Semantic Approach to Knowledge Processing

MLADEN STANOJEVIĆ, SANJA VRANEŠ
The Mihailo Pupin Institute
Volgina 15, 11060 Belgrade
SERBIA
Mladen.Stanojevic@institutepupin.com http://www.institutepupin.com

Abstract: The processing of semantic information requires the adequate knowledge representation and ability to interpret semantically related knowledge. The majority of present day approaches to knowledge representation and processing are based on symbolic approach, i.e. on describing the meaning of represented domain knowledge and procedural knowledge used to process this domain knowledge. Hierarchical Semantic Form (HSF) implements the semantic approach to semantic knowledge representation and processing, which is not based on naming as a means to describe the meaning of the knowledge, but on semantic contexts that enable an implicit way to define the meaning of represented knowledge, and on simple and complex semantic categories used to interpret the semantics of represented knowledge. HSF facilitates the automatic translation of knowledge expressed in natural language into structured form and vice versa with no loss of information and its processing including natural language understanding, semantic search and question answering.

Key-Words: - Semantic Knowledge Processing, Knowledge Representation, Connectionist Model, Question Answering

1 Introduction

The predominant strategy in all existing knowledge processing techniques is based on the symbolic approach, which was introduced by the advent of first high-level programming languages. In this approach the information is represented using symbols, i.e. named variables and their values, and then processed by the computer program to provide the needed results.

Symbolic knowledge consists of independent symbols, which are then combined to produce the values of resulting symbols, while in semantic knowledge data are mutually related, where the processing of these data requires finding and searching the relationships between them. To support the representation of semantic knowledge, symbols become more complex by enabling the representation of internal structure (table-fields, class-attributes) and external structure (named relations). However, the essence of symbolic approach is still preserved, because names are used to explicitly define the meaning of represented knowledge.

The existing knowledge representation techniques, be they classical (e.g. relational [1] and object-oriented [2]) databases), AI techniques [3], [4] (e.g. logic formalism, semantic nets, conceptual dependencies, frames, scripts, rules, etc.), possibilistic approach based on Bayesian Networks [5], Semantic Web ontology and schema languages [6] (e.g. XOL [7], SHOE [8], OML [9], RDFS [10], DAML+OIL [11], OWL [12]), or distributed approach of the connectionist model, are all based on the symbolic approach. In these techniques, the meaning of the represented knowledge is described by a human expert through a designing process, whereby this process cannot be in any way automated. An example of application of symbolic approach to question answering can be found in [13].

In pure semantic approach to knowledge representation, names are not used to explicitly describe the meaning of represented knowledge, but semantic contexts [14]. These semantic contexts, which provide an implicit way to define the meaning of represented knowledge, are then interpreted using simple and complex semantic categories to determine their meaning.

One implementation of pure semantic approach to knowledge representation is represented by Hierarchical Temporal Memory (HTM) [15], while another solution [16] relies on the Hopfield-like neural networks.

Hierarchical Semantic Form (HSF) [14], [17], represents a modification of localist approach [18] and provides support for semantic approach to semantic knowledge representation and processing. The implementation of HSF described in [14] was a hybrid solution, where semantic approach was used in knowledge representation, while symbolic approach was used in knowledge processing [19]. In [17] a solution based on pure semantic approach has

been outlined and in this paper, some more information about applying semantic approach to knowledge processing will be given, by defining the states of HSF nodes and signals exchanged between nodes.

2 HSF Nodes

There are four types of HSF nodes [12]: specific groups, specific links, generic groups and generic links (Fig. 1).

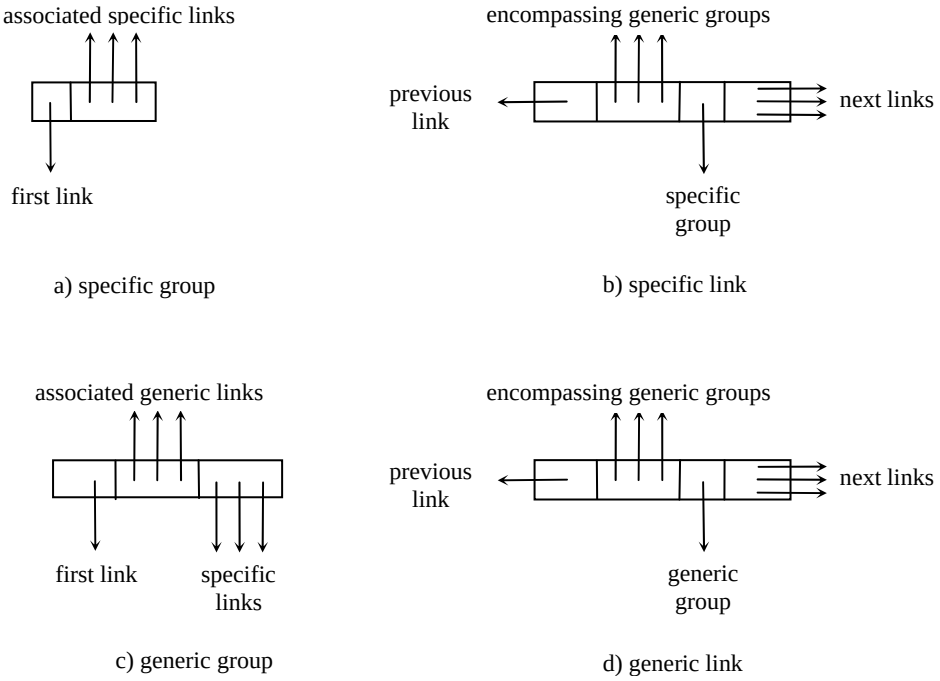

Fig. 1. HSF nodes

The specific group node designates characters, a group of characters, words, a group of words, sentences, etc. Except at the lowest level, where groups represent single characters, this data abstraction is used to represent sequences (semantic contexts) at different levels of hierarchy. One group can appear in different semantic contexts, so it can have many associated semantic links (for each context – one link).

The specific link node enables the creation of semantic contexts at different hierarchy levels (sequences of characters, words, group of words, sentences, etc.). The main role of specific links is to represent language categories (groups) in different semantic contexts.

The generic group node is used in HSF to represent a semantic category, whereby simple semantic categories generalize words, while complex semantic structures generalize complex language structures (phrases, statements, paragraphs, etc.) and corresponding semantic categories.

The generic link node enables the representation of semantic contexts comprised of semantic categories. Together with generic groups, they support the generalized structures corresponding to all language structures

If we would like to apply HSF to represent the following sentences:

John is a boy.
Mary is a girl.
John loves Mary.

we would have first to feed single words to HSF: “John”, “Mary”, “is”, “boy”, “girl”, “loves” (Fig. 2).

We can then feed the whole sentences to HSF and it will be modified correspondingly, by identifying and representing all semantic relationships between words (Fig. 3). It notices that the phrase “is a” occurring in the first statement is repeated in the second one, so it will create a new group representing this phrase. Each word and phrase “is a” is uniquely represented in HSF and for each statement they appear in, the corresponding link is created.

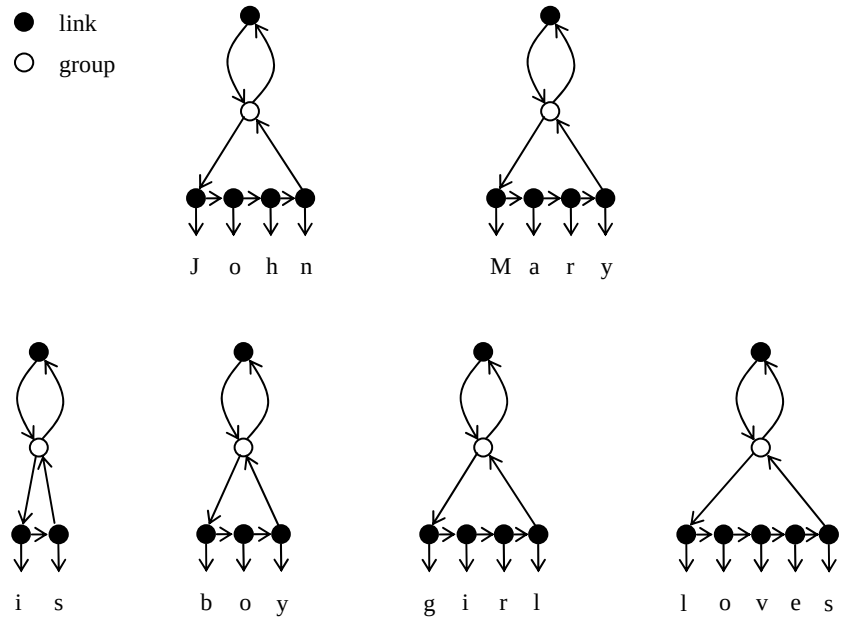

Fig. 2. Representation of single words in HSF

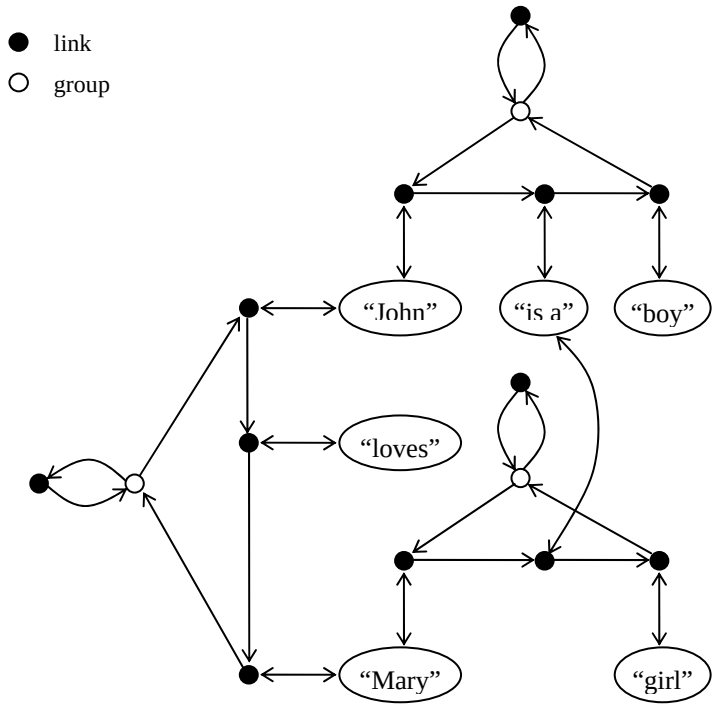

Fig. 3. Representation of statements in HSF

The main difference between HSF and declarative techniques is that no names are used (groups in Fig. 2 are not named) and no designing is needed to get the representation in HSF. HSF automatically creates the HSF structure from plain text. However, a computer is still not able to understand the meaning of knowledge represented by HSF.

3 States and Signals

In HSF each group and link can be in one of the four states (Table 1): inactive, semi-active, excited and active.

Table 1. Types of states

State	Indication
inactive	0
semi-active	½
excited	¾
active	1

There are also seven types of signals that can be exchanged between links and groups (Table 2): reset, relaxed, inhibitory, no signal, semi-active, excited and active.

Table 2. Types of signals

Signal	Indication
reset	-1
relaxed	-¾
inhibitory	-½
no signal	0
semi-active	½
excited	¾
active	1

A specific link can receive a signal from previous link, generic group and attached specific group and it can send a signal to generic group, specific group and next links (Fig. 4).

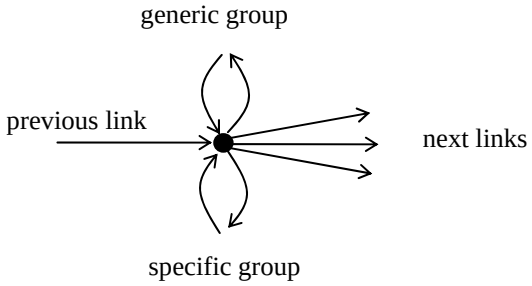

Fig. 4. Input and output signals for specific link

In Table 3 are given the combinations of input signals, state transitions and output signals for of input signals that cause the state transition of The last changed input signals are marked by '*'.

Table 3. Input signals, state transitions and output signals for specific links

Input Signals			State Transitions		Output signals		
Prev. Link	Spec. Group	Gen. Group	Prev. State	New State	Next Links	Spec. Group	Gen. Group
*½, 1	0, ½	0, ½	0	½	½	0	0
*½, 1	0, ½	1	½	0	-½	0	0
½, 1	0, ½	*1	½	0	-½	0	0
*½	1	1	1	1	1	0	0
0	*½	0, ½, 1	0	½	½	½	½
½, 1	*½	0, ½, 1	½	½	0	½	½
0, ½	*1	0, ½, 1	0, ½	1	½	1	1
1	*1	0, ½, 1	0, ½	1	1	1	1
*1	1	0, ½, 1	0, ½	1	1	0	0
*¾	½	½	½	½	0	¾	0
*¾	1	1	1	1	¾	0	0
½, ¾, 1	*¾	½, ¾	½	¾	¾	0	¾
½, 1	½	*¾	½	½	0	¾	0
*-½	0, ½, 1	0, ½, 1	½, 1	0	-½	0	0
¾	¾	*-¾	¾	¾	0	-¾	0
*-¾	¾	¾	¾	¾	0	-¾	0
*-¾	1	1	1	1	-¾	0	0
0, ¾	*-¾	-¾, ¾	¾	0	-¾	0	-¾
*-1	0, ½, 1	0, ½, 1	½, 1	0	-1	-1	-1
0, ½, 1	*-1	0, ½, 1	½, 1	0	-1	-1	-1
0, ½, 1	0, ½, 1	*-1	½, 1	0	-1	-1	-1

A generic link can receive a signal from previous link and attached specific group, and it can send a signal to specific group and next links (Fig. 5).

In Table 4 are given the combinations of input signals, state transitions and output signals for generic links.

A specific group can receive a signal from the attached specific links and from the last link, and can generate a signal for the first link and for the attached specific links (Fig. 6).

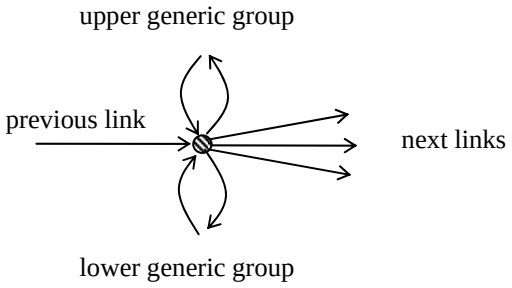

Fig. 5. Input and output signals for generic link

Table 4. Input signals, state transitions and output signals for generic links

Input Signals			State Transitions		Output signal		
Prev. Link	**L.G. Group**	**U.G. Group**	**Prev. State**	**New State**	**Next Links**	**L.G. Group**	**U.G. Group**
*½, 1	0, ½	0, ½	0	½	½	0	0
*½, 1	0, ½	1	½	0	-½	0	0
*½, 1	0	0, ½	0	½	½	0	0
*½, 1	0	1	½	0	-½	0	0
*1	½	½	½	½	0	¾	0
*½, 1	0, ½	1	½	0	-½	0	0
½, 1	0, ½	*1	½	0	-½	0	0
*½	1	1	1	1	1	0	0
0	*½	0, ½, 1	0	½	½	½	½
½, 1	*½	0, ½, 1	½	½	0	½	½
0, ½	*1	0, ½, 1	0, ½	1	½	1	1
1	*1	0, ½, 1	0, ½	1	1	1	1
*1	1	0, ½, 1	0, ½	1	1	0	0
*¾	½	½	½	½	0	¾	0
*¾	1	1	1	1	¾	0	0
½, ¾, 1	*¾	½, ¾	½	¾	¾	0	¾
½, 1	½	*¾	½	½	0	¾	0
*-½	0, ½, 1	0, ½, 1	½, 1	0	-½	0	0
¾	¾	*-¾	¾	¾	0	-¾	0
*-¾	¾	¾	¾	¾	0	-¾	0
*-¾	1	1	1	1	-¾	0	0
0, ¾	*-¾	-¾, ¾	¾	0	-¾	0	-¾
*-1	0, ½, 1	0, ½, 1	½, 1	0	-1	-1	-1
0, ½, 1	*-1	0, ½, 1	½, 1	0	-1	-1	-1
½, 1	0, ½, 1	*-1	½, 1	0	-1	-1	-1

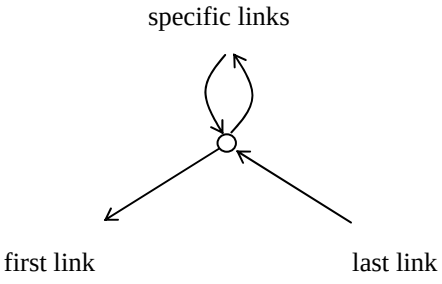

Fig. 6. Input and output signals for specific group

In Table 5 are given the combinations of input signals, state transitions and output signals for specific groups.

A generic group can receive a signal from the attached generic and specific links, and can generate a signal for the attached generic and specific links (Fig. 7). In Table 6 are given the combinations of input signals, state transitions and output signals for generic groups.

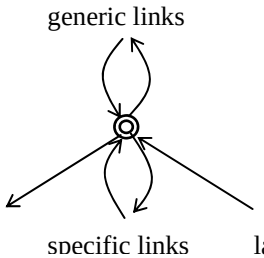

Fig. 7. Input and output signals for generic group

Table 5. Input signals, state transitions and output signals for specific groups

Input Signals		State Transitions		Output signal	
Spec. Link	**Last Link**	**Prev. State**	**New State**	**First Link**	**Spec. Links**
*¾	½	½	½	¾	0
¾	*¾	½	¾	0	¾
0, ½	*½	0	½	½	½
0, ½	*1	0, ½	1	1	1
½, 1	*-½	½, 1	0	-½	-½
*-¾	¾	¾	¾	-¾	0
-¾	*-¾	¾	0	0	-¾
*-1	½, 1	½, 1	0	-1	-1

Table 6. Input signals, state transitions and output signals for generic groups

Input Signals			State Transitions		Output signal		
Spec. Link	**Last Link**	**Gen. Link**	**Prev. State**	**New State**	**First Link**	**Gen. Links**	**Spec. Links**
*1	0, ½	0, ½	0, ½	1	1	1	1
0, ½	*1	0, ½	0, ½	1	0	1	1
*½	0, ½	0, ½	0	½	½	½	½
½	½	*¾	½	½	¾	0	¾
*¾	½, ¾	¾	½	¾	0	¾	0
¾	¾	*-¾	¾	¾	0	0	-¾
*-¾ (½)	¾	-¾	¾	½	½	-¾	0
*-¾ (0)	¾	-¾	¾	0	-¾	-¾	0
-1	0, ½, 1	0, ½, 1	½, 1	0	-1	-1	-1
0, ½, 1	-1	0, ½, 1	½, 1	0	-1	-1	-1
0, ½, 1	0, ½, 1	-1	½, 1	0	-1	-1	-1

Query answering represents the most complex kind of semantic processing, which includes natural language processing and semantic search. In HSF it is performed in four phases:

1. **Matching.** In this phase, words and phrases from the query are matched with the corresponding semantic categories. These semantic categories constitute the complex semantic category standing for the general form of the query, but also represent parts of general form of the answer. At the same time as the query is matched against the complex semantic category representing general query form, statements representing potential answers to this query are also identified. The states of the matched semantic category corresponding to general query form will be set to active and the states of statements representing potential answers will be set to semi-active.
2. **Excitation.** If the complex semantic category representing the general query form is matched against the query, the excitation phase will start. The complex semantic category representing the general answer form is only partially matched and the constituting semantic categories that are still not matched will, in this phase, be matched with words and phrases from the statements identified in the matching phase as potential answers. In the excitation phase the first answer to the query will be selected and the state of statements representing this answer will be set to excited.
3. **Relaxation.** In the relaxation phase statements in the excited state will be relaxed, i.e. their state will be set to inactive. During this phase, statements that are relaxed can be presented as an answer to the query. If there are some other answers, they will be identified in the repeated excitation phase. Excitation and relaxation phases will be repeated as many times as there are valid answer to the query.
4. **Resetting.** If there are no more valid answers, the states of all nodes in HSF will be set to inactive in the resetting phase.

4 Example

To illustrate the use of HSF in semantic knowledge processing, we will take a simple example. To enable HSF to understand a question “Who does John love?”, we would have to define the following semantic categories: <interrogative-pronoun>

("who"), <present-tense-do> ("does"), <emotional-relationship> ("love").

When we have defined these semantic categories, we can feed the question "Who does John love?" to HSF and the corresponding representation will be created (Fig. 8).

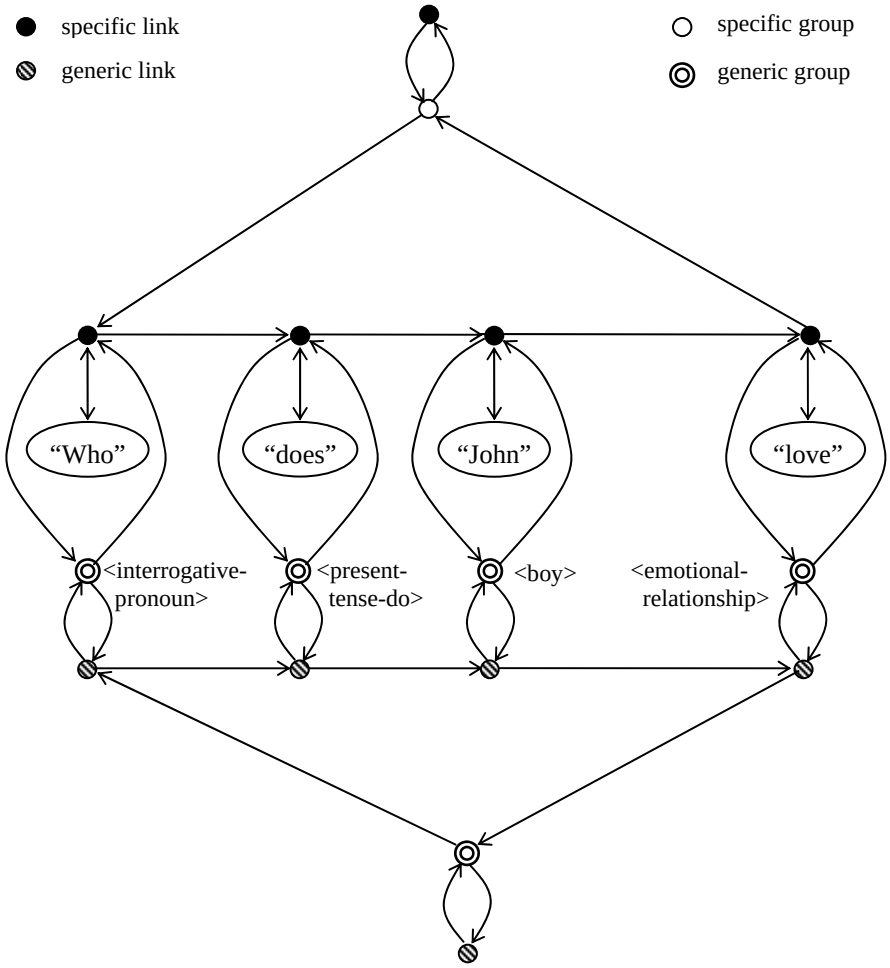

Fig. 8. Complex semantic category

In the process of understanding this query, HSF acts as a recurrent neural network. After the word "Who" is matched and the signals are propagated through the HSF, nodes and connections will be in the state as represented in Fig. 9. The generic group representing the complex semantic category will be in the semi-active state indicating that this group may potentially become active if the expected input is fed to HSF. The same holds for the specific group that represents the whole question.

After the whole question is fed, all nodes and connections of the HSF will become active. By matching all constituting semantic categories, the meaning of the complex semantic category representing the question is understood.

The understanding of complex semantic categories is dependent only on the constituting semantic categories and not on their order. This provides a great flexibility of understanding, because not only syntactically correct inputs can be recognized, but also the ones such as "John does love who?". Moreover the queries containing some unknown words can also be recognized (e.g. "Could you please tell me who John does love?".

The understanding of questions can be used to find the corresponding answers by propagating the signals through the rest of HSF, but due to the limited space this will be described in greater details in some other paper.

4 Advantages of HSF

The advantages of HSF in the representation of semantic knowledge are the following:

- **Representation of knowledge which is not domain specific**. HSF does not use any kind of naming or tagging for groups and links, so knowledge represented using HSF is not domain limited.
- **Automatic semantic knowledge acquisition from plain text**. Semantic knowledge represented in HSF is equivalent to its plain text

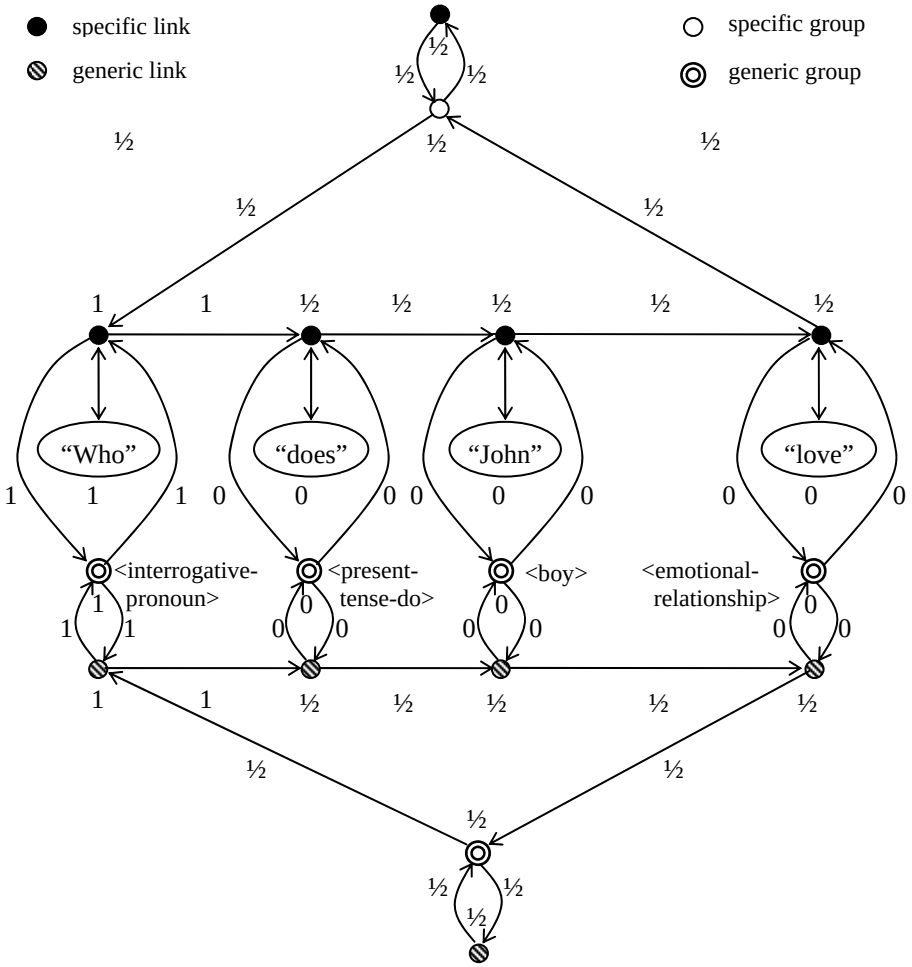

Fig. 9. State of HSF after partial matching

form. Knowledge in plain text form can be translated into Hierarchical Semantic Form and vice versa with no loss of information.

- **Learning of language structures and semantic relationships between these structures**. HSF has the ability to recognize and learn repeated sequences at various hierarchical levels (word, groups of words, statements), to identify all semantic relationships between them and to represent them using HSF.
- **Extending the existing HSF semantic knowledge repository is easy**. HSF can be easily extended by simply feeding new statements to HSF without any need for knowledge repository redesign. As new statements are fed to HSF, the existing HSF structures are reorganized, so that each structure is represented uniquely and that all semantic information is preserved.
- **Merging of existing semantic knowledge repositories in HSF can be performed automatically**. When two knowledge repositories in HSF need to be merged, one of them can be transformed into a plain text form and then statements can simply be added to the other one.

The advantages of using HSF in natural language understanding and query answering are the following:

- **Ability to learn to understand statements in natural language**. Instead of defining a fixed grammar for a subset of natural language, HSF is able to learn basic and complex semantic

categories defining the meaning of statements and questions.

- **Flexible understanding**. Since understanding in HSF is supported using semantic categories, a great flexibility of understanding is achieved, because it can process and understand statements that contain words in arbitrary order or unknown words, as well as statements that are syntactically incorrect.
- **Efficient information retrieval**. The efficiency of information retrieval using HSF is achieved by the hierarchical representation of knowledge in HSF and neural network capability for parallel processing of semantic categories appearing in question and in potential answers. This means that at the same time the question is processed, some potential answers to this question are also found.

When compared with a relational database model or ontology languages, the main advantage of HSF is that, unlike these descriptive semantic knowledge representation techniques, they enable knowledge representation that is not domain limited, which means that they do not require designing to describe the meaning of represented knowledge, nor programming to retrieve the needed information. Instead, designing is replaced with unsupervised learning used to organize the represented knowledge, while programming is replaced by supervised learning of basic and complex semantic categories used in question-answer forms.

5 Conclusion

Standard techniques for semantic knowledge representation and processing are based on symbolic approach where naming is used to explicitly define the meaning of represented knowledge. As a consequence, to describe and define domain specific knowledge a designing process is required, which makes the development of semantic processing applications quite expensive.

Another way to facilitate semantic knowledge representation and processing, proposed in this paper, is based on semantic approach. In this approach names are not used to explicitly define the meaning of represented knowledge. Instead, the meaning of the represented knowledge is implicitly defined by semantic contexts and interpreted using simple and complex semantic categories.

We have described briefly in the paper Hierarchical Semantic Form, a modification of the localist approach that can be used to implement semantic approach to semantic knowledge representation and processing. HSF supports four types of nodes: specific groups, specific links, generic groups and generic links to be able to represent specific natural language constructs and their corresponding generic forms expressed by simple and complex semantic categories.

Furthermore, we have introduced seven types of signals that can be exchanged between HSF nodes: reset, relaxed, inhibitory, no signal, semi-active, excited and active. These signals are used in HSF to enable complex semantic processing including natural language processing, semantic search and question answering. A simple example is given to illustrate the use of HSF in semantic knowledge processing.

References:

[1] C.J. Date, *Database in Depth: Relational Theory for Practitioners*, O'Reilly Media, Inc., Sebastopol, CA, 2005.

[2] C. Russell et al, *The Object Data Standard: ODMG 3.0.*, Morgan Kaufmann, San Francisco, CA, 2000.

[3] J. Sowa, *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole Publishing Co., Pacific Grove, CA, 2000.

[4] S. Vraneš, M. Stanojević, "Prolog/Rex - A Way to Extend Prolog for Better Knowledge Representation", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6 No. 1, 1994, pp. 22-37.

[5] Abdelkader Heni , Mohamed-Nazih Omri, Adel M. Alimi, "Fuzzy Knowledge Representation Based on Possibilistic and Necessary Bayesian Networks", *WSEAS Transaction on Information Science & Applications*, Vol. 3, No. 2, February 2006, pp. 224-231.

[6] D. Fensel, J.A. Hendler, H. Lieberman, W. Wahlster (Eds.), *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, MIT Press, Cambridge, MA, 2003.

[7] R. Karp et al: XOL: An XML-Based Ontology Exchange Language (version 0.4). www.ai.sri.com/pkarp/xol/. (February 26, 2007)

[8] Heflin, J. et al, *SHOE: A Knowledge Representation Language for Internet Applications*, Technical Report, CS-TR-4078 (UMIACS TR-99-71), Dept. of Computer Science, Univer-sity of Maryland, 1999.

[9] R. Kent, *Ontology Markup Language Version 0.3*. www.ontologos.org/OML/OML%200.3.htm. (April 30, 2008)

[10] D. Brickley, R.V. Guha (Eds.), *RDF Vocabulary Description Language 1.0: RDF Schema*, W3C Recommendation. www.w3.org/TR/rdf-schema/. (April 30, 2008)

[11] D. McGuinness, R. Fikes, J. Handler, L. Stein, *DAML+OIL: An Ontology Language for the Semantic Web*, *IEEE Intelligent Systems*, Vol. 17, No. 5, 2002, pp. 72-80.

[12] D. McGuinness, F. van Harmelen (Eds.), *OWL Web Ontology Language – Overview*, W3C Recommendation. www.w3.org/TR/owl-features/. (April 30, 2008)

[13] Sheng-Yuan Yang Chun-Liang Hsu Dong-Liang Lee Lawrence Y. Deng, "FAQ-master: An Ontological Multi-Agent System for Web FAQ Services", *WSEAS Transactions on Information Science and Applications*, Vol. 5, No. 1, 2008, pp. 221-228.

[14] M. Stanojević, S. Vraneš, "Applying Neural Networks to Knowledge Representation and Determination of its Meaning", F. Melle et al. (Eds.), BVAI 2007, *Lecture Notes in Computer Science* 4729, Springer, 2007, pp. 523-532.

[15] J. Hawkins, "Learning Lik1 A Human", IEEE Spectrum, April 2007, pp. 17-22.

[16] A. Kharlamov, V. Raevsky, "Networks Constructed of Neuroid Elements Capable of Temporal Summation of Signals", in J. Rajapakse, L. Wang (Eds.), *Neural Information Processing: Research and Development*, Springer, 2004, 56-76.

[17] M. Stanojević, S. Vraneš, "Knowledge Representation with SOUL", *Expert Systems With Applications*, Vol. 33, No. 1, 2007, pp. 122-134.

[18] G.E. Hinton, "Mapping Part-Whole Hierarchies into Connectionist Networks", *Artificial Intelligence*, Vol. 46, No. 1-2, 1990, pp. 47-75.

[19] Mladen Stanojević, Sanja Vraneš, „Applying Connectionist Model to Natural Language Understanding", *WSEAS Transactions on Information Science and Applications*, Vol. 3, No. 8, 2006, pp. 1501-1507.