All for One, One for All

by

C. J. Date

Mathematically, there is a correspondence
between mappings and relationships

——from reference [8]

May 31st, 2006

ABSTRACT

This paper has a lot to say about perhaps not very much. Its aim is to pin down the exact nature of one-to-one, many-to-one, one-to-many, and many-to-many relationships. In particular, it tries to come up with precise definitions for these concepts. The paper might thus be dismissed by some as mere pedantry; it might also be accused of making very heavy weather over what are essentially very simple concepts. But a survey of the literature certainly betrays confusion and lack of systematic thinking in this area, and as I've had occasion to remark elsewhere, in the field of computing in general, and in the field of database management in particular, clarity, accuracy, and precision are surely paramount; without them we're doomed. So you be the judge.

INTRODUCTION

It might just be me, but I'm never quite certain what people mean when they use expressions like *one-to-one relationship, many-to-one mapping,* and so forth. For example, consider this quote from reference [5]:

> There is a one-to-one ... relationship between flight segments and aircraft. A particular flight segment on a given day uses one and only one aircraft.

Well, clearly a particular flight segment on a given day does use one and only one aircraft. Surely, however, a particular aircraft can be used on one, two, ... or any (reasonable) number of flight segments on a given day——possibly even none at all? So isn't the relationship not one-to-one but, rather, many-to-one ("Many flight segments on a given day use a given aircraft"), where "many" includes the zero case? Or is it perhaps one-to-many ("A given aircraft is used on many flight segments on a given day"), where again "many" includes the zero case?

This paper is an attempt to clarify such matters and inject a little rigor into the terminology. *Note:* I should warn you that some of the definitions in what follows are inevitably a little complicated, or at least might appear so on a first reading. *Caveat lector.*

GROUNDWORK AND ASSUMPTIONS

I'll begin by laying a little groundwork and spelling out some assumptions.

- I assume you have an intuitive understanding of what's meant by the term *relationship*. Please note that I'm not using that term in any loaded kind of way; in particular, I'm not giving it any specifically relational interpretation, in the sense of the relational model, nor am I using it in any specific "entity/relationship modeling" kind of sense (if there can be said to be any such sense).

- Mention of the term *entity* reminds me: It's possible, and I think highly desirable, to discuss this subject without using that fuzzy term at all. As you'll quickly find, therefore, my definitions (of *relationship* and the like) are all phrased in terms of the mathematical concept of a *set.* That term itself can be defined as follows:

  set A collection of objects, or elements, with the property that given an arbitrary object *x,* it can be determined whether or not *x* appears in that collection.

  In particular, observe that the mathematical term (and the term I'll be using) for the objects that are members of sets is *elements*. *Note:* Many of the definitions in this paper, including the foregoing definition of the term *set* in particular, are taken from reference [2], as are many of the examples also (at least the more mathematical ones).

- For simplicity, I assume all relationships are binary. All of the ideas to be discussed extend to ternary, quaternary, etc., relationships in a straightforward manner.

- For definiteness, I assume all relationships are directed. As a consequence, if there's a relationship from *A* to *B,* there's an inverse relationship (also directed, of course) from *B* to *A* as well. *Note:* It would be possible to talk of relationships as bidirectional and thus avoid the need to introduce the notion of inverse relationships, but I think clarity would suffer.

- As in the introduction, I'll take "many" to include the zero case throughout (i.e., "many" means zero or more), barring explicit statements to the contrary.

  Here then is a generic definition of the term *relationship:*

relationship Let *A* and *B* be sets, not necessarily distinct. Then a relationship from *A* to *B* is a rule pairing elements of *A* with elements of *B.* (Equivalently, we might just say the relationship is the pairing itself.)

Note: Since I'm taking relationships to be directed, I do think it makes better sense to think of a relationship as being *from* one set *A to* the other set *B* than it does to think of it being *between* those two sets. Also, as already indicated, I'll distinguish the relationship in question from its inverse, which is a relationship from set *B* to set *A.*

By way of illustration, let *A* be the letters of the alphabet A-Z and let *B* be the digits 0-9. Then the following pairing (spelled out in detail on telephone handsets) is a well known example of a relationship from *A* to *B:*

nil	0
nil	1
A-C	2
D-F	3
G-I	4
J-L	5
M-O	6
P-S	7
T-V	8
W-Z	9

There's also an inverse relationship from *B* to *A* (in which, as you can see, it happens that certain elements of *B* are actually paired with no elements of *A* at all).

I'll refer to the foregoing example as "the telephone example" in what follows.

HOW MANY CASES ARE THERE?

Given the notion of a relationship as I've defined it, along with the notion that any given relationship has an inverse relationship, how many distinct kinds of relationships——or combinations thereof, rather——can reasonably exist involving two sets *A* and *B?* Well (using lowercase *a* and *b* to refer to arbitrary elements of *A* and *B,* respectively), it should be clear that:

- For a given *a* there could be at most one *b;* exactly one *b;* at least one *b;* or many *b*'s (i.e., M *b*'s for some M $\geq$ 0).

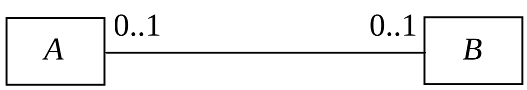
- In each of the foregoing cases, for a given *b* there could be at most one *a;* exactly one *a;* at least one *a;* or many *a*'s (i.e., M *a*'s for some M $\geq$ 0).

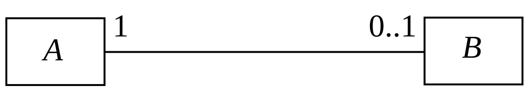
On the face of it, therefore, there are 16 possible combinations. Fig. 1 presents a summary of the 16 cases in the form of a matrix; Fig. 2 shows them pictorially, as they might be represented in, e.g., a

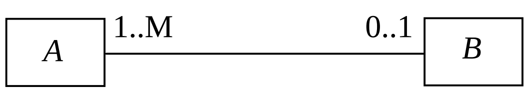
UML diagram——though I should say in the interests of accuracy that the notation in Fig. 2 is not exactly that of UML, though it's close (see reference [1]).

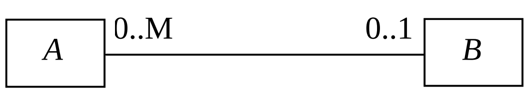
a has ↓ / *b* has →	at most one *a*	exactly one *a*	at least one *a*	M *a*'s ($M \geq 0$)
at most one *b*	Case 1.1	Case 1.2	Case 1.3	Case 1.4
exactly one *b*	Case 2.1	Case 2.2	Case 2.3	Case 2.4
at least one *b*	Case 3.1	Case 3.2	Case 3.3	Case 3.4
M *b*'s ($M \geq 0$)	Case 4.1	Case 4.2	Case 4.3	Case 4.4

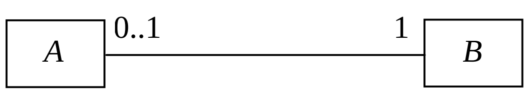
Fig. 1: The 16 cases in matrix form

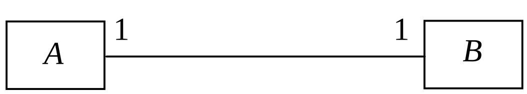
Case 1.1:


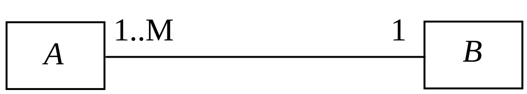
Case 1.2:


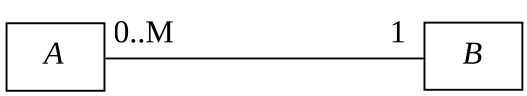
Case 1.3:


Case 1.4:


Case 2.1:


Case 2.2:


Case 2.3:


Case 2.4:


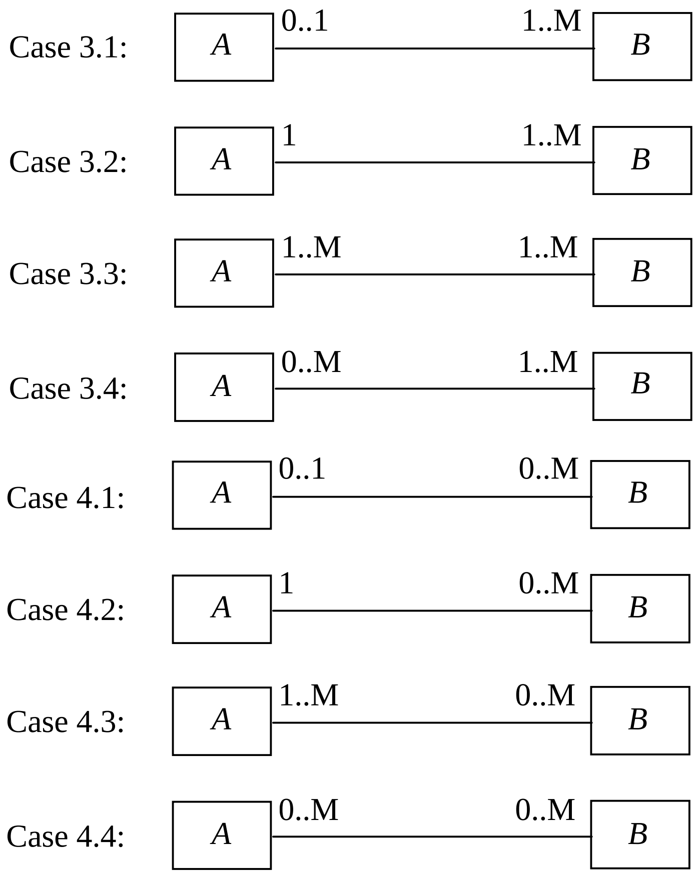


Fig. 2: The 16 cases in pictorial form

Let's consider a realistic—well, fairly realistic—example of each of the 16 cases. *Note:* You might want to make a copy of either Fig. 1 or Fig. 2 and use it as an aid to memory as you read through the remainder of this paper.

1.1 *Each* a *has at most one* b, *each* b *has at most one* a.

Example: At a given time, a given man has at most one wife and a given woman has at most one husband (assuming no polygamy); however, some men have no wife and some women have no husband.

1.2 *Each* a *has at most one* b, *each* b *has exactly one* a.

Example: In a given company at a given time, each employee manages at most one department and each department has exactly one manager (i.e., employee who manages the department); however, some employees aren't managers. *Note:* I'd like to point out in passing that this case occurs ubiquitously in connection with type inheritance as described in, e.g., reference [4]. For example, let *A* and *B* be types RECTANGLE and SQUARE, respectively, and let SQUARE be a subtype of supertype RECTANGLE. Then the supertype/subtype relationship is such that a given rectangle "has" a corresponding square if and only if the rectangle in question is in fact that square, and a given square "has" a corresponding rectangle that's precisely the rectangle that the square in question happens to be. (Of course, every square is a rectangle, but many rectangles aren't squares.) In this example, the relationship from RECTANGLE to SQUARE and the inverse relationship from SQUARE to RECTANGLE are both *identity* (i.e., "is the same rectangle as," or simply "is equal to").

1.3 *Each* a *has at most one* b, *each* b *has at least one* a.

Example: In a given company at a given time, each employee has at most one department (but some employees have no department at all) and each department has at least one employee.

1.4 *Each* a *has at most one* b, *each* b *has many* a*'s.*

Example: In a given company at a given time, each employee has at most one department (but some employees have no department at all) and each department has any number of employees (possibly none at all).

2.1 *Each* a *has exactly one* b, *each* b *has at most one* a.

Example: This is just Case 1.2 with *A* and *B* interchanged. Hence, an example is: In a given company at a given time, each department has exactly one manager (i.e., employee who manages the department), and each employee manages at most one department.

2.2 *Each* a *has exactly one* b, *each* b *has exactly one* a.

Example: In a given shipping company, a given shipment has exactly one corresponding invoice and a given invoice has exactly one corresponding shipment.

2.3 *Each* a *has exactly one* b, *each* b *has at least one* a.

Example: In a given company at a given time, each employee has exactly one department and each department has at least one employee.

2.4 *Each* a *has exactly one* b, *each* b *has many* a*'s.*

Example: In a given company at a given time, each employee has exactly one department and each department has any number of employees (possibly none at all). As another example, every person has exactly one biological mother, and every woman is the biological mother of zero or more children.

3.1 *Each* a *has at least one* b, *each* b *has at most one* a.

Example: This is Case 1.3 with *A* and *B* interchanged; see that case for an example.

3.2 *Each* a *has at least one* b, *each* b *has exactly one* a.

Example: This is Case 2.3 with *A* and *B* interchanged; see that case for an example.

3.3 *Each* a *has at least one* b, *each* b *has at least one* a.

Example: Every book has at least one author, every author (that is, every *book* author) is by definition author of at least one book.

3.4 *Each* a *has at least one* b, *each* b *has many* a*'s.*

Example: Every sporting event has at least one winner (possibly more than one if ties are allowed), every competitor is the winner of zero or more such events.

4.1 *Each* a *has many* b*'s, each* b *has at most one* a.

Example: This is Case 1.4 with *A* and *B* interchanged; see that case for an example.

4.2 *Each* a *has many* b*'s, each* b *has exactly one* a.

Example: This is Case 2.4 with *A* and *B* interchanged; see that case for an example.

4.3 *Each* a *has many* b*'s, each* b *has at least one* a.

Example: This is Case 3.4 with *A* and *B* interchanged; see that case for an example.

4.4 *Each* a *has many* b*'s, each* b *has many* a*'s.*

Example: In the well known suppliers-and-parts database, each supplier is located in the same city as zero or more parts and each part is located in the same city as zero or more suppliers.

So there aren't really 16 distinct cases after all but only 10. Notice, by the way, that several of the examples correspond to business policies or business rules (see, e.g., Cases 1.4 and 2.3). Others might be merely fortuitous (see, e.g., Case 4.4), but can be interesting nevertheless──some user might certainly want to "exploit the relationship," as common parlance has it, in order to determine which parts are located in the same city as a given supplier, for example.

Now, I opened this section by asking: How many distinct kinds of relationships can reasonably exist involving two sets *A* and *B?* Of course, the emphasis here is on *reasonably.* It would obviously be possible to classify the various "many" cases further (at most two, exactly two, at least two, at most three, and so on). To quote reference [8]: "For example, a meeting must have at least two people, and tango contests must have an even number of participants." However, such further classification doesn't seem particularly rewarding (it adds a lot of complexity, and the law of diminishing returns comes into play very quickly), and I won't consider such possibilities any further in this paper.

EXAMPLES OF CONFUSION

Now let's take a look at some actual quotes from the literature that demonstrate, at least arguably, a certain amount of confusion in this area. The first is a longer extract from reference [5] (the quote in the introductory section was taken from this extract, as you can see).

<quote>

For example, consider the following relationship descriptions:

- There is a one-to-one equipment-use relationship between flight segments and aircraft. A particular flight segment on a given day uses one and only one aircraft.

- There is a one-to-one departure relationship between flight segments and airports, and another one-to-one arrival relationship between flight segments and airports. Each flight segment has exactly one departure airport and one arrival airport.

- There is a one-to-many publication relationship between publishers and books. Each publisher can publish many books, and each book can be published by only one publisher.

- There is a many-to-many authoring relationship between papers and people. Each paper can be authored by one or more people, and each person can author one or more papers.

</quote>

Let's analyze each of these examples in turn. I'll repeat the examples to make the analysis easier to follow.

- There is a one-to-one equipment-use relationship between flight segments and aircraft. A particular flight segment on a given day uses one and only one aircraft.

As noted in the introductory section, the true situation here is as follows: Each flight segment ("on a given day") has exactly one aircraft; each aircraft has many flight segments (possibly none at all)——again, on a given day. Thus, we're dealing here with an example of Case 2.4 (or Case 4.2, if we look at the example in the inverse direction).

- There is a one-to-one departure relationship between flight segments and airports, and another one-to-one arrival relationship between flight segments and airports. Each flight segment has exactly one departure airport and one arrival airport.

Actually, each flight segment has exactly one departure airport, but each airport has many flight segments departing from it (where, I presume, "many" does *not* include the zero case); so I think we're dealing here with Case 2.3 (or Case 3.2). Similarly for arrivals.

- There is a one-to-many publication relationship between publishers and books. Each publisher can publish many books, and each book can be published by only one publisher.

Accurate analysis of this example depends on exactly what's meant by the term *book.* For example, most people would surely say that *Pride and Prejudice* is a book, but it's been published by many distinct publishers (even simultaneously; at the time of writing, it's available from several distinct publishers). And some books are published jointly by two or more publishers, anyway; I have in my own professional library at least one book published jointly by the ACM Press and Addison-Wesley. So I would say the situation is that each publisher has many books and each book has many publishers, where neither of those "many"s includes the zero case: Case 3.3.

- There is a many-to-many authoring relationship between papers and people. Each paper can be authored by one or more people, and each person can author one or more papers.

Actually, each paper has at least one author, while each person is an author for zero or more papers: Case 3.4 (or Case 4.3).

Note: Just to show that I don't mean to pick unfairly on one particular publication or one particular writer as a target for criticism, here are a couple more examples that demonstrate confusions similar to those just discussed:

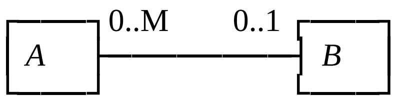
- From reference [6]: "*One-to-one relationship* ... A typical example might be ... that one and only one FACTORY manufactures the particular PRODUCT." Surely FACTORY to PRODUCT is many-to-one, not one-to-one? *Note:* I take the liberty of using the term *many-to-one* here (and in the rest of the present section) even though I still haven't defined it precisely. I'll do so eventually, I promise.

- From reference [7]: "[A] one-to-one mapping ... means that *at every instance* [sic] *in time, each value of A has one and only one value of B associated with it.* There is a one-to-one mapping between EMPLOYEE NAME and SALARY." Again, surely the relationship (here from EMPLOYEE NAME to SALARY) is many-to-one, not one-to-one?

To return for a moment to the examples from reference [5], I do think part of the confusion is due to the writer's use of the word *between*—as in "a one-to-many ... relationship *between* publishers and books," for example (my italics). As I said earlier, relationships (at least, directed relationships) are better thought of as being *from* one set to another, not *between* two sets. Thus, it might be plausible to suggest that (for example) the relationship from *A* to *B* is one-to-one while the inverse relationship from *B* to *A* is many-to-one (i.e., Case 2.4, possibly); but I think it's hard to come up with a clear and unambiguous characterization of "the relationship *between*" *A* and *B* in such a situation.

Incidentally, I said in the abstract to this paper that "a survey of the literature [also displays some] lack of systematic thinking in this area." One example of that lack of systematic thinking occurs in connection with this issue of whether "many" includes the zero case. Consider Fig. 2 once again; for definiteness, consider Case 1.4 from that figure (repeated here for convenience):

Case 1.4:



The line between the two boxes here represents both the relationship from *A* to *B* and the inverse relationship from *B* to *A*, and the annotation on that line represents what are often called the *cardinality constraints* on those relationships. To be precise, the specification 0..1 indicates that for one *a* there can be from zero to one (i.e., at most one) *b;* the specification 0..M indicates that for one *b* there can be any number of *a*'s, from zero to some undefined upper bound M. The other specifications in Fig. 2 have analogous interpretations.

Incidentally, I feel obliged to mention in passing that the UML term for the concept under discussion is not cardinality but *multiplicity*. As reference [1] puts it, "multiplicity" is "a specification of the range of allowable cardinality values—the size—that a set may assume." Well, I've complained before about our field's cavalier way with terms, but this has to be a fairly grotesque example ... The fact is, *multiplicity* simply doesn't mean what UML wants it to mean (and I say this in full knowledge of Humpty Dumpty's pronouncements on such matters). *Chambers Twentieth Century Dictionary* defines the term thus:

> **multiplicity** the state of being manifold: a great number

(*Manifold* here just means "many in number.") Thus, we might reasonably say that, e.g., "UML suffers from a multiplicity of terms——far more terms than it has concepts" (to pick an example more or less at random). However, we can't reasonably say that, e.g., "the multiplicity of the set {*a*,*b*,*c*} is three." (The *cardinality* of that set is three, of course.) *Note:* For an extended discussion of UML's troubles over terms and concepts, see reference [3].

Anyway, to get back to cardinality constraints as such: The terms *mandatory* and *optional* are often used in an attempt to get at the question of whether or not "many" includes the zero case. But there doesn't seem to be any consensus on exactly what these terms mean. Consider the following quotes:

- From reference [8]: "The zero indicates that each object *may* not map to any object. The one indicates that each object *must* map to at least one object. Because of the *may* and *must* aspects of these two minimum constraints, they [*sic*] are sometimes referred to respectively as *optional* or *mandatory* mappings."

- From reference [9]: "*Optional:* The existence of either entity in the relationship is not dependent on the relationship ... *Mandatory:* The existence of both entities is dependent upon the relationship."

Do you think these definitions display "systematic thinking"? (I don't.) Do you think they're clear? (Surely not.) Do you think they're saying the same thing? (Very hard to say.)

Well, enough of this griping. One concept I think can help us arrive at more precise definitions in this area is the concept of a function. Let's take a closer look.

FUNCTIONS

Functions are ubiquitous in computing contexts, and I'm sure you have at least an intuitive understanding of what a function is. Like many other terms in computing that have their origins in mathematics, however, the term *function* has unfortunately had its meaning muddied and diluted somewhat over the years, to our loss. Here I'd like to get back to the original mathematical definition, because as I've already said I think it can be quite helpful in clarifying the relationship concept. Here is such a definition:

> **function** Let *A* and *B* be sets, not necessarily distinct. Then a function *f* is a rule that pairs each element of *A* (the *domain* of *f*) with exactly one element of *B* (the *codomain* of *f*); equivalently, we might just say *f* is that pairing itself. The unique element *b* of *B* corresponding to element *a* of *A* is the *image* of *a* (under *f*), and the set of all such images is the *range* of *f*. Note that the range is a subset, often a proper subset, of the codomain.

The telephone example quoted earlier in this paper illustrates this definition: The function is a rule that pairs letters with integers (equivalently, it's that pairing per se of letters with integers); the domain is the set of letters A-Z, the codomain is the set of integers 0-9, and the range is the set of integers 2-9 (a proper subset of the codomain). Under that function, the image of (for example) the letter H is the integer 4.

As another example, let *f* be the rule that pairs nonnegative integers *x* with their squares x^2. Then *f* is a function with domain and codomain both the set of all nonnegative integers and range that proper subset of the codomain (the domain too, in fact) that consists only of perfect squares. *Note:* From this point forward I'll tend to favor examples like this one (i.e., examples with a slightly mathematical flavor) because their semantics are, or should be, crystal clear——we don't have to get into arguments about what exactly is meant by less well defined concepts such as *department* or *book* or *flight segment* (etc.).

Note: Two further terms that mean exactly the same thing in mathematics as *function* are *mapping* and *transformation* (abbreviated *map* and *transform,* respectively). Again, however, you should be aware that the meanings of these terms have become somewhat muddied and diluted in computing contexts, to our loss, and you can certainly find conflicting definitions in the literature. Again, *caveat lector.*

So what does the notion of a function have to do with that of a relationship? Well, if *f* is a function with domain *A,* codomain *B,* and range *C* (where *C* is a subset of *B*), it should be clear that:

- Function *f* defines a relationship from *A* to *B* according to which, for each element *a* in *A,* there exists exactly one element *b* in *B* (the image of *a* under *f*).

- Likewise, function *f* defines a relationship from *A* to *C* according to which, for each element *a* in *A,* there exists exactly one element *c* in *C* (the image of *a* under *f*).

- Function *f* also defines, at least implicitly, an inverse relationship from *B* to *A* according to which, for each element *b* in *B,* there exist M elements *a* in *A* such that *b* is the image of *a* under *f* (M ≥ 0). *Note:* In general, of course, that inverse relationship won't be a function; in fact, it'll be a function if and only if M = 1 for all elements *b* in *B.*

- Likewise, function *f* also defines, at least implicitly, an inverse relationship from *C* to *A* according to which, for each element *c* in *C,* there exist M elements *a* in *A* such that *c* is the image of *a* under *f* (M > 0). *Note:* Again that inverse relationship will be a function if and only if M = 1 for all elements *c* in *C.*

These facts give some hint as to how the notion of a function can help in pinning down the meaning of terms such as *many-to-one relationship* more precisely, as I'll now try to show.

By definition, any given function *f* has the property that an arbitrary number M (M > 0) of elements *a* from its domain *A* can map to a given element *c* in its range *C.* For that reason, we can and do say that the relationship from *A* to *C* is **many-to-one** ("many" here *not* including the zero case, please note). Furthermore, we often describe that relationship (for emphasis) more specifically as many-to-one **onto,** since by definition there's no element *c* in *C* that isn't the image of something in *A.* As for the relationship from *A* to the codomain *B,* we describe that relationship as many-to-one **into,** since in general there are some elements *b* of *B* that aren't the image of anything at all in *A*. *Note:* Many-to-one onto is a special case of many-to-one into, of course. Also, I should make it clear that in the case of many-to-one into (though not many-to-one onto), the qualifier *many-to-one* is being used in a slightly sloppy sense. See the section "Correspondences" (subsection "Many-to-One Correspondence"), following the present section.

Examples:

1. The telephone example (where *A* is the set of letters A-Z, *B* is the set of integers 0-9, and *C* is the set of integers 2-9) provides an example of a many-to-one into relationship from *A* to *B*——and, of course, a many-to-one onto relationship from *A* to *C*.

2. Likewise, the function that maps nonnegative integers x to their squares x^2 (where *A* and *B* are both the set of all nonnegative integers and *C* is the set of all perfect squares) also provides an example of a many-to-one into relationship from *A* to *B* and a many-to-one onto relationship from *A* to *C*.

More terminology: A many-to-one onto relationship is also known as a *surjective* function (or mapping), or just a *surjection* for short. For completeness, here's the definition:

> **surjection** Let *f* be a function with domain *A* and codomain *B*. Then *f* is a surjection (also known as a *many-to-one onto relationship*) if and only if each element *b* in *B* is the image of at least one element *a* in *A*——in other words, if and only if the range is equal to the codomain.

(In general, the range of a function is a subset of the codomain. But since by definition every set is a subset of itself, having the range and the codomain be equal, which is what happens with a surjection, doesn't count as a violation of this requirement on functions in general.)

Now let's consider the "squares" example again, where the domain *A* and range *C* are the set of all nonnegative integers and the set of all perfect squares, respectively, and *f* is the function that maps nonnegative integers x to their squares x^2. This particular function satisfies the property——as most functions do not——that if *a1* and *a2* are distinct elements in *A*, then their images *c1* and *c2* are distinct elements in *C;* in other words, each element in *C* is the image of exactly one element in *A*. In such a case, we can and do say that the relationship from *A* to *B* is **one-to-one,** or one-to-one **onto** to be more precise. As for the relationship from *A* to the codomain *B*, we describe that relationship as one-to-one **into,** since in general there are some elements *b* of *B* that aren't the image of anything at all in *A*. *Note:* One-to-one onto is a special case of one-to-one into, of course, and one-to-one in general is a special case of many-to-one. Also, I should make it clear that in the case of one-to-one into (though not one-to-one onto), the qualifier *one-to-one* is being used in a slightly sloppy sense. See the section "Correspondences" (subsection "One-to-One Correspondence"), following the present section.

Further terminology: A one-to-one onto relationship is also known as a *bijective* function (or mapping), or just a *bijection* for short; a one-to-one into relationship is also known as an *injective* function (or mapping), or just an *injection* for short. For completeness, here are the definitions:

bijection Let f be a function with domain A and codomain B. Then f is a bijection (also known as a *one-to-one onto relationship*) if and only if each element b in B is the image of exactly one element a in A.

injection Let f be a function with domain A and codomain B. Then f is an injection (also known as a *one-to-one into relationship*) if and only if each element b in B is the image of at most one element a in A.

Note in particular that a function is a bijection if and only if it's both an injection and a surjection. Equivalently, a bijection is a function (from domain A to codomain B) for which the inverse relationship (from B to A) is a function—in fact a bijection—too. Here's a simple example: Let A and B both be the set of all integers, and let f be the function that maps integers x to their successors x+1. Then f is a bijection from A to itself (and so the inverse function, which maps integers x to their predecessors x-1, is a bijection from A to itself as well).

Note further that a bijection can be regarded as a *nonloss* mapping in the sense that, given an arbitrary element a in the domain, we can always get back to that element a from its image b in the codomain by using the inverse mapping, which is a bijection also. *Note:* Bijections are sometimes also referred to as *isomorphic* mappings, though this terminology is deprecated (see the section "Closing Remarks").

Let me now relate the terms I've defined in this section to the cases identified in the section "How Many Cases Are There?" (where possible). It should be clear without too much discussion that:

- Case 2.1 is an injection, or one-to-one into relationship, from A to B (and Case 1.2 is an injection from B to A).

- Case 2.2 is a bijection, or one-to-one onto relationship, from A to B (and also from B to A).

- Case 2.3 is a surjection, or many-to-one onto relationship, from A to B (and Case 3.2 is a surjection from B to A).

- Case 2.4 is a function from A to B that isn't an injection, bijection, or surjection (and Case 4.2 is a function from B to A that isn't an injection, bijection, or surjection).

So now we've dealt with four of the ten distinct cases, or seven of the 16 cases overall. What about the others? Obviously the notion of function can only take us so far; in order to deal with the other cases properly, we need some more concepts, and more definitions.

CORRESPONDENCES

The mathematical notion of a correspondence is more general than that of a function—in fact, it can be seen as a formalization of the intuitive notion of a directed binary relationship (some but not all such relationships being functions, as we've seen). As you'd probably expect, correspondences come in four flavors: one-to-one, many-to-one, one-to-many, and many-to-many (though many-to-one and one-to-many are really just the same thing looked at from two different points of view, again as you'd probably expect). I'll treat each case in a subsection of its own.

One-to-One Correspondence

Here's a precise definition:

> **one-to-one correspondence** Let *A* and *B* be sets, not necessarily distinct. Then a one-to-one correspondence from *A* to *B* is a rule that pairs each element of *A* with exactly one element of *B* and each element of *B* with exactly one element of *A*. Equivalently, we might just say the one-to-one correspondence is that pairing itself.

By way of example, let *A* be the set of all integers. Then the pairing of elements x with their successors x+1 is a one-to-one correspondence from *A* to itself, and so is the pairing of elements x with their predecessors x-1.

Now, I hope it's obvious that (as the example suggests) a one-to-one correspondence is nothing more nor less than a bijection as previously defined. The trouble is, the term *one-to-one correspondence* is often used, loosely, to mean something slightly different from a bijection as such. To be specific, it's often used to mean one or other of the following (note the various subtle differences):

1. A pairing such that each element of *A* corresponds to at most one element of *B* (while each element of *B* corresponds to exactly one element of *A*)

2. A pairing such that (while each element of *A* corresponds to exactly one element of *B*) each element of *B* corresponds to at most one element of *A*

3. A pairing such that each element of *A* corresponds to at most one element of *B* and each element of *B* corresponds to at most one element of *A*

I'll refer (for the moment) to these three interpretations of the term *one-to-one correspondence* as *Type 1*, *Type 2*, and *Type 3*, respectively, and I'll refer to the strict (bijective) interpretation as *Type 0*. Let me now relate the four interpretations to the cases identified in the section "How Many Cases Are There?" (where possible). In fact, it should be clear that:

- Case 2.2 (previously identified, correctly, as a bijection, or one-to-one onto relationship, from *A* to *B* and also from *B* to *A*) can also be identified as a *Type 0* one-to-one correspondence from *A* to *B* (and also from *B* to *A*).

- Case 1.2 (previously identified, correctly, as an injection, or one-to-one into relationship, from *B* to *A*) can also be identified as a *Type 1* one-to-one correspondence from *A* to *B*.

- Case 2.1 (previously identified, correctly, as an injection, or one-to-one into relationship, from *A* to *B*) can also be identified as a *Type 2* one-to-one correspondence from *A* to *B*.

- Case 1.1 can be identified as a *Type 3* one-to-one correspondence from *A* to *B* (and also from *B* to *A*).

Note: Given that (as I've said) the *Type 1, Type 2,* and *Type 3* interpretations are loose, you can now see why I said earlier (in the section "Functions") that in the case of a one-to-one into relationship in particular, the qualifier *one-to-one* was being used in a slightly sloppy sense.

Many-to-One Correspondence

Again I'll start with a definition:

> **many-to-one correspondence** Let *A* and *B* be sets, not necessarily distinct. Then a many-to-one correspondence from *A* to *B* is a rule that pairs each element of *A* with exactly one element of *B* and each element of *B* with at least one element of *A.* Equivalently, we might just say the many-to-one correspondence is that pairing itself.

By way of example, let *A* and *B* be the set of all integers and the set of all nonnegative integers, respectively. Then the pairing of integers *x* with their absolute values $|x|$ is a many-to-one correspondence from *A* to *B*.

Now, I hope it's obvious that a many-to-one correspondence is nothing more nor less than a surjection as previously defined. The trouble is, the term *many-to-one correspondence* is often used, loosely, to mean something slightly different from a surjection as such. To be specific, it's often used to mean one or other of the following:

1. A pairing such that each element of *A* corresponds to at most one element of *B* (while each element of *B* corresponds to at least one element of *A*)

2. A pairing such that (while each element of *A* corresponds to exactly one element of *B*) each element of *B* corresponds to any number of elements of *A* (possibly none at all)

3. A pairing such that each element of *A* corresponds to at most one element of *B* and each element of *B* corresponds to any number of elements of *A* (possibly none at all)

I'll refer for the moment to these three interpretations of the term *many-to-one correspondence* as *Type 1, Type 2,* and *Type 3,* respectively, and I'll refer to the strict (surjective) interpretation as *Type 0*. Let me now relate the four interpretations to the cases identified in the section "How Many Cases Are There?" (where possible). Again it should be clear that:

- Case 2.3 (previously identified, correctly, as a surjection, or many-to-one onto relationship, from *A* to *B*) can also be identified as a *Type 0* many-to-one correspondence from *A* to *B*.

- Case 1.3 can be identified as a *Type 1* many-to-one correspondence from *A* to *B.*

- Case 2.4 (previously identified, correctly, as a function that's not an injection, bijection, or surjection) can also be identified as a *Type 2* many-to-one correspondence from *A* to *B.*

- Case 1.4 can be identified as a *Type 3* many-to-one correspondence from A to B.

Note: Given that (as I've said) the *Type 1*, *Type 2*, and *Type 3* interpretations are loose, you can now see why I said earlier (in the section "Functions") that in the case of a many-to-one into relationship in particular, the qualifier *many-to-one* was being used in a slightly sloppy sense.

One-to-Many Correspondence

Again I'll start with a definition:

> **one-to-many correspondence** Let A and B be sets, not necessarily distinct. Then a one-to-many correspondence from A to B is a rule that pairs each element of A with at least one element of B and each element of B with exactly one element of A. Equivalently, we might just say the one-to-many correspondence is that pairing itself.

By way of example, let A and B be the set of all nonnegative numbers and the set of all numbers, respectively. Then the pairing of integers x with their square roots $\pm\sqrt{x}$ is a one-to-many correspondence from A to B.

The trouble is, the term *one-to-many correspondence* is often used, loosely, to mean something slightly different from the concept as just precisely defined. To be specific, it's often used to mean one or other of the following:

1. A pairing such that each element of A corresponds to any number of elements of B, possibly none at all (while each element of B corresponds to exactly one element of A)

2. A pairing such that (while each element of A corresponds to at least one element of B) each element of B corresponds to at most one element of A

3. A pairing such that each element of A corresponds to any number of elements of B, possibly none at all, and each element of B corresponds to at most one element of A

I'll refer for the moment to these three interpretations of the term *one-to-many correspondence* as *Type 1*, *Type 2*, and *Type 3*, respectively, and I'll refer to the strict interpretation as *Type 0*. Let me now relate the four interpretations to the cases identified in the section "How Many Cases Are There?" (where possible). Once again it should be clear that:

- Case 3.2 (previously identified, correctly, as a surjection, or many-to-one onto relationship, from B to A) can also be identified as a *Type 0* one-to-many correspondence from A to B.

- Case 4.2 (previously identified, correctly, as a function from B to A that isn't an injection, bijection, or surjection) can also be identified as a *Type 1* one-to-many correspondence from A to B.

- Case 3.1 can be identified as a *Type 2* one-to-many correspondence from A to B.

- Case 4.1 can be identified as a *Type 3* one-to-many correspondence from *A* to *B.*

Many-to-Many Correspondence

Again I'll start with a definition:

> **many-to-many correspondence** Let *A* and *B* be sets, not necessarily distinct. Then a many-to-many correspondence from *A* to *B* is a rule that pairs each element of *A* with at least one element of *B* and each element of *B* with at least one element of *A.* Equivalently, we might just say the many-to-many correspondence is that pairing itself.

By way of example, let *A* be the set of all positive integers. Consider the pairing of positive integers *x* and *y* defined as follows: Positive integers *x* and *y* are paired if and only if they have the same number of digits in conventional decimal notation. Then that pairing is a many-to-many correspondence from *A* to itself.

The trouble is, the term *many-to-many correspondence* is often used, loosely, to mean something slightly different from the concept as just precisely defined. To be specific, it's often used to mean one or other of the following:

1. A pairing such that each element of *A* corresponds to any number of elements of *B,* possibly none at all (while each element of *B* corresponds to at least one element of *A*)

2. A pairing such that (while each element of *A* corresponds to at least one element of *B*) each element of *B* corresponds to any number of elements of *A* (possibly none at all)

3. A pairing such that each element of *A* corresponds to any number of elements of *B* (possibly none at all) and each element of *B* corresponds to any number of elements of *A* (possibly none at all)

I'll refer to these three interpretations of the term *many-to-many correspondence* as *Type 1, Type 2,* and *Type 3,* respectively, and I'll refer to the strict interpretation as *Type 0*. Let me now relate the four interpretations to the cases identified in the section "How Many Cases Are There?" (where possible). Yet again it should be clear that:

- Case 3.3 can be identified as a *Type 0* many-to-many correspondence from *A* to *B* (and also from *B* to *A*).

- Case 4.3 can be identified as a *Type 1* many-to-many correspondence from *A* to *B.*

- Case 3.4 can be identified as a *Type 2* many-to-many correspondence from *A* to *B.*

- Case 4.4 can be identified as a *Type 3* many-to-many correspondence from *A* to *B* (and also from *B* to *A*).

Summary

All 16 cases have now been covered. Fig. 3, an edited form of Fig. 1, summarizes the situation at this point. *Note:* I've used several self-explanatory abbreviations in that figure (for example, M:1 for many-to-one). Also, some of the entries in the matrix could be stated in more than one way; in such cases, I've chosen the form of text that I personally find most helpful (for example, I prefer "1:1 onto" to "1:1 *Type 0*"). The notation $A \rightarrow B$ means "from *A* to *B*"; the notation $A \leftrightarrow B$ means "from *A* to *B* and also from *B* to *A*."

a has ↓ / *b* has →	at most one *a*	exactly one *a*	at least one *a*	M *a*'s (M ≥ 0)
at most one *b*	1:1 *Type 3* $A \leftrightarrow B$	1:1 into $B \rightarrow A$	M:1 *Type 1* $A \rightarrow B$	M:1 *Type 3* $A \rightarrow B$
exactly one *b*	1:1 into $A \rightarrow B$	1:1 onto $A \leftrightarrow B$	M:1 onto $A \rightarrow B$	M:1 *Type 2* $A \rightarrow B$
at least one *b*	M:M *Type 2* $A \rightarrow B$	M:1 onto $B \rightarrow A$	M:M strict $A \leftrightarrow B$	M:M *Type 2* $A \rightarrow B$
M *b*'s (M ≥ 0)	1:M *Type 3* $A \rightarrow B$	1:M *Type 1* $B \rightarrow A$	M:M *Type 1* $A \rightarrow B$	M:M *Type 3* $A \leftrightarrow B$

Fig. 3: The 16 cases summarized

CLOSING REMARKS

I'd like to close with a series of recommendations. I'll start with a couple of small ones. First, I said earlier that it makes more sense to think of a relationship as being from one set to another than it does to think of it being between those sets; I even described the use of "between" as a source of some of the confusions we observe in this area. However, there are a couple of cases where "between" is acceptable, and possibly even the better term: namely, Cases 2.2 and 3.3——where, just to remind you, Case 2.2 is the strict one-to-one case and Case 3.3 is the strict many-to-many case. And an argument could be made (I don't think I'd subscribe to it, but it's hardly an earth-shattering issue) that "between" is also acceptable in Cases 1.1 and 4.4. In all other cases, however, I would definitely recommend staying with "from" and "to" and avoiding "between."

Second, I noted earlier that a bijection is sometimes said to be an isomorphic mapping——but I also said that term was deprecated. The reason it's deprecated is that, mathematically speaking, an isomporphism is more than just a bijection (loosely, it's a bijection plus operators). Here's the definition:

> **isomorphism** Let *A* and *B* be sets, not necessarily distinct, and let *f* be a bijective mapping from *A* to *B*. Let *OpA* be an operator that takes elements of *A* as its operands and yields an element of *A* as its result. Then *f* is an isomorphism if and only if, for all such operators *OpA*, there exists an analogous operator *OpB* that takes elements of *B* as its operands and yields an element of *B* as its result such that, whenever *OpA* applied to *a1, a2, ..., an* yields *a*, then *OpB* applied to *b1, b2, ...,*

bn yields *b,* where *b1, b2, ..., bn,* and *b* are the images of *a1, a2, ..., an,* and *a*, respectively, under *f*.

In other words, a bijective mapping is an isomorphism if and only if it preserves the algebraic structure of the domain *A* in the codomain *B*. Of course, if a given bijective mapping *f* is an isomorphism, then its inverse is an isomorphism as well.

Now I turn to a more serious recommendation. I've shown in this paper that the situation regarding relationships in general is quite complex——more complex than it's usually given credit for. In fact, I think it's sufficiently complex that our usual reliance in this area on somewhat simplified terminology is a trifle dangerous, or at least misleading. Instead of relying on such simplified (or oversimplified) terminology, therefore, I think, as suggested in the abstract to this paper, that we should strive for a better set of terms——terms that are clearer, more accurate, and more precise than the ones we usually use. To be more specific:

- I don't think we should refer to the various different cases by reference numbers ("Case 3.2" and the like), because such reference numbers always need some kind of guide or key by which they can be interpreted. Of course, I used such reference numbers in the body of the paper, but that was because I was trying to imposing a systematic structure on the subject matter for purposes of analysis. I'd be very surprised if you didn't have difficulty in remembering which case was which, especially if you didn't keep a copy of Fig. 1 or Fig. 2 by you as a key.

- I don't think we should use labels like *optional* and *mandatory,* because their meanings are essentially arbitrary (even if we can agree on them, something I'm not at all sanguine about anyway). *Note:* I remark in this connection that several other similar terms can also be found in the literature: *conditional* relationships, *contingent* relationships, *complex* relationships, *singular* relationships, *exclusive* relationships, and so on (this isn't an exhaustive list). Do you think the precise meanings of these terms are all obvious? Or all universally agreed upon?

- I don't think we should use mathematical terms like *function, injection, surjection,* or *bijection,* even though those terms do at least have precise definitions. One problem is that few people are familiar with those definitions (and in any case, I think the meanings of *injection* and *surjection,* at least, are difficult to remember). Another is that——rather annoyingly!——the terms don't cover all cases. Yet another is that the term *function,* at least, is often used in a very imprecise manner, despite its having a precise definition, thus increasing the likelihood of breakdowns in communication.

- Similar remarks apply to the terms *one-to-one onto* (and *into*) and *many-to-one onto* (and *into*).

- I've pointed out that each of the terms *one-to-one correspondence, many-to-one correspondence, one-to-many correspondence,* and *many-to-many correspondence* has three loose interpretations as well as one strict one. As a consequence, I think these terms should be avoided too, except in situations where there's no possibility of misunderstanding (you might like to meditate on whether any such situations exist). I certainly wouldn't use terms like "*Type 2* one-to-many correspondence" at all! Like the terms "Case 3.2" (etc.), I used those terms in the body of the paper only because I was trying to imposing a systematic structure on the subject matter for purposes of analysis, and I'd be very surprised if you didn't have difficulty in remembering exactly what they all meant.

So what terms do I think we should use? Well, I certainly think we should use terms that are *absolutely explicit*——thereby saying things like "zero or one to one or more," if that's what we really mean, instead of using an approximation like "one-to-many" and hoping that our audience knows that "one" here includes the zero case and "many" doesn't. So here are the terms I would recommend. Note that they can all be immediately derived from the annotation in Fig. 2. Note too that I do have to use the terminology of "Case 3.2" (etc.) in this list!——but, I hope, for the very last time.

- Case 1.1: at most one to at most one
- Case 1.2: exactly one to at most one
- Case 1.3: one or more to at most one
- Case 1.4: zero or more to at most one
- Case 2.1: at most one to exactly one
- Case 2.2: exactly one to exactly one
- Case 2.3: one or more to exactly one
- Case 2.4: zero or more to exactly one
- Case 3.1: at most one to one or more
- Case 3.2: exactly one to one or more
- Case 3.3: one or more to one or more
- Case 3.4: zero or more to one or more
- Case 4.1: at most one to zero or more
- Case 4.2: exactly one to zero or more
- Case 4.3: one or more to zero or more
- Case 4.4: zero or more to zero or more

Let me admit immediately that the foregoing terms are still not as transparently clear as I could wish them to be. In the case of "zero or one to one or more," for example, it has to be clearly understood that the "zero or one" means that each *b* has at most one *a* and not the other way around, and the "one or more" means each *a* has at least one *b* and not the other way around. So perhaps I'm still in the market for some better terms ... but until such terms come along, I think the foregoing ones should serve. *Note:* In this connection, Hugh Darwen has suggested the following numeric substitutions for the phrases "at most one" (etc.):*

at most one	1^-
exactly one	1
one or more	1^+
zero or more	0^+

Using these replacements, "at most one to zero or more" (for example) becomes "1^- to 0^+" (or, perhaps better, just "$1^-:0^+$"). This notation is at least more succinct, avoids the use of English words, and works well for more unusual specific cases such as "exactly one to at most 2" ($1:2^-$); but I still don't think it's perfect.

I'd like to leave you with some questions to ponder. In this paper, I've offered precise definitions of (among other things) the terms *function, one-to-one correspondence, many-to-one correspondence, one-to-many correspondence,* and *many-to-many correspondence*. Here now is a precise definition of a *binary relation* (in the mathematical sense of that term):

> **binary relation** Let *A* and *C* be sets, not necessarily distinct. Then a binary relation *r* is a rule that pairs each element of *A* (the *domain* of *r*) with at least one element of *C* (the *range* of *r*); equivalently, we might just say *r* is that pairing itself.

So what exactly do you think the relationship is between relationships as I've defined them in this paper and the mathematical concept of a binary relation? And what do you think the relationship is between relationships and relations as defined in the relational model? And why do you think the relational model is so called?

REFERENCES

1. Grady Booch, James Rumbaugh, and Ivar Jacobson, *The Unified Modeling Language User Guide.* Reading, Mass.: Addison-Wesley (1999).

2. C. J. Date: *The Relational Database Dictionary* (2006, to appear).

3. C. J. Date: "Basic Concepts in UML: A Request for Clarification," in two parts, *www.dbdebunk.com* (December 2000; January 2001). Republished in *Date on Database: Writings 2000-2006*. Berkeley, Calif.: 2006 (to appear).

4. C. J. Date and Hugh Darwen: *Databases, Types, and the Relational Model: The Third Manifesto* (3rd edition). Reading, Mass.: Addison-Wesley (2006).

5. Mary E. S. Loomis: *Object Databases: The Essentials.* Reading, Mass.: Addison-Wesley (1995).

---

* In passing, let me acknowledge Hugh's careful review of an earlier draft of this paper.

6. Michael M. Gorman: *Database Management Systems: Understanding and Applying the Technology*. Wellesley, Mass.: QED Information Sciences (1991).

7. James Martin: *Computer Data-Base Organization* (2nd edition). Englewood Cliffs, N.J.: Prentice-Hall (1977).

8. James Martin and James J. Odell: *Object-Oriented Methods: A Foundation* (2nd edition). Englewood Cliffs, N.J.: Prentice-Hall (1997).

9. Toby J. Torey and James P. Fry: *Design of Database Structures*. Englewood Cliffs, N.J.: Prentice-Hall (1982).

***** End *** End *** End *****