On the Working Definition of Intelligence

Pei Wang
Center for Research on Concepts and Cognition, Indiana University
510 North Fess, Bloomington, IN 47408, USA
pwang@cogsci.indiana.edu

Abstract

This paper is about the philosophical and methodological foundation of artificial intelligence (AI). After discussing what is a good "working definition", "intelligence" is defined as "the ability for an information processing system to adapt to its environment with insufficient knowledge and resources". Applying the definition to a reasoning system, we get the major components of *Non-Axiomatic Reasoning System (NARS)*, which is a symbolic logic implemented in a computer system, and has many interesting properties that are closely related to intelligence. The definition also clarifies the difference and relationship between AI and other disciplines, such as computer science. Finally, the definition is compared with other popular definitions of intelligence, and its advantages are argued.

1 To Define Intelligence

1.1 Retrospect

The attempts of clarifying the concept "intelligence" and discussing the possibility and paths to produce it in computing machinery can be backtracked to Turing's famous article in 1950, in which he suggested an imitation test as a sufficient condition of being intelligent [38].

The debate on this issue has been going on for decades, and there is still little sign of consensus [15]. As a matter of fact, almost every one in the field has his/her own ideas about how the word "intelligence" should be used, and these ideas in turn influence the choice of research goals and methods, as well as serve as standards to judge other researchers' works.

Following are some representative opinions:

> "Intelligence is the power to rapidly find an adequate solution in what appears *a priori* (to observers) to be an immense search space." (Lenat and Feigenbaum, [17])

> "Artificial intelligence is the study of complex information-processing problems that often have their roots in some aspect of biological information-processing. The goal of the subject is to identify interesting and solvable information-processing problems, and solve them." (Marr, [18])
>
> "AI is concerned with methods of achieving goals in situations in which the information available has a certain complex character. The methods that have to be used are related to the problem presented by the situation and are similar whether the problem solver is human, a Martian, or a computer program." (McCarthy, [19])
>
> Intelligence usually means "the ability to solve hard problems". (Minsky, [22])
>
> "By 'general intelligent action' we wish to indicate the same scope of intelligence as we see in human action: that in any real situation behavior appropriate to the ends of the system and adaptive to the demands of the environment can occur, within some limits of speed and complexity." (Newell and Simon, [24])
>
> "Intelligence means getting better over time." (Schank, [32])

Here we do perceive something in common among the statements, however, their difference is equally obvious.

1.2 Do we need a definition?

Maybe it is too early to define intelligence. It is obvious that, after the decades of study, we still do not know very much about it. There are more questions than answers. Any definition based on the current knowledge is doomed to be revised by future works. We all know that a well-founded definition is usually the result, rather than the starting point, of scientific research.

However, there are still reasons for us to be concerned about the definition of intelligence at this time. Though clarifying the meaning of a concept always helps communication, this problem is especially important for AI.

As a community, AI researchers need to justify their field as a scientific discipline. Without a (relatively) clear definition of intelligence, it is hard to say why AI is different from, for instance, computer science or psychology. Is there really something novel and special, or just fancy labels on old stuff?

More vitally, every researcher in the field needs to justify his/her research paradigm according to such a definition. For a concept as complex as "intelligence", no direct study is possible, especially when an accurate and rigid tool, namely the computer, is used as the research medium. We have to specify our aim clearly, then try to solve it. Therefore, anyone who wants to work on artificial intelligence is facing a two-phase task: choosing a working definition of intelligence, and producing it on the computer.

A *working definition* is a definition that is concrete enough that you can directly work with it. By accepting a working definition of intelligence, it does not mean that you really believe it fully captures the concept "intelligence", but that you will take it as a goal for

your current research project. Such a definition is not for an AI journal editor who needs a definition to decide what papers are within the field, or a speaker of the AI community who needs a definition to explain to the public what is going on within the domain.

Therefore, the lack of a consensus on what intelligence is does not prevent each researcher from picking up (consciously or not) a working definition of intelligence. Actually, unless you keep one (or more than one) definition, you cannot claim that you are working on artificial intelligence. It is your working definition of intelligence that relates your current research, no matter how domain-specific, to the AI enterprise.

The paper is about such a working definition of intelligence (for an individual researcher), rather than other types of definition (for a teacher, a reviewer, or a grant agency).

1.3 Are there better definitions?

A group of people wants to climb a mountain. They do not have a map, and the peak is often covered by clouds. At the foot of the mountain, there are several paths leading into different directions. When you join the group, some of the paths have been explored for a while, but no one has reached the top.

If you want to get to the peak as soon as possible, what should you do? You cannot sit at the foot of the mountain until you are absolutely sure which path is the shortest — you have to explore. On the other hand, taking an arbitrary path is also a bad idea. Though it is possible that you make the right choice from the beginning, it definitely would be advisable to use your knowledge about mountains, and also to study other people's reports about their explorations, so as to avoid a bad choice in advance.

There are three kinds of "wrong paths": those which lead nowhere, those which lead to interesting places (even to unexpected treasures) but not to the peak, and those which eventually lead to the peak but are much longer than some other paths. If the only goal is to climb the peak as early as possible, a climber should use all available knowledge to choose a better path to explore. Although switching to another path is always possible, it is time-consuming.

We are facing a similar situation in choosing a working definition for intelligence. There are already many such definitions, which are quite different, though still related to each other (so hopefully we are climbing the same mountain). As a scientific community, it is important that competing paradigms are followed at the same time, but it does not mean that all of them are equally justified, or will be equally fruitful.

By accepting a working definition of intelligence, the most important commitments a researcher makes are on the acceptable assumptions and desired results, which bind all the concrete works that follow. The defects in the definition can hardly be compensated by the research, and improper definitions will make the research more difficult than necessary, or lead the study away from the original goal.

1.4 What are the criteria of a good definition?

Before studying concrete working definitions of intelligence, we need to set up a general standard for what makes a definition better than others.

Carnap met the same problem when he tried to clarify the concept "probability". The task "consists in transforming a given more or less inexact concept into an exact one or, rather, in replacing the first by the second", where the first may belong to everyday language or to a previous stage in the scientific language, and the second must be given by explicit rules for its use [3].

According to Carnap [3], the second concept, or the *working definition* as it is called in this paper, must fulfill the following requirements:

1. It is *similar* to the concept to be defined, as the latter's vagueness permits.
2. It is defined in an *exact* form.
3. It is *fruitful* in the study.
4. It is *simple*, as the other requirements permit.

If we agree that these requirements are reasonable and suitable for our purpose, let us see what they mean concretely to the working definition of intelligence:

Similarity. Though *intelligence* has no exact meaning in everyday language, it does have some common usages which the working definition should follow. For instance, normal human beings are intelligent, but most animals and machines (including ordinary computer systems) are either not intelligent at all or much less intelligent than human beings.

Exactness. According to the working definition, whether (or how much) a system is intelligent should be clearly decidable. For this reason, *intelligence* cannot be defined in terms of other ill-defined concepts, such as *mind, thinking, cognition, intentionality, rationality, wisdom, consciousness*, though these concepts do have close relationships with intelligence.

Fruitfulness. The working definition should provide concrete instructions for the research based on it, for instances, on what assumptions can be accepted, what phenomena can be ignored, what properties are desired, and so on. Especially, the working definition of intelligence should contribute to the solving of the fundamental problems in AI.

Simplicity. Although intelligence is surely a complex phenomena, the working definition should be simple. For a theoretical reason, a simple definition makes it possible to explore a paradigm in detail; for a practical reason, a simple definition is easy to use.

For our current purpose, there is no "right" or "wrong" definition for intelligence, but there are "better" and "not-so-good" ones. When comparing proposed definitions, the four requirements may conflict with each other. For example, one definition is more fruitful, while another is simpler. In such a situation, some weighting and trade-off is needed. However, there is no evidence that shows the requirements cannot be satisfied at the same time.

2 A Working Definition of Intelligence

2.1 The definition

Following the preparation in the previous section, we propose here a working definition of intelligence:

> *Intelligence is the ability for an information processing system to adapt to its environment with insufficient knowledge and resources.*

An *information processing system* is a system whose internal activities and interactions with its environment can be studied *abstractly*, that is, without mentioning the physical events that carry out the activities and interactions. Usually, such a system has certain *tasks* (assigned by the environment, or generated by the system itself) to fulfill. To do that, the system takes some *actions*, guided by its *knowledge* about how the actions and tasks are related. All the internal activities cost the system some *resources*. The interaction between the system and its environment can be described by its *experience* and *response*, which are streams of input and output, respectively. An input may be a task or a piece of new knowledge, and an output is usually a result of a task. The valid patterns of input and output consist of the *interface language* of the system. According to this description, all human beings and computer systems, as well as many animals and automatic control systems, can be referred to as information processing systems.

To *adapt* means the system learns from its experiences. It fulfills tasks and adjusts its internal structure to improve its resource efficiency under the assumption that future situations will be similar to past situations. Not all information processing systems adapt to its environment. For instance, a traditional computing system gets all of its knowledge during its designing phase (or before its "birth"). After that, its experience contains tasks only, and the results do not depend on the experience of the system. To acquire new knowledge, the system needs to be redesigned which is not done by communicating in its interface language. On the other hand, not all experience-related changes can be called "adaption". Briefly, the change should make the system work *better*, if the environment is relatively stable.

Insufficient knowledge and resources means the system works with respect to the following restrictions:

Finite. The system has a constant information processing capacity,

Real-time. All tasks have time requirements attached, and

Open. No constraints are put on the knowledge and tasks that the system can accept, as long as representable in the interface language.

There are two components in the working definition: *adaption* and *insufficient knowledge and resources*. They are related. An adaptive system must admit some insufficiency in its knowledge and resources, otherwise it need not change. On the other hand, without adaption, a system may admit that its knowledge and resources are insufficient, but make no attempt

to improve the situation. Actually, such an admission makes no practical difference from a claim that the knowledge and resources are already sufficient.

Not all information processing systems completely take the insufficiency of knowledge and resources into consideration. Non-adaptive systems simply inhibit or ignore new knowledge in its interaction with its environment, and most artificial adaptive systems are not finite, real-time, and open:

1. Though all implemented systems are finite, many theoretical models neglect the possibility that the requirements for processors and memory space may go beyond the supply capacity of the system.
2. Most current AI systems do not consider time constraints at run-time. Most "real-time" systems only process time constraints in the form of deadline [37].
3. Various constraints are imposed on what the system can experience. For example, only questions that can be answered by retrieval and deduction from current knowledge are acceptable, new knowledge cannot conflict with previous knowledge, and so on.

Many computer systems are designed under the assumption that their knowledge and resources, though *limited* or *bounded*, are still *sufficient* to fulfill the tasks that they need to handle. When facing a situation where this assumption fails, such a system simply panics, and asks for external interfere.

To design a system under the *Assumption of Insufficient Knowledge and Resources* (hereforth *AIKR*), it does not mean that the knowledge and resources of the system are *always insufficient* for all tasks, but that the knowledge and resources are *not always sufficient* for all tasks. To work with respect to the assumption, a system should have mechanisms to handle the following situations:

A new processor is required when all of them are occupied;
A piece of memory is required when the working space is already full;
A task comes up when the system is busy with something else;
A task comes up with a time requirement, so an exhaustive search is not affordable;
New knowledge conflicts with previous knowledge;
A question is presented for which no sure answer can be deduced from available knowledge;
... .

For traditional computing systems, these situations usually cause external interferes and rejections of the task or knowledge involved. However, for a system designed under AIKR, these are *normal situations*, and should be managed smoothly.

According to above definition, intelligence is a strong form of adaption. This assertion is consistent with the usages of the two words in natural language: we are willing to call many animals, computer systems, and automatic control systems "adaptive" but not "intelligent".

2.2 An intelligent reasoning system

To make our discussion more concrete and fruitful, let us apply the above working definition of intelligence to a special type of information processing system — reasoning system.

A *reasoning system*, in a broad sense, is an information processing system that has the following components:

1. *a formal language*, defined by a grammar, for the communication between the system and its environment and the internal representation of the system;
2. *a semantics* of the formal language that explains the meaning of the words and the truth value of the sentences in the language;
3. *a set of inference rules* that is defined formally, and can be used to match questions with knowledge, to infer conclusions from promises, to derive subquestions from questions, and so on;
4. *a memory* that systematically stores the questions and knowledge, and to provide a work place for inferences.
5. *a control mechanism* that is responsible for resources management, such as to choose premises and inference rules for each step of inference, and to process space requirements.

The first three components are usually referred to as a *logic*.

Being reasoning system is neither necessary nor sufficient for being intelligent, but we can see that an *intelligent reasoning system* provides a suitable object for the study of intelligence.

Before seeing how such an intelligent reasoning system can be designed, let us first see its opposite: a reasoning system designed under the assumption that its knowledge and resources are sufficient to answer the questions asked by its environment (so no adaption is needed). By definition such a system has the following properties:

1. No new knowledge is necessary. All the system needs to know to answer the questions is already there at the very beginning, represented by a set of axioms and postulates.
2. The axioms and postulates are *true*, and will remain true, in the sense that they correspond to the actual situation of the environment.
3. The system answers questions by applying a set of formal rules on the axioms and postulates. The rules are sound and complete (with respect to the valid questions), so they guarantee correct answers for all questions.
4. The memory of the system is so big that all axioms, postulates, and intermediate results can be put into it.
5. There is an algorithm that can carry out any required inference in finite time, and it runs so fast that it can satisfy all time requirements that may be attached to the questions.

This is a system dreamed by Leibniz, Boole, Hilbert, and many others. It is usually referred to as "decidable axiomatic system" or "formal system". The attempt to build such systems has dominated the study of logic for a century and strongly influenced the research of artificial intelligence. Many researchers believe that such a system can serve as a model of human thinking.

However, if intelligence is defined as "to adapt under AIKR", what we want is the *contrary*, in some sense, to an axiomatic system, though it is still *formalized* or *symbolized* in a technical sense. That is why *Non-Axiomatic Reasoning System*, NARS for short, is chosen to name an intelligent reasoning system to be introduced in the following section.

3 The Components of NARS

Non-Axiomatic Reasoning System, or *NARS*, is designed to be an intelligent reasoning system, according to the working definition of intelligence described previously.

In the following, let us see how the major components of NARS (formal language, semantics, inference rules, memory, and control mechanism) are determined, or strongly suggested, by the definition, and how they differ from the components of an axiomatic system. Because this paper is concentrated in the philosophical and methodological foundation of the NARS project, formal descriptions and detailed discussions for the components are left to other papers (such as [40, 42]).

3.1 Experience-grounded semantics

The traditional model-theoretic semantics is no longer applicable to NARS. Due to AIKR, no knowledge in NARS is "true" in the sense that it corresponds to "state of affairs" in the real world. Knowledge comes, directly or indirectly, from the experience of the system, and it is always revisable by future experience. Therefore, the relationship of the expressions in the language and the environment (that is what semantics is about) is revealed by how the expressions ground on the experience. A model and an interpretation is no longer needed.

In NARS, the *truthfulness* of a statement is judged according to its relationship with the experience, rather than according to its relationship with a model. Similarly, the *meaning* of a term, that is, what makes the term different from other terms to the system, is determined by its relationships to other terms, according to the experience, rather than by an interpretation that mapping it into an object in a model. The new semantics is discussed in more detail, and applied to a formal language, in [40].

The basic differences between experience-grounded semantics and model-theoretic semantics are:

1. As descriptions of an environment, the former is partial, developing in time, and not conflict-free, whereas the latter is complete, static, and consistent.
2. The former is accessible to the system itself, whereas the latter is supplied by an observer, so usually unknown to the system.

3. With insufficient resources, the truth-value of each statement and the meaning of each term in NARS is usually grounded on part of the experience. As a result, even without new experience, the inference activity of the system will change the truth-values and meanings, by taking previously available-but-ignored experience into consideration. In contrary, according to model-theoretic semantics, the internal activities of a system have no effects on truth value and meaning of the language it uses.

"Without an interpretation, a system has no access to the semantics of a formal language it uses" is the central argument in Searle's "Chinese room" thought experiment against strong AI [33]. His argument is valid for model-theoretic semantics, but not for experience-grounded semantics. For an intelligent reasoning system, the latter is more appropriate.[1]

3.2 Uncertainty measurement

As mentioned above, in NARS the truth value of a statement indicates its relationship with the experience of the system, so always revisable in light of new knowledge. When answering a given question, there is no "correct" result in the absolute sense, but there are "better" results in a relative sense.

To have a general, domain-independent method to compare competing answers, a numerical truth value, or a *measurement of uncertainty*, becomes necessary for NARS, which quantitatively records the relationship between a statement and available experience.

In its simplest form, the relationship can be measured by the *weight* of positive and negative evidence of a statement (according to available experience). This measurement of uncertainty and its variations are discussed in detail in [40].

What makes this measurement different from other proposed measurements of uncertainty is that it compares a statement with the experience of the system rather than with a model. As a result, the evaluation is changeable and system-dependent.

The weight of evidence is defined in such a way that it can be used to indicate *randomness* (see [39] for a comparison with Bayesian network [26]), *fuzziness* (see [41] for a comparison with fuzzy logic [44]), and *ignorance* (see [43] for a comparison with Dempster-Shafer theory [34]). Though different types of uncertainty have different origins, they usually co-exist, and are tangled with one another in practical situations. Since NARS makes no restrictions on what can happen in its experience, and needs to make justifiable decisions when the available knowledge is insufficient, such a unified measurement is necessary.

3.3 Term-oriented language

To support the above *weight of evidence*, we need (1) to determine what is positive and negative evidence for a given statement, and (2) to find a natural unit for the measurement.

This problem is hard in *first order language*, the dominating formal language in AI. Used in first order predicate logic, this language is designed for the study of "foundations of

[1] A more complete discussion on this issue is left for a future paper.

mathematics", and thus intrinsically related to binary deductive inference. We all know what a *proof* or a *disproof* of a statement is in first order predicate logic, however, as revealed by Hempel's famous "confirmation paradox" [9], to introduce the concepts of "positive evidence" and "negative evidence" into first order language is hard, if not impossible.

Fortunately, we have an alternative language family: the formal language used by term logics. Let us call it *term-oriented languages*.

Term-oriented language, as used in Aristotle's logic, is characterized by the use of subject-predicate sentences. Though widely believed to be "too restricted" and outstripped as a language for mathematical logic, term-oriented language is more suitable for an intelligent reasoning system, as shown by the practice of NARS.

NARS uses a formally defined language (see [40] for a simple version), in which each sentence has the form "$S \subset P$", where S is the subject term of the sentence, and P is the predicate term of the sentence. The copula "$\subset$" is the extension of a reflexive and transitive (binary) relation "$\sqsubset$" between two terms, and can be intuitively understood as "are".

As a multi-valued extension of "$\sqsubset$", "$\subset$" is transitive to a certain degree, which is measured by the truth value. For a term T, ideally there are three possibilities (a more general and formal description is in [40]):

1. If either both "$T \sqsubset S$" and "$T \sqsubset P$" are true, or both "$P \sqsubset T$" and "$S \sqsubset T$" are true (according to the experience of the system), T is counted as a piece of *positive evidence* for "$S \subset P$", because T confirms the transitive relation.
2. If either "$T \sqsubset S$" is true but "$T \sqsubset P$" is false, or "$P \sqsubset T$" is true but "$S \sqsubset T$" is false (according to the experience of the system), T is counted as a piece of *negative evidence* for "$S \subset P$", because T disconfirms the transitive relation.
3. If neither "$T \sqsubset S$" nor "$P \sqsubset T$" is true, T provides *no evidence* for "$S \subset P$", because it cannot be used to test the transitive relation.

Another important property of term-oriented language is: it is possible for a term to be subject in one sentence, and predicate in another. The distinction between "predicates" (indicating abstract properties or relations) and "arguments" (denoting concrete objects) in predicate logic no longer exists. A term may serve as an instance for another term, and represent a property for a third term, at the same time. In this way, inferences about *intensions* (properties) and inference about *extensions* (instances) are processed similarly (see [40] for detail).

3.4 Plausible inferences

Due to insufficient knowledge, the system needs to do "ampliative inferences", such as induction, abduction, and analogy. Even deductions are no longer "truth-preserving", in the sense that a conclusion may be revised by new knowledge, even if the premises remain unchallenged.

A major advantage of term logics over predicate/propositional logics is: multiple types of inference can be naturally put into the format of *syllogism* [27, 40]. For example,

	deduction	**induction**	**abduction**
$GIVEN:$	$dove \subset bird$	$dove \subset bird$	$bird \subset flyer$
$GIVEN:$	$bird \subset flyer$	$dove \subset flyer$	$dove \subset flyer$
$CONCLUSION:$	$dove \subset flyer$	$bird \subset flyer$	$dove \subset bird$

Because all knowledge is certain to some extent, the inference rules need to include functions calculating the truth values of the conclusions from those of the premises. Different types of inference have different truth value functions. Generally speaking, abductive and inductive conclusions are supported by less evidence than deductive conclusions. The functions are determined according to the semantics of the language [40].

In axiomatic logics, an inference rule is *valid* if it is truth-preserving. In NARS, an inference rule is *valid* if its conclusion summarizes correctly (according to the semantics) the experience carried by the premises. Due to insufficient resources, all the rules are *local* in the sense that each of them only takes a constant amount of knowledge into consideration, and all the conclusions are *partial* in the sense that each of them are based on part of the system's experience.

3.5 Bi-directional reasoning

Beside the forward reasoning by which conclusions are derived from two pieces of knowledge, NARS also reasons backward, that is, to use a question and a piece of knowledge as premises. If the knowledge happens to provide an answer for the question, the answer is accepted as a tentative result, otherwise a derived question may be generated, whose solution, combined with the knowledge, will provide a solution to the original question. In this way, the reasonings are goal-directed, and the system's resources efficiency can be improved.

Due to insufficient resources, the system cannot consult all relevant knowledge for each question. On the other hand, to set up a static standard for a satisficing answer [37] is too rigid a solution, because the resources may still be variable for a better answer. What NARS does is: to report a best-so-far answer [40], and to continue looking for better answers, permitted by the system's current resources situation [42].

3.6 Controlled concurrency

To support different types of time requirements, in NARS the "time pressure" on each inference task (forward or backward) is not represented by an absolute deadline, but by a relatively defined *urgency*, which indicates the time quota the task can get by comparing it with other tasks, and a *decay*, which indicates how fast the urgency decreases in time.

Because new tasks can appear at any time, and the question-answering activities are usually open-ended (as described above), NARS cannot answer questions one by one, but have to work on many of them in a time-sharing manner.

NARS use a control mechanism named *controlled concurrency* [42], which is similar to the *parallel terraced scan* strategy [13]. In NARS, the processor time is allocated according to the urgency distribution of the tasks, so different tasks are processed at different speeds.

The urgency of a task is adjusted dynamically, according to the decay rate and whether a good answer is already found. When the urgency is decreased to a certain threshold, the task will be removed from the system, no matter how good an answer has been found for it.

3.7 Chunk-based memory

Because only part of the system's knowledge is used in answering each question and because it is possible for a question-answering activity to stop after any number of inference steps (like *anytime algorithms* [2]), it is important for the system to organize its knowledge in such a way that the more relevant and important knowledge is made more accessible.

By using a term logic, it is very natural for NARS to divide its memory into "chunks", each of which corresponds to a term that appeared in the system's past experience. Knowledge and questions are put into the two chunks that are labeled by their subject and predicate. For example, the knowledge "$dove \subset bird$" (with its truth value) is put into chunk *dove* and chunk *bird*. Because in term logics, all forward and backward inferences require the premises share at least one common term, they must happen *within a chunk*. As a result, chunk becomes a natural unit for time and space scheduling [42].

Within a chunk, knowledge is also organized according to a relative *importance* evaluation. Knowledge with a higher importance value is more "accessible" for the system, that is, has a higher probability to be used to process the current tasks. When the storage space is in short supply, some knowledge with low importance is removed.

Importance values of knowledge decay, too. They are also adjusted according to how useful they are in processing tasks. As a result, the more "useful" a pieces of knowledge is, the more "important" it is to the chunk.

3.8 Concepts generating and removing

In NARS, each "concept" has a term as its name, and a chunk as its body. All terms that appear in the system's experience have a corresponding concept within the system. In addition, the system generates compound concepts by using a set of pre-determined operators on given concepts, to summarize its experience more efficiently.

As mentioned in the discussion of semantics, such concepts not necessarily correspond to external objects, but they must be coined and maintained in response to the patterns that repeatedly appear in the system's experience.

For example, when the system notices that a *swan* is both a $flyer$ and a *swimmer*, it concludes that "a *swan* is a $flyer \cap swimmer$", where the predicate, "$flyer \cap swimmer$", is a compound concept. If the concept is already known to the system, this conclusion is put into its body; otherwise a new concept is generated.

Most concepts generated in this way are short-lived. Due to insufficient resources, the system is constantly removing the chunks that are not very useful. Only the compound concepts that correspond to repeatedly appearing patterns in the experience can survive the competition for resources, and develop into stable, full-fledged concepts.

3.9 Working routine

In summary, NARS works by repeatedly carrying out inference steps, each of which consist of the following operations [42]:

1. Check for input tasks (new knowledge or question provided by environment). If there are input tasks, put them into corresponding chunks, increase the priority of the involved chunks, and generate new chunks (if necessary).
2. Pick up a chunk according to their priority distribution. A chunk with a higher priority has a higher probability to be chosen.
3. Pick up a task (knowledge or question) and a piece of knowledge from the chunk, according to the priority distributions among tasks and knowledge.
4. Do different types of inferences (revision, deduction, induction, abduction, backward reasoning, and so on), according to the combination of the task and the knowledge.
5. Adjust the priority for the involved task, knowledge, and chunk, according to how they behave in this inference step.
6. Process the inference results by sending derived tasks and knowledge to correspond chunks, and generating new chunks (if necessary). If the task is an input question, and the knowledge happens to provide a best-so-far answer for it, the answer is reported to the environment.

4 The Properties of NARS

The following is a list of properties shown by NARS, produced by the components described above. Again, formalization and implementation details are omitted.

4.1 Internal conflicts

There maybe conflicts in NARS, in the sense that the same sentence is attached to different truth values when derived from different parts of the experience. Under AIKR, NARS cannot find and eliminate all potential conflicts within its knowledge pool. What it can do is: when a conflict is found, to generate a summarized sentence whose truth value reflects the combined evidence. These conflicts are *normal*, rather than *exceptional*. Actually, their existence is a major driving force of learning, and only by their solutions some types of inference, like induction and abduction, can get their results accumulated [40]. In first order predicate

logic, a pair of conflicting propositions imply all propositions. This does not happen in a term logic, such as NARS.

4.2 Multiple results

Conventional algorithms provide a single answer to each question, then stop working on it. In contrary, NARS reports each answer that is the best one found, then looks for a better one (usually with a lower urgency). Of course, eventually the system will end its working for the question, but the reason is neither that a satisficing answer has been found, nor that a deadline is reached, but that the question-answering task has lost in the resources competition.

As a result, like *trial and error procedures* [16], NARS may provide no, one, or more than one answer(s) to a question. In the last case, a later answer is "better" than a previous one, because it is based on more knowledge, but not necessarily "closer to the objective fact".

When an answer is found, usually there is no way to decide whether it is the last the system can get. In NARS, there is no "final conclusion" that cannot be updated by new knowledge and further consideration, because all conclusions are based on partial experience of the system. This *self-revisable* feature makes NARS a more general model than the various *non-monotonic logics*, in which only binary statements are processed, and only the conclusions derived from default rules can be updated, but the default rules themselves are not effected by experience of the system.

4.3 Reasonable answers

With insufficient knowledge and resources, NARS cannot guarantee that all the answers are *correct*, in the sense that they will not be challenged by the system's future experience. However, the answers are *reasonable* in the sense that they are the best summaries of the past experience, given the current resources supply.

NARS often makes "reasonable mistakes" that are caused by the insufficiency of knowledge and resources, rather than by the errors in the designing or functioning of the system.

4.4 Massive parallelism

Many processes coexist at the same time in NARS. The system not only processes input tasks in parallel, but also does so for the derived subtasks. The fact that the system can be implemented in a single-processor machine does not change the situation, because what matters here is not that the processes run exactly at the same time on several pieces of hardware (though it is possible for NARS to use multiple processors), but that they are not run in an one-by-one way, that is, one process begins after another ends.

4.5 Internal competition

The system does not treat all processes as equal. It distributes its resources among the processes, and only allows each of them progress at certain speed and to certain "depth" in the knowledge pool, according to how much resources it has. Also due to insufficient knowledge, the resource distribution is maintained dynamically (adjusted while the processes are running), rather than statically (determined before the processes begin to run), because the distribution depends on how they work.

As a result, the processes compete with one another for resources. To speed up one process usually means to slow down the others. The urgency value of a task reflects its priority in the competition, but does not determine its actual resources consumption, which is also influenced by the urgency of other tasks.

4.6 Attention and forgetting

With insufficient time, it is impossible for all the knowledge and questions to be treated equal. Some of them (the higher urgency or importance values) get more *attention*, that is, are more active or accessible, while some others are relatively forgotten. With insufficient space, some knowledge and questions will be absolutely forgotten — eliminated from the memory.

Like in human memory [21], in NARS forgetting is not a deliberate action, but a side-effect caused by resource competition.

4.7 Flexible time-consuming

In traditional computing systems, how much time is spent on a task is determined by the system designer, and the user provides tasks without time requirements. On the other hand, many real-time systems allow users to attach a deadline to a task, and the time spent on the task is determined by the deadline [37]. A variation of this approach is that the task is provided with no deadline, but the user can interrupt the process at any time to get a best-so-far answer [2].

NARS uses a more flexible manner to decide how much time is spent on a task, and both the system and the user (environment) influence the result. The user attaches a (relative) urgency to the task, which determines its priority in the resource competition, but the actual allocation also depend on the current situation of the system. As a result, the same task, with the same initial urgency, will get more processing when the system is "idle" than when the system is "busy".

4.8 Distributed representation

Knowledge in NARS is represented distributedly in the sense that there is no one-to-one correspondence between the input/output in the experience/response and the knowledge in

the memory [10].

When a piece of new knowledge is provided to the system, it is not simply inserted into the memory. Spontaneous inferences will happen, which generate derived conclusions. Moreover, the new knowledge may be revised when it is in conflict with previous knowledge. As a result, the coming of new knowledge may cause non-local effects in memory.

On the other hand, the answer of a question can be generated by non-local knowledge. For example, in answering the question "Is *dove* an instance of *bird* ?", a piece of knowledge "$dove \subset bird$" (with its truth value) stored in chunks *dove* and *bird* provides a ready-made answer, but the work does not stop. Subtasks are generated (with lower urgencies) and sent to related chunks. Because there may be internal conflicts within the knowledge base, the previous "local" answer may be revised by knowledge stored somewhere else.

Therefore, the digestion of new knowledge and the generation of answers are both non-local events in memory, though chunks corresponding to terms that appear directly in the input knowledge/question usually have larger contributions. How "global" such an event can be is determined both by the available knowledge and the resources allocated to the task.

4.9 Redundancy

In NARS, information is not only stored distributedly and with duplications, but also processed through multiple pathways. Under AIKR, when a question is asked or a piece of knowledge is told, it is usually impossible to decide whether it will cause redundancy, so multiple copies and pathways become inevitable. Redundancy can help the system recover from partial damage, and also make the system's behaviors depend on statistical facts. For example, if the same question is repeatedly asked from different sources, it will have multiple active copies in the resources competition, and, as a result, get more processor time.

4.10 Robustness

Unlike many symbolic AI systems, NARS will not be easily "killed" by improper inputs. NARS is open and domain-independent, so any knowledge and question, as long as they can be represented in the system's interface language, can be provided to the system. The conflict between new knowledge and previous knowledge will not cause the "implication paradox" (that is, from an inconsistence, any propositions can be derived). Any mistakes in input knowledge can be revised by future experience. The questions beyond the system's current capacity will no longer cause a "combinatorial explosion", but will be abandoned gradually by the system, after some futile efforts. In this way, the system may fail to answer a certain question, but such a failure will not cause a paralysis.

4.11 Knowledge driven

How an answer is generated is heavily dependent on what knowledge is available and how it is organized. Facing a question, the system does not choose a method first, then collect

knowledge accordingly, but lets it interact with available knowledge. In each inference step, what method is used to process a task is determined by the type of knowledge that happens to be picked up at that point.

As a result, the question-answering method for a task is determined dynamically at run-time, by the current memory structure and resource distribution of the system, not by a predetermined problem-oriented algorithm.

4.12 Decentralization

According to the working manner of NARS, each chunk as a processing unit only takes care of its own business, that is, only does inferences where the concept is directly involved. As a result, the answering of a question is usually the cooperation of several concepts.

However, like in connectionist models [30], there is no "global plan" or "central process" that is responsible for each question. The cooperation is carried out by message-passing among chunks. The generating of a specific answer is the *emergent result* of lots of local events, not only caused by the events in its derivation path, but also by the activity of other tasks that adjust the memory structure and compete for the resources. For this reason, each event in NARS is influenced by all the events that happen before it.

4.13 Context sensitivity

In the way that NARS processes questions, it is not surprising that the answer to a specific question is context-sensitive, that is, it not only depends on the question itself and the knowledge the system has, but it also depends on how the knowledge is organized and how the resources are allocated at that time. The *context* under which the system is asked a question, that is, what happens before and after the question in the system's experience, strongly influences what answer the question receives.

Therefore, if the system is asked the same question twice, the answers may be (though not necessarily) different, even though there is no new knowledge provided to the system in the interval.

4.14 Unpredictable behaviors

In principle, the behavior of NARS is unpredictable from an input task along, though still predictable from is initial state and complete experience.

For practical purposes, the behavior of NARS is not accurately predictable to a human observer. To exactly predict the system's answer to a specific question, the observer must know all the details of the system's initial state, and closely follow the system's experience until the answer is actually produced. When the system is complex enough (compared with the information processing capacity of the predictor), nobody can actually do these. However, it does not mean that the system works in a random manner. Its behaviors are still determined by its initial state and experience, so approximate predictions are still possible.

4.15 Spontaneous concepts forming

There are two types of concepts in NARS: those which appear in the system's experience and those generated by the system from available concepts. The need for compound concepts directly comes from AIKR: with insufficient knowledge, the system must consider similar things as equivalent for certain purposes, so to extend past experience into current situation; with insufficient resources, the system must summarize specific experience into general rules, so to save time and space. As a result, the generated concepts not necessarily correspond to external "objects", but to the perceived patterns in the system's experience.

What concept to generate is also a knowledge-driven decision. NARS does not search interesting concepts in a predetermined "concept space" and check each one out by certain standards. Instead, generating a concept is triggered by a pattern noticed by the system in its experience, and how long a generated concept can survive is determined by its relationship with the future experience of the system.

4.16 Fluid concepts

To the system itself, the meaning of a concept is not determined by an interpretation that links it to an external object, but by its relations with other concepts. The relations are in turn determined by the system's experience and its processing on the experience. When a concept is involved in the processing of a task, usually only part of the knowledge associated with the concept is used. Consequently, concepts become "fluid" [13]:

1. No concept has a clear-cut boundary. Whether a concept is an instance of another concept is a matter of degree.[2]
2. The membership evaluations are revisable. Therefore, what a concept actually means to the system is also variable.
3. However, the meaning of a concept is not arbitrary or random, but relatively stable, given the system's experience.

4.17 Autonomy and alienation

The global behavior NARS is determined by the "resultant of forces" of its internal processes. Initially, the system is driven only by input tasks (knowledge and question). The system then derives subtasks recursively according to available knowledge.

However, it is not guaranteed that the achievement of the derived processes will turn out to be really helpful or even related to the original processes, because the knowledge, on which the derivation is based, is defeasible. On the other hand, it is impossible for the system to always determine correctly which processes are more closely related to the original processes. As a result, the system's behavior will to a certain extent depend on "its own

[2] Therefore, all the concepts in NARS are "fuzzy" [44], however, NARS is not a "fuzzy logic", according to the current usage of the term. See [41] for more discussions.

tasks", which are actually more or less independent of the original processes, even though historically derived from them. This is the *functional autonomy* phenomena [22].

In the extreme form, the derived tasks may become so strong that they even prevent the input tasks from being fulfilled. In this way, the derived tasks are *alienated*.

4.18 Creativity

The alienation and unpredictability sometimes result in the system to be "out of control", but at the same time, they lead to *creative and original* behaviors, because the system is pursuing goals that are not directly assigned by its environment or its innateness, with methods that are not directly deduced from given knowledge.

By *creativity*, it does not mean that all the results of such behaviors are of benefit to the system, or excellent according to some outside standards. It does not mean that these behaviors come from nowhere, or a "free will" of some sort, neither. In contrary, it means that the behaviors are novel to the system, and cannot be attributed either to the designer (who determined the system's initial state and skills) or to a teacher (who determined part of the system's experience) alone. Designers and teachers only make the creative behaviors possible. What turns the possibility into reality is the system's experience, and for a system that lives in a complex environment, its experience is not completely determined by any other systems (human or computer). For this reason, these behaviors, with their results, are better to be attributed to the system *itself*, than to anyone else [11].

4.19 Own life

Traditional computer systems always repeat the following "life cycle": waiting for problems $\rightarrow$ accepting a problem $\rightarrow$ working on it $\rightarrow$ getting a solution for it $\rightarrow$ waiting for problems $\cdots$.

In contrary, NARS has a "life-time of its own" [7]. When the system is experienced enough, there will be lots of tasks for the system to process. On the other hand, new input can come at any time. Consequently, the system's history is no longer like the previous loop. The system usually working on its "own" tasks, but at the same time, it is always ready to respond to new tasks. Each piece of input usually attracts the system's attention for a while, and also cause some long-term effects. The system never reaches a "final state" and stops there, though it can be reseted by a human user to its initial state.

4.20 Summary

In this paper, it is impossible to discuss each of the properties in detail for its own interest. What we are doing here is to show that all of them are closely related to the working definition introduced previously. The following is a list of what is shared by the properties:

1. Few of the properties are proposed by axiomatic logical systems and ordinary computing systems.

2. They appear in human thinking, and are often recognized as related to intelligence.
3. Most of them have been discussed and produced by different AI theories and systems, but separately.
4. They are often judged as impossible by the critics of AI (for examples, see [6, 33]).

The interesting point is: now all of them can be derived (to a certain extent) from the previous working definition of intelligence, and many of them have been shown, to a different extent, by an implementation of NARS [42]. In NARS, these properties, no matter whether they are referred to as advantage or disadvantage, become inevitable "epiphenomena" [11] of a unified architecture, which is based on several simple principles.

5 What Is Unintelligent?

5.1 The need to exclude something

When defining intelligence, many authors ignore the complementary question: what is unintelligent? For AI to be a branch of science, this question must be clearly answered, as pointed out by Searle [33].

As any concept, if everything is intelligent, then this concept is empty. We need to rule out something to study the remaining objects. Even if we agree that intelligence, like almost all properties, is a matter of degree, we still need criteria to indicate what makes a system more intelligent than another.

Further more, for AI to be a (independent) discipline, we require the concept "intelligence" to be different from other established concepts, otherwise, we are only talking about some well-known stuff with a new name, which is not enough to set up a *new* branch of science. For example, if every computer system is intelligent, it is better to stay within the theory of computation. "Intelligent system" does not mean a faster and bigger computer. It should be different from some better understood concepts like "non-numerical computing", "parallel inference", or "complex system", otherwise we would use those concepts instead, to avoid confusion.

The distinction should also be consistent with the way the word "intelligence" is used in everyday language, otherwise we would better use another word. Intuitively, normal humans are intelligent, but traditional computing systems and most animals are not, or much less intelligent. As Searle said [33], a definition of intelligence can rule out candidates like stomach, adding machine, or telephone.

Though under the flag of AI, different people are actually doing quite different things, we can still feel something in common in the field, at least shared by the problems, if not by the suggested solutions. For one thing, hard AI problems are usually easy for human beings, which make AI different from other sub-domains of computer science, where computer systems do better than people.

Therefore, an unintelligent system is not necessarily incapable or gives only wrong results. Actually, most ordinary computer systems and many animals can do something that human

beings cannot. However, their ability usually cannot earn the title "intelligent" for them. What is missing in these capable-but-unintelligent systems?

5.2 Pure-axiomatic, semi-axiomatic, and non-axiomatic

According to the working definition of intelligence introduced previously, an *unintelligent* system is one that does not adapt to its environment. Especially, in artificial systems, a *unintelligent* system is one that is designed under the assumption that it only works on problems for which the system has sufficient knowledge and resources.

Let us concentrate on reasoning systems for more details. Generally, we can distinguish three types of reasoning systems:

Pure-axiomatic systems. They are designed under the assumption that both knowledge and resources are sufficient (with respect to the questions), so adaption is not necessary. A typical example is a "formal system" suggested by Hilbert (and many others): all answers are deduced from a set of axioms by a determined algorithm, when applied to a practical domain through a model-theoretical semantics. Such a system is based on sufficient knowledge and resources, because all relevant knowledge is already embedded in the axioms, and questions have no time constraints, as long as they are answered in finite time. If a question goes beyond the scope of the axioms, it is not the system's fault, but the user's, so no attempt is made for the system to improve its capacity and to adapt to its environment.

Semi-axiomatic systems. They are designed under the assumption that knowledge and resources are insufficient in some, but not all, aspects. Consequently, adaption is necessary. Most current AI approaches fall into this category. For example, non-monotonic logics consider the revision of defeasible conclusions (such as "Tweety can fly") caused by new evidence (such as "Tweety is a penguin"), but usually make default rules (such as "Birds normally can fly") unchangeable, and do not take time pressure into account [29]. Many learning systems attempt to improve the behaviors of a system, but still work with binary logic, and look for best solutions of problems. Various heuristic searching systems give up optimum results by assuming the existence of time limit, but they usually do not attempt to learn from experience, and the change of time pressure is beyond consideration.

Non-axiomatic systems. Such a system has been briefly introduced in the previous sections. It is not to say that its knowledge and resources are *always* insufficient, but it is the *normal* situation, so the system needs to be designed under such an assumption and works in such a way, though for a specific question, its knowledge and resources may happen to be sufficient to answer it.

By our working definition, we say that pure-axiomatic systems are not intelligent at all, non-axiomatic systems are intelligent, and semi-axiomatic systems are intelligent in certain aspects.

An intelligent system is not always "better" than an unintelligent system for practical purposes. Actually, it is the contrary: when a problem can be solved by both of them, the unintelligent system is usually better, because it guarantees a correct solution. As Hofstadter said, for tasks like adding two numbers, a "reliable but mindless" system is better than an "intelligent but fallible" system [11].

Pure-axiomatic systems are very useful in mathematics, where the aim is to idealize knowledge and questions to such an extent that the revision of knowledge and the deadline of questions can be ignored. In such situations, questions can be answered in a way that is so accurate and reliable that the procedure can be reproduced by a Turing machine.

We need intelligence only when no such pure-axiomatic method can be used, due to the insufficiency of knowledge and resources. For the same reason, the performance of a non-axiomatic system is not necessarily better than that of a semi-axiomatic system, but it can work in environments where the latter cannot be used.

Under the above definitions, intelligence is still (as we hope) a matter of degree. Not all systems in the "non-axiomatic" and "semi-axiomatic" categories are equally intelligent. Some systems may be more intelligent than some other systems for having a higher resources efficiency, using its knowledge in more ways, communicating with its environment in a richer language, adapting more rapidly and thoroughly, and so on. For instance, there are many ways that NARS can be extended from its current design [42], though it is already a non-axiomatic system.

5.3 Intelligence and computation

What is the relationship of artificial intelligence (AI) and computer science (CS)? What is the position of AI in the whole science enterprise?

Traditionally, AI is referred to as a branch of CS. According to our previous definitions, AI can be implemented with the tools provided by CS, but from a *theoretical* point of view, they make opposite assumptions: CS focuses on pure-axiomatic systems, but AI focuses on non-axiomatic systems.

The fundamental assumptions of computer science can be found in mathematical logic (especially, first order predicate logic) and computability theory (especially, Turing machine). These theories take the sufficiency of knowledge and resources as implicit postulates, so adaption, plausible inference, and tentative solutions of problems are neither necessary nor possible.

Similar assumptions are often accepted by AI researchers with the following justification: "We know that the human mind usually works with insufficient knowledge and resources, but if you want to set up a *formal* model and then a *computer* system, you must somehow *idealize* the situation."

It is true that any formal model is an idealization, and so is NARS. The problem is what to omit and what to preserve in the idealization. In the current implementation of NARS, many factors that should influence reasoning are ignored, but AIKR is strictly assumed throughout. Why? Because AIKR is a *definitive* feature of intelligence, so if it were lost

through the "idealization" the resulting study would be about something else.

If NARS is implemented in a von Neumann computer, can it go beyond the scope of CS? Yes, it is possible because a computer system is a hierarchy with many levels [11]. Some critics implicitly assume that *because a certain level of a computer system can be captured by first order predicate logic and Turing machine, these theories also bind all the performances the system can have.* This is not the case. When a system A is implemented by a system B, the former does not necessarily inherit all properties of the latter. For example, we cannot say that a computer cannot process decimal numbers (because they are implemented by binary numbers), cannot process symbols (because they are coded by digits), or cannot use functional or logical programming language (because they are eventually translated into procedural machine language).

As a virtual machine, NARS can be based on another virtual machine, which is a pure-axiomatic system [42], and this fact does not make the system less "non-axiomatic". Obviously, with its fluid concepts, revisable knowledge, and fallible inference rules, NARS breaks the regulations of classic logics. Being context-dependent and open-ended, the question-answering activities are also no longer *computations*. On the other hand, if we take the system's complete *experience* and *response* as input and output, then NARS is still a Turing machine that definitely maps inputs to outputs in finite steps. What happens here has been pointed out by Hofstadter: "Something can be computational at one level, but not at another level." [12]. On the contrary, traditional computer systems are Turing machines either globally (from experience to response) or locally (from question to answer).

Many arguments proposed against logical AI (for example, [1, 20]), symbolic AI (for example, [6]), or AI as a whole (for example, [33]), are actually against a more specific target: pure-axiomatic systems. Designed as a *reasoning system*, but not a "logicist" one [25], NARS actually shares more philosophical opinions with the *sub-symbolic*, or *connectionist* movement [12, 14, 30, 36], but chooses to formalize and implement these opinions in a framework that looks more close to the traditional symbolic AI tradition. The practice of NARS shows that such a framework has its advantages, such as more general and abstract, more closely related to the old problems in the domain, and more suitable for the studies of high-level phenomena that are related to intelligence.[3]

6 Compared with Other Definitions

There are just too many different opinions about what intelligence is and what the best methodology for AI is, that it is impossible to compare our ideas to each of them. Instead, in the following these opinions are classified into several categories, and their relationship with the working definition of intelligence introduced previously are discussed.

Generally speaking, the research of artificial intelligence has two major motivations. As a field of science, we want to learn how human mind, or "mind" in general, works; as a branch

[3]In addition, NARS can also be interpreted as a network by taking terms as nodes, and judgments as links [40]. The possibility of interpreting NARS both as a symbolic reasoning system and an associative network ease the comparisons between NARS and other systems.

of technology, we want to apply computers to domains where only the human mind works well currently. Intuitively, both goals can be achieved if we can build computer systems that are "similar to the human mind".

But in what sense are they "similar"? To different people, the similarity may be in the *structure, performance, capacity, function,* or *principle*. In the following, we discuss typical opinions in each of the five categories, to see where these definitions of intelligence will lead AI to.

6.1 To simulate human brain

Intelligence is produced by brain, so maybe AI should attempt to simulate a brain in a computer model as faithful as possible. Such an opinion is put in its extreme form by neuroscientists Reeke and Edelman, who argue that "the ultimate goals of AI and neuroscience are quite similar" [28].

Though it sounds reasonable to identify AI with *brain model*, few AI researchers exactly take such an approach. Even the "neural network" movement is "not focused on *neural modeling* (i.e., the modeling of neurons), but rather ... focused on *neurally inspired* modeling of cognitive process" [30].

Why? One obvious reason is the *complexity* of this approach. The current technology is still not powerful enough to simulate a huge neural network, not to mention that there are still many mysteries in brain.

Moreover, even if we were able to build a brain model at the neuron level to the desired accuracy, it could not be called the success of AI, though it would be the success of neuroscience. AI is more closely related to "model of mind", that is, a *high-level* description of brain activity in which biological concepts do not appear.

A high-level description is preferred, not because a low level description is impossible, but because it is usually more simple and general. A distinctive character of AI is the attempt to "get a mind without a brain", that is, to describe mind in a medium-independent way. This is true for all "models": by the building of a model, we concentrate on certain properties of an object or process, and ignore irrelevant aspects, so, as a result, we can get insights that are hard to see in the object or process itself. For this reason, an accurate duplication is not a model, and a model including unnecessary details is not a good model.

If we agree that "brain" and "mind" are different concepts, then *a good model of brain is not a good model of mind*, though the former is useful for its own sake, and helpful for the building of the latter.

6.2 To duplicate human behaviors

Because we always judge the intelligence of other people by their behaviors, it is natural to use "reproducing behaviors of human brain as accurate as possible" as the aim of AI. In this way, we can draw "a fairly sharp line between the physical and the intellectual capacities of a man" (Turing in [38]).

Such a working definition of intelligence asks researchers to use passing the Turing test as a sufficient *and necessary* condition for having intelligence, and to take psychological evidence seriously, as Soar does [23].

Such a working definition can be criticized from different directions:

Is it sufficient? Searle argues that even if a computer system can pass the Turing test, it still cannot *think*, because it lacks the *causal capacity* of brain to produce *intentionality*, which is a biological phenomenon [33]. However, he does not demonstrate convincingly why thinking, intentionality, and intelligence cannot have a high-level (higher than biological level) description. His "Chinese room" thought experiment is based on the assumption that a formal system can only get meaning according to model-theoretic semantics, but it is not the case, as discussed previously.

Is it possible? Due to the nature of the Turing test and the resources limitation of a concrete computer system, it is impossible for the system to remember all possible questions and proper answers in advance, then pretend to be a human being by searching such a list. To imitate human performance in a conversation, it has to produce the answers in a "human-way". To do this, it not only needs some cognitive facilities, but also a "human experience" [8]. Therefore, it must have a body that feels like human, it must have all human motivations (including the biological ones), and it must be treated by people as a human being — so it must simply be an "artificial human", rather than a computer system with artificial intelligence.

Is it necessary? As French points out, by using behaviors as evidence, the Turing test is for human intelligence, not for intelligence in general [8]. As a working definition for intelligence, such an approach can lead to good psychological models, which are valuable for many reasons, but suffer from a "human chauvinism" [11] — we have to say, according to the definition, that E. T. is not intelligent, because it will definitely fail a Turing test. Furthermore, we have to say that no other animal except a human has vision, if we define "vision" as "indistinguishable to human in response to light stimulus to eye" or something like that. It is a very unusual and unfruitful way to use concepts.

6.3 To solve hard problems

In everyday language, "intelligent" is usually applied to people who can solve hard problems. Many definitions of intelligence come from this usage. According to this type of definition, intelligence is the *capacity* of solving hard problems, and *how* the problems are solved is not very important.

What problems are "hard"? In the early days of AI, many researchers worked on typical intellectual activities, such as game-playing and theorem-proving. Nowadays, many people turn to "real problems" appearing in various domains to build "expert systems". Obviously, experts are usually intelligent, so if a computer system can solve problems that only experts can solve, the computer system must be intelligent, too.

This movement has produced many practically useful systems, and attracted financial and manpower investments, and thus made important contributions to the development of AI enterprise. Usually, the systems are developed by analyzing domain knowledge and expert strategy, then building them into a computer system.

Though often profitable, these systems do not provide much insight about how the mind works. No wonder people ask, after knowing how such a system works, "Where's the AI?" [32] — these systems look just like ordinary computer application systems, and suffer from rigidity and brittleness (something AI wants to avoid).

Sometimes computer systems are referred to as "intelligent" by some people, because the use of techniques that developed or widely used by AI workers, for example, to represent knowledge in frames or sematic networks, to program in Lisp or Prolog, or to organize the system as an inference engine or production system. However, the use of these techniques does not cause a fundamental difference — usually the same capacity can be got, though not as conveniently, by using traditional data structures, system organizations and programming languages in computer science.

If intelligence is defined as the "capacity of solving *hard* problems", then the next question is: "*Hard* to whom?" If we say "hard to human beings", then most existing computer softwares are already intelligent — no human can manage a database as good as a database management system, or substitute a word in a file as fast as an editing program. If we say "hard to computers", then AI becomes "whatever hasn't been done yet", which is called "Tesler's Theorem" by Hofstadter [11] and "gee whiz view" by Schank [32]. Such a definition cannot lead to a proper distinction between intelligent and unintelligent systems. It is not a good approach, if the aim of study is intelligence in general, rather than concrete domain problems.

6.4 To carry out cognitive functions

According to this type of opinion, intelligence is characterized by a set of cognitive functions, such as reasoning, perception, memory, problem solving, language use, and so on. Researchers with this idea usually concentrate on one of the functions, with the believe that the works on different functions can be combined together in the future to get a whole picture of intelligence.

A "cognitive function" is often defined in a general and abstract manner, independent to the brain mechanisms that carry it out, the specific performances that it can produce, and the practical domains that it can be applied to. The direct aim of the study is to build a computer system with the desired function(s).

This approach has produced, and will produce more, information processing tools in the form of software packages and even specialized hardware, each of which can carry out a function that is similar to certain mental skills of human beings, and therefore can be used in various domains for practical usage.

However, this kind of success is not enough for claiming that it is the proper way for AI study. To define intelligence as a "toolbox" of functions has the following weaknesses:

1. Even if we can get the desired tools, it does not mean that we can easily combine them, because the tools are usually based on different postulations.
2. When specified in isolation, an implemented function is often quite different from its "natural form" that happens in the human mind. For example, to study analogy without perception leads to distorted cognitive models [4].
3. Having a certain cognitive function is not enough to make a system intelligent. For example, problem-solving by exhaustive searching is usually not considered intelligence, and many unintelligent animals have perception.

The basic problem of the "toolbox" approach is: without a "big picture" in mind, the study of a cognitive function in an isolated, abstracted, and often distorted form does not necessarily contribute to our understanding of intelligence.

A common defense goes like this: "Intelligence is very complex, so we have to start from a single function to make the study simple." For many systems with weak internal connections, this is often a good choice, but for a system, like a mind, whose complexity comes directly from its tangled internal interactions, the situation may be just the opposite. When the so called "functions" are actually phenomena produced by a complex-but-unified mechanism, to reproduce all of them together (by duplicating the mechanism) is even simpler than to reproduce only one of them. We have evidence to believe that intelligence is such a problem.

6.5 To develop new principles

According to this type of opinions, what distinguishes intelligent systems and unintelligent systems are their *postulations*, applicable *environments*, and basic *principles* of information processing.

The working definition of intelligence introduced in this paper belongs to this category. As a reasoning system adapting to its environment with insufficient knowledge and resources, NARS has many cognitive *functions*, but they are better referred to as closely related external phenomena, rather than as well-defined tools used by the system. By leaning from its experience, NARS has a potential *capacity* to solve hard problems[4], but it has no built-in capacity, so, without proper training, no capacity is guaranteed, and even acquired capacities can be lost. With similar principles, we expect NARS behave similarly to human beings, but the similarity exists at a more abstracted level than concrete *performance*. Due to the fundamental difference in experience, NARS is not expected to accurately reproduce psychological data or to pass a Turing test. Finally, the internal *structure* of NARS has some properties in common with a description of the human mind at the sub-symbolic level, but it is not an attempt to build an artificial neural network.

In summary, the *structure* approach contributes to neuroscience, the *performance* approach contributes to psychology, the *capacity* approach contributes to various application

[4] Actually, *hard problems* are exactly those for them a solver (human or computer) has insufficient knowledge and resources. Once the answer is known in advance, or all possible answers are known and there is enough time to check them one by one, no problem is hard any more.

domains, and the *function* approach contributes to computer science. These approaches are not as good as the *principle* approach, when *intelligence* is the peak to climb, though they are still valuable for other purposes, and helpful for the study of AI.

As a matter of fact, what has been proposed in our definition is not entirely new to the AI community. It seems that no one will argue against the opinion that "adaption", or "learning", is essential for intelligence, and "insufficient knowledge and resources" is the focus of many subfields of AI, such as heuristic search, reasoning under uncertainty, real-time planning, and machine learning.

We can also find similar attempts to base intelligence on certain basic principles (as some kind of rationality) in the following ideas:

> Simon's *bounded rationality*: "Within the behavioral model of bounded rationality, one doesn't have to make choices that are infinitely deep in time, that encompass the whole range of human values, and in which each problem is interconnected with all the other problems in the world." [35]
>
> Cherniak's *minimal rationality*: "We are in the finitary predicament of having fixed limits on our cognitive resources, in particular, on memory capacity and computing time." [5]
>
> Russell and Wefald's *limited rationality*: "Intelligence was intimately linked to the ability to succeed as far as possible given one's limited computational and informational resources." [31]

Medin and Ross even have made it so clearly: "Much of intelligent behavior can be understood in terms of strategies for coping with too little information and too many possibilities." [21]

With all of the above already said, what is *new* in NARS? We claim that the following makes NARS different from other AI projects, philosophically and methodologically:

1. To explicitly and unambiguously define intelligence as "adaption with insufficient knowledge and resources".
2. To further specify "with insufficient knowledge and resources" as *being finite, real-time, and open.*
3. To choose a *reasoning system* as the framework for applying the definition *completely* in an all-encompassing manner.
4. To invent proper techniques, such as term-oriented language, experience-ground semantics, extended syllogism, chunk-based memory structure, controlled concurrency, to formalize the definition in a symbolic logic, then to implement the logic into a computer reasoning system.

7 A Primary Evaluation

After all the descriptions and discussions, let us compare our working definition of intelligence with the requirements set up at the beginning of the paper:

Similarity. Obviously, natural information processing systems, i.e., humans and animals, are adaptive, and often have to work with insufficient knowledge and resources. Being more adaptive, human beings are much more intelligent than other animals. On the other hand, though traditional computing systems also have *limited* knowledge and resources, they usually limit the problems to be processed, so to make the knowledge and resources *sufficient* for those problems. Therefore, our definition draws a line between intelligent and unintelligent systems in such a way that is similar to the common usage of the word "intelligence".

Exactness. Our definition is exact, because whether a system is adaptive can be determined by testing whether its behaviors depend on its experience. For a computer system, whether it is designed under AIKR can be determined by testing the three properties: finite (Can the system forget?), real-time (Can the system respond to different time requirements?), and open (Does the system restrict what it can be told or asked?). Like Turing's idea, we also decide whether a system is intelligent by "talking" with it, but the standards are different — we do not require that the system talk like a human.

Fruitfulness. As described previously, the definition is instructive in determining the major components of NARS, which produce many desired properties. Based on the definition, NARS addresses many problems in AI in a consistent manner, and also provides a foundation for AI that clearly distinguishes it from other related disciplines, such as computer science, psychology, and neuroscience.

Simplicity. Our definition is quite simple, so it is easy to be discussed and applied to research. Its direct result, NARS, is also not very complex in its structure (compared with other AI systems), though the system's behavior can be very complex due to its interaction with its environment.

With the above properties, we believe that the working definition of intelligence introduced in this paper is *better* than many others accepted by AI researchers. However, we do not claim that the definition is *the correct one*. Obviously, there are many intelligence-related phenomena that have not been explained by the definition. These phenomena suggest further extensions of NARS, which may cause future revisions to the definition, but cannot be used as evidence at this time against the definition. Since (as discussed at the beginning of the paper) working definitions correspond to the choice of a paradigm *before*, not *after*, carrying out research in the paradigm, a working definition can only be rejected by providing a better one, rather than by finding a weakness in it. We hope in the near future our definition can be replaced by a better one, according to the requirements of *similarity, exactness, fruitfulness*, and *simplicity*.

Acknowledgment

This work has been supported by a research assistantship from the Center for Research on Concepts and Cognition, Indiana University.

Thanks to Helga Keller for correcting my English.

References

[1] L. Birnbaum. Rigor mortis: a response to Nilsson's "Logic and artificial intelligence". *Artificial Intelligence*, 47:57–77, 1991.

[2] M. Boddy and T. Dean. Deliberation scheduling for problem solving in time-constrained environments. *Artificial Intelligence*, 67:245–285, 1994.

[3] R. Carnap. *Logical Foundations of Probability.* The University of Chicago Press, Chicago, 1950.

[4] D. Chalmers, R. French, and D. Hofstadter. High-level perception, representation, and analogy: a critique of artificial intelligence methodology. *Journal of Experimental and Theoretical Artificial Intelligence*, 4:185–211, 1992.

[5] C. Cherniak. *Minimal Rationality.* The MIT Press, Cambridge, Massachusetts, 1986.

[6] H. Dreyfus. *What Computers Still Can't Do.* The MIT Press, Cambridge, Massachusetts, 1992.

[7] J. Elgot-Drapkin, M. Miller, and Perlis D. Memory, reason, and time: the step-logic approach. In R. Cummins and J. Pollock, editors, *Philosophy and AI*, chapter 4, pages 79–103. The MIT Press, Cambridge, Massachusetts, 1991.

[8] R. French. Subcognition and the limits of the Turing test. *Mind*, 99:53–65, 1990.

[9] C. Hempel. A purely syntactical definition of confirmation. *Journal of Symbolic Logic*, 8:122–143, 1943.

[10] G. Hinton, J. McClelland, and D. Rumelhart. Distributed representation. In D. Rumelhart and J. McClelland, editors, *Parallel Distributed Processing: Exploration in the Microstructure of cognition, Vol. 1, Foundations*, pages 77–109. The MIT Press, Cambridge, Massachusetts, 1986.

[11] D. Hofstadter. *Gödel, Escher, Bach: an Eternal Golden Braid.* Basic Books, New York, 1979.

[12] D. Hofstadter. Waking up from the Boolean dream, or, subcognition as computation. In *Metamagical Themas: Questing for the Essence of Mind and Pattern*, chapter 26. Basic Books, New York, 1985.

[13] D. Hofstadter and M. Mitchell. The Copycat project: a model of mental fluidity and analogy-making. In K. Holyoak and J. Barnden, editors, *Advances in Connectionist and Neural Computation Theory, Volume 2: Analogical Connections*, pages 31–112. Ablex Publishing Corporation, Norwood, New Jersey, 1994.

[14] J. Holland. Escaping brittleness: the possibilities of general purpose learning algorithms applied to parallel rule-based systems. In R. Michalski, J. Carbonell, and T. Mitchell, editors, *Machine Learning: an artificial intelligence approach*, volume II, chapter 20, pages 593–624. Morgan Kaufmann, Los Altos, California, 1986.

[15] D. Kirsh. Foundations of AI: the big issues. *Artificial Intelligence*, 47:3–30, 1991.

[16] P. Kugel. Thinking may be more than computing. *Cognition*, 22:137–198, 1986.

[17] D. Lenat and E. Feigenbaum. On the thresholds of knowledge. *Artificial Intelligence*, 47:185–250, 1991.

[18] D. Marr. Artificial intelligence: a personal view. *Artificial Intelligence*, 9:37–48, 1977.

[19] J. McCarthy. Mathematical logic in artificial intelligence. *Dædalus*, 117(1):297–311, 1988.

[20] D. McDermott. A critique of pure reason. *Computational Intelligence*, 3:151–160, 1987.

[21] D. Medin and B. Ross. *Cognitive Psychology*. Harcourt Brace Jovanovich, Fort Worth, 1992.

[22] M. Minsky. *The Society of Mind*. Simon and Schuster, New York, 1985.

[23] A. Newell. *Unified Theories of Cognition*. Harvard University Press, Cambridge, Massachusetts, 1990.

[24] A. Newell and H. Simon. Computer science as empirical inquiry: symbols and search. The Tenth Turing Lecture, March 1976. Frist published in *Communications of the Association for Computing Machinery* 19.

[25] N. Nilsson. Logic and artificial intelligence. *Artificial Intelligence*, 47:31–56, 1991.

[26] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann Publishers, San Mateo, California, 1988.

[27] C. Peirce. *Collected papers of Charles Sanders Peirce*, volume 2. Harvard University Press, Cambridge, Massachusetts, 1931.

[28] G. Reeke and G. Edelman. Real brains and artificial intelligence. *Dædalus*, 117(1):143–173, 1988.

[29] R. Reiter. Nonmonotonic reasoning. *Annual Review of Computer Science*, 2:147–186, 1987.

[30] D. Rumelhart and J. McClelland. PDP models and general issues in cognitive science. In D. Rumelhart and J. McClelland, editors, *Parallel Distributed Processing: Exploration in the Microstructure of cognition, Vol. 1, Foundations*, pages 110–146. The MIT Press, Cambridge, Massachusetts, 1986.

[31] S. Russell and E. Wefald. *Do the Right Thing*. The MIT Press, Cambridge, Massachusetts, 1991.

[32] R. Schank. Where is the AI. *AI Magazine*, 12(4):38–49, 1991.

[33] J. Searle. Minds, brains, and programs. *The Behavioral and Brain Science*, 3:417–424, 1980.

[34] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, New Jersey, 1976.

[35] H. Simon. *Reason in Human Affairs*. Stanford University Press, Stanford, California, 1983.

[36] P. Smolensky. On the proper treatment of connectionism. *Behavioral and Brain Sciences*, 11:1–74, 1988.

[37] J. Strosnider and C. Paul. A structured view of real-time problem solving. *AI Magazine*, 15:45–66, 1994.

[38] A. Turing. Computing machinery and intelligence. *Mind*, LIX:433–460, 1950.

[39] P. Wang. Belief revision in probability theory. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, pages 519–526. Morgan Kaufmann Publishers, San Mateo, California, 1993.

[40] P. Wang. From inheritance relation to non-axiomatic logic. Technical Report 84, Center for Research on Concepts and Cognition, Indiana University, Bloomington, Indiana, 1993. Available via WWW at *http://www.cogsci.indiana.edu/farg/pwang_papers.html*. A revised version is going to appear in the *International Journal of Approximate Reasoning*.

[41] P. Wang. The interpretation of fuzziness. Technical Report 86, Center for Research on Concepts and Cognition, Indiana University, Bloomington, Indiana, 1993. Available via WWW at *http://www.cogsci.indiana.edu/farg/pwang_papers.html*.

[42] P. Wang. Non-axiomatic reasoning system (version 2.2). Technical Report 75, Center for Research on Concepts and Cognition, Indiana University, Bloomington, Indiana, 1993. Available via WWW at *http://www.cogsci.indiana.edu/farg/pwang_papers.html*.

[43] P. Wang. A defect in Dempster-Shafer theory. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 560–566. Morgan Kaufmann Publishers, San Mateo, California, 1994.

[44] L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.