wholly owned S corporation, Entity B. As a flowthrough entity, the profits and losses of B pass through to the owner A and are reported on A's tax return.
- A and B form a partnership with a third entity, Entity C. C is chosen in such a way that C has losses from another operation.

- A, B, and C in turn form a partnership entity, Entity D. Because of C's apparent business expertise, the partnership agreement allocates 100 percent of the profits to C and 100 percent of the losses to B.
- D then executes foreign currency transactions that generate a gain of $100M and a loss of the same amount, and at the same time. Thus, as the result of those two transactions, no money is gained or lost, but accounting records are created.
- As per the partnership agreement, C takes the paper gain but pays no taxes on it because the losses from its other operation offset the gains.
- B takes the paper loss of $100M which flows through to A—this is only an artificial loss because the currency trades canceled each other.
- Thus, A receives 100 percent of the tax loss which offsets the actual gain that A made.

This transaction can be modeled by the graph shown in Figure 4. In an actual abusive transaction, additional specifics may be very important to the overall identification. Items like the size and type (i.e., ordinary income versus capital gain) of dollar amounts; the number and type of entities involved; the State or country of each entity; return characteristics like initial year and/or final year; and even name or industry) can all be critical components of the ultimate pattern.

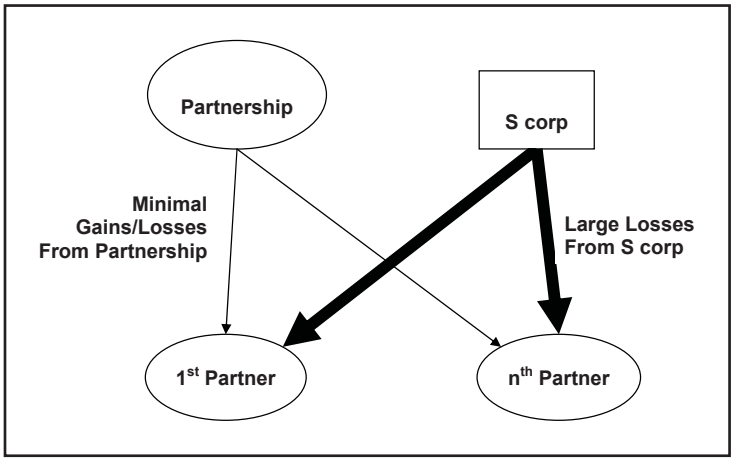

Figure 4

Entities involved in an ATAT, and depicted by nodes in a graph, can represent one of several possible entity types from the tax domain: individuals (Form 1040), businesses (Form 1120), partnerships (Form 1065), S corporations (Form 1120S), trusts (Form 1041), locations (any form with an address), etc. Similarly, vertices can represent Schedule K1, affiliations (Form 851), etc.

Son of BOSS is an ATAT scheme that was once very popular. Figure 5 shows how a Son of BOSS scheme, involving a partnership (P), an S corporation (S), and two individuals (I), can be modeled using graphs.

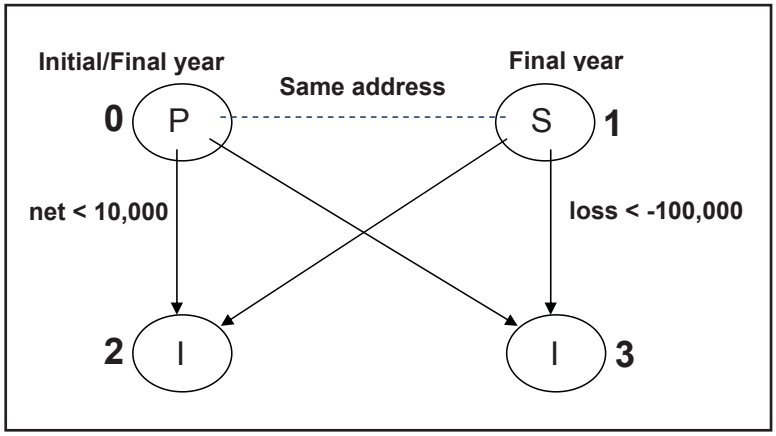

Figure 5

Graph Matching Process

Graph Query consists of three main components: the front-end (which provides an interface for the user to specify graphs), the graph query engine (which performs the task of finding matching graphs in a relational database), and the database itself (which holds the data of all the entities and their associations).

The process begins with the user specifying the pattern of interest in the form of a graph. This is accomplished via a drag-and-drop feature of the front-end interface. The user is presented with a palette of nodes (1040, 1120, etc.) from which nodes can be dragged onto a canvas. Nodes can further be customized by imposing conditions on them. Edges can be used to connect two or more nodes. Just like nodes, a palette of edges is presented to

the user, and they can be further customized by imposing conditions on them. A snapshot of the user interface with a graph drawn is shown in Figure 6.

The next step in the graph-matching process is to convert the user-defined graph into a language called the Graph Representation Language (GRL). GRL is a powerful language that is used to describe a graph. It includes notations to specify nodes and links, along with conditions and constraints on them. Users comfortable with GRL can finetune the graph and its conditions—this creates a more powerful graph, something that may not be possible via the front-end.

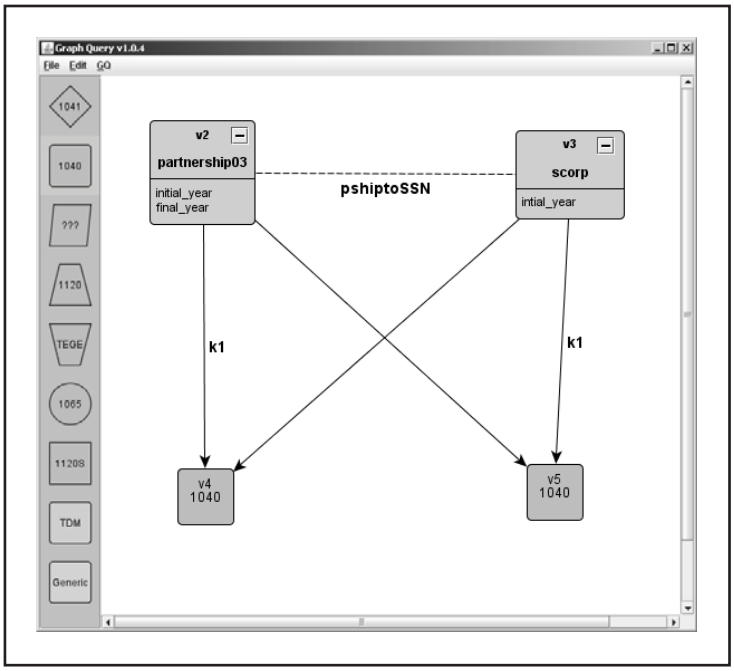

Figure 6

A complete GRL describing a graph consists of a sequence of statements. Each statement describes either a node and its conditions or an edge and its conditions. A fragment from a GRL representing the graph in the snapshot in Figure 5. Statements that begin with "v" represent the vertices, and those that begin with "d" represent links.

v 0 partnership where init_year and final_year;
v 1 scorp where init_year;
v 2 individual;
v 3 individual;
d 0 2 k1 where net < 10000;
d 0 3 k1 where net < 10000;
d 1 2 k1 where loss < –100000;
d 1 3 k1 where loss < –100000;

The function of the graph query engine is to take the GRL containing the description of a graph and to run queries against the database to find matching entities. In order to accomplish this, the engine transforms GRL into another language, termed intermediate language (IL). The reason behind this is to replace the user-defined node and edge names and conditions with the actual table and column names from the database.

The IL bears a strong resemblance to the database language SQL. Each line in the GRL becomes a query to the database. To optimize the processing of queries, the statements in the IL are arranged according to the number of records that each statement is likely to retrieve. Each statement in the IL is then translated into SQL and executed against the database after which a list of entity TINs is returned.

These TINs become the input to the next IL and so on. Finally, the list of TINs returned by the last IL query would be of the entities that participate in the relationship described by the graph that was input to the tool. Figure 7 summarizes the process of processing a graph that was just described.

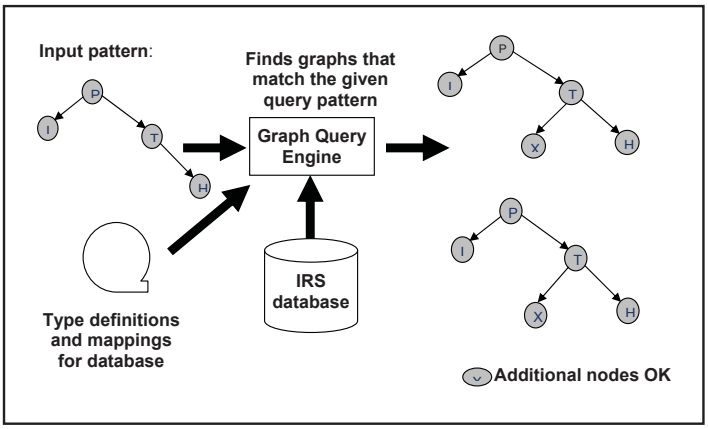

Figure 7

Conclusions and Future Work

Graph Query is a powerful tool in the modeling and detection of abusive transactions and the identification of entities that participate in such transactions. A part of its power comes from the fact that it enables end-users not familiar with database technologies to specify complex and sophisticated ATAT patterns. Further, the tool has the potential to uncover vast amounts of fraud and interesting ways that are being used to avoid paying tax. In addition, the tool can be applied to a variety of problem domains. For example, if it were possible to model the characteristics of individuals who are likely to have offshore accounts, Graph Query could be used to find such individuals by changing the database against which the queries are executed.

There are many future avenues that can be pursued with Graph Query. One such opportunity involves the problem of frequent substructure discovery. As opposed to giving the tool a pattern and asking it to find entities that participate in that pattern, in this particular case, the tool is used to search a database for patterns that seem to be occurring frequently without knowing in advance what they look like. Some of these may well turn out to be ATATs that were not discovered before.

Also, enforcement workload selection can be aided by the concept of enterprise risk (rather than simply the risk associated with a single return). In this situation, once an enterprise has been defined, it may be possible to define the concept of risk associated with an enterprise. Workload selection processes will then involve looking for enterprises with the greatest risk. Graph Query could then be used to identify enterprises in the database that meet or exceed a specified risk threshold. As Graph Query is used in more situations, there will be many more problem-solving opportunities where it can be applied.