# Modelling grammatical processing with spreading activation and default inheritance.

## Previous ESRC projects

The proposed project will be the third phase of a larger project whose first two phases have already been funded by ESRC:

- Software for an inflectional network (£40,398, 2001)
- Modelling spreading activation and default inheritance in Word Grammar (£92,283, 2003-4).

The evaluators of the first phase rated it outstanding and good respectively. The second phase is still ongoing. This series of projects arose out of a more ambitious proposal to ESRC:

- Parsing by spreading activation in Word Grammar (£276,740, 2001-3).

This proposal was rejected, but the Board Assessor encouraged resubmission:

> "This is a project that has considerable promise in that it may well provide a more realistic model of human syntactic processing than the mainstream approach that has been provided in theoretical linguistics. However, the project as it stands is very ambitious, very expensive particularly given the number of potential problems that developing it may well imply. It is thought that a more piecemeal approach, which inevitably will mean a more modest proposal, may be a better way forward for this very innovative and interesting approach to modelling."

## Theoretical background

The overall goal of the rejected proposal was to produce a software package which would allow us to model a novel theory of parsing – that is, of how humans analyse the grammatical structure of a sentence. The underlying assumption is this:

(1) Language is a network.

Although this claim is fairly controversial in theoretical linguistics, it is widely espoused in 'cognitive' theories such as Cognitive Grammar and Construction Grammar (, ), and is particularly well developed in formal terms in the theory that we are testing, Word Grammar (, , ).

Hypothesis (1) also fits well with the views of many psychologists and psycholinguistics. In psychology it is normal to view knowledge as a network in which nodes are activated, and in which this activation spreads, in a blind fashion, to neighbouring nodes (:259, :16, ). Evidence from speech errors and priming experiments shows that this spreading activation plays an important part in speaking and listening, so it seems safe to conclude that language must be a network that can support spreading activation. The goal of the rejected proposal was to explore the implications of this general idea in an area where spreading activation has not been invoked: recognising grammatical structure. Two projects later, this still seems a realistic target, and the present proposal is for a third project which will achieve it.

In outline, the idea is this:

(2) Wherever there is a choice, spreading activation will generally ensure that the most active alternative is the one that should be chosen; and when this

assumption in fact produces the wrong answer, the result will be a human-like error.

The idea in (2) may, of course, be wrong, but until it has been tested it cannot be rejected. It is so general that it can apply to any area of cognition from general problem-solving to language, and within language it can apply to all areas from phonology to semantics, and even to both directions of processing, production and perception. However our main goal in the proposed project will be to apply it to parsing – the understanding of grammatical structure.

Another fundamental tenet of Word Grammar is that generalisations are achieved by means of **default inheritance**, which allows default generalisations to be overridden by exceptions. The network hypothesis means that default inheritance is combined with spreading activation, and this pairing will play an important part in the proposed project. So far as we know, Word Grammar is the only theory that integrates default inheritance and spreading activation into a single system which could be modelled computationally. The logic of default inheritance is displayed diagrammatically in Figure 1, where each small triangle indicates an "is-a" relation; for example, A' is-a A. This diagram illustrates another theoretical tenet which is unique to Word Grammar: the assumption that relations as well as nodes are related in is-a hierarchies. In processing, default inheritance interacts with spreading activation in a way which we shall explain below.

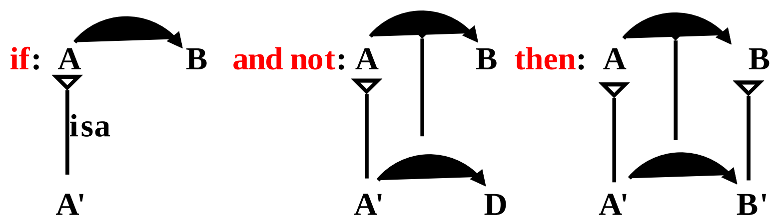


**Figure 1**

The theoretical foundations for the overall project were mostly in place before it started, but some of the details have evolved under pressure from this work. We can now define more precisely the kind of network that we are trying to model, which we call an '**inheritance network**':

(3) In an inheritance network, every node and link belongs to one or more classification hierarchies, which allow default inheritance of properties.

We also have a clearer idea about how default inheritance interacts with spreading activation, as we explain below; and the project has changed and clarified our view of some details of linguistic structure.

## Previous work funded by ESRC

### Phase 1: WGNet

The first step towards testing (2) was to build a general-purpose network editor, which we called WGNet. This was the goal of the first project, and as the project's title suggested, the aim was to apply the editor to one small area of English grammar: inflectional morphology. WGNet allows a user to display any inheritance network on screen in a convenient format, and to interact with the network by adding, deleting or changing nodes and links. Figure 2 shows a simple (non-linguistic) network with the WGNet toolbar above it. The left half of the screen contains a hierarchy of relations – in this case, family relations such as ‘mother’ and ‘daughter’ – while the right half contains a hierarchy of people (John, Bill, Mary and Ann) and types of people (e.g. man, adult). The diagonal lines link one person to another via some named relation. The numbers show the distance, counted in links, of every node from the currently selected one, the one for Mary. The system will accept and display any inheritance network, though we plan to improve the display during the current project.

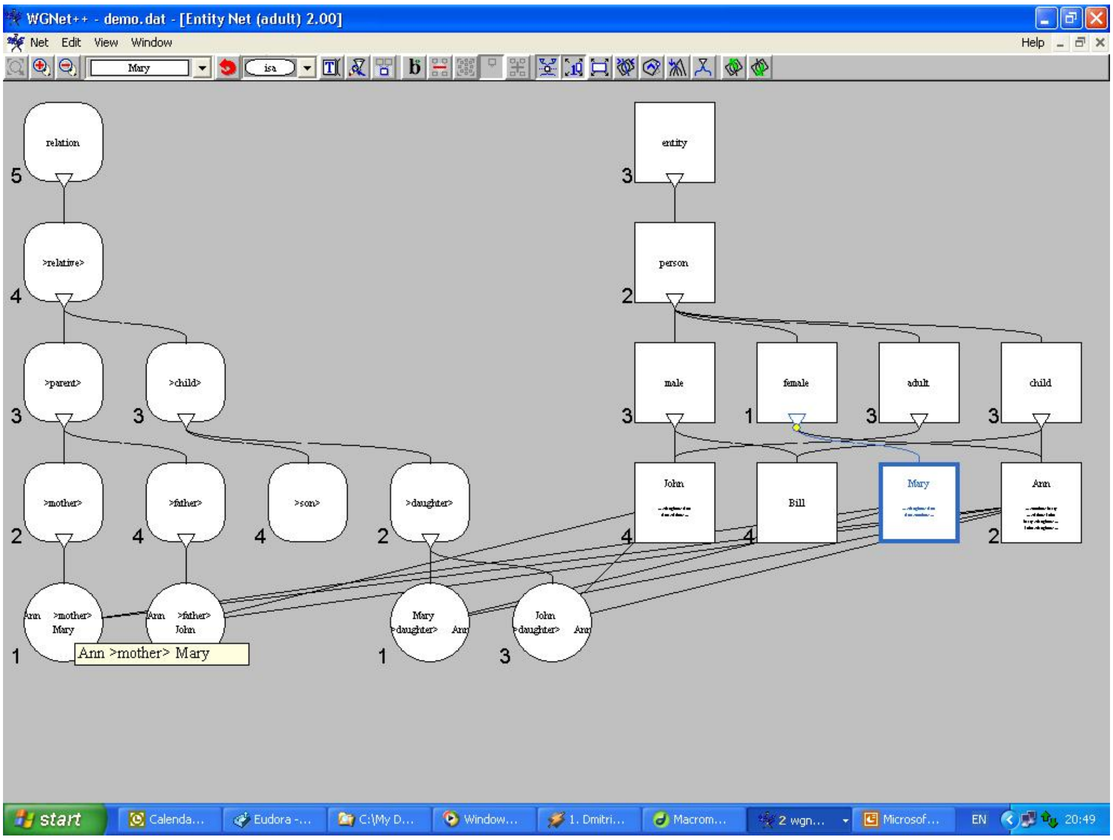


**Figure 2**

We have also built a database for a significant portion of English inflectional morphology, and this has been our main test for the effectiveness of the program. Figure 3 illustrates the problem of presenting even a small fragment of this data in a usable way on screen.

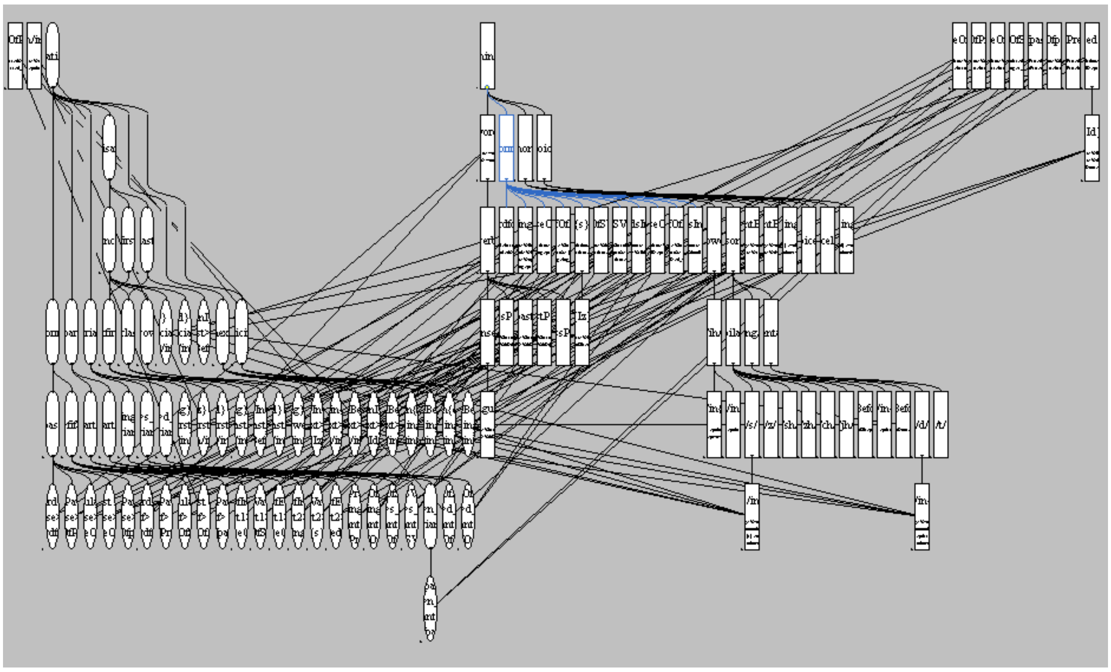

**Figure 3**

However, the complexity is easily controlled by the user, who can convert this display into Figure 4 with just five key-strokes so as to focus on the immediate environment of just one node (the one for the suffix {ing}). The rather complex names, such as “%fifOfIng”, denote ‘variables’, a notion explained below.

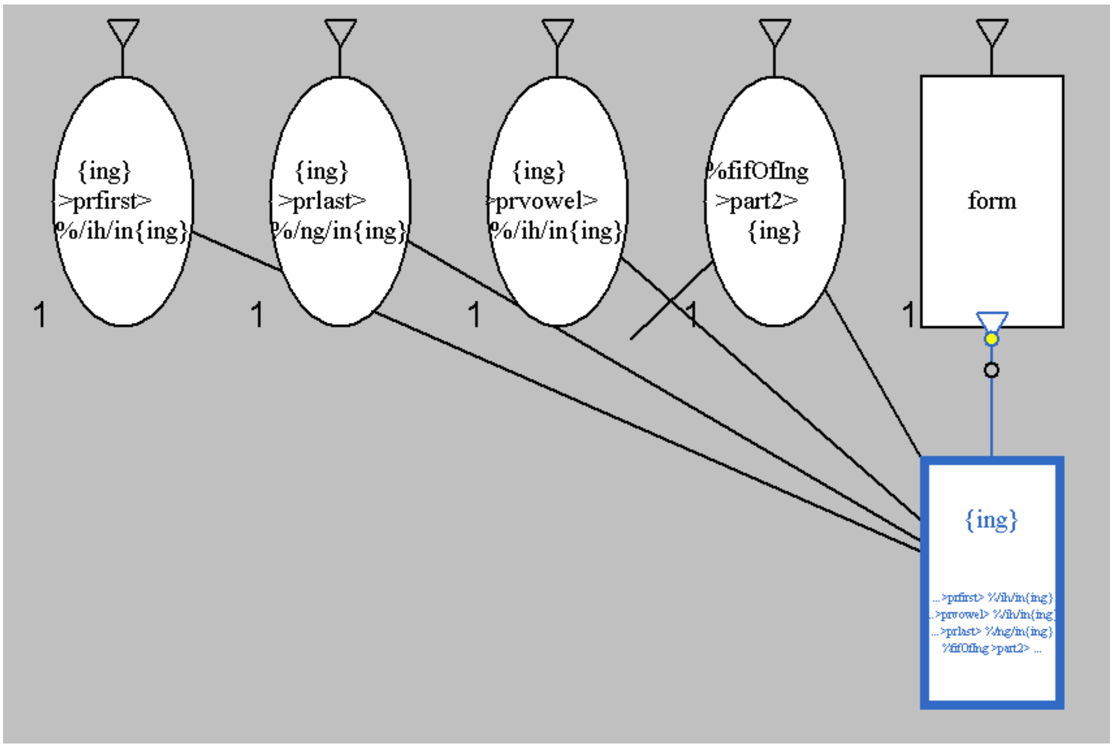


**Figure 4**

## Phase 2: WGNet++

Although it could still be improved, the interface is ready for use, so the current project is targeted at the functionality of the processes for exploiting networks: default inheritance and spreading activation. This expansion of WGNet is what we call "WGNet++". Unfortunately, the present grant-application has to be made less than half-way through the project, but we already have a primitive system for displaying **inheritance** in a small 'property-inspection' window, which lists all the links for the currently selected node. This can be seen in Figure 5 , where the circled property is inherited rather than stored. We expect the default inheritance system to be fully operational in the next few weeks.

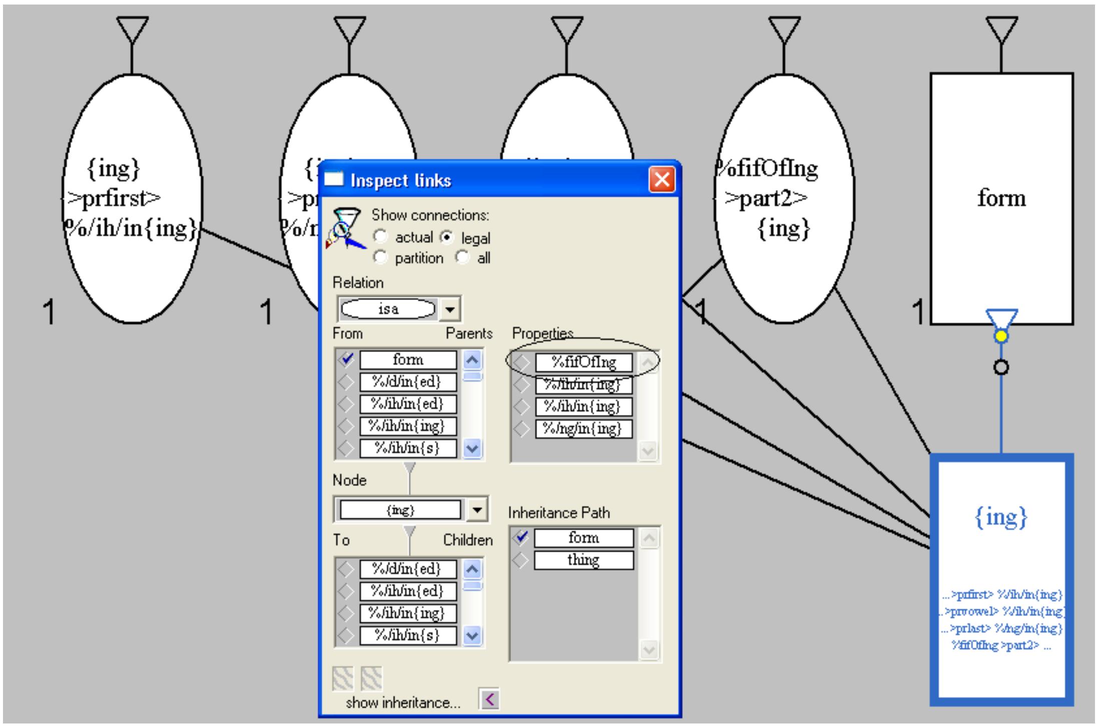


**Figure 5**

**Spreading activation** will extend the system in a different way. What is needed is, first, a mechanism for recording for each node a level of activity, such as a figure between 0 and 1, and second, a mechanism for allowing the activity levels to change through time as activation spreads through neighbouring nodes. No doubt many nodes are changing activation simultaneously in the human mind, but in this research tool we shall simulate parallel processing by dividing time into a series of small notional transitions and updating all node activities at each transition. This will allow the researcher to 'step' through the changes slowly enough to study the effects on the network.

Ultimately the model will need a formula for updating activity levels comparable to the one built into WEAVER++ (), but it would be premature to select a formula at this stage. What we need is a facility (in the WGNet++ program) to allow a researcher to define formulae as needed, so we aim to define a wide range of parameters which could possibly be relevant. For example, we assume we shall need the following functions (among others):

- a function for degrading all activity levels through time,

- a threshold for each node (which may vary between nodes) which determines when it ‘fires’,
- a function for spreading activity from firing nodes to their neighbours,
- a function for adding this activity to the activity of the targets and subtracting it from that of the source.

We shall be able to test formulae on the growing database of inflectional morphology, so we expect to have a tentative formula by the end of phase 2. We also expect to have expanded the database considerably by incorporating a large list of words and word-frequencies from the Survey of English Usage. The frequencies are likely to be relevant to the spreading activation formula, since activation speed is correlated with frequency.

## The proposal for Phase 3: Grammatical processing

The present grant proposal is for Phase 3 of the overall project, which will develop WGNet++ into a model of grammatical processing. The main questions that we shall try to answer are the following:

- What is the formula for spreading activation?
- How can spreading activation and default inheritance be controlled so that they produce the desired outcomes without unwanted side-effects?
- When applied to single ambiguous words, does the model select the interpretation indicated by some pre-activated semantic node?
- When applied to sentences, does the model generate appropriate dependency links between the words?

We discuss these goals separately below.

### The formula for spreading activation

As explained earlier, we assume that spreading activation is a uniform process that applies in the same way to all kinds of knowledge, so we expect to find a single general formula. The research literature offers many examples of formulae that may be relevant, but none were designed specifically with inheritance networks in mind so we probably cannot adopt any existing formulae without change. The most important peculiarity of our networks is that the links are themselves treated as concepts which can spread activation to one another. It is easy to imagine a range of formulae which would fit what we know about spreading activation (e.g. regarding the fan effect) but we hope that WGNet++ will provide a test bed for selecting among them.

### The control algorithm for spreading activation and default inheritance

What is missing from the model described so far is any overall mechanism for controlling the flow of information and directing it usefully. Default inheritance allows information to be inherited – but when should it apply? Spreading activation passes activity around the network – but how does activation help the hearer or speaker? The following paragraphs explain the algorithm that we propose to apply.

The algorithm rests on two fundamental (and familiar) distinctions:

- between **types** and **tokens**
- between **constants** and **variables**.

The type/token distinction contrasts the permanent contents of the network (types) from temporary entities and relations which are specially added for handling individual items of experience (tokens). For example, the previous sentence contained three tokens of the word type *the*, each of which must be represented by a distinct node in the network; so anyone reading that sentence must add three temporary nodes to their network. The token nodes must belong to the same network as the permanent stored type because the aim of parsing is to establish an "is-a" relation between each word token and at least one word type. However, although tokens are part of the network, they are different from types because they are temporary, and as such they must be recognisable by WGNet++. (An obvious way to use the network to model learning would be to allow some token nodes to become permanent, but this idea would need a separate research project.)

Constants and variables are already distinguished in WGNet++. Variables all have names starting with "%" (e.g. "%fifOfIng", quoted earlier) in contrast with constants such as Noun, DOG (i.e. the lexeme DOG) and "word 3". This is merely a notational trick since in a network, names should strictly speaking not carry any information – all the information should lie in the relations. In handcrafted Word Grammar diagrams, variables are generally shown simply as a dot or question mark, because in these diagrams identity is shown entirely by relations rather than by labels. For example, Figure 6 shows that by default a word's base is the same as its "fif" (fully inflected form), and also that by default its referent "isa" (is an example of) its sense. We can safely use "?" for all these variables because they are distinguished by the relation arcs that point at them.

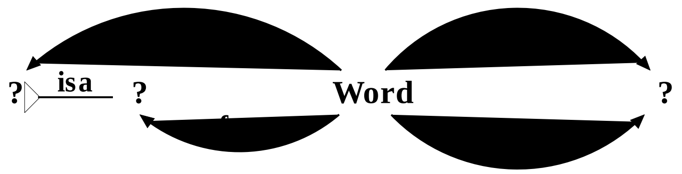


**Figure 6**

The type/token and constant/variable contrasts will play an important part in our processing algorithm, which is summarised in (4) . The term "activated" means "above its activity threshold", and a sister node is one which belongs to (isa) the same supercategory.

(4) If a token node is activated, apply the following procedures to it:

a. if it is a variable, **bind** it to its most active sister constant.
b. **inherit** values for all its activated relations.

Variable binding is important when processing tokens, because the inheritance process assigns them variables, which need to be bound to constants of various kinds. Indeed, the first task of processing any token of experience is to represent it as a distinct entity, along with a variable for the concept of which it is an example. The procedure for word tokens is sketched in Figure 7 :

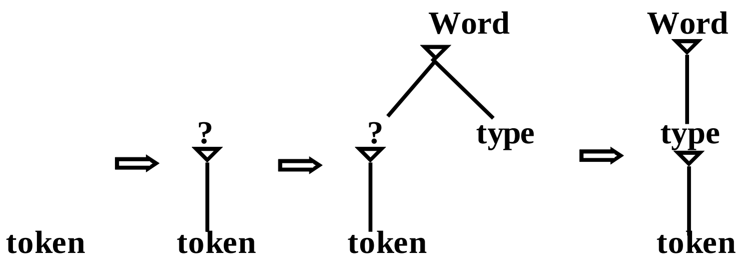


Figure 7

- As for any item of experience, build a new node for the token,
- then add a variable ("?") with an is-a link to the token.
- If it is recognisable as a word, add an is-a link from "?" to Word, making it a "sister" of every stored word-type.
- Meanwhile, whatever observable properties the token has (e.g. its spelling) will have activated the word types that have those properties; the final step is to bind "?" to the most active type (by merging the two nodes).

Here is a simple example from inflectional morphology, showing how variable binding selects DOG and Plural as the models for the word spelt <dogs>. The spelling activates the individual letters, but these will only pass the activity on to those words in which the letters occur in this order. (Order is shown in a primitive way by numbered "part" relations: "1", "2", and so on, supplemented by "first" and "last".) Figure 8 shows an early stage in this process leading to the final outcome, in which the word token is recognised as an example of both DOG and Plural. The more complex morphological structure is inherited from the latter.

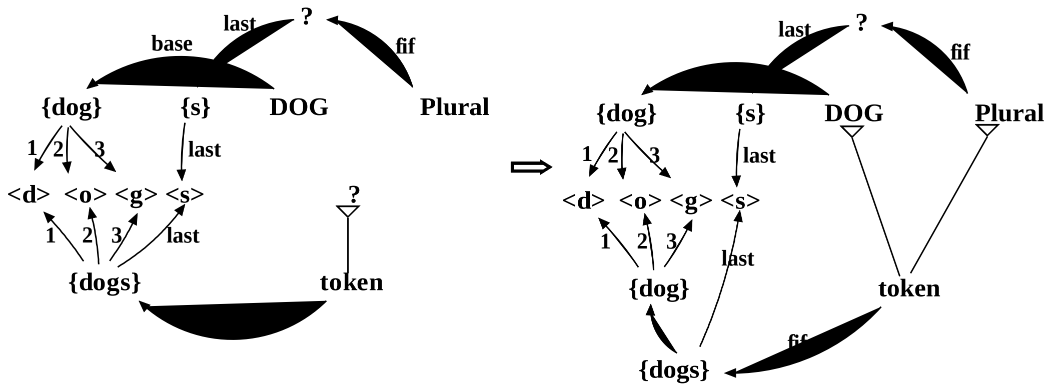


Figure 8

## Contextual disambiguation of single ambiguous words

In the previous example it was assumed that the selection procedure is driven entirely bottom-up by the spelling (or pronunciation), but this is obviously not how readers and hearers actually handle words. For one thing, many word-forms are ambiguous; for example, <dogs> could spell either the plural noun (as above) or the singular verb (as in "Trouble dogs all his efforts"). And for another, when words are ambiguous we choose interpretations which fit the expected meaning as well as the actual word-form. The more active the node for Dog (the idea of a typical dog) is, the more likely we are to select the noun interpretation. Indeed, it is possible for contextual expectations to outweigh pure form, so that (for example) we might take a word as an example of DOG even if it is mispronounced so that it sounds more like DUG.

Our hypothesis is that WGNet++ will be able to model the disambiguating effects of context because it will always favour the most active word type, regardless of whether this activity came from form or from context. In Figure 9 , the ovals show the sources for activation, and the prediction is that the activation from Dog is enough to make DOG more active than DUG in spite of the activation from the form. Of course it remains to be seen whether, and how, WGNet++ will automatically produce this outcome.

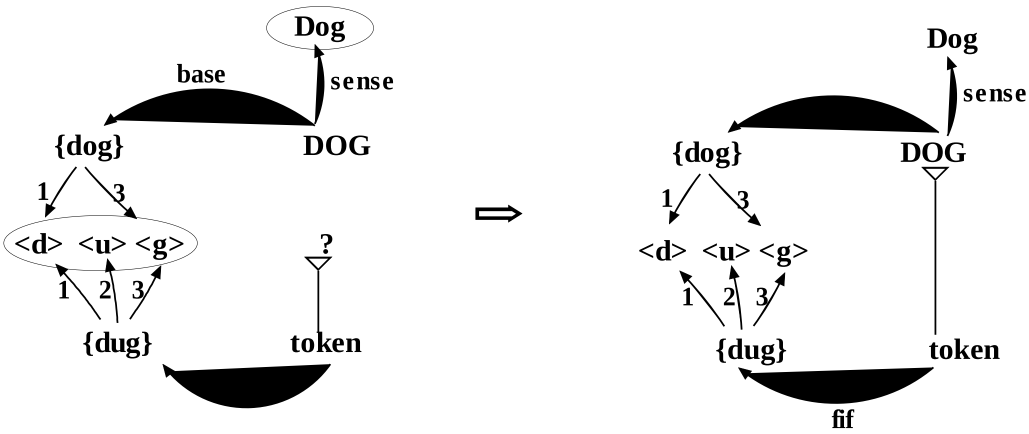


**Figure 9**

## Sentence analysis

Parsing by spreading activation is still our ultimate target, as it was in the original (rejected) proposal. The example given in that proposal was “John snores”. Can we use WGNet++ to add a subject link between *John* and *snores*? The example is, of course, quite trivial, and no challenge at all for a purpose-built parser; but the point about WGNet++ is that it is not a parser, but a general-purpose network model. Our hypothesis is that the processes outlined above will be sufficient for parsing not only easy sentences like this but also more demanding ones.

We believe that WGNet++ will be able to parse “John snores” by the following steps:

- Analyse *John* as an example of the noun JOHN, leaving the token in a high state of activity because it is still expecting a word on which it can depend.
- Add *snores* and analyse it as an example of SNORE and Singular. From the latter let it inherit a subject variable, which is-a noun.
- Variable binding looks for the most active noun, and, on selecting *John*, binds the subject variable to it.

The steps are presented in Figure 10 .

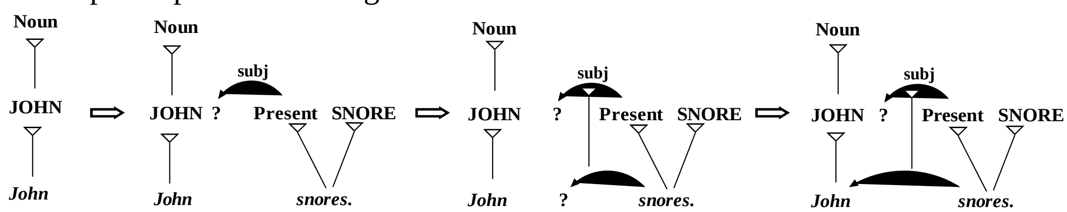

**Figure 10**

## Significance of the research

We believe this project opens new research ground in a number of ways, which we list below.

- It is the first attempt to apply a general-purpose reasoning model to the details of language structure.
- It is by no means the first general-purpose model, but it is unlike its predecessors (e.g. ACTR and SOAR) in various fundamental respects such as the subclassification of relations.
- It is the first attempt to develop a single processing model for all areas of language which (by hypothesis) will model production as well as perception.
- The algorithm in (4) integrates spreading activation with binding and default inheritance, thereby avoiding a number of well-known problems in computer systems. For example, default inheritance is generally regarded as a very costly mechanism because each inherited property has to be checked for possible overriders. When combined with spreading activation this problem disappears because the only nodes that need to be checked are the most active ones.

The project will allow the model to be evaluated, but its different strands will carry different weights.

- The software for WGNet++ itself is already a sophisticated tool, and by the end of the proposed project it will be ready for wider use.
- The selected formula for spreading activation may be more tentative, as it will need to be tested against a wide range of different databases and problems before we can claim it to be truly general.
- The databases for English vocabulary and grammar will necessarily be very incomplete even if we supplement the vocabulary as mentioned above. This is a serious problem for evaluating WGNet++ and the formula because of the unknown effects of scaling the database up to realistic proportions.