Introduction

This paper addresses recent advances to Tridbit technology. The paper describes a conversational personal information manager, JotChat, based on the patent-pending natural language understanding system, Tridbit technology. The development and usability testing of a JotChat prototype was conducted as part of the United States Department of Education's National Institute on Disability and Rehabilitation Research (NIDRR) SBIR grant H133S080032. A report on the success of untrained users in interacting with JotChat to complete scenarios asking them to enter and retrieve personal information can be found at:

http://www.tridbits.com/pubs/ConversInterfaceReport2.pdf

Tridbit technology had already achieved a remarkable level of natural language understanding prior to the grant, as demonstrated by the example dialog in Figure 1. This paper describes the key enhancements made to Tridbits during the grant that allowed JotChat to perform in the usability tests with sufficient understanding that users embraced it as a conversation partner.

What follows is a technical discussion for readers interested in the details of how Tridbit technology works. Prior knowledge of Tridbit technology is not required. Sufficient background in Tridbits is provided to build up to the enhancements that will be described. A background in logic, knowledge representation and/or philosophy may be helpful, but the main requirement is the ability to follow abstract logical reasoning. Examples and illustrations are used generously to try to make the representations as grounded as possible.

It is not feasible to describe all of the enhancements made to the Tridbit engine in a single paper. This paper focuses on the aspects of Tridbit technology called infervals and general qualified concepts (GQC). These tridbit configurations reflect naturally occurring language constructs that encode information in powerful but flexible ways. Below is an example of a complex scenario whose resolution depends heavily on the use of infervals and GQCs, as will be demonstrated in the final section of this paper.

Phil is the karate teacher of the mother of Mary.

Mary's mom's karate teacher's studio phone number is 555-1111.

What is Phil's phone number?

JotChat is able to resolve this scenario, but what is most remarkable is that it is able to do this with no prior knowledge of karate teachers, studio phone phones or who Mary's mother is.

The approach we'll take to describe how JotChat resolves this scenario will be will like a math lesson. The karate teacher example is the hard problem solved at the end of the chapter. To begin the chapter, we'll introduce the tools needed to solve the problem. The journey will cover a lot of territory, though it will by no means be an extensive review of the entire Tridbit system.

June likes computers.
They are smarter than typewriters.
She drove the sporty gold car to the store that sells computers.
Who likes computers?
June
What color was the car?
gold

Robyn is the mother of Tyler.
Who is the child of Robyn?
Tyler

Spiders have eight legs.
Spiders have how many legs?
8

The television under the table is broken.
The television is under what?
table

What is broken?
television

The car that Jeff drove to the store that sold electronics crashed.
What crashed?
car

What sold electronics?
store

What is smarter than typewriters?
computer

Figure 1: Tridbit demo prior to grant. User input is gray and Tridbit response is red.

One thing that differs with the math analogy is, that if this were calculus and the problem was an optimal solution in a complex system, as a novice you'd have no intuition as to what the answer was. As a human being with incredible natural language understanding (NLU) ability built in, you have little difficulty solving the karate teacher problem. The trouble is, no one understands well enough how we do this to reproduce the behavior with a computer. In fact humans are so good at natural language, most people barely realize there is a difference seeing words and understanding them.

Those that have tried to work on NLU problems know too well what I am saying. For others, please try to disconnect yourself with the idea that reciting some well-known grammar lesson solves anything. These lessons are meant for entities that already have language understanding abilities built in. Computers are truly blank slates. So to make a computer simulate language behavior in any sort of self-directed way entails building a very complex system starting with the most fundamental notions of what information is.

Tridbits is a system for representing and processing (extracting, reasoning, expressing, etc.) information conveyed via natural language. Some of the processing, especially the reasoning, resembles a kind of algebra for transforming meaning. Other aspects are more like chemistry, defining tridbit structures, how tridbits combine and the resulting behavior given various configurations. It would be surprising if Tridbit technology introduced radically new concepts as our understanding of the world is like a fractal where the same elements are recombined at different levels. Tridbits reside at the most basic level, describing the initial structuring of phenomenon into information. Human minds, not computers, are still the sole domain of actually doing the structuring. But once the structuring is done, humans describe and work with this information via natural language. That is the starting point at which Tridbits takes information.

To start the journey we need to review the basic elements of a tridbit representation and introduce its notation. We'll start with a simpler sentence than the karate teacher scenario and see how it is represented:

The phone is green.

If we type this sentence into JotChat, we can ask it to draw a tridbit diagram showing how it processed the sentence. The tridbit diagram for "The phone is green" is shown in Figure 2.

Each word in the sentence is shown at the far left, in a row containing a drop down indicating the word use type selected for this interpretation of the sentence. Tridbit word use is similar, but not identical to, traditional part of speech categorization. Appendix A defines all of the Tridbit word use categories.

In this sentence "phone" is categorized as an N1, basically a subtype of a noun reserved for words that name a category of things. "Things" is used in a technical sense here, in that it is the term for one of the three basic metaphysical categories, which are reviewed in the next section. The other two metaphysical categories in the Tridbit model are events and properties.

The word "is" in this sentence is categorized as a V2, which is essentially a verb category created just for this use of the verb to be. (This is somewhat of a legacy category. As the model has evolved "is" would probably be better categorized as a tag word)

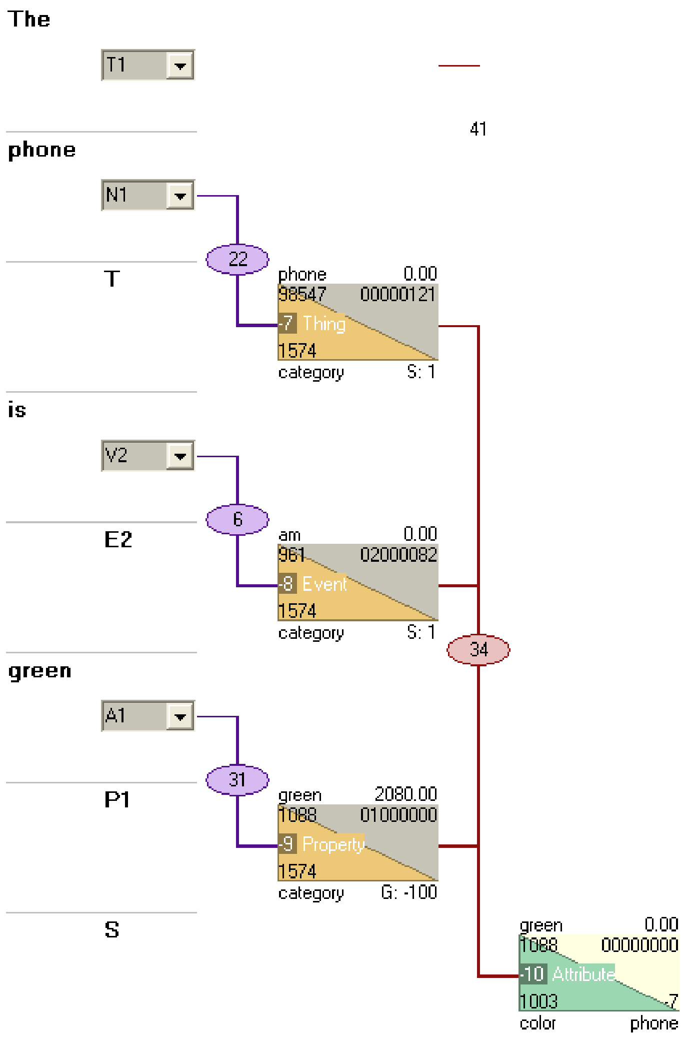


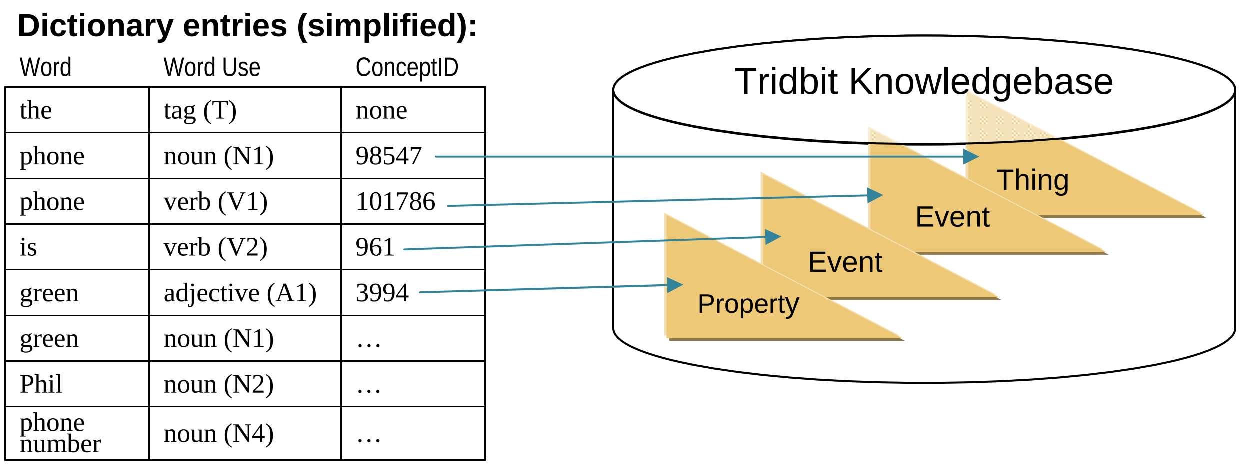
Figure 2: Tridbit diagram of "The phone is green."

The word "the" in this sentence is categorized as a tag word. Tag words in the Tridbit system can come from a variety of traditional part of speech categories such as articles, conjunctions, some propositions, etc. The characteristic that distinguishes tag words is that they do not represent a referent. Tag words provide information about how the referents relate to each other.

A **referent** is just the phenomenon that a word or expression refers to. In the Tridbit model a referent is either a thing, event or property as indicated by its referent type. Descriptions of these basic metaphysical categories are given below:

A **thing** exists, such as a person, an idea, a phone or phones in general
An **event** occurs, such as specific episodes of going, seeing, breaking, meeting or meetings in general
A **property** describes another tridbit, such as loud, green or big

JotChat makes word use choices by looking up each word in the Tridbit dictionary. If multiple word uses exist for a word, an initial choice is made based on past associations between the word in question and the words occurring before and after this word. If the chosen word uses do not produce an interpretation of the sentence that makes sense, other choices can be tried until a sensible interpretation is found.



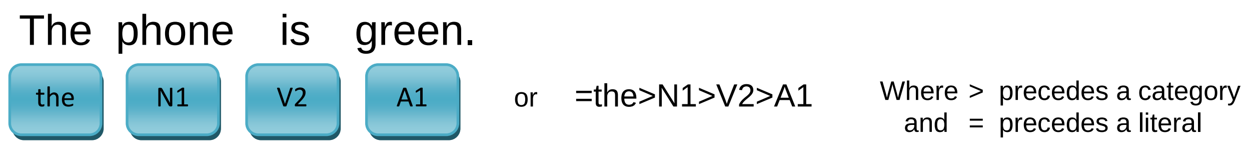
Dictionary entries (simplified):

Word	Word Use	ConceptID
the	tag (T)	none
phone	noun (N1)	98547
phone	verb (V1)	101786
is	verb (V2)	961
green	adjective (A1)	3994
green	noun (N1)	…
Phil	noun (N2)	…
phone number	noun (N4)	…

Figure 3: Diagram of linkage between words in Tridbit dictionary and concepts in Tridbit knowledgebase.

Once word use choices have been made, the Tridbit system starts an iterative process of looking for patterns and applying transformations based on the patterns found. A sentence's initial pattern consists of just word use categories (N, V and A) and subcategories, plus literals for any tag words. As the pattern is processed, the word use tokens are consumed and replaced by referent tridbit tokens, which are then also processed and replaced by assert tridbits.

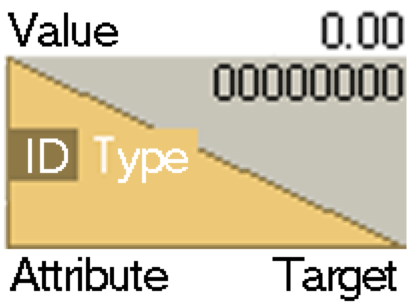
Below is the optimal starting pattern assignment for "The phone is green."



We need to know a little more about what a tridbit is to understand the next step in processing "The phone is green." This document will only describe what is needed to process the examples, but there is more to the Tridbit model. The paper Babble: Simple Conversations With a Computer, linked below discusses the foundation of the Tridbit system in more detail, especially comparing some of the fundamental concepts such as scope and meaning to similar ideas in logic, philosophy and other natural language work.

http://www.tridbits.com/pubs/simpleconvers.pdf

Tridbits are the basic unit of information in the Tridbit model. The most basic type of tidbit represents a referent and thus is called a referent tridbit. Because there are three types referents in the Tridbit metaphysical model, things, events and properties, there are three corresponding types of referent tridbits. Figure 4 below represents the general structure of a tridbit as drawn in the tridbit diagram.



The elements of the tridbit shown at left are:

ID: unique value for identifying the tridbit.

Type:
- 3 types of referent tridbits: Thing, Event or Property
- 2 types of assert tridbits: Attribute or Compare

Target: The "subject" of the tridbit, it is the referent a tridbit is "about."
Since a referent tridbit defines the referent, its target would be itself.
Rather than "wasting" this element the target of a referent tridbit stores:
- **Scope**: **G**eneral or **S**pecific
- **Count**: numeric value representing how many

Attribute: A referent defining the type of information being conveyed about the target. For a referent tridbit, this is how the referent is defined, typically by Name, Category or various "place holding" types.

Value: A referent that completes the attribute – target information.

Figure 4: Diagram and description of the elements of a tridbit.

The pair of numbers in the top right corner conveys different information depending on the tridbit type, which is not needed for this discussion. Tridbits have the ability to link to another tridbit which is used for a variety of purposes, including creating the tridbit configurations described later.

Now we're ready to look at the next step in the process that goes from the words of the sentence "The phone is green" to the tridbits that represent the referents of the sentence. In the tridbit diagram for the sentence, referent tridbits are drawn as gold triangles. The middle column contains 3 referent tridbits, shown in Figure 5 below.

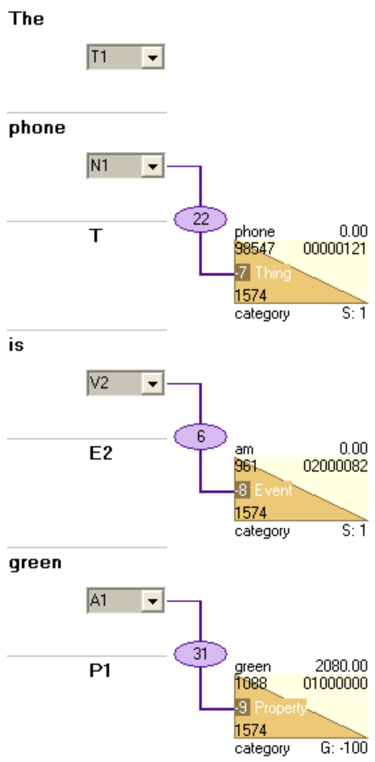


Figure 5: Tridbit diagram of "The phone is green" showing referent tridbit generation only.

The Tridbit system looks for patterns that trigger syntax rules within the starting pattern for the sentence:

=the>N1>V2>A1

Patterns involving word use categories tend to be short, with few multiple matches. In this case, each of the last three tokens is matched individually as described below. Note that the expression on the right side of the ⟶ is the notation for defining tridbits. A tridbit definition consists of 4 values separated by comas and enclosed in <>s. The first value is the tridbit type, followed by the tridbit's attribute, target and value. The "P#" indicates that element is taken from the #th token in the pattern.

Rule 22 matches the word use category (N1) of the word "phone" and generates the thing referent whose ID is -7.

Rule 22 :

>N1 ⟶ <Ref/T Category G:-100 P1>

Rule 6 matches the word use category (V2) of the word "is" and generates the event referent whose ID is -8.

Rule 6 :

>V ⟶ <Ref/E Category S:1 P1>

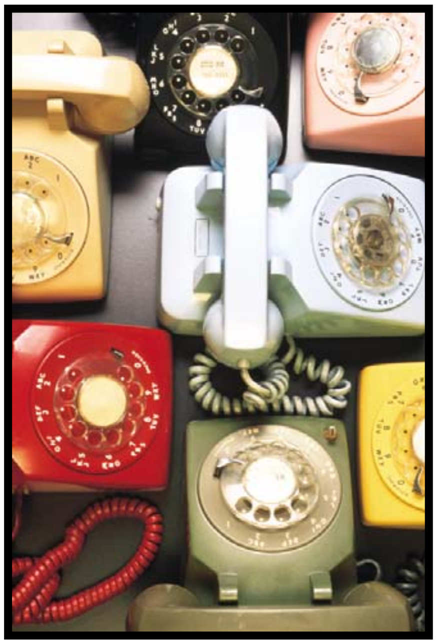
Rule 31 matches the word use category (A1) of the word "green" and generates the property referent whose ID is -9.

Rule 31 :

>A ⟶ <Ref/P Category G:-100 P1>

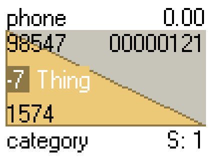
Scope

An important aspect of the Tridbit metaphysical model is the concept of scope. Every referent must have a scope which will be either specific or general.

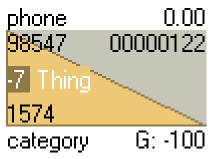


If a referent tridbit is specific in scope, it represents a specific referent that has been picked out or identified. Tridbit -7, which represents the green phone, is specific in scope. The scope:count, which is the bottom right element in a referent tridbit, is set to S:1 to indicate this.

Referents that are general in scope represent any or all referents (depending on the count) that satisfy the criteria by which the referent is defined. Thus "All phones are green," the referent tridbit representing "all phones" would have a scope:count of G:-100. -100 indicates all members and since the referent tridbit is defined by category, the tridbit represents all members of the category, in other words it represents the category.



This tridbit represents a specific phone.



This tridbit represents all phones.

A specific phone referent tridbit does more than declare that a phone exists. It represents exactly the phone being referred to. This is important in maintaining the knowledgebase and assigning properties. If we pick out a specific phone and assign it a property of green, the property will only apply to that specific phone. We can infer, or reason by analogy, but we cannot definitively conclude anything about phones in general. On the other hand, if we say all phones are green, we can conclude that any member of that category has the property of green. If we say there exists a phone that is green, we have not picked out a specific referent to which we can assign the property of green. Instead, the property is assigned to a general referent that represents an arbitrary subset of phones with at least one member.

Scope and category are the result of perhaps the two most basic functions human consciousness performs on the information it processes. First, it organizes information into categories. We distinguish phones from bells and radios and spoons from forks and knives. Second we pick out specific individuals or groups of individuals. It is this act of picking out or enumerating that distinguishes specific from general. Naming something is, by definition, picking it out, so referents defined by name are always specific. But when the referent's definition involves its category, it is hard to know whether the speaker has picked out specific members of the category or not. Thus English has various types of articles, syntax and other devices to help the listener determine the scope of a referent defined by category. For example the definite article "the" generally lets the listener know the referent that follows is specific in scope.

Scope also allows the speaker to inform the listener whether it's important to indentify the referent in order to understand or comply with what the speaker said. If you have the silverware drawer open and your companion says to you "Bring me a spoon," any spoon will do. But if they say "Bring me the spoon," you'll have to get clarification unless you've already established what "the spoon" means. In most cases (but certainly not all), "the" indicates a specific referent. "A" is more complex. Consider the difference between "I used a spoon" and "I used the spoon." Because the verb used is past tense, common sense says there is a specific spoon used in each event. When the speaker chooses "the spoon" they inform the listener that to fully understand the sentence the listener needs to identify the specific spoon the speaker intends. This can be a fairly complex task, a process referred to as “referent reduction” in Tridbits. If the identity of the spoon is not crucial to understanding the sentence, the speaker chooses “a spoon.”

Determining a referent’s scope and matching it to the speaker’s intentions is much more complex than it seems to those of us with built-in natural language understanding. Every referent goes through referent reduction, even referents of general scope, in order to avoid multiple tridbits representing the same referent – something that will quickly confuse any attempt at knowledge representation. Consider the two statements “Spoons are used to eat soup” and “Spoons are used for measurement.” In each sentence the referent of “spoons” is general, but the second referent needs to be reduced so it does not create a new category of spoons used for measurement. JotChat is able to perform a reasonable job of referent reduction using the strategies described above and others that are beyond the scope of this document. Several relatively complex examples of referent reduction will be discussed in working through the karate example. However, it is a complex area that will continue to be developed.

Getting back to processing the example sentence “The phone is green,” we are now ready to process the three referent tridbits generated by rules 22, 6 and 31. The three referent tridbits form a new pattern which will match a new set of rules, ultimately consuming all the word tokens and referent tridbits and leaving just assert tridbits, representing the information in the sentence.

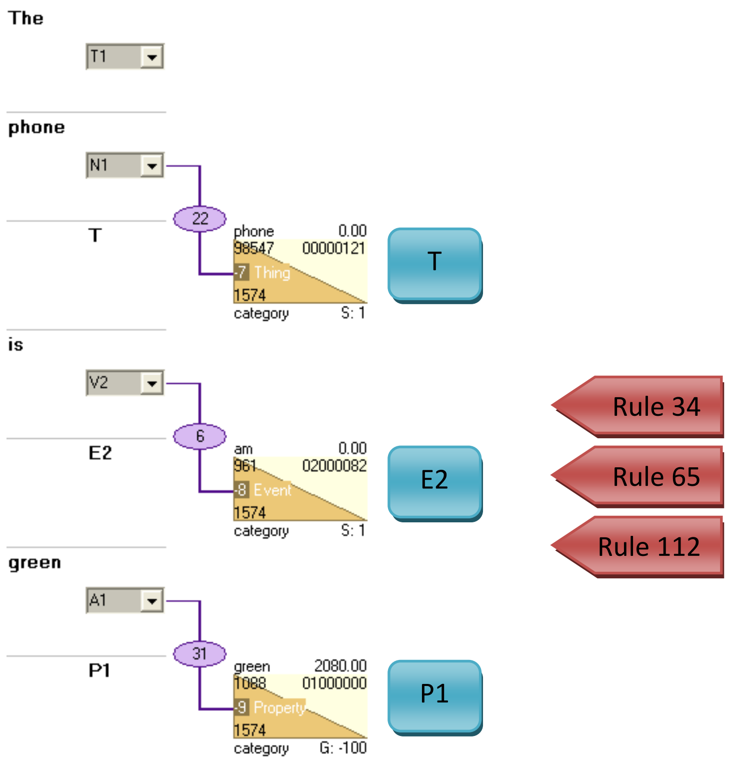


Figure 6: Augmented Tridbit diagram showing referent tridbits forming a new pattern

There is no formal sequence of first generating referent tridbits followed by assert tridbits. Instead it’s more of an organic process where a rule can be applied any time it matches. More like a soup of chemical fragments looking for other fragments to bond with to form a complex structure incorporating all the available material. Like a chemical soup, where different types of fragments may try to bond without success, there are usually multiple rules that can apply to a pattern but some will fail immediately and other may fail as the larger component that incorporates them reaches a dead end.

We will walk through applying a rule that fails, followed by the successful rule, but first we need to describe assert tridbits. It is the constraints within assert tridbits that are largely responsible for causing rules to fail, thereby eliminating many nonsensical interpretations.

Assert tridbits have the same simple structure shown in Figure 4 as referent tridbits. Most important are the tridbit’s three interrelated elements: attribute, target and value. An assert tridbit asserts a relationship between its three elements. There are two types of assert tridbits: assert attribute tridbits, which are discussed below and assert compare tridbits, which are mentioned briefly at the end.

Tridbits impose unique constraints that require certain relationships between the elements of an assert tridbit in order for the tridbit to be valid. Figure 7 at right depicts the fundamental constraint within an assert attribute tridbit. Four possible examples are shown for each element, resulting in the four valid assertions:

1. Green is the color of the phone.
2. Betty is the mother of Mary.
3. Phil is the subject of a teach event.
4. Betty is the object of a teach event.

Examples 3 and 4 are normally combined in a sentence such as "Phil teaches Betty." Note that the constraints preclude a verb from being used as an attribute, which is usually not the case with regular triples.

As tridbits are generated by applying rules, they are first tested to see if they make "structural sense." If not, the rule is rejected. By understanding the constraints of an assert attribute tridbit, it is possible to disambiguate attribute information, even though the grammar may not prescribe a fixed position for each element, or leave one out.

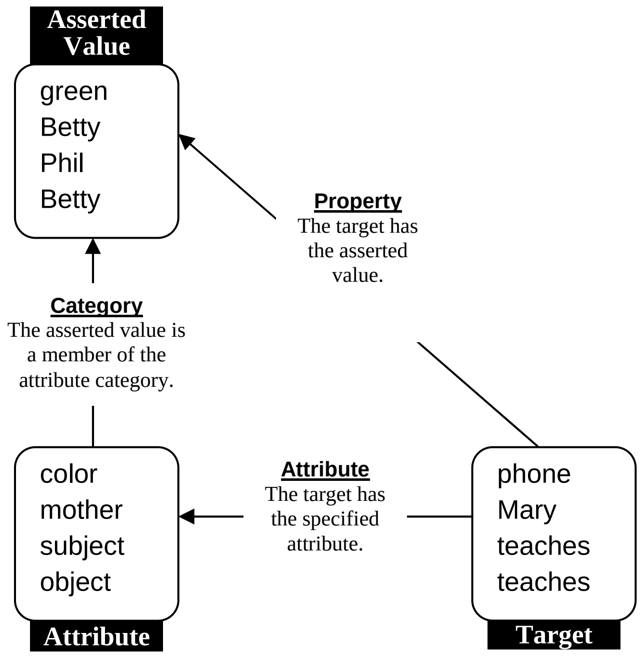

Asserted Value
green
Betty
Phil
Betty
Property
The target has the asserted value.
Category
The asserted value is a member of the attribute category.
color
mother
subject
object
Attribute
Attribute
The target has the specified attribute.
phone
Mary
teaches
teaches
Target

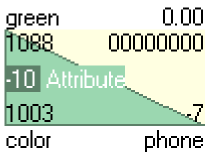
Figure 7: Constraints within an assert attribute tridbit.

Let's see how this works to eliminate possible interpretations in the example sentence "The phone is green." Below are the 3 rules that match the pattern >T>E2>P1, formed after processing the words into referent tridbits as shown in Figure 6. The expression on the right side of the → is the notation for defining tridbits. A tridbit definition consists of 4 values separated by comas and enclosed in <>s. The first value is the tridbit type. In this case they are all assert attribute tridbits. The remaining 3 values are the tridbit's attribute, target and value. The "P#" indicates that element is taken from the #th token in the pattern. "U0" indicates the element is undefined.

1st line: **Rule** 2nd line: **Assert tridbit generated**	**Does it make structural sense?**
34 >T>E2>P1 → <Ast/A, Property, P1, P3> <Ast/A, Property, phone, green>	Yes, phones have properties and green is a property, further the type of property can be filled in with color since color is the only category that green belongs to that makes sense here.
65 >T>E2>P1 → <Ast/A, P1, U0, P3> <Ast/A, phone, U0, green>	No, if phone is the attribute, green would need to be a member of the phone category and that doesn't make sense.
112 >T>E2>P1 → <Ast/A, Equivalent, P1, P3> <Ast/A, Equivalent, phone, green>	No, to be equivalent the target and value need to be the same referent types and phone is a thing while green is a property.

Figure 8: Result of applying the 3 matching rules, 34, 65, and 112 to the pattern: T **(phone)** E2 **(is)** P1 **(green)**

Applying the rules in the manner shown in the table above eliminates 2 of the 3 matching rules. Thus the final interpretation uses rule 34 to generate the assert tridbit shown below. In doing so, all the word tokens and referent tridbits are consumed, providing an interpretation of the sentence that makes structural sense.



The result of processing the sentence "The phone is green" is the assert attribute tridbit shown at left. This tridbit simply asserts that the color attribute of a specific phone referent has the value of green. This is the most basic type of assertion where the attribute is a category of observable properties. One might speculate that the need to convey this basic type of information might evolve a Tridbit-like capability. But the use of these structures seems to have evolved beyond encoding simple observations to a complex system of encoding more abstract information.

Now that we've covered the basics, let's look at the first tridbit configuration, which is referred to as an inferval in Tridbits. Names and categories are two ways to refer to a referent. Infervals can be thought of as a way to *define a referent using an assertion*. Inferval s are another way that we pick out a referent in the world when we don't have or want to give the referent a name and the category alone is not discriminating enough. Infervals are also what is formed when the value element of an assert attribute tridbit is missing.

The surface structure of English makes infervals easy to distinguish using what's commonly referred to as possessive forms. However if you read through the examples in Figure 9, the only normal ownership going on here may be between John and his car.

The color of the phone	The phone's color
Mother of Mary	Mary's Mother
The capital of Wisconsin	Wisconsin's capital
The date of the broadcast	The broadcast's date
The president of the bank	The bank's president
The car of John	John's car
The nose of the boy	The boy's nose
The phone number of Phil	Phil's phone number

Figure 9: Examples of infervals expressed 2 ways.

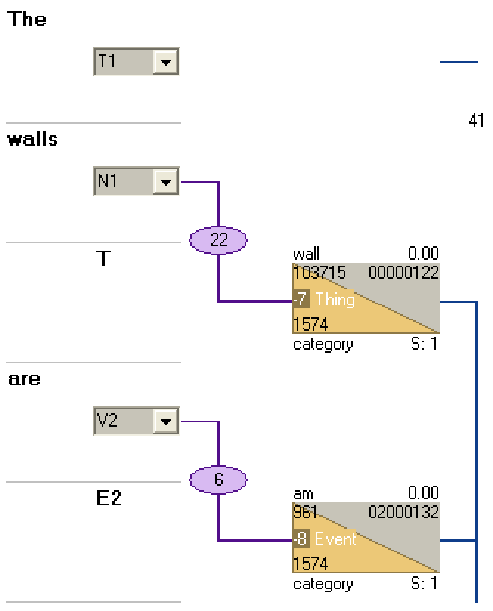
Lets revisit the assert attribute generated for "The phone is green." If value element was not filled in, the expression would still imply the color green since it refers to the color attribute of a specific phone, which has a specific value. Because of the constraints imposed on an assert attribute tridbit, we know at least one important thing about what can be filled in this element. Namely it must be a member of the attribute's category. In this specific case, that means the referent defined by this inferval will be an instance of a color. Since "color" is a category of properties, and properties are always general in scope, this inferval will be general in scope. However, most attributes are categories whose members are specific, so it is more common for infervals to be specific in scope.

If we want to refer to values that are defined by an assertion, such as the color of the phone, we need to have a way to link a referent tridbit to the value of the assertion. That is exactly what an inferval does.

Consider the example sentence:

The walls are the color of the phone.

This sentences uses an inferval (color of the phone) to assign a property to the walls. The tridbit diagram in Figure 10 shows how Tridbits processes this sentence to generate the inferval and what an inferval looks like.



<-34> Inferval representing "color of the phone"

Rule 110 matches the pattern >T=of>T and generates an inferval. Infervals are composed of two tridbits. There is a referent tridbit (-34) that is used to refer to the inferval. The inferval's referent tridbit has an attribute of "InferVal" to specify this specific way of defining a referent. The inferval's referent is linked to a second tridbit that is the inferval's defining assertion (-33).

An inferval is resolved by filling in its defining assertion's value element. Before it is resolved, the value element of the defining assertion points to the inferval's referent and the value element of the inferval's referent tridbit contains the attribute of the defining assertion.

If an inferval is resolved, the value elements are replaced with the resolved value and the inferval is subject to referent reduction. Many infervals are not resolved because they are the best way to refer to things like John's house, the boy's leg, or Suzie's cold.

<-38> Assertion uses the inferval

The inferval's referent tridbit is used like any other referent tridbit. Assert tridbit -38 uses the inferval "color of the phone" to specify a property of the wall.

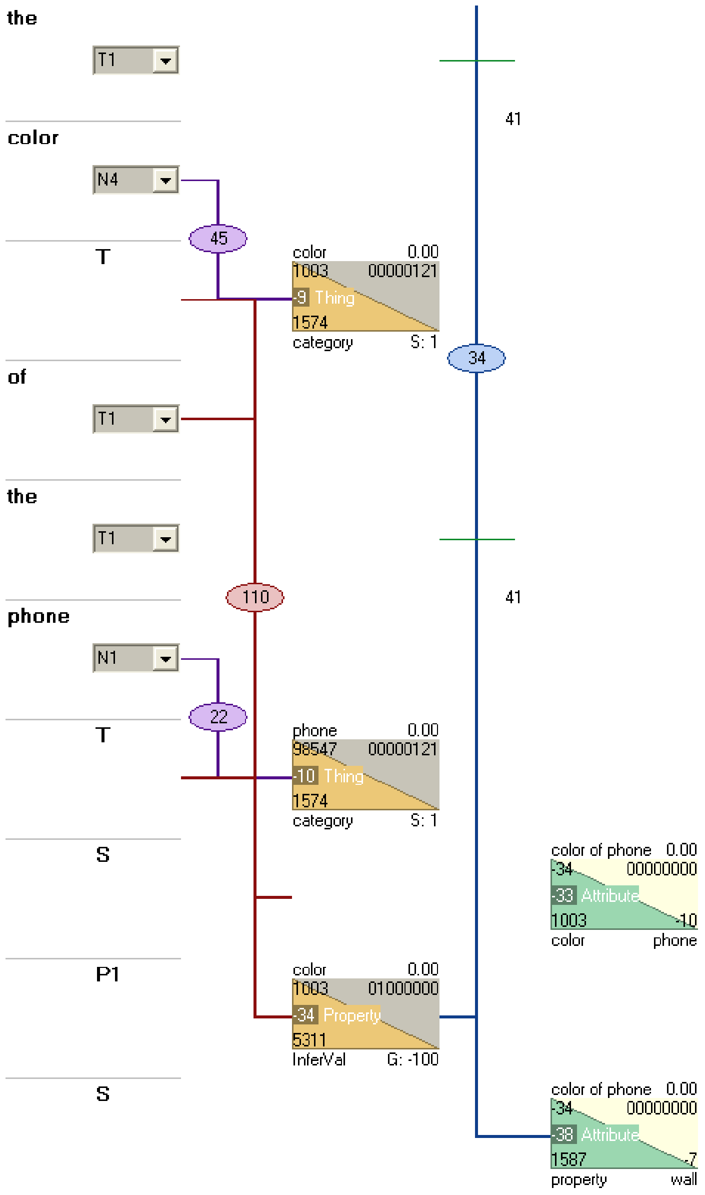


Figure 10: Tridbit diagram of "The walls are the color of the phone."

Infervals are an extremely powerful construction for encoding information. Each tridbit in the pair fits naturally in the tridbit system. Very little needed to be changed for these new linked tridbits to participate in searches, referent reduction, reasoning, and other tridbit processes.

With infervals in place well before the grant started, JotChat easily handled questions like "What is Phil's phone number?" With the start of the grant we were looking for a better way to handle Phil's cell phone number or work number. JotChat could answer those too - as long as multi-word concepts were defined before hand for each type of phone. Besides the work and inflexibility of predefining categories, the new concepts wouldn't automatically be categorized as phones; that information would have to be entered as well. Since humans make up new categories on the fly, it seemed there should be a natural construction Tridbits could utilize to represent this construction. The structure is called a General Qualified Concept (GQC).

As the name implies, general qualified concepts refer to general concepts. Thus only the second sentence below refers to a GQC:

Neal liked the chocolate cake.

Neal likes chocolate cake.

The first sentence is much easier to represent, as it refers to a specific instance of cake that has the attribute of being chocolate. The GQC in sentence 2 refers to the general concept of cakes having the property of being chocolate. One issue is that the attribute of being chocolate cannot be asserted as a property of the general concept of cakes or the result would be that all cakes are chocolate. Also one cannot assume that Neal likes all cakes, one can only be assured that he will like cakes that are chocolate.

Because of the distinct behavior of GQCs they need to be distinguished from other types of referents. Tridbits had done this for a long time, but not using an optimal structure that retains the category characteristics and combines naturally with infervals to represent the complicated expressions in the karate teacher example, coming up next.

It was the observation that GQCs are actually quite similar to infervals that suggested that they be represented in a similar structure. Consider these pairs:

	Inferval	**GQC**
1	John’s car	American cars
2	Hamster’s cage	Hamster cage
3	Barry’s cats	House cats
4	The bank’s president	Bank presidents
5	Alice’s disease	Addison disease
6	Joseph’s family	Upper class families
7	Harry’s legs	Hairy legs
8	World War II’s story	World War II story
9	Beer’s king (i.e. king of beer)	Beer drinking kings

Figure 11: Comparison of similar Infervals and general qualified concepts (GQC).

The second concept in each pair is the same and in some instances, such as hamster and cage, both concepts are the same! The difference is that in the infervals, the second concept is the attribute of the inferval’s defining assertion, so the referent will be a member of the attribute’s category. Thus, infervals will tend to be specific in scope, unless modified to be general.

GQCs are also represented by a referent tridbit/ assert tridbit pair. The assertion is called the GQC’s qualifying assertion because it uses the first concept to qualify the second concept in a way that may or may not be discernable a priori. The result is a subset of the second concept rather than a specific member, as it was for infervals. Thus, GQCs will be general in scope, unless modified to be specific.

Here are some examples that show how one can force a different scope on what is normally an inferval or GQC. Making GQCs specific is usually done with definite articles as described above in order to pick out one or more specific members of the GQC such as *these* American cars or *the* house cat. If the speaker really wants to pick out every member of the GQC they might say *all the* American cars. Consider the difference between:

American cars have windshields.   vs.   All the American cars have windshields.

The first sentence is more likely to be construed as a requirement of the set of cars that are American, whereas the second is more of an observation about a specific group of cars.

It can also make sense to force an inferval to be general when one really doesn't want to pick a specific referent. Consider:

I will drive John's car.   vs.   I will drive one of John's cars or I will drive a car of John's

The first sentence uses the standard inferval construction so even if John has multiple cars, it is assumed I have a specific car in mind that I will drive. By adding the phrase "one of" or using the indefinite article in the second sentence I modified the referent to be general in scope, indicating the specific car has not been picked out. Language gives the speaker many ways to convey to the listener the precise referent she has in mind, whether the referent is specific or general and how important it is to identify it.

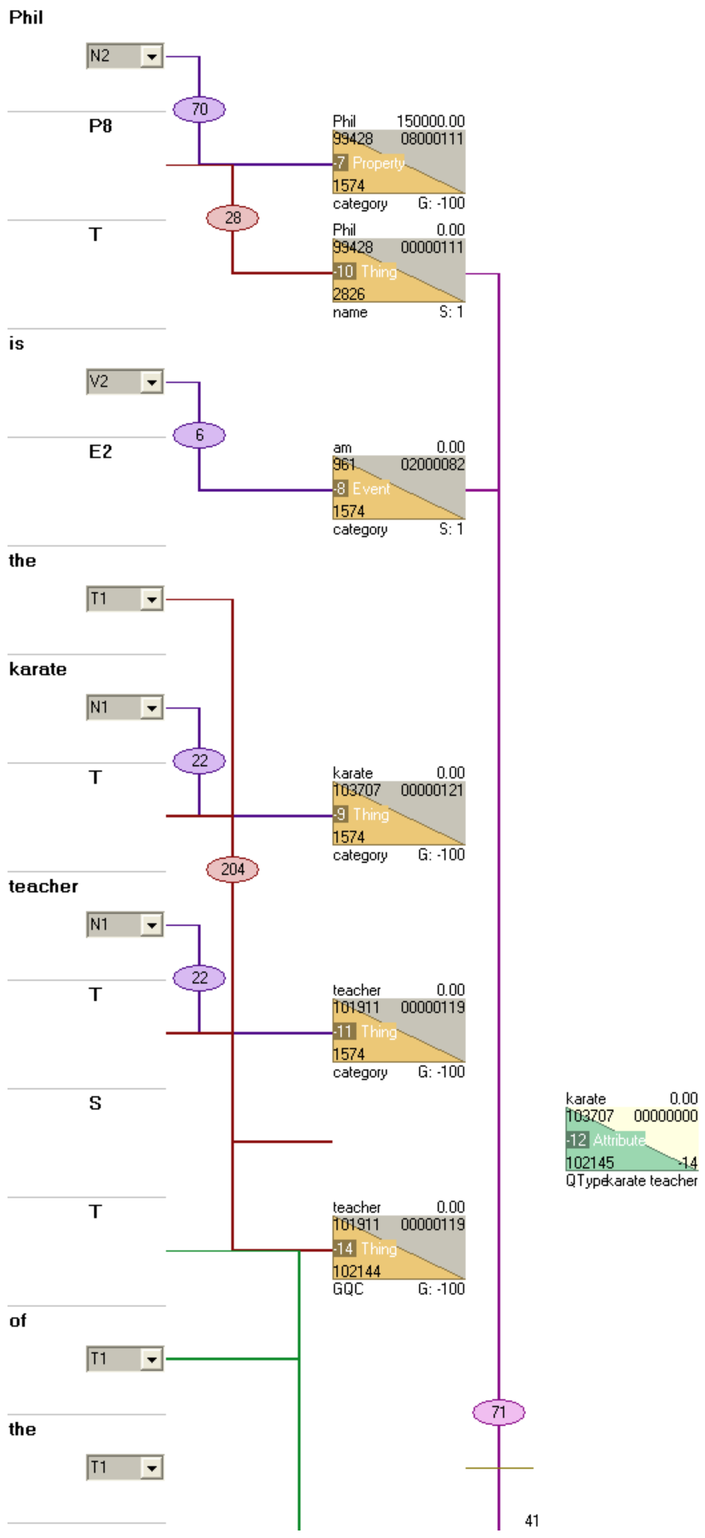
Now lets look at the tridbit diagrams for the sentences in the karate teacher example and see how the inferval and GQC representations work together to allow scope and category information to combine in an optimal way.

The first sentence in karate example is:

Phil is the karate teacher of the mother of Mary.

This sentence contains two infervals and a general qualified concept. Its tridbit diagram is shown at right and spans 2 pages. The appendices contain all the word use categories and rule definitions used in these tridbit diagrams so you can follow the parsing if you are so inclined.

I will walk through each of these diagrams, pointing out how they captured the meaning, use of infervals and GQCs and other things of interest.



<-7> Names generate property tridbits

Names (word use category N2), in this example "Phil," generate a property tridbit that represents the name Phil. Because it is a property referent, it will be general in scope. All entities that are named Phil share the same name property, Tridbits does not instantiate a specific property for each one.

<-10> Name is transformed to a thing tridbit

Usually names refer to a thing referent so the generation of a property tridbit is most often followed by the application of rule 28, which transforms a name property tridbit to a thing tridbit.

<-14> Karate teacher is a GQC

The concept of "Karate teacher" is created on the fly by qualifying the base category "teacher" (tridbit -11) with the qualifying type (QType) of "karate" (tridbit -9). QType is the attribute used to represent the qualifying relationship in a GQC. QType differs from a simple subset in that the value referent alone is not a subset of the target.

The new GQC referent for karate teacher is linked to the assert tridbit (-12) that qualifies it.

<-67> 1st inferval: mother of Mary

The phrase “the mother of Mary” prompts the first inferval to be generated. We never find out who the mother of Mary is, but the inferval (-67) is a referent tridbit used to refer to her. The inferval is linked to an assert tridbit (-66) that defines the inferval by specifying an attribute of a target, in this case the attribute is mother and target is Mary.

<-74> 2nd inferval: karate teacher of the mother of Mary

This inferval combines the previously generated GQC (karate teacher) and inferval (mother of Mary). This time we do find out who Mary’s mother’s karate teacher is, so the referent tridbit for Phil (-10) is put in to the value element of both the inferval (-74) and its defining assertion (-73).

Note that the scope of the 2 infervals is specific. This is common but not always true. The scope of an inferval depends on what kinds of members belong to the defining attribute’s category. In other words, mother of Mary is specific because the members of the mother category are specific individuals. An inferval such as the dog’s breed would be general because breed is a category of categories.

<-77> Equivalence: tridbits with the same referent

The point of the sentence is to tell us that the person named Phil and the person who is Mary’s mother’s karate teacher is one and the same. The assertion of equivalence (-77) represents that fact. Since each unique referent is ideally represented by one consistent referent tridbit, the preferred referent will replace the other.

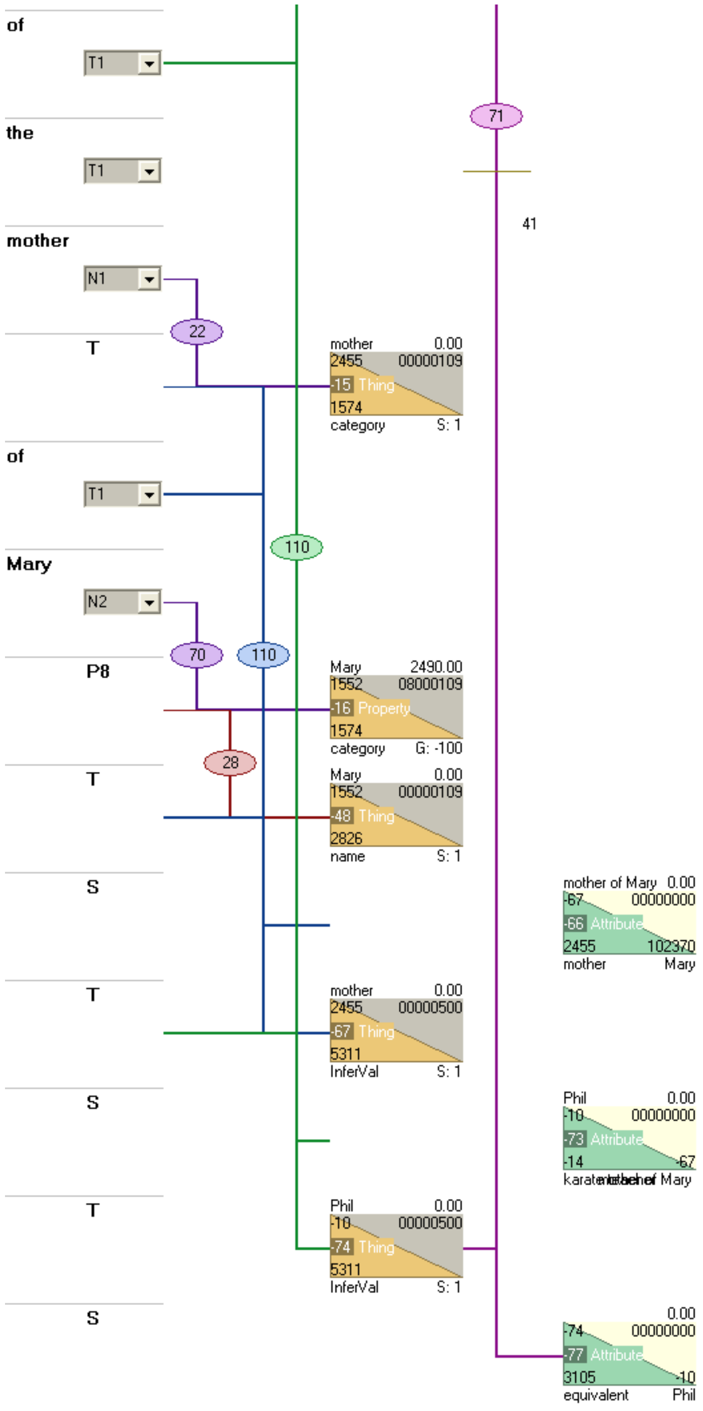

of
T1
the
T1
mother
N1
T
of
T1
Mary
N2
P8
T
S
T
S
T
S
71
41
22
110
70
110
28
mother 0.00 2455 00000109 -15 Thing 1574 category S: 1
Mary 2490.00 1552 08000109 -16 Property 1574 category G: -100
Mary 0.00 1552 00000109 -48 Thing 2826 name S: 1
mother 0.00 2455 00000500 -67 Thing 5311 InferVal S: 1
Phil 0.00 -10 00000500 -74 Thing 5311 InferVal S: 1
mother of Mary 0.00 -67 00000000 -66 Attribute 2455 102370 mother Mary
Phil 0.00 -10 00000000 -73 Attribute -14 -67 karate teacher Mary
0.00 -74 00000000 -77 Attribute 3105 -10 equivalent Phil

Figure 12: Tridbit diagram for "Phil is the karate teacher of the mother of Mary."

The next sentence in the scenario is a bit more complex than the first:

Mary's mom's karate teacher's studio phone number is 555-1111.

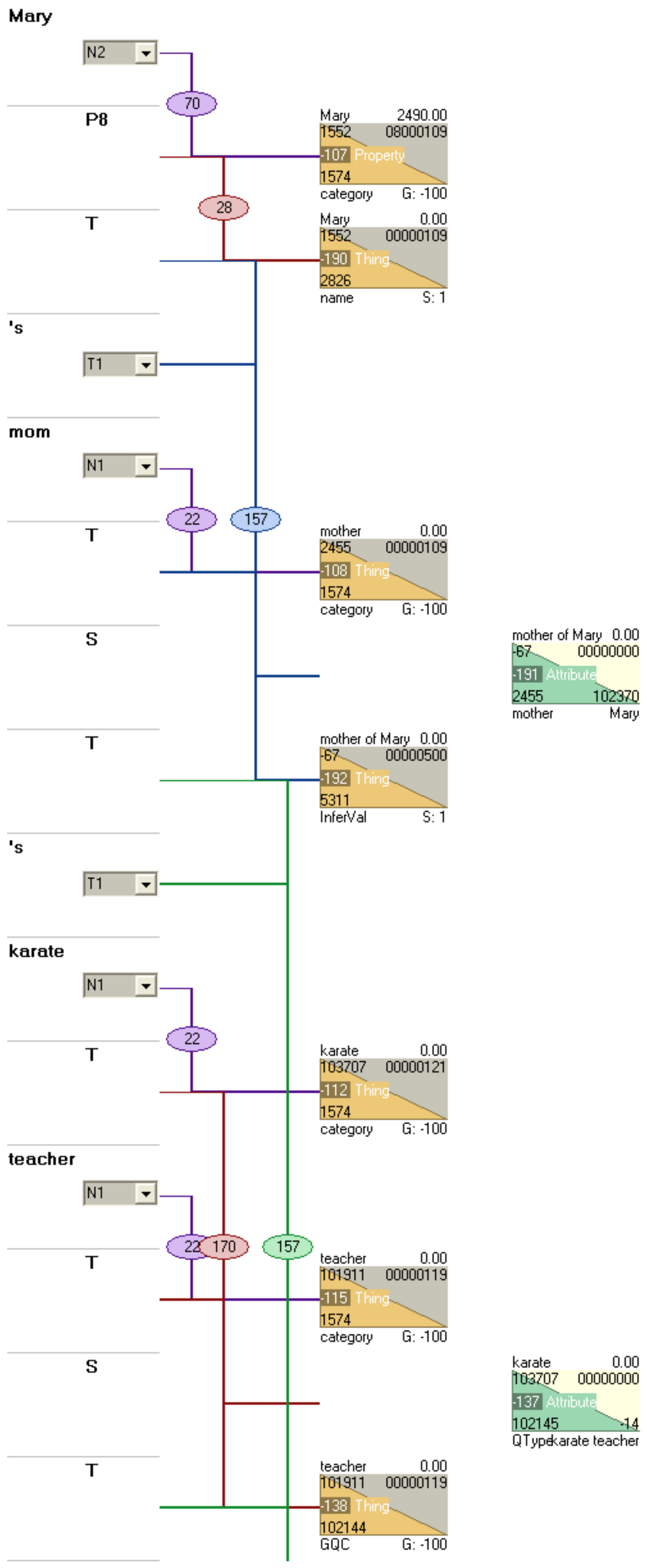
This sentence contains three infervals and three general qualified concepts. It also refers to referents established in the first sentence that will need to be reduced.

<-190> Mary must be reduced

Like the first sentence, this sentence also produces a thing tridbit to represent Mary. If we allow a referent, in this case Mary, to be represented by more than one tridbit, it becomes much too complex to manage the information concerning the referent. Thus referent reduction is performed to detect this case and enforce consistent use of the preferred referent tridbit, which is normally the first mention of the referent in the conversation. You can't tell from the drawing of the second tridbit representing Mary (-190) that it has been reduced. However the tridbit ID that is used to refer to Mary in other tridbits, such as the target of tridbit -191 will be the ID of the first Mary tridbit, which has been reduced to its ID in the knowledgebase.

<-192> Mary's mom is reduced

The first sentence used the phrase "mother of Mary" to refer to the same person as "Mary's mom" in this sentence. While they were processed using different syntax rules, both generated infervals (67, 192) with identical defining assertions (66, 191). We still don't know who the mother of Mary is, nonetheless referent reduction is able to detect that the infervals have the same referent and uses the ID of the first inferval (-67) to represent Mary's mother.



<-194> Mary's mom's karate teacher is reduced

Referent reduction also detects that the two expressions for Mary's mom's karate teacher represent the same referent. In this case we do know who the karate teacher is, so the referent tridbit for Phil (-10) is put in to the value element of both the inferval (-194) and its defining assertion (-193).

<-136>Studio phone number combines defined with on-the-fly GQC

Studio phone number is a GQC created on the fly as a category of phone numbers qualified by the QType studio. You can't tell from the tridbit drawing, but the concept of phone number is also defined as a GQC. In this case however, the GQC is stored in the knowledgebase as a referent tridbit and its qualifying assertion. The dictionary entry for "phone number" points to this GQC in the knowledgebase.

Because of the consistent use of structure and constraints across tridbits, A GQC such as studio phone number is inherently identified as being a phone number and a number. Any additional information derived based on the QType at either level can be applied to the more highly qualified concept.

<-196>Phil's studio phone number

What starts out as a very complex inferval, "Mary's mom's karate teacher's studio phone number", is simplified in the defining assertion (-195) to "the studio phone number of Phil" thanks to referent reduction. This is a case where the inferval is general in scope because phone number is a category of properties, which have only one instantiation.

<-197> Equivalent phone numbers

This equivalence is between Phil's studio phone number, and a property generated by an expression that qualifies as a phone number. Both are property referents of the same sub type and thus compatible.

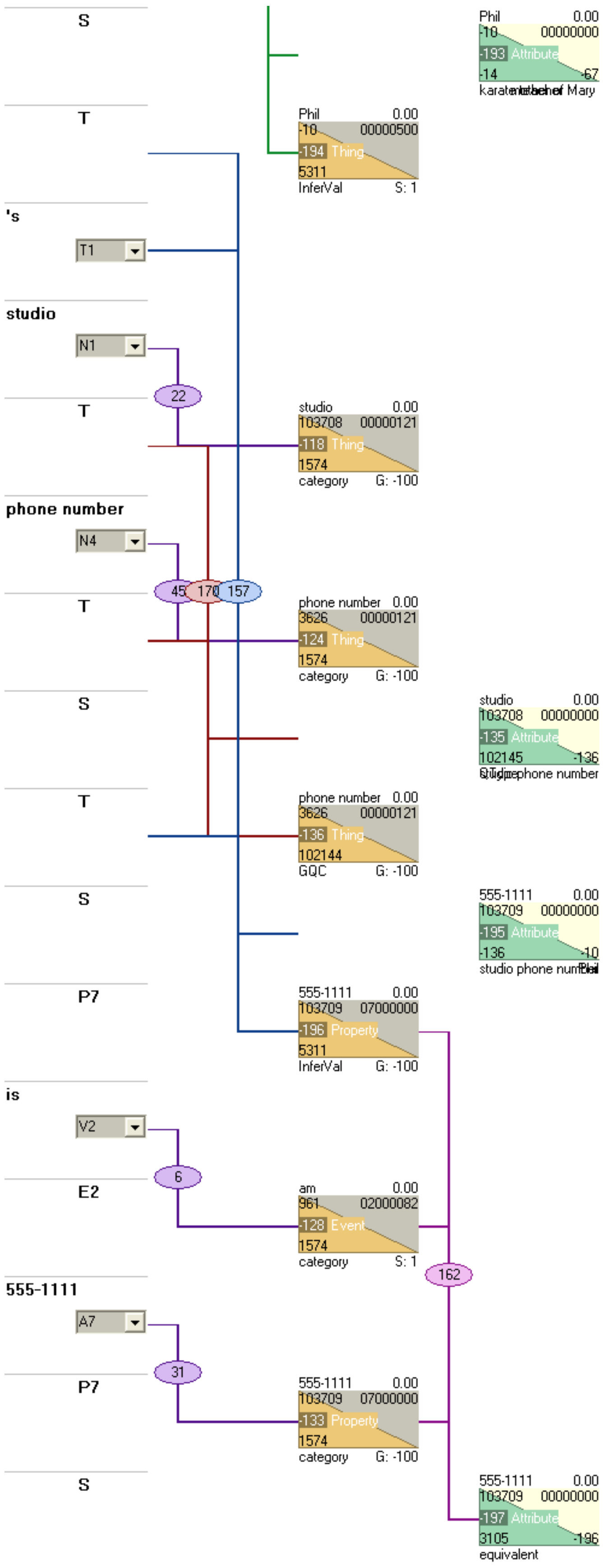


Figure 13: Tridbit diagram for "Mary's mom's karate teacher's studio phone number is 555-1111."

Here is the last sentence of the scenario:

What is Phil's phone number?

Let's see how a question like this is represented and answered.

<-208> "What" generates an inquiry placeholder

"What" is a tag word. Tag words don't represent phenomenon in the world, so they do not, as a category, generate tridbits. In other words, there are no syntax rules that refer to tag words as a category. Rather the function of a tag word is defined by its involvement in syntax rules that reference it as a literal. (Which is why defining them is so difficult! Other examples of tag words include "the", "a", "of", "how", "if", "but", etc.) In this case, "what" generates a referent tridbit that acts as a place-holder tridbit, which needs to be filled in to answer the question.

<-218> "What" is equivalenced to an inferval

The sentence equivalences the placeholder generated by "what" with the inferval (-217) that represents the phone number of Phil, telling us they are one and the same. So to fill in the place holder, we need to resolve the inferval "phone number of Phil." (i.e. find its value element)

Because referent reduction is constantly reducing referents to their simplest previous mention in the conversation, the information is relatively easy to find in the previous sentence.

Inferval -196 represents the studio phone number of Phil, Phil being the reduction of Mary's mom's karate teacher.

JotChat needs to apply the reasoning that a studio phone number is a phone number to verify that the value element of the inferval's definition is the type of information we are looking for. This reasoning is easy thanks to the structure of GQC's. The answer then is the value element of assert tridbit -195 or 555-1111.

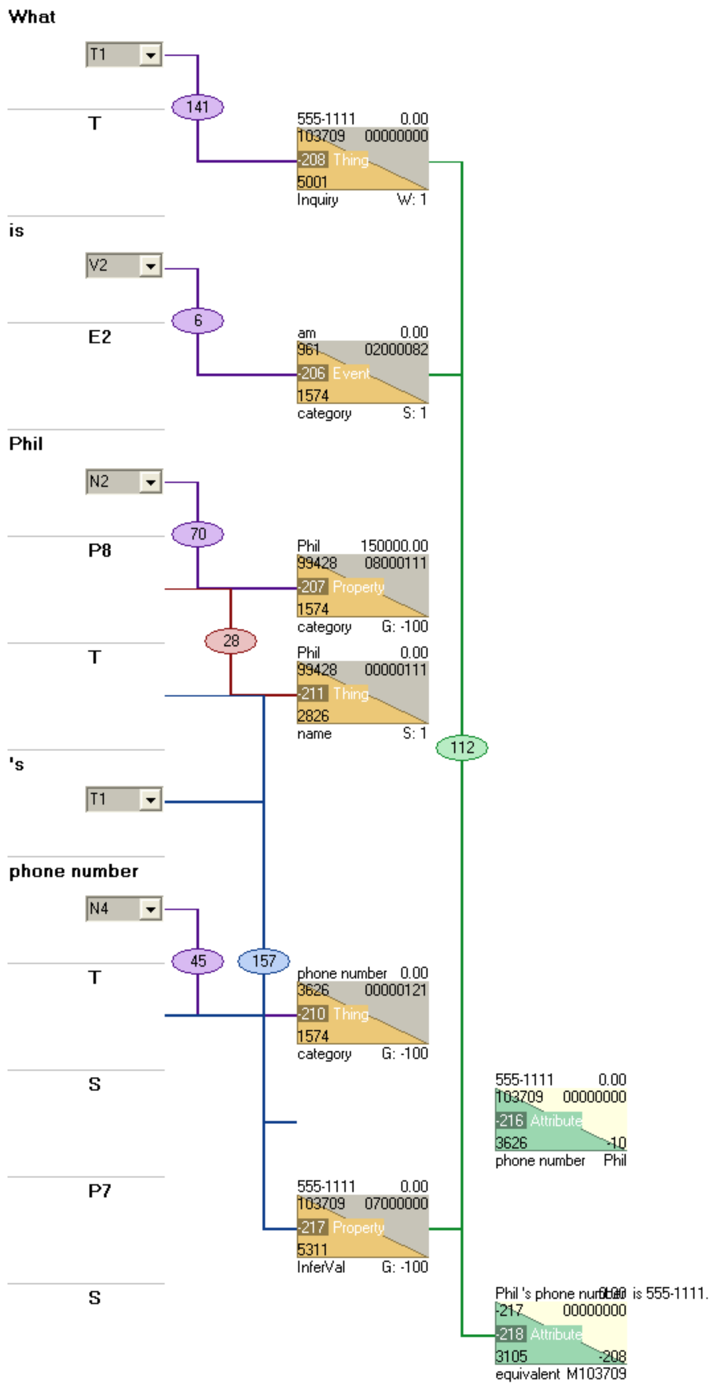


Figure 14: Tridbit diagram for "What is Phil's phone number?"

The Tridbit solution to the karate teacher exercise utilizes very simple structures and processes that leverage the fundamental concepts of category and scope to reflect the powerful organizational tools that they are.

GQCs are a powerful language construct that represents a significant advance in Tridbit technology. The combination of infervals and GQCs provides an important mechanism used by natural language for encoding information. We have not fully realized all the benefits of these constructs. Further work will involve integrating them more fully in the reasoning system to pull in contextual knowledge.

Another intriguing finding is the essentially identical configurations that appear in assert compare tridbits. Assert compare tridbits are a second type of assertion used to represent comparative information whose constraints differ from assert attribute tridbits. Much like infervals represent an assert attribute tridbit with the value element missing (e.g. the height of Karen), compvals represent an assert compare tridbit with the target element missing (e.g. shorter than Dan). GQCs also have a complement in assert compare tridbits that looks to be a powerful construct for reasoning with comparative information.

Compvals have been integrated into the understanding process sufficiently to validate their usefulness in representing clauses, location and other information that tends to be comparative in nature. While the details will have to wait for a future report, the knowledge of their existence emphasizes the role that these structures play in understanding natural language.

Appendix A - TRIDBIT WORD USE CATEGORIES (AS CURRENTLY IMPLEMENTED)

Noun subcategories include:

Thing category **N1**	A *thing category* is the most common type of noun, representing a category of things. **Examples are rocks, cities, water, people and ideas.**
Event category **N5**	An *event category* is a noun that represents a category of events. **Examples are dances, conversations, concerts, repairs, singing and thinking**.
Property category **N4**	A *property category* is a noun that represents a category of properties. **Examples are color, size, temperature and weight.**
Assertion category **N7**	An *assertion category* is a noun that represents a category of assertions. **Examples are causes, reasons, results and beliefs.**
Proper noun **N2**	*Proper nouns* are names used to refer to specific things or events. **Examples are the Mississippi River, Patrick Henry, Woodstock and the Rose Bowl.**
Pronoun (various types) **N3, N6**	*Pronouns* are used to refer to something, without providing a name or category of the referent. **Examples are you, me, they it, who and what.**

Verb subcategories include:

Event **V1**	Most verbs are used as *events*, which refer to a specific event. **Examples are give, laugh, drive, think, like** and all forms of person and tense, such as give, gives, gave, given and giving.
Assignment **V2**	The verb type *Assignment*, is given to the verb "is" when used to assign properties. In the future "is" may be reclassified as a tag word rather than a special type of verb.
Subject property **V3**	A *subject property* is the form of a verb that is used as an adjective to indicate the referent being described is the subject of the event. **Examples are dancing, throwing, chewing, painting and knowing**.
Object property **V4**	An *object property* is the form of a verb that is used as an adjective to indicate the referent being described is the object of the event. **Examples are thrown, chewed, painted and known**.

Adjective/Adverb subcategories include:

Relative property **P1**	*Relative properties* name a relative position within a property category such as brightness where the category is scalar in nature. **Examples are bright, red, big, silly and heavy, brightest, reddest, biggest, etc.**
Compare operator **P2**	*Compare operators* indicate a ranking within a property category. **Examples are more, better, brighter, redder, bigger, hotter and heavier.**
Quantity **P3**	*Quantities* names an amount, independent of any units. **Examples include five, a third, ninety-nine and 5,467,892.45.**
Date/Time **P5**	*Date/Time* names a point in time. **Examples include July 4, 1776, 6:00, tomorrow, noon or 11/2/2004 23:59.**
Spatial location **P6**	Defines a point or region in space that applies to anything at that location such as **an address or coordinate**, but not a country, planet, etc which are things that can participate in a relative location.
Designation **P7**	A symbol that contains meaning or a pattern but differs from the normal conventions of looking up words in the dictionary. Unlike dates and numeric quantities, designations do not resolve to a value based on the symbol. **Examples include phone numbers, email addresses and social security numbers.**

Appendix B - TRIDBIT SYNTAX RULES REFERENCED IN THIS DOCUMENT (OUT OF A CURRENT TOTAL OF 143)

	Rule	Example phrase that triggers it
6	>V → <Ref/E Category S:1 P1>	“is”
22	>N1 → <Ref/T Category G:-100 P1>	“phone”
28	>P8 → <Ref/T Name S:1 P1>	Phil
31	>A → <Ref/P Category G:-100 P1>	“green”
34	>T>E2>P1 → <Ast/A, Property, P1, P3>	the phone is green
45	>N4 → <Ref/T Category G:-100 P1>	“phone number”
65	>T>E2>P1 → <Ast/A, P1, U0, P3>	the color is green
70	>N2 → <Ref/P Category G:-100 P1>	“Phil”
71	>T>E2>T → <Ast/A, Equivalent, P1, P3>	he is Phil
110	>T=of>T → <Ast/A, P1, P3, N0> <Ref/T Inferval P-1 M0>	mother of Mary
112	>T>E2>P → <Ast/A, Equivalent , P1, P3 >	What is Phil’s phone number
141	=what → <Ref/T Inferent W:1 M0>	“what”
157	>T=’s>T → <Ast/A, P3, P1, N0> <Ref/T Inferval P-1 M0>	Mary’s mom
162	>P>E2>P → <Ast/A, Equivalent, P1, P3>	Phil’s phone number is 555-1111
170	>T>T → <Ast/A, QType, P2, P1 > <Ref/T GQC P-1 P2>	studio phone number
204	=the>T>T → <Ast/A, QType, P3, P2 > <Ref/T GQC P-1 P3>	the karate teacher