Constrained Line Of Identity: An Approach To Conditional Joins *

Harry S. Delugach and Thomas H. Hinke

Computer Science Department
The University of Alabama in Huntsville
Huntsville, AL. 35899
Email: delugach@cs.uah.edu
Phone: (205) 895-6614
FAX: 895-6239

Keywords: Conceptual graphs, line of identity, co-referent link, implicit referent, embedded referent, type definition expansion/contraction, conditional co-referent

Abstract. Conceptual graphs support a notion called a "line of identity" (also called a "co-referent link") that specifies a single concept which is denoted by two or more concept boxes. The multiple concepts can be joined into a single concept (if they are in the same context). This work proposes a richer notion, expressible using existing conceptual graph constructs, that allows finer distinctions to be made about what concepts are allowed to be joined, and also addresses the problem of joining between different contexts. In the course of describing an entire relational database with multiple graphs (one for each database relation). We encountered a problem in joining these graphs into one large graph (the universal join relation); namely, the existence of some instance constraints on joins. In some cases, foreign key attributes in one relation's graph cannot be joined unconditionally with their corresponding attributes in another relation. We describe a new notion, a conditional join, to handle this problem and others as well. We provide a rationale, description and examples of such a join.

1 Introduction

Conceptual graphs support a notion called a "line of identity" (also called a "co-referent link") that denotes a single concept represented by two or more concept boxes. The multiple concepts can be joined into a single concept (if they are in the same context). In the course of our work, however, we have encountered some limitations with the co-referent link as described in [Sow84]. John Esch has described other problems [Esc92, CGl]. This work proposes our solution, expressible using existing conceptual graph constructs, that allows finer distinctions to be made about what concepts are allowed to be joined, and also addresses the problem of joining between different contexts.

* This work was supported under U.S. Department of Defense, Maryland Procurement Office Contract No. MDA904-92-C-5146.

This research arose during the AERIE research project, whose purpose is to develop a classification of various types of inferences using the AERIE model and to apply conceptual graphs to the inference detection problem: i.e., how to prevent the inferring of classified information from unclassified information [DHC93, HDC93]. In the course of our work, we have encountered the need to represent lines of identity that are dependent upon other graphs; i.e., the line of identity cannot be drawn between instances without their meeting some additional constraints. We have solved this problem using what we call a *constrained line of identity* or CLOI. It is not an addition to existing conceptual graph notation, merely a new use for existing constructs.

This paper describes our development of the CLOI notion. We outline problems with the current interpretation of the line of identity, and describe the analysis example with which we are working. We then show unconditional joins through examples in order to set the stage for describing the problem of conditional joins, in the context of combining a set of graphs from database relations. We then discuss the CLOI notion and present some issues we have discovered concerning its application.

2 Problems With Lines Of Identity

The co-referent link has been studied in depth by John Esch [Esc92]. In his work, he deals generally with lines of identity that link concepts in two different contexts, where one dominates the other. His examples illustrate several uses of lines of identity. In our work we are interested primarily in lines of identity connecting concepts in the same context; as we explain later, in our work we obtain many graphs which are conveniently linked together through lines of identity.

A co-referent link connects two or more concepts, as in Fig. 1(a). Under normal circumstances, a co-referent link within the same context connects two or more concepts that can be joined unconditionally; i.e., the concepts refer to the same real-world thing. The two or more concepts can therefore be joined into the same concept, thereby joining the two or more graphs in which they appear, as shown in Fig. 1(b). Concepts X, Q, and Y can all be joined together as long as their referents do not conflict (if instance graphs are considered).

Some problems arise from using lines of identity. We will now describe two important problems: the first is where a line of identity may or may not be drawn depending on individual referents, and the second is where a line of identity actually links embedded co-referents.

2.1 Generic Vs. Individual Graphs

The problem of lines of identity depending on individual referents came about through our work in representing a set of database relations in conceptual graphs. For our application purposes, we began by focusing on a particular database that is to be inference analyzed. Our analysis involves the collection of various types of inference-relevant data by focusing on one relation at a time. This collection we call a *microanalyzed knowledge chunk* or MKC. The details of MKC's are beyond the scope of this paper; more information can be found in [HDC93, HDC94]. With

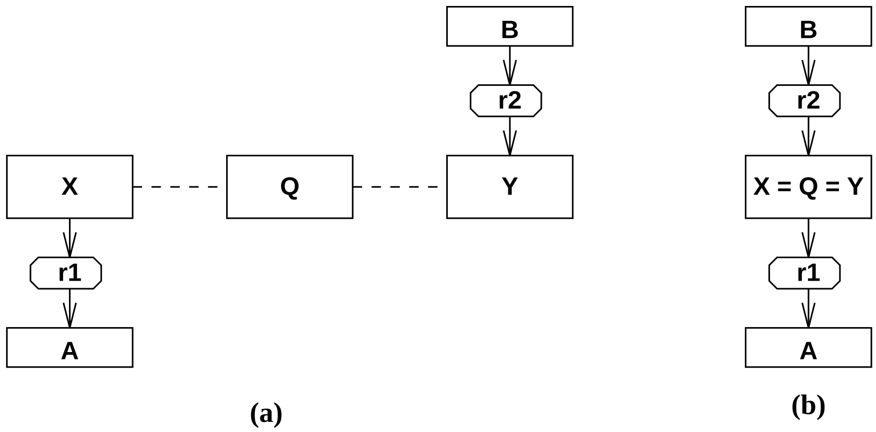

Fig. 1. Unconstrained Line of Identity.

respect to this paper, the relevant results of this microanalysis are a set of graphs that are associated with different relations in the database.

Even though the AERIE project uses the database relation as the focus of an these graphs, we have argued elsewhere [HDC94] that ours is a general approach that can be used in a wide variety of knowledge acquisition and representation applications. All that is required is to have a knowledge engineer in possession of the AERIE methodology described in [HDC94]. For this paper, it is sufficient to be aware of the existence of a set of related graphs with some shared concepts.

As an example database, we will present a sample company that stocks heavy equipment and parts. Its database consists of a number of relevant relations; we will consider only three of them.

The relation Parts Catalog, shown in Fig. 2, is the schema for a relation that contains the catalog of parts stocked by the company. Note that PartNo, being underlined, is a primary key for the relation. The corresponding conceptual graph representation of the schema is shown in Fig. 3(a). Two instances of the Parts Catalog relation are shown in Fig. 3(b) and (c). Note that generic concepts of the schema in Fig. 3(a) are replaced by individual concepts in Fig. 3(b) and (c).

Parts Catalog Relation				
PartNo	**Description**	**Class**	**WholeItem**	**UnitPrice**
5500	Cotton Picker	005	YES	35467.00
G874-22	12V Battery	160	NO	45.67
43729C	Spindle	505	NO	1550.00
...	...	...	...	...

Fig. 2. Parts Catalog Relation Schema and Instances.

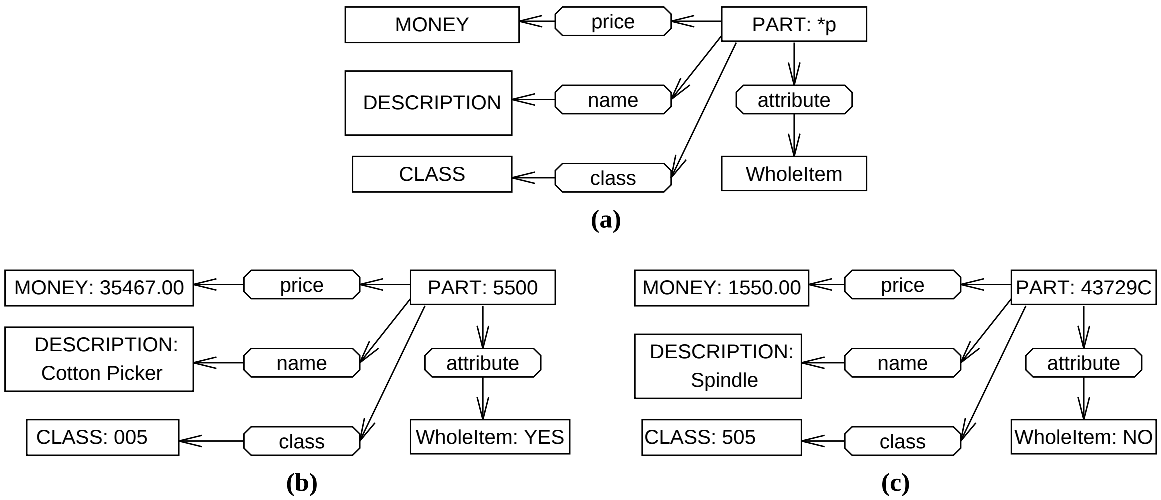

Fig. 3. Graphs Of The Parts Catalog Relation.

The company database contains information about stock on hand, keyed by part number, as shown in the relation Parts Inventory in Fig. 4. The corresponding conceptual graph representation of the schema is shown in Fig. 5(a). An instance of the Parts Breakdown relation is shown in Fig. 5(b).

Parts Inventory Relation			
PartNo	**Site**	**OnHand**	**OnOrder**
5500	Huntsville	4	3
G874-22	Nashville	39	25
43729C	Lincoln	7	4
...	...	...	...

Fig. 4. Parts Inventory Relation Schema and Instances.

An equipment company's database will also contain information about the parts breakdown or "explosion" for each whole good item. That is, a composite part's component parts will be known. Sometimes the breakdown is shown in a drawing to show the relative physical arrangement of parts in the whole; we assume that the part-of information is kept in a database relation, as shown in Fig. 6. The corresponding conceptual graph representation of the schema is shown in Fig. 7(a). An instance of the Parts Breakdown relation is shown in Fig. 7(b).

We can now characterize the problem of co-referent links depending on instances by considering the two schema graphs in Fig. 5(a) and Fig. 7(a). Since there is a [PART] concept in both graphs, they could normally be joined around the [PART] concept to form a single graph. This resulting graph would be incorrect, however,

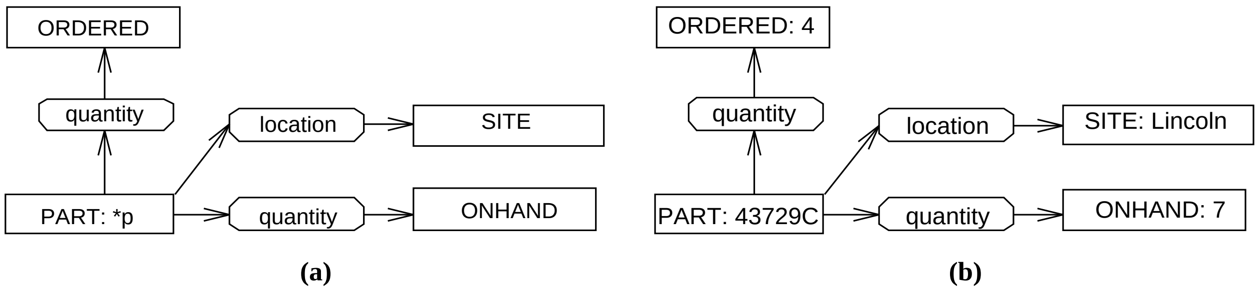

Fig. 5. Graphs Of The Parts Inventory Relation.

Parts Breakdown Relation		
WholeNo	**PartNo**	**QtyOfParts**
5500	43729C	1
5500	G874-22	2
...	...	...

Fig. 6. Parts Breakdown Relation and Instances.

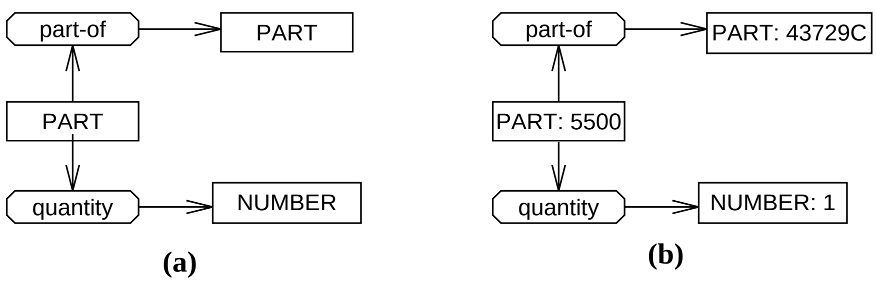

Fig. 7. Graphs Of The Parts Breakdown Relation.

because we would then be able to instantiate (from the database) individual graphs whereby any part would be a part of some whole. We must somehow constrain the join so that we can only join parts which are *bona fide* components of some other part. We will show in the next section that the constrained line of identity easily addresses this problem.

2.2 Embedded Co-Referents

The problem of embedded (or implicit) co-referents has been identified by John Esch and others [CGl]. The problem is illustrated by the difference between the two derivations shown in Fig. 8 and in Fig. 9.

In Fig. 8, there is no problem. Starting with Fig. 8(a), we can split the concept [STUDENT: Linda], linking the two copies via a line of identity as in Fig. 8(b).

Given the type definition for HONOR-STUDENT in Fig. 8(c), we obtain the graph in Fig. 8(e); likewise, given the type definition for SENIOR in Fig. 8(d), we obtain the graph in Fig. 8(f). This derivation is unambiguous because both definitions' base concepts are the concepts that are linked by a line of identity.

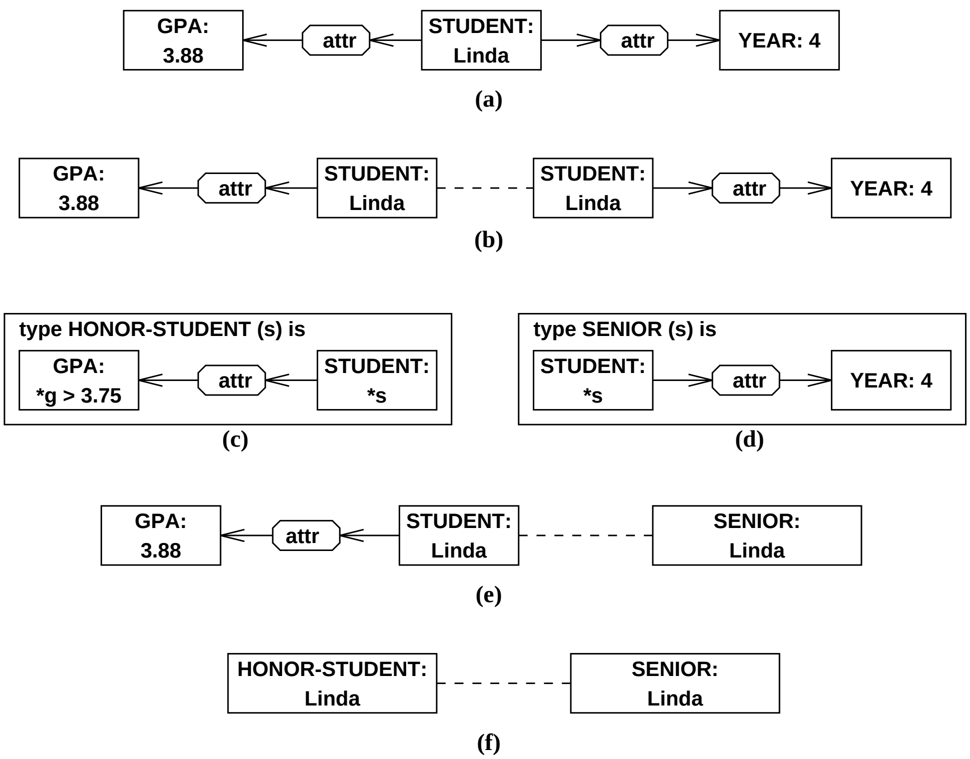

Fig. 8. Type Contraction Involving Base Concepts.

In Fig. 9, however, there is a problem. As before, we can split the concept [STUDENT: Linda] and contract the left part of the graph using the HONOR-STUDENT type definition in Fig. 9(c). We can also contract the right part using the type definition for SENIOR-YEAR in Fig. 9(d). The problem with the line of identity now becomes apparent. The base concepts of each are different than the lined concepts. It is certain that *something* is being linked, but what? We should not be able to simply draw a line of identity, as shown by the ?? in Fig. 9(e) because Honor-Student Linda and Senior-Year 4 are clearly not the same concept, Yet the rules of the line of identity would allow us to join them.

Esch [Esc92] proposed labeling the line of identity with the symbolic referent from the definition, as in Fig. 9(f), to indicate that the line of identity links the concept [HONOR-STUDENT: Linda] with an embedded concept whose referent is *s embedded in the definition of SENIOR-YEAR. Here the s indicates that the co-reference is not to the referent of [SENIOR-YEAR] but to the concept of its differentia [STU-

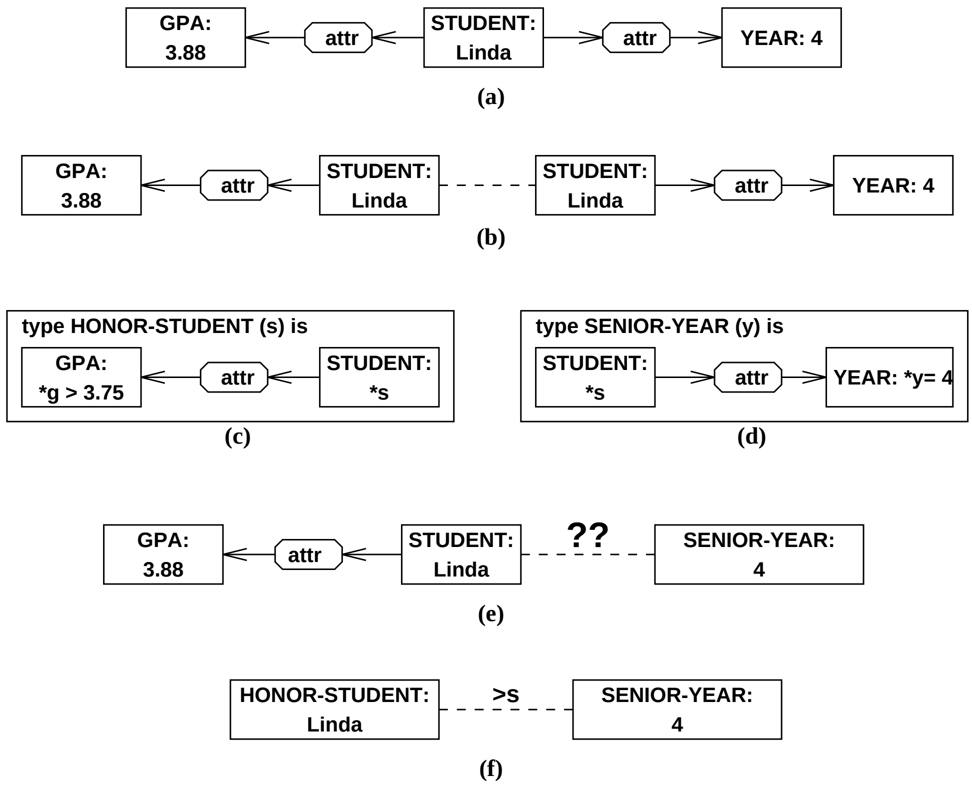

Fig. 9. Type Contraction Problems Using Lines Of Identity.

DENT] with s as its variable. Esch notes the obvious undesirability of its syntactic clumsiness, but there are additional problems with this approach that are even more important. These additional problems are illustrated by Fig. 10.

The derivation steps of Fig. 10(a) through (e) are similar to the steps of Fig. 8. From the definition in Fig. loi-defn-2-contraction(f), the graph in Fig. loi-defn-2-contraction(e) can be rewritten (using the mechanism proposed by Esch) by further labeling the line of identity with the referent d of [T6: *d]. The d is now ambiguous; to which d does it refer? The d could refer to the referent of [T2] in Fig. loi-defn-2-contraction(b) and (d), or to the referent of [T6] in Fig. loi-defn-2-contraction(e) or (f). Even if the variable names were unique, a lesser problem is the notational ugliness of having more than one label on a co-referent link.

Esch's approach involves adding a new meaning to variable names in type definitions. Determining a name for each concept's referent in a type definition creates a naming problem analogous to pikcing variable names in a programming language with dynamic scoping; i.e., a reference to a local variable may occur in ANY scope. This means that either (a) some global renaming process would have to occur (to ensure uniqueness), or (b) no symbolic referent (e.g., *y) could be used more than once!

There is an important theoretical problem here as well. The label on the line

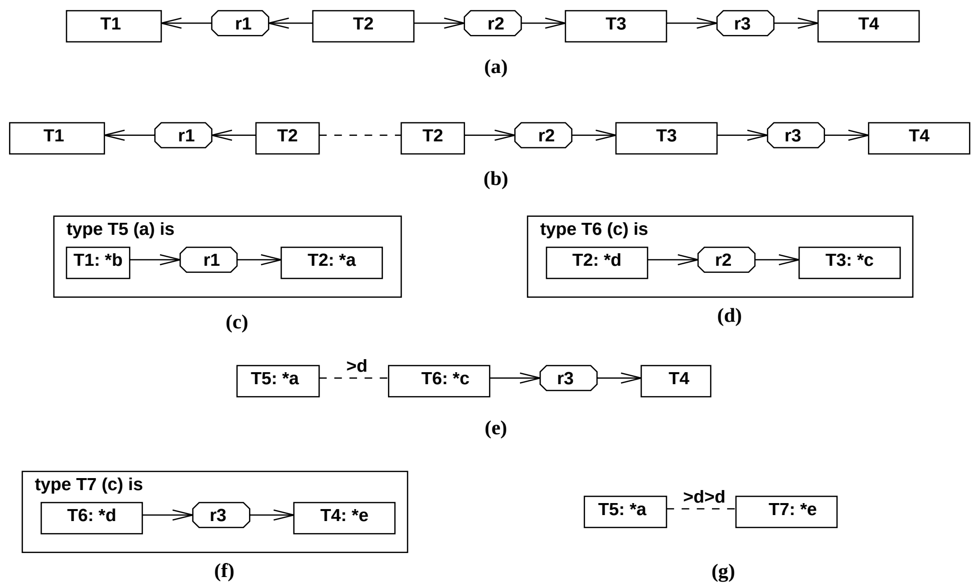

Fig. 10. Problems With Labeled Lines Of Identity.

of identity actually contains meta-information that has no meaning for the graphs as shown. If a concept actually denotes more than it represents, then using ...>d... conveys the intent: *I am explicitly referring to otherwise-undenoted internal details of some concept.* This issue is analogous to allowing accessing implementation details of some abstract data type in a programming language. That is, should we be allowed access to internals that we aren't showing? In programming languages, such practices are generally discouraged; we argue below that such access to internals should also be discouraged in conceptual graph notation.

These problems can be better addressed by the constrained line of identity (CLOI) which we are proposing in this paper. The next section describes the CLOI and shows our solutions to these two problems.

3 Constrained Line Of Identity

The previous section showed two difficulties with the line of identity (or co-referent link) as presently interpreted. This section outlines the constrained line of identity as our proposed solution to the problems we have mentioned, and shows how the problem examples would be expressed.

3.1 Unconstrained vs. Constrained Line of Identity

A constrained line of identity connects two or more concepts via some concept that appears in a constraint. For example, in Fig. 11, concepts X and Y are connected to concept Q which appears inside a constraint. The meaning of this construct will be based on applying the beta rules. As shown, X and Y cannot be joined, even if their referents are compatible, because Q is in a nested context. In the example, if P were asserted in the same context as X and Y, then P could be removed from inside the negated context (through deiteration) and the two negations could be removed, thereby placing Q in the same context as X and Y. Since Q now appears in the same context as X and Y, and all three are connected via a line of identity, then all three concepts can be joined.

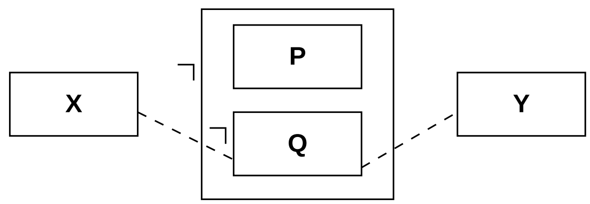

Fig. 11. Constrained Line of Identity.

This is a constrained line of identity because the line of identity only exists between all three concepts if P is true. Therefore the truth of P forms the condition under which the line of identity forms a true co-referent link.

3.2 Handling Generic Vs. Individual Graphs

In this section, we show how we join the graphs derived from the relations. We assume in the parts inventory, that the primary key PartNo represents the same information as PartNo in the parts catalog. We therefore can draw a line of identity connecting the two schema (generic) graphs through the [PART: *p] concept, as shown in Fig. 12(a).

Because these two graphs are in the same context (i.e., level 0), we can perform an unconditional join on the two instance graphs as well, combining the two [PART: 43729C] concepts to form the single graph of Fig. 12(c).

Contrast this situation with the more interesting one; namely, where lines of identity can be drawn, but only between concepts in certain instance graphs. For example, the part number of the Parts Breakdown relation can be joined with the part number of the Parts Catalog – but not in all cases! A cotton picker has parts, but a battery does not. If we drew a simple line of identity, this constraint would not be captured. One solution (albeit tedious and inefficient) is to enumerate all the instance graphs, and draw lines of identity only where a whole-item's catalog graph appears. A more concise solution is to use the constrained line of identity. Fig. 13 shows how the constraint is denoted. In this graph, the IF-THEN rule of the negated context (interpreted through the beta rules) means: *If the part "p" has a WholeItem attribute with value "YES," then the part "q" can be asserted in the outer context.*

Given the constraint in Fig. 13, we can show an instance from the two relations, one from Fig. 3(b), the other from Fig. 7(b). Fig. 14(a) shows the schema instantiated for an instance of Part 5500. Fig. 14(b) shows the negated context after insertion of the left-hand graph, followed by a join inside the negated context. Fig. 14(c) shows

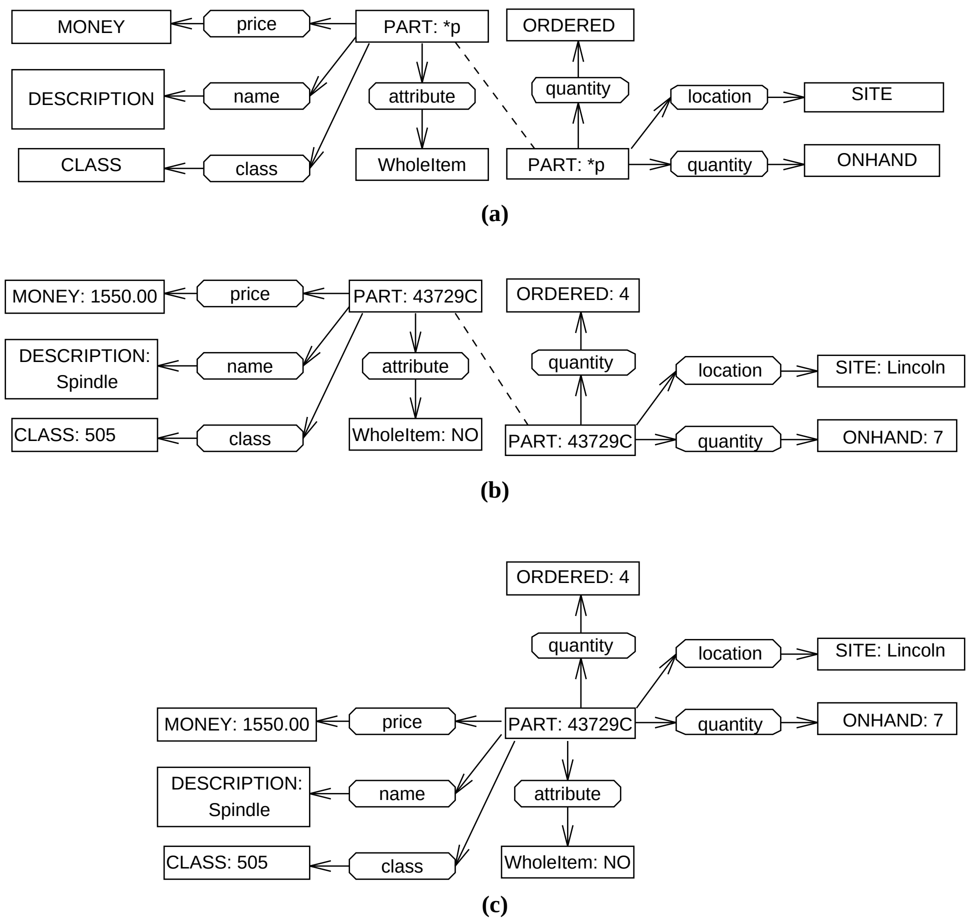

Fig. 12. Unconstrained Line Of Identity Between Same Context Graphs.

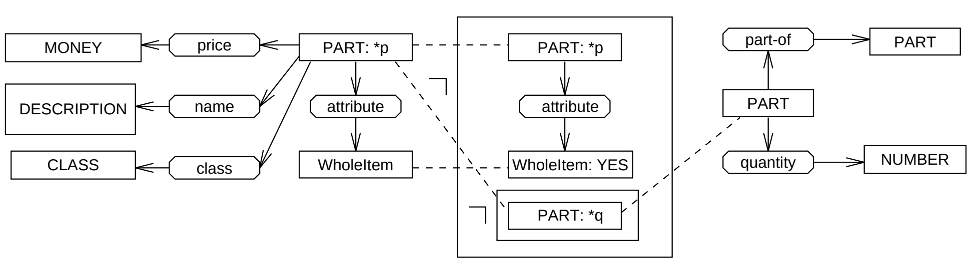

Fig. 13. Constrained Line Of Identity Between Generic Graphs.

the graph after deiteration from inside the negated context, and finally Fig. 14(d) shows removal of the double negation to perform the join.

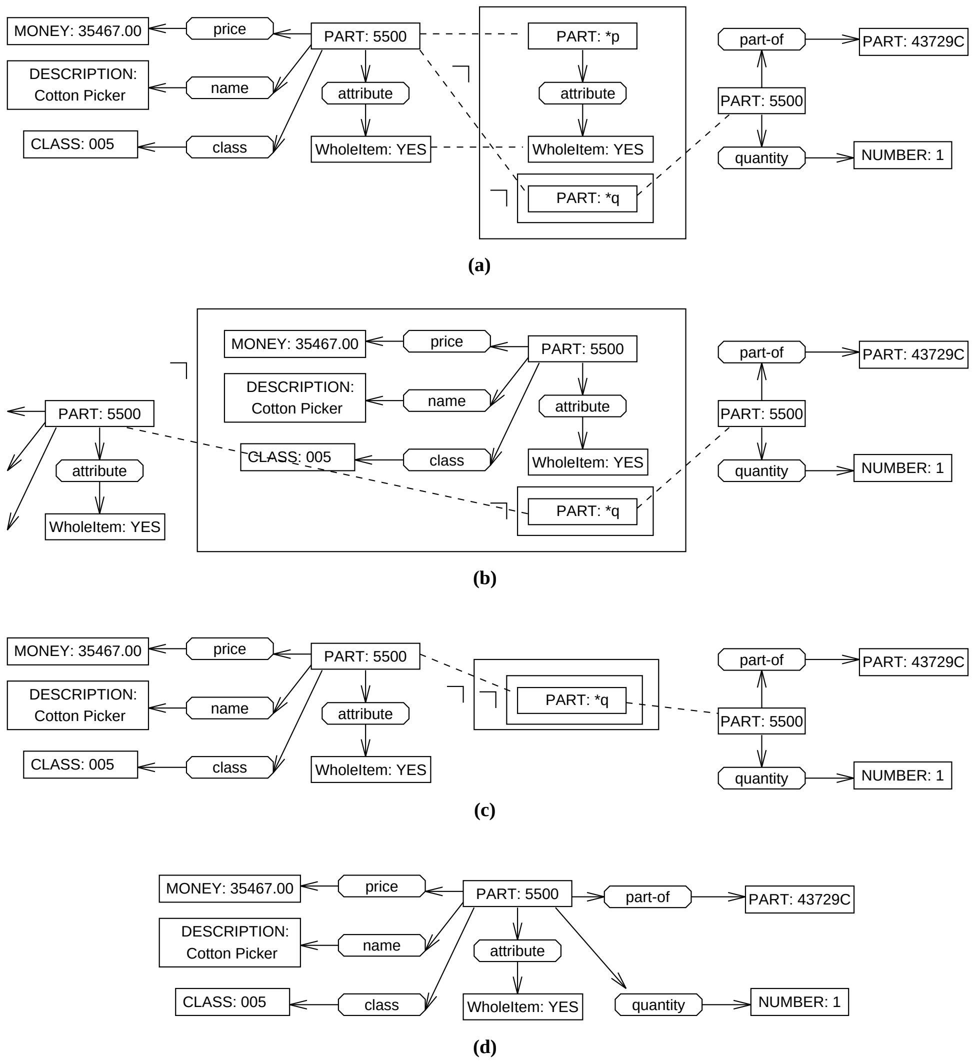

Fig. 14. Removal Of Constrained Line Of Identity Between Instance Graphs.

3.3 Handling Embedded Co-Referents

The handling of embedded co-referents has several philosophical approaches. The most restrictive philosophy would be to assume that a line of identity may only link a base concept of a definition; if the base concept does not participate in the line of identity before contraction, then the contracted result has no line of identity. This policy is quite safe, since we never have to deal with links between any but a true set of co-referents.

The restrictive policy unfortunately also results in the loss of some information. Some would argue that this is in fact what definition contraction is supposed to do; namely, denote a higher level of abstraction. It should be pointed out that removing the line of identity unless base concepts are involved also means that the process of contraction followed by expansion is not symmetrical.

A second approach is to use Esch's solution of providing meta-information on the co-referent link. This solution distinguishes lines of identity involving base concepts from lines of identity which do not. As previously noted, however, this approach has several drawbacks, such as requiring unique naming to avoid ambiguities and allowing access to internals of a concept, not to mention its modification of the basic syntax of conceptual graphs.

Our approach is to use the existing definition of the line of identity, and preserve its semantics; namely, that co-referent links are only allowed to link concepts that can be joined. The conditional line of identity serves this purpose, as shown in Fig. 15. The meaning of the negated context alone in Fig. 15(a) is as follows: *If the T6 called "c" exists, then the T2 called "d" exists.* This corresponds to its definition; namely that there is a T2 in T6's definition. Since T2 is coreferent with T5, we can also say that T5 exists. Likewise Fig. 15(b) shows the corresponding rule from T7's definition.

Note there are actually three lines of identity! This is required by our philosophy that only true co-referents may be linked by a single line of identity. All the lines of identity conform to the already-existing rules for their use; no additional semantics or syntax is required.

4 Discussion

We have identified several issues with respect to the constrained line of identity (CLOI).

Despite its usefulness, the constrained line of identity also suffers from a lack of symmetry in derivations with implied co-referents. Although special rules could be devised for specifically interpreting the IF-THEN rule along the constrained line of identity, in fact the IF-THEN rule is just another subgraph in a larger graph whose meaning must be conveyed by its entire contents. It is therefore possible that the original line of identity may not re-appear in its original simple form.

There are some problems in expressing the CLOI in the linear form of conceptual graphs. Ordinarily, co-referent information is conveyed by sharing an identical symbolic referent. Fig. 11 would hence be written as:

```
[X: *a]  [ [P]  [Q: *a] ] [Y: *a]
```

In the CLOI, the common referent is somewhat misleading since two common referents are not unconditionally co-referent. Of course, being in different nested contexts changes the meaning, but nested contexts are harder to understand in the linear form. Although not a central issue for the use of CLOI's, it is important for practical purposes that any construct we can draw using the display form also be

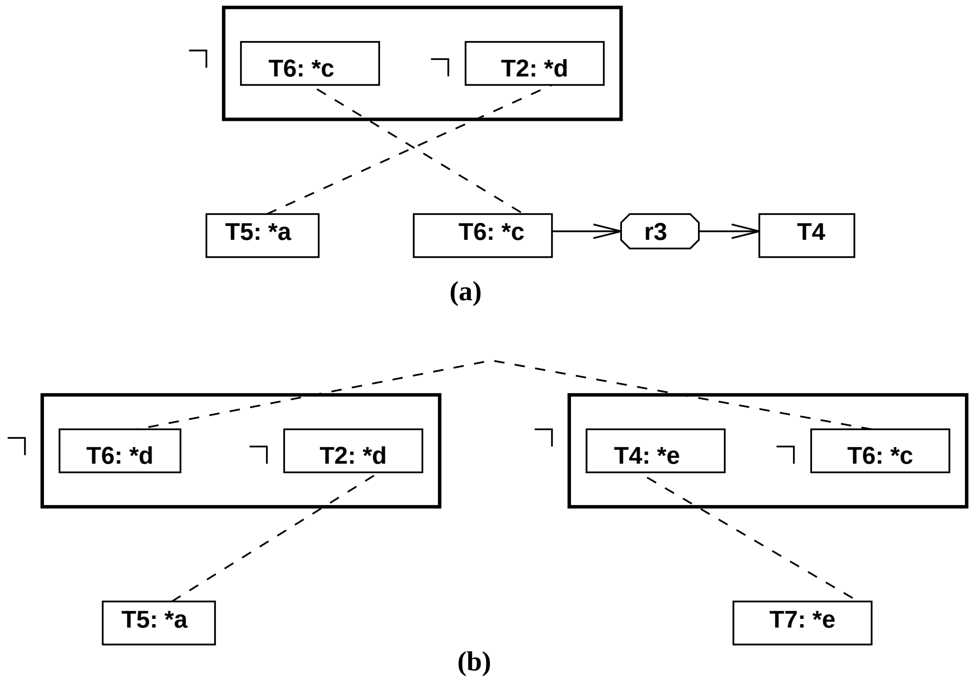

Fig. 15. Representing Embedded Co-Referents Using Conditional Line Of Identity.

expressible in the linear form. The = sign in a referent has been proposed to handle the problem [Sow84].

The constrained line of identity expresses an important class of relationships in conceptual graphs that are not easily expressed with the original co-referent link idea. Without adding any new syntax or semantics, the constrained line of identity captures the notion of conditional co-reference in a natural and easily understood way.

Two important problems are solved using the constrained line of identity: instance-dependent co-referents and embedded (implicit) co-referents. For the first problem, it provides more power and precision than a mere enumeration of instance graphs (some of which would have a line of identity, some would not). In time, we would expect that a constrained line of identity would become a well-known idiom in conceptual graph structures, similar to the $\neg(P \vee \neg Q)$ idiom for if-then rules.

References

[CGl] Conceptual Graph Electronic Mailing List (cg@umn.cs.edu). Contact Bosco Tjan (tjan@cs.umn.edu).

[DHC93] Harry S. Delugach, Thomas H. Hinke, and Asha Chandrasekhar. Applying Conceptual Graphs for Inference Detection Using Second Path Analysis. In *Proc. ICCS93, Intl. Conf. on Conceptual Structures*, pages 188–197, Laval University, Quebec City, Canada, Aug. 4-7 1993.

[Esc92] John Esch. The Scope Of Co-reference In Conceptual Graphs. In Heather Pfeiffer, editor, *Proceedings of the 7th Annual Conceptual Graphs Workshop*, pages 129–138. Springer Verlag, 1992. Las Cruces, NM, U.S.A., July 8-10.

[HDC93] Thomas H. Hinke, Harry S. Delugach, and Asha Chandrasekhar. Layered Knowledge Chunks For Database Inference Detection. In *Proc. 7th IFIP WG 11.3 Working Conference on Database Security*, Huntsville, Alabama, Sept. 1993.

[HDC94] Thomas H. Hinke, Harry S. Delugach, and Asha Chandrasekhar. Microanalyzed Knowledge Chunks For Database Inference Detection. *Jour. Computer Security*, 1994. (submitted).

[Sow84] John F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine.* Addison-Wesley, Reading, MA, 1984.

This article was processed using the LaTeX macro package with LLNCS style