On Attention Mechanisms for AGI Architectures: A Design Proposal

Helgi Páll Helgason[1], Eric Nivel[1] & Kristinn R. Thórisson[1,2]

[1]Center for Analysis and Design of Intelligent Agents / School of Computer Science, Reykjavik University, Menntavegur 1, 101 Reykjavik, Iceland
[2]Icelandic Institute for Intelligent Machines, Menntavegur 1, 101 Reykjavik, Iceland
helgih09@ru.is, eric@ru.is, thorisson@ru.is

Abstract. Many existing AGI architectures are based on the assumption of infinite computational resources, as researchers ignore the fact that real-world tasks have time limits, and managing these is a key part of the role of intelligence. In the domain of intelligent systems the management of system resources is typically called "attention". Attention mechanisms are necessary because all moderately complex environments are likely to be the source of vastly more information than could be processed in realtime by an intelligence's available cognitive resources. Even if sufficient resources were available, attention could help make better use of them. We argue that attentional mechanisms are not only nice to have, for AGI architectures they are an *absolute necessity*. We examine ideas and concepts from cognitive psychology for creating intelligent resource management mechanisms and how these can be applied to engineered systems. We present a design for a general attention mechanism intended for implementation in AGI architectures.

Keywords: artificial intelligence, attention, resource management, architecture, cognition, system design

1 Introduction

Most higher intelligences in nature have a built-in mechanism for deciding how to apply their brainpower from moment to moment. We call it *attention*, and by that we mean cognitive resource management of some type. As the real world is generally a source of much more information than any single intelligent agent could ever hope to cognitively ingest and process in any given period of time, even the smartest being of them all must come equipped with attentional mechanisms of some sort. Powerful methods for cognitive resource management are critical if we intend to create more capable AI systems than seen to date, systems capable of learning to solve novel tasks and adapting to unforeseen changes in environments of real-world complexity, while operating under time constraints – systems we refer to as artificial general intelligence (AGI) systems. Given the short shrift this subject has gotten in the AI literature, it can hardly be overemphasized that an AGI operating in the real world will have *limited* *resources at all times*. Ignoring how to design attention will only delay the day when AGI arrives on the scene. Natural attention is a cognitive function – or a set of them – that allow animals to focus their limited resources on relevant parts of the environment as they perform various tasks, while remaining reactive to unexpected events. Without it we could for example not stay alert to environmental events while finishing an important task, or manage multiple tasks at the same time. This cognitive function is not any less critical for AGI systems than it is for humans. In this paper we present a high-level design of an attention mechanism and discuss how prior work in cognitive psychology serves as a backdrop and inspiration. First we survey selected work on human attention from cognitive psychology and extract ideas we consider useful for implementing of attention in AGI systems. We review implementations of attention within some existing cognitive architectures and discuss their benefits and limitations. We then outline our attention mechanism designs, which is based on a holistic approach to attention, addressing data and process prioritization, and featuring simultaneous top-down and bottom-up control. The design makes few and fairly high-level requirements for the underlying architecture but is otherwise architecture-independent. The design proposal presented here is just that – a proposal for a design – but the basic principles on which it rests have already been proven in prior architecture implementations (Nivel 2007 & 2008, Thórisson 2009a & 2009b). Our work so far has not only resulted in the new attention mechanism presented here but also greatly affected the kinds of architectures we consider to be relevant to AGI research – architecture and attention are co-dependent. In that respect we discuss how the attention mechanism presented can be used for managing meta-cognitive operation and architectural self-growth, two fundamental functions of AGI systems (Thórisson & Helgason 2012).

2 Attention in Cognitive Psychology

The beginning of modern attention research is frequently associated with the work of Colin Cherry on the “cocktail party effect” (Cherry 1953), which examines how humans can focus on specific sensory data in the presence of distractions and background noise while still staying alert to relevant and/or important information that unexpectedly appears in the background. This ability implies simultaneous operation of a selective filter and deliberate steering mechanism which together perform allocation of cognitive resources. Deliberate, task-driven functionality is referred to here as *top-down attention,* reactive, stimulus-driven functionality as *bottom-up attention*. A number of psychological models for attention have been proposed that typically fall into one of two categories. **Early selection**: Selection of sensory information occurs early in the sensory pipe-line and is based on primitive physical features of the information (shallow processing) and little or no analysis of meaning. **Late selection**: Selection is performed after some level of non-trivial analysis of meaning at later stages of the sensory pipeline. The Broadbent filter model (Broadbent 1958) is one of the best known early-selection (filter) models. It assumes information filtering based on primitive physical features, with information that is not selected by the filter receiving no further processing. The Deutsch-Norman model (Norman 1969) is a well-known late-selection model. In contrast to the filter model, it proposes *gradual processing* of information to the point where memory representations are activated. Competitive selection is performed at the level of these representations, with the most active ones being selected for further processing. Some obvious problems are apparent for early selection models; they fail to account for commonly-observed human behavior such as noticing unexpected but relevant information – the cocktail party effect. The acoustic features alone of someone calling our name from the other side of a crowded room are not likely to be sufficient to attract our attention – some analysis of meaning must be involved. More recent models of attention focus on the interaction between top-down and bottom-up attention, such as the Knudsen attention framework (Knudsen 2007; see Figure 1). It consists of four interacting processes: *working memory, top-down sensitivity control*, *bottom-up filtering* and *competitive selection*. This framework seems to capture the major necessary parts for attention and be a promising starting point for AGI systems, from which some important issues for consideration can be extracted. **Systems that are expected to perform tasks while remaining reactive to unexpected events require both deliberate, top-down attention as well as reactive, bottom-up attention.** Top-down attention is responsible for ensuring that information relevant to current goals will receive processing. A system equipped with only this type of attention will frequently fail to notice (process) unexpected events that might be important for goals currently being pursued or necessary triggers for the generation of new ones. Bottom-up attention is responsible for detecting such events. This process is not (or less) influenced by current goals of the system, evaluating incoming information based on novelty, general relevance/familiarity to the perceiver, and unexpectedness. Systems implementing only bottom-up attention are unable to perform tasks beyond those that are simple and reactive, making tasks consisting of multiple steps (requiring some form of planning) problematic. **Managing the balance between top-down and bottom-up attention, in terms of resource allocation, is part of the role of attention.** Combining these two “types” (or roles) of attention can give rise to flexible, interruptible systems capable of performing complex tasks. Finding an acceptable balance in resource allocation between these processes or goals is a necessary function of attention. Over-assigning resources to top-down attention will introduce operational risk, as probability of missed important events is increased. Conversely, over-assignment of resources to bottom-up attention will adversely affect task performance, making it more time-consuming and difficult to accomplish goals. Balance between the two is difficult to specify in advance, as it depends on the environment and context of the system. This leads us to conclude that reaching and maintaining such a balance is a continuous and dynamic process that must be learned by the system from experience. **Late selection models provide a more reliable measure of importance of information than early selection models.** The shortcomings of early selection models were highlighted above. In the case of AGI systems, no assumptions can be made in advance with regards to the environment and system tasks; any incoming information is potentially important. While primitive physical features and signal characteristics may give rough clues to the importance of information, this information seems insufficient to guide informed resource management decisions. Operational risk may result when information is ignored without being related in any way to the operational experience (knowledge) of the system to determine meaning. For an example of why this may be problematic, consider subtle changes (in terms of basic information characteristics) in the environment that are precursors to important events – these are not likely to pass through classical early selection filters, potentially making the system unprepared to deal with critical scenarios. **Competitive selection is more desirable than filtering**. Viewing attention as a single-step process that decides whether information should be processed or not, is problematic in terms of resource utilization. Such decisions must be made in light of current availability of resources. It seems more reasonable to let attention evaluate the importance of incoming information, deferring processing decisions to actual execution time at which time resource availability is fully known and information competes for processing based on attention-steered priority evaluation.

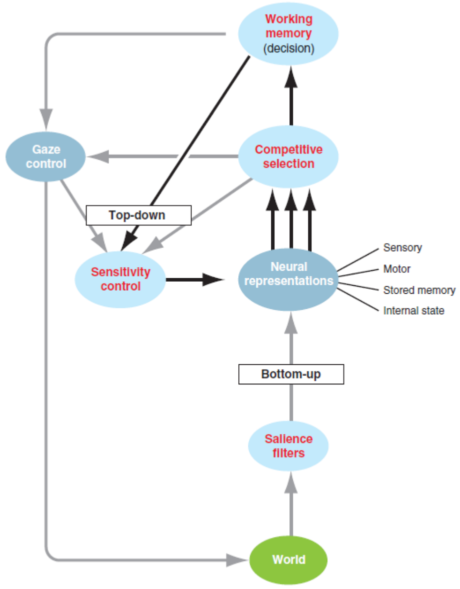

Fig. 1. The Knudsen attention framework (from Knudsen 2007). Information flows up from the environment and passes through saliency filters that detect important or unusual stimuli. Information that is passed through the filters then activates memory representations that encode knowledge. Memory representations are also activated by top-down sensitivity control, this process is influenced by the contents of working memory and adjusts activation thresholds of representations. Representations compete for access to working memory, with the most active ones being admitted.

3 Prior work

Some work has targeted attention in parts of AI systems, focusing on specific tasks and/or modalities (c.f. Schmidhuber 1991) and limited aspects of attention (c.f. Skubic 2004). A key difference between that work and ours is that we target attention in a complete sense, as needed for a whole cognitive architecture. Second, we exclusively target architectures that have a goal of being *general*, i.e. targeting artificial general intelligence (AGI). Third, implementability of both attention itself, and the architecture in which it operates, is a primary concern. Here we thus limit the discussion to the AGI domain. Only a handful of existing AGI architectures specifically implement some form of attention functionality, including NARS, LIDA and CLARION[1]. This chapter gives a brief overview of how these architectures implement attention and examines to what degree they satisfy some necessary requirements. NARS (Wang, 1995) is a cognitive architecture implemented as a general-purpose reasoning system, targeting operation in realtime with insufficient knowledge and resources. The system implements attention using a computational control strategy called controlled concurrency where task execution is controlled by two prioritization parameters: urgency and durability. The urgency parameter is the main priority parameter and decays over time in relation to the value of the durability parameter, which is used to specify if a task is long- or short-term. The result is dynamic resource management where tasks compete for execution based on their priority value. While priority of internally-generated goals is assigned by the system, original goals (provided by the developer) are assumed to have pre-assigned priority values. This delegates part of process prioritization – an integral role of attention – to an outside control mechanism, which is problematic with regards to achieving autonomy. LIDA (Baars, 2009) is another cognitive architecture based on a theory of human consciousness and targets intelligent, autonomous software agents. Attention is a core process of each operating cycle, consisting of three phases: sensing, attending, and action selection. During the attending phase, selection of data for further processing is performed by a collection of attentional codelets (small programs) which form coalitions of data that proceed to compete for system resources. LIDA thus implements both filtering and competitive selection for data. Attention is a learnable process in LIDA, allowing the system to improve its data-filtering over time. The attention functionality of LIDA does not take resource availability into account, making realtime operation somewhat problematic and potentially introducing resource utilization issues. Additionally, prioritization and selection is applied only to the data side without consideration of process prioritization. The CLARION cognitive architecture (Sun 2006) features a dedicated meta-cognitive subsystem (MCS) responsible for information selection, dynamic selection of learning methods for different situations, and modifying control parameters of other system modules. The MCS does not have integrated temporal management as required for realtime processing, and control processes are not affected by availability of resources at any given time, although attention can be said to be involved with process control via tuning of control parameters as mentioned earlier. **Data Selection.** The most widely accepted function of attention is selection of data for processing. The architectures address this in different ways. LIDA and CLARION implement information selection (filtering, and competitive selection in LIDA) in special phases of the sensory pipeline; NARS opts for a prioritization-based approach, where information is processed in decreasing order of urgency values as opposed to being filtered, resulting in a pure competitive selection control mechanism. None of these architectures address both top-down and bottom-up attention, focusing largely on the top-down side. **Control and Process Selection.** While attention is often viewed as an information filtering process, we argue that it must address process control as an

[1] See Thórisson & Helgason (2012) for a more general review of these architectures.

equally important aspect. The control of an AGI system is not limited to information selection – it must include selection of proper processes at any point in time, based on the context of the system, which includes time and resource constraints, in light of constraints imposed by tasks and context. In CLARION there is some overlap between attention and process control, but none of the three architectures take a fully integrated approach to data and process selection. **Realtime Processing.** For AGI systems, one of the core goals of attention must be to allocate resources in light of internal and external temporal constraints. This requires some form of temporal reasoning as well as consideration of resource availability, as tasks become increasingly urgent when their deadlines approach and ongoing tasks may interfere with access to resources. NARS does temporal reasoning using relative timings between events; the system can represent order of tasks and events and specify the temporal aspects of tasks using the durability parameter. Relative handling of time is clearly better than no temporal management, but reasoning with absolute timings allows for more fine-grained and precise control. NARS is implemented as a reasoning system and does not focus on perceptual nor action-related processes (inputs and outputs of the system are logical statements), emphasizing instead anytime performance. Integrated temporal reasoning is missing in both LIDA and CLARION and the availability of resources does not affect process control or data selection.

4 Attention Mechanism Design

We now present our design of an attention mechanism for AGI systems. The holistic, inclusive approach to attention we have taken includes top-down goal-derived control, bottom-up filtering and novelty interruption processes, and includes internal process control as part of the mechanism’s operation. While a general-purpose attention mechanism, applicable to any AI architecture, could be a goal to strive for, we do not believe this is possible, as resource management touches on too many fundamental issues in the structure and operation of an architecture to make it practically viable. Our proposed solution is only tractable if the following requirements are satisfied: ***Data-driven***: All processing occurs as a result of the occurrence of data; individual processes are executed only when paired with data that fits the input specification of the process. This eliminates the need for fixed control loops, allowing for operation on multiple time scales, greater flexibility, and above all, high operating efficiency. ***Fine-grained***: Processing and data units of the system are small and numerous (Thórisson & Nivel 2009a). Many such elements must collaborate to solve complex tasks. Reasoning about small, simple components and their effects on the overall system (e.g. in terms of resource usage) is more tractable than for larger, more complex components. ***Predictive capabilities***: Capacity to generate predictions with regards to future expectations must be supported. Predictions are necessary control data for (top-down) attention in addition to goals. ***Unified sensory pipeline***: Data originating from inside or outside the system is given equal treatment, allowing cognitive functions of the architecture – attention, in particular – to be applied equally to task performance and meta-cognitive processing (e.g. self-configuration). Systems satisfying these requirements will be built from small units of data and processing, with processes being executed when their input data specification is matched by an existing data item. Pattern matching is a practical method for determining matches as it allows each process some flexibility. New data are continuously created as the external environment is sampled by the system's sensors, triggering processes to run, resulting in either further data items being produced or in commands for the system's actuation devices, producing an action in the external environment, the effects of which are observed by the system via environmental sampling, closing the perception-action loop. For basic resource management, data and process need priority parameters; the main role of attention is, however it is implemented, to determine appropriate values for theses given the current operating situation. We refer to the priority parameters as *activation* in the case of processes, and *salience* in the case of data. System resources are managed to execute processing units with highest activation, on data units with the highest salience (no processing unit will execute without a compatible data unit). Processes can take the role of data, and data can describe processes. Adjusting activation and salience is the main role of attention; this is viewed as a *biasing* task. In our system, four parallel attention processes perform these tasks, as described below. The components in figure 2 that are involved in each process are indicated in parentheses. While this is probably not the only high-level system architecture that can meet the architectural requirements above, it explains well the operation of our attention mechanism. Note that the above architectural requirements are probably neither complete nor sufficient; for some AGI-acceptable attention mechanisms (unknown to us) they might not even all be necessary. That said, we have reason to believe that our proposed attention mechanism, and the requirements it rests on, represent a valid and useful step in the direction of more capable AGI systems. ***Top-down data biasing*** (*Attentional Patterns, Matching*): At some level, the goals and predictions of the system must be specified in operational terms, identifying particular states (inside or outside the system) that are desired (goals) or expected (predictions). Information contained in goals and predictions is used by this process to create attentional templates: Patterns that target data to varying levels of specification, from information related to a particular entity to all information coming from a single modality (e.g. auditory). For example, if the system has a goal of having object O1 in position P1, an attentional template is created that matches all information related to O1 (e.g. all data units referring to O1). This works identically for predictions. A unified sensory pipeline allows external and internal data to be targeted identically. When a data unit matches an attentional template, it receives a positive bias relative to priority of the goal that spawned the template. Data units that do not match any active attentional template will not receive bias from this top-down data biasing process. ***Bottom-up data biasing*** (*Bottom-up Attentional Processes, Evaluation*): Events that are novel and unexpected (in terms of prior experience or in a particular context), yet not directly related to task-driven goals, will almost certainly occur during operation. As top-down processes only target expected and goal-related data, such events are by their nature unlikely to be caught by it. The bottom-up process is responsible for determining a quantitative measure of novelty and unexpectedness for incoming data items, and providing saliency bias to them accordingly. The underlying idea is that novel data are likely to be useful in some way

– e.g. for learning or to detect events that threaten success of current goals. This process is not responsible for determining actual relevance of data, but rather to give these data units greater chance of receiving processing. Novelty and unexpectedness are *evaluated* based on the operating experience of the system, data or patterns of data that have occurred before receive lower bias than previously unseen data. To accomplish this task under tight temporal (and likely also memory) constraints, it is necessary to compress prior experience of the system in some way, preferably in data structures that allow for efficient look-up and comparison. Consequently, this process must constantly generate and update its control data based on incoming information in order for satisfactory evaluation of novelty and unexpectedness to occur. Habituation is an emergent operational property of this process, as novel or unexpected information will cease to be so automatically after having been observed on an increasing number of occasions.

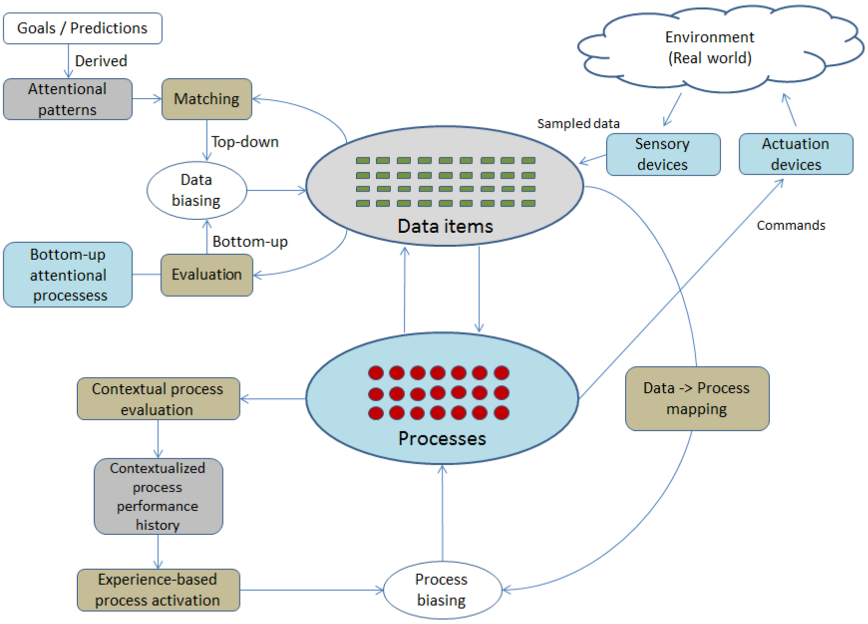

Fig. 2. Overview of the proposed attention mechanism.

Top-down process biasing (*Contextual Process Evaluation, Contextualized Process Performance History, Experience-based Process Activation*): While relationships between goals and processes are not obvious, these may be extracted from operational experience by tracking and maintaining history of the contribution of individual processes to the achievement of individual goals. While this is a non-trivial task, as many goals will be achieved by the collaboration of a number of processing units, it is nevertheless tractable using e.g. back-propagation from goal achievement through the operational chain which it resulted in (using some form of ampliative reasoning - c.f. Wang 1995). Furthermore, this process must have the capability to determine the similarity of goals, as goals are stated in precise operational terms and exactly identical goals are unlikely to occur multiple times. When a new goal is generated within the system, this process must search some compressed form of the operational history in order to find a sufficiently similar goal that has been previously achieved. The best such match (if one is found) results in positive biasing of processes that contributed to goal achievement on previous occasions. ***Bottom-up process biasing*** (*Data -> Process mapping*): To ensure processing of most salient data units (especially early on in the operation of the system, when top-down process biasing has insufficient experience to perform efficiently), this process works to assign positive bias to processes that are capable of processing the currently most salient data. The main purpose of this attentional function is practical, as efficient operation of the system may be highly problematic when no processing bias values are available due to the large number of processing units assumed to be present. The control of this process follows directly from the operation of top-down data biasing.

Although the design itself does not feature processes directly dedicated to realtime operation, it facilitates realtime operation as it is based on small processing units and can better make predictions (including temporal ones) about its own operation. The significance of small processing units with homogenous computational complexity is that most *processes take roughly the same amount of time to execute*, making temporal aspects of performance predictable, and that *the system is highly interruptible* and preemptive, never having to wait for time-consuming processes to complete before knowing how long it takes, or reacting to new data. Another important feature of the attentional design approach presented is that it can be applied directly to systems that manage their own growth and expansion – constructivist architectures (Thórisson 2009c). As the sum of internal system activity is likely to constitute a large amount of information, it is desirable that the attention mechanism be used to manage resources for self-reconfiguration – in much the same way as it is used for other task performance. The mechanism presented here already assumes a unified sensory pipeline: attention operates identically on environmental data and internal data. By generating internal goals supporting directed self-reconfiguration of the system and targeting internal states, AGI systems can be envisioned that simultaneously perform tasks in complex environments and manage their own growth, while operating under realtime constraints with limited resources.

5 Conclusions

Surprisingly little work focusing on attention has been performed in the field of AI, although we have seen that existing attention models from cognitive psychology can be mapped to AGI architectures in a useful way. In our work to design an AGI-ready attention mechanism we have found a large overlap between the functionalities of attention and the control mechanisms of the underlying architecture. This is an indication that retrofitting an existing architecture with the resource management capabilities stated will be highly problematic; on close examination attention reveals itself as a ubiquitous function of a cognitive architecture, influencing operation and structure across all levels. So, while this work had the goal of designing an attention mechanism, the result is also a near-complete control mechanism for cognitive architectures.

6 Acknowledgements

This work has been supported in part by the EU-funded project HUMANOBS: Humanoids That Learn Socio-Communicative Skills Through Observation, contract no. FP7-STREP-231453 (www.humanobs.org), and by grants from Rannis, Iceland.

7 References

1. Baars, B. J., Franklin, S. 2009. Consciousness is computational: The LIDA model of Global Workspace Theory. Intl. Journal of Machine Consciousness, 2009, 1(1): p. 23-32.
2. Broadbent, D. E. 1958. Perception and Communication. London: Pergamon.
3. Cherry, E. C. 1953. Some experiments on the recognition of speech, with one and two ears. *Journal of the Acoustical Society of America*, p. 975-979.
4. Nivel, E. 2007. Ikon Flux 2.0. Reykjavik University technical report, available online at http://www.ru.is/media/skjol-td/RUTR-CS07006.pdf
5. Nivel, E., Thórisson, K. R. 2008. Self-Programming: Operationalizing Autonomy. In *Proceedings of the 2nd Conf. on Artificial General Intelligence.*
6. Norman, D. A. 1969. Memory while shadowing. Quarterly Journal of Experimental Psychology, vol. 21, pages 85-93.
7. Schmidhuber, J., Huber R. Learning to generate artificial fovea trajectories for target detection. International Journal of Neural Systems, 2(1 & 2):135-141, 1991.
8. Skubic, M., Noelle, D., Wilkes, M., Kawamura, K., Keller, J.M. 2004. A biologically inspired adaptive working memory for robots. AAAI Fall Symp., Workshop on the Intersection of Cognitive Science and Robotics. Washington D.C. 2004.
9. Sun, R. 2006. The CLARION cognitive architecture: Extending cognitive modelling to social simulation. In: Ron Sun (ed.), Cognition and Multi-Agent Interaction. Cambridge University Press, New York.
10. Thórisson, K. R., Nivel, E. 2009a. Achieving artificial general intelligence through peewee granularity. In *Proc. of the 2nd Conf. on Artificial General Intelligence,* pages 220-221.
11. Thórisson, K. R., Nivel, E. 2009b. Holistic intelligence: Transversal skills and current methodologies. *Proc. of the 2nd Conf. on Artificial General Intelligence,* pages 222-223.
12. Thórisson, K. R. 2009c. From Constructionist to Constructivist A.I. Keynote, Technical Report, FS-90-01, AAAI press, Menlo Park, California.
13. Thórisson, K. R., Helgason, H. P. 2012. Cognitive Architectures and Autonomy: A Comparative review. Journal of Artificial General Intelligence, vol. 3, pages 1-30.
14. Wang, P. 1995. Non-Axiomatic Reasoning System: Exploring the Essence of Intelligence. Ph.D. diss., Dept. of Computer Science, Indiana Univ., CITY, Indiana.