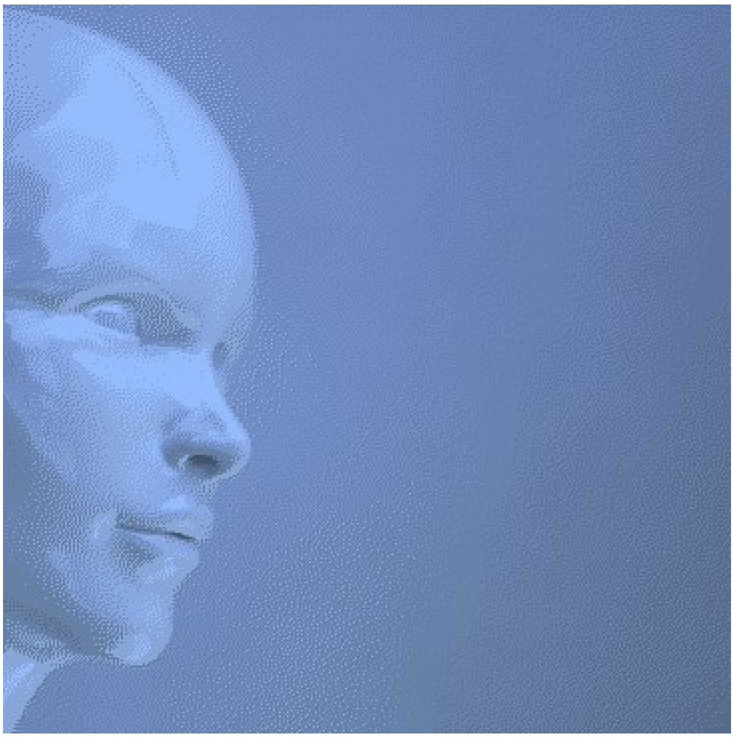
Scientific challenge:

Beat the simplest results of my Controlled Natural Language (CNL) reasoner



Introduction

Autonomous reasoning requires both natural intelligence and natural language. Without knowing, Aristotle applied natural intelligence to natural language roughly 2,400 years ago:

> Given: “All men are mortal.”
> Given: “Socrates is a man.”
•
• Logical conclusion:
< “Socrates is mortal.”

Roughly 200 years ago, such reasoning constructions were formalized through Predicate Logic. And since the start of this century, these reasoning constructions are implemented in software through Controlled Natural Language (CNL) reasoners. CNL reasoners are able to autonomously derive new knowledge from previously unknown knowledge, and to express the derived knowledge in readable sentences (with a limited grammar).

Problem description 1: Reasoning in the past tense

The reasoning example mentioned above was true during the life of Socrates. But now, after the ultimate proof of his morality – his death in the year 399 BC – we should use the past tense form:

> Given: “All men are mortal.”
> Given: “Socrates **was** a man.”
•
• Logical conclusion:
< “Socrates **was** mortal.”

The tense of a verb tells us about the state of the involved statement:

- “Socrates **is** a man” tells us that Socrates is still alive;
- “Socrates **was** a man” tells us that Socrates isn't amongst the living anymore.

In regard to the conclusion:

- “Socrates **is** mortal” tells us that the death of Socrates is inevitable, but that his mortality isn't proven yet by hard evidence;
- “Socrates **was** mortal” tells us that his mortality is proven by hard evidence.

In the past 2,400 years, scientists have "forgotten" to define algebra for the past tense. So, reasoning in the past tense form is not described in any scientific paper, while it is implemented in my CNL reasoner.

Problem description 2: Possessive reasoning

Besides past tense reasoning, also possessive reasoning – reasoning using possessive verb "has/have" – is not supported by predicate logic (algebra). For example:

> Given: "Paul is a son of John."
•
• Logical conclusion:
< "John **has** a son, called Paul."

Or the other way around:

> Given: "John **has** a son, called Paul."
•
• Logical conclusion:
< "Paul is a son of John."

So, why doesn't predicate logic (algebra) support past tense reasoning – nor possessive reasoning – in a natural way? Why should any predicate beyond present tense verb "is/are" be described in an artificial way, like has_son(john,paul)? Why is algebra still not equipped for natural language, after those centuries of scientific research?

Problem description 3: Generation of questions

There is even more: Algebra describes the Exclusive OR (XOR) function in a natural way, while CNL reasoners still don't implement its linguistic equivalent: conjunction "or". CNL reasoners are therefore unable to generate the following question:

> Given: "Every person is a man **or** a woman."
> Given: "Addison is a person."
•
• Logical question:
< "Is Addison a man **or** a woman?"

So, 2,400 years after Aristotle, scientists don't even understand the basics of natural intelligence and natural language:

Words like definite article "the" (see Block 6), conjunction "or" (see Block 5), possessive verb "has/have" (see Block 1, Block 2 and Block 3) and past tense verbs "was/were" and "had" (see Block 4) have a naturally intelligent function in language.

Generally accepted workaround

The generally accepted workaround in the field of Artificial Intelligence (AI) and knowledge technology (NLP), to enter knowledge containing verb "have", is to program it directly into a reasoner, like: has_son(john,paul). However, this is **not** a generic solution (=science), but a specific solution to a specific problem (=engineering). Because it requires to program each and every noun directly into the reasoner (has_daughter, has_father, has_mother, and so on), and for each and every new language. As a consequence, there is no technique available to convert a sentence like "Paul is a son of John" to "John **has** a son, called Paul" in a generic way – from natural language, through an algorithm, to natural language – by which noun "son" and proper nouns "John" and "Paul" don't have to be programmed into the reasoner. It is just the first example of this challenge (see Block 1).

Below, a contribution I received from a student, in an attempt to solve this problem. With his permission, his Excel implementation for the English language:

```
= IF(ISERROR(SEARCH("has a";A1));MID(A1;SEARCH("of";A1)+3;999) & " has a" & IF(ISERROR(SEARCH("is an";A1));" ";"n ") & MID(SUBSTITUTE(A1;"is an";"is a");SEARCH("is a"; SUBSTITUTE(A1;"is an";"is a"))+5;SEARCH("of"; SUBSTITUTE(A1;"is an";"is a"))-SEARCH("is";SUBSTITUTE(A1;"is an";"is a"))-6) & " called " & LEFT(A1;SEARCH("is";SUBSTITUTE(A1;"is an";"is a"))-1);MID(SUBSTITUTE(A1;"has an";"has a");SEARCH("called";SUBSTITUTE(A1;"has an";"has a"))+7;999) & " is a" & IF(ISERROR(SEARCH("has an";A1));" ";"n ") & MID(SUBSTITUTE(A1;"has an";"has a");SEARCH("has a"; SUBSTITUTE(A1;"has an";"has a"))+6;SEARCH("called"; SUBSTITUTE(A1;"has an";"has a"))-SEARCH("has";SUBSTITUTE(A1;"has an";"has a"))-7) & " of " & LEFT(A1;SEARCH("has";SUBSTITUTE(A1;"has an";"has a"))-1))
```

This solution doesn't check for word types, as explained in paragraph 2.3.4. The function of word types in reasoning of my fundamental document. Besides that, this logic needs to be copied for each language, while a generic solution has only one logical implementation. Moreover, this implementation can't be expanded to process for example multiple specifications words, like in: "Paul is a son of John and Anna" or "John has 2 sons, called Paul and Joe". So, this implementation is not flexible, and therefore not generic, and not scientific.

The field of AI and NLP is "inspired by nature". But it has no foundation in nature. Therefore, this field is limited to deliver specific solutions to specific problems (=engineering), like Excel implementation mentioned above. However, this challenge is about uplifting this field of engineering towards a science, by developing generic solutions, based on a foundation in nature, like I am developing:

My fundamental approach shows that verb "has/have" is complementary to verb "is/are". So, verb "has/have" can also be used in predicate logic. In order to utilize the naturally intelligent function of non-keywords (structure words), I have defined natural intelligence first. Then I have identified a few (Natural Laws of) Intelligence in embedded in Grammar. And by implementing these laws of nature as a set of structuring algorithms is my system able to structure the knowledge of the system autonomously.

The rules of this challenge

- There are 6 blocks to beat the most basic techniques of my system. Your implementation should deliver the results of at least one block listed below;
- Your implementation should not have any prior knowledge. Instead, it should derive its knowledge from the input sentences of the examples listed below, from natural language, through an algorithm, to natural language;
- Preferable: The nouns and proper nouns of the listed examples are unknown upfront. (I use grammar definitions and an algorithm instead of a words list);
- Your implementation should be implemented as generic as can be, in such a way that all examples of this challenge can be integrated into one single system. The reasoning screen shots of my CNL reasoner illustrate how multiple reasoning constructions reinforce each other. The Screen shots of this challenge – which are added at the end of this document – show the execution by my software of the examples listed below;
- Your implementation should be published as open source software, so that its functionality is transparent. My software is open source too;
- Your implementation should be accepted by a scientific committee (conference or journal);
- In case your results are slightly different, you need to explain why you have chosen differently;
- It is an on-going challenge, until my system has been beaten on all blocks;
- I am the jury.

Your rewards

- A small gesture from me: € 250 for each scientifically accepted block;
- You will be the first one to have described in a scientifically accepted way, the logic of language that I have discovered.

You can contact me via the contact page of my website, or via LinkedIn.

Block 1: Direct conversions

The algebra of language listed below, applies structure words: “is”, “has”, “called”, “of”, “every” and “part of”.

“{proper noun 1} is {indefinite article + singular noun} of {proper noun 2}”
equals to
“{proper noun 2} has {indefinite article + singular noun} called {proper noun 1}”

“Every {singular noun 1} has {indefinite article + singular noun 2}”
equals to
“{indefinite article + singular noun 2} is part of every {singular noun 1}”

Examples:

> Given: “Paul is a son of John.”
•
• Generated conclusion:
< “John has a son, called Paul.”
>
> Given: “Anna has a daughter, called Laura.”
•
• Generated conclusion:
< “Laura is a daughter of Anna.”
>
> Given: “Every car has an engine.”
•
• Generated conclusion:
< “An engine is part of every car.”
>
> Given: “A sail is part of every sailboat.”
•
• Generated conclusion:
< “Every sailboat has a sail.”

Block 2: Indirect conversions

The algebra of language listed below, applies structure words: "is", "are", "has", "and", "every" and "part of".

"Every {singular noun 1} has {indefinite article + singular noun 2} and {indefinite article + singular noun 3}"

from which can be concluded

"{indefinite article + singular noun 2} and {indefinite article + singular noun 3} are part of every {singular noun 1}"

"Every {singular noun 1} has {indefinite article + singular noun 2} and {indefinite article + singular noun 3}"

and

"{proper noun} is {indefinite article + singular noun 2 or 3}"

from which can be concluded

"{proper noun} is part of {indefinite article + singular noun 1}"

"Every {singular noun 1} has {indefinite article + singular noun 2} and {indefinite article + singular noun 3}"

and

"{proper noun} is {indefinite article + singular noun 2}"

from which can be assumed

"{proper noun} has probably {indefinite article + singular noun 3}"

"Every {singular noun 1} has {indefinite article + singular noun 2} and {indefinite article + singular noun 3}"

and

"{proper noun} is {indefinite article + singular noun 3}"

from which can be assumed

"{proper noun} has probably {indefinite article + singular noun 2}"

"Every {singular noun 1} has {indefinite article + singular noun 2} and {indefinite article + singular noun 3}"

and

"{proper noun} has {indefinite article + singular noun 2 or 3}"

from which can be assumed

"{proper noun} is probably part of {indefinite article + singular noun 1}"

"Every {singular noun 1} has {indefinite article + singular noun 2} and {indefinite article + singular noun 3}"

and

"{proper noun} has {indefinite article + singular noun 2}"

from which can be assumed

"{proper noun} is probably {indefinite article + singular noun 3}"

"Every {singular noun 1} has {indefinite article + singular noun 2} and {indefinite article + singular noun 3}"

and

"{proper noun} has {indefinite article + singular noun 3}"

from which can be assumed

"{proper noun} is probably {indefinite article + singular noun 2}"

Examples:

> Given: “Every family has a parent and a child.”
•
• Generated conclusion:
< “A parent and a child are part of every family.”
>
> Given: “Michael is a parent.”
•
• Generated conclusion:
< “Michael is part of a family.”
•
• Generated assumption:
< “Michael has probably a child.”
>
> Given: “Adam is a child.”
•
• Generated conclusion:
< “Adam is part of a family.”
•
• Generated assumption:
< “Adam has probably a parent.”
>
> Given: “Peter has a parent.”
•
• Generated assumptions:
< “Peter is probably a child.”
< “Peter is probably part of a family.”
>
> Given: “Ronald has a child.”
•
• Generated assumptions:
< “Ronald is probably a parent.”
< “Ronald is probably part of a family.”

Block 3: Grouping of knowledge

The algebra of language listed below, applies structure words: “is”, “has”, “of”, “and” and “called”.

“{proper noun 1} is {indefinite article + singular noun} of {proper noun 2}”

equals to

“{proper noun 2} has {indefinite article + singular noun} called {proper noun 1}”

“{proper noun 1} has {indefinite article + singular noun 1} called {proper noun 2}”

and

“{proper noun 1} has {indefinite article + singular noun 1} called {proper noun 3}”

equals to

“{proper noun 1} has {number: 2} {plural form of singular noun 1} called {proper noun 2} and {proper noun 3}”

Examples:

> Given: “John is a parent of Paul.”
•
• Generated conclusion:
< “Paul has a parent, called John.”
>
> Given: “Anna is a parent of Paul.”
•
• Generated conclusion:
< “Paul has 2 parent [plural of 'parent' is unknown], called John and Anna.”
>
> Given: “Paul has 2 parents, called John and Anna.”
•
• Detected that the generated conclusion is confirmed:
< “Paul has 2 parent [plural of 'parent' is unknown], called John and Anna.”
•
• Detected: You have Given plural noun “parents”, which was unknown to me.

Block 4: Past tense reasoning

The algebra of language listed below, applies structure words: “the”, “was”, “has”, “had”, “of”, “called” and “no […] anymore”.

“{proper noun 1} was {**definite** article + singular noun} of {proper noun 2}”

from which can be concluded

“{proper noun 2} has no {singular noun} anymore”

“{proper noun 1} was {**definite** or indefinite article + singular noun} of {proper noun 2}”

from which can be concluded

“{proper noun 2} had {indefinite article + singular noun} called {proper noun 1}”.

“Every {singular noun 1} is {indefinite article + singular noun 2}”

and

“{proper noun} was {indefinite article + singular noun 1}” [1]

from which can be concluded

“{proper noun} was {indefinite article + singular noun 2}”

Examples:

> Given: “James was the father of Peter.”
•
• Generated conclusions:
< “Peter has no father anymore.”
< “Peter had a father, called James.”
>
> Given: “Every father is a man.”
•
• Generated conclusion:
< “James was a man.”

1 Sentence “James was the father of Peter” must be recognized as “James was a father”.

Block 5: Detection of a conflict and generation of a question

The algebra of language listed below, applies structure words: "is", "or", "and", "every" and "not".

"Every {singular noun 1} is {indefinite article + singular noun 2} **or** {indefinite article + singular noun 3}"

conflicts with

"{proper noun 1} is {indefinite article + singular noun 2} **and** {indefinite article + singular noun 3}

"Every {singular noun 1} is {indefinite article + singular noun 2} or {indefinite article + singular noun 3}"

and

"{proper noun 1} is {indefinite article + singular noun 1}"

from which can be concluded

"{proper noun 1} is {indefinite article + singular noun 2} or {indefinite article + singular noun 3}"

"{proper noun 1} is {indefinite article + singular noun 2} or {indefinite article + singular noun 3}"

equals to

"Is {proper noun 1} {indefinite article + singular noun 2} or {indefinite article + singular noun 3}?"

"Is {proper noun 1} {indefinite article + singular noun 1} or {indefinite article + singular noun 2}?"

and

"{proper noun 1} is not {indefinite article + singular noun 1}"

from which can be concluded

"{proper noun 1} is {indefinite article + singular noun 2}"

“Is {proper noun 1} {indefinite article + singular noun 1} or {indefinite article + singular noun 2}?”

and

“{proper noun 1} is not {indefinite article + singular noun 2}”

from which can be concluded

“{proper noun 1} is {indefinite article + singular noun 1}”

Examples:

> Given: “Every person is a man or a woman.”
> Given: “Addison is a man and a woman.”
!
! Detected conflict: This sentence is not accepted, because it is in conflict with itself.
!
> Given: “Addison is a person.”
•
• Generated question:
< “Is Addison a man or a woman?”
>
> Given: “Addison is not a woman.”
•
• Detected that the generated question has been answered:
< “Is Addison a man or a woman?”
•
• Generated conclusion:
< “Addison is a man.”

Block 6: Archiving of knowledge

The algebra of language listed below, applies structure words: “the”, “is”, “has”, “of”, “called”, “previous” and “new”.

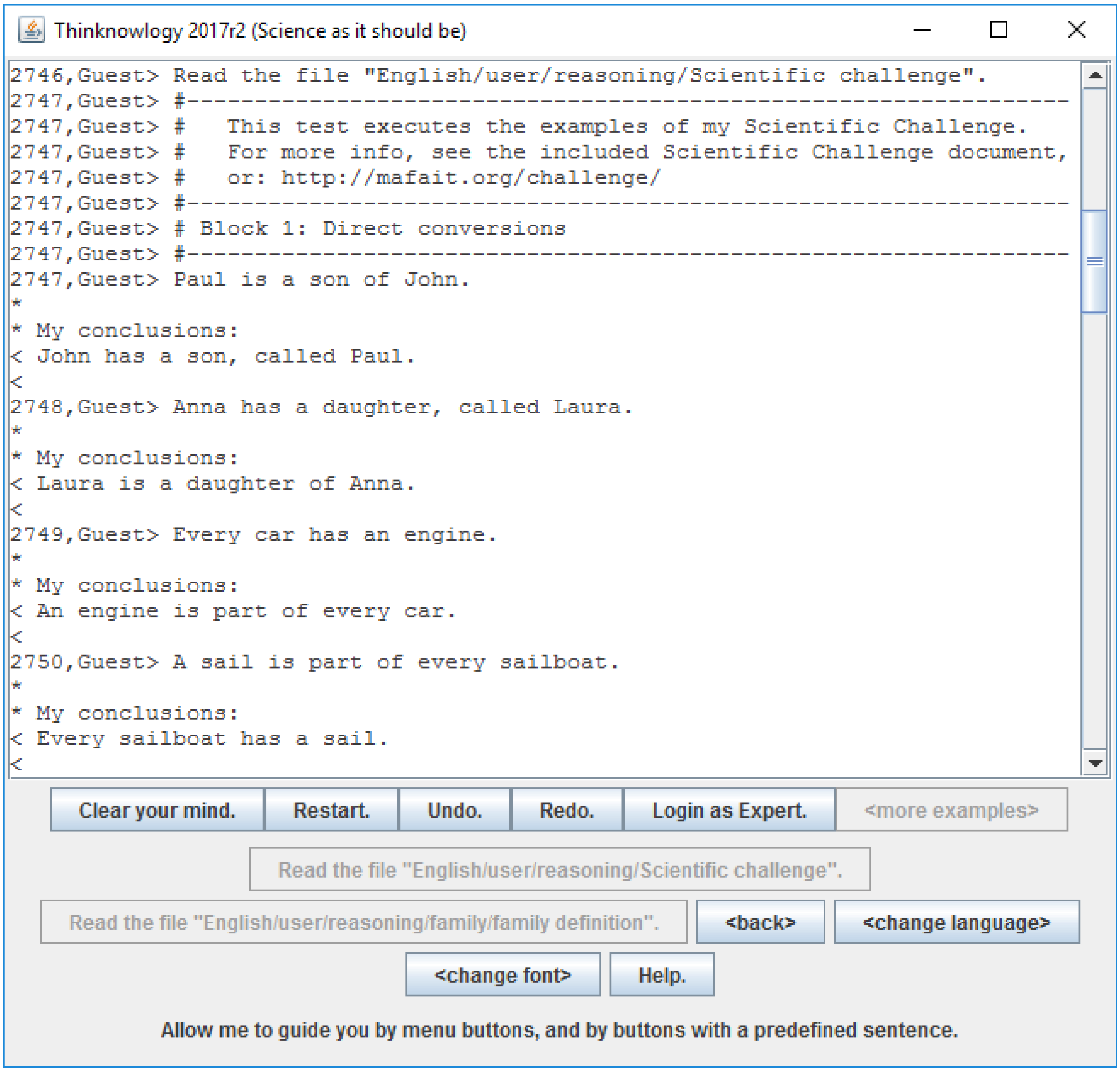
“{proper noun 1} is {**definite** article + singular noun} of {proper noun 2}”
from which can be concluded
“{proper noun 2} has {indefinite article + singular noun} called {proper noun 1}”

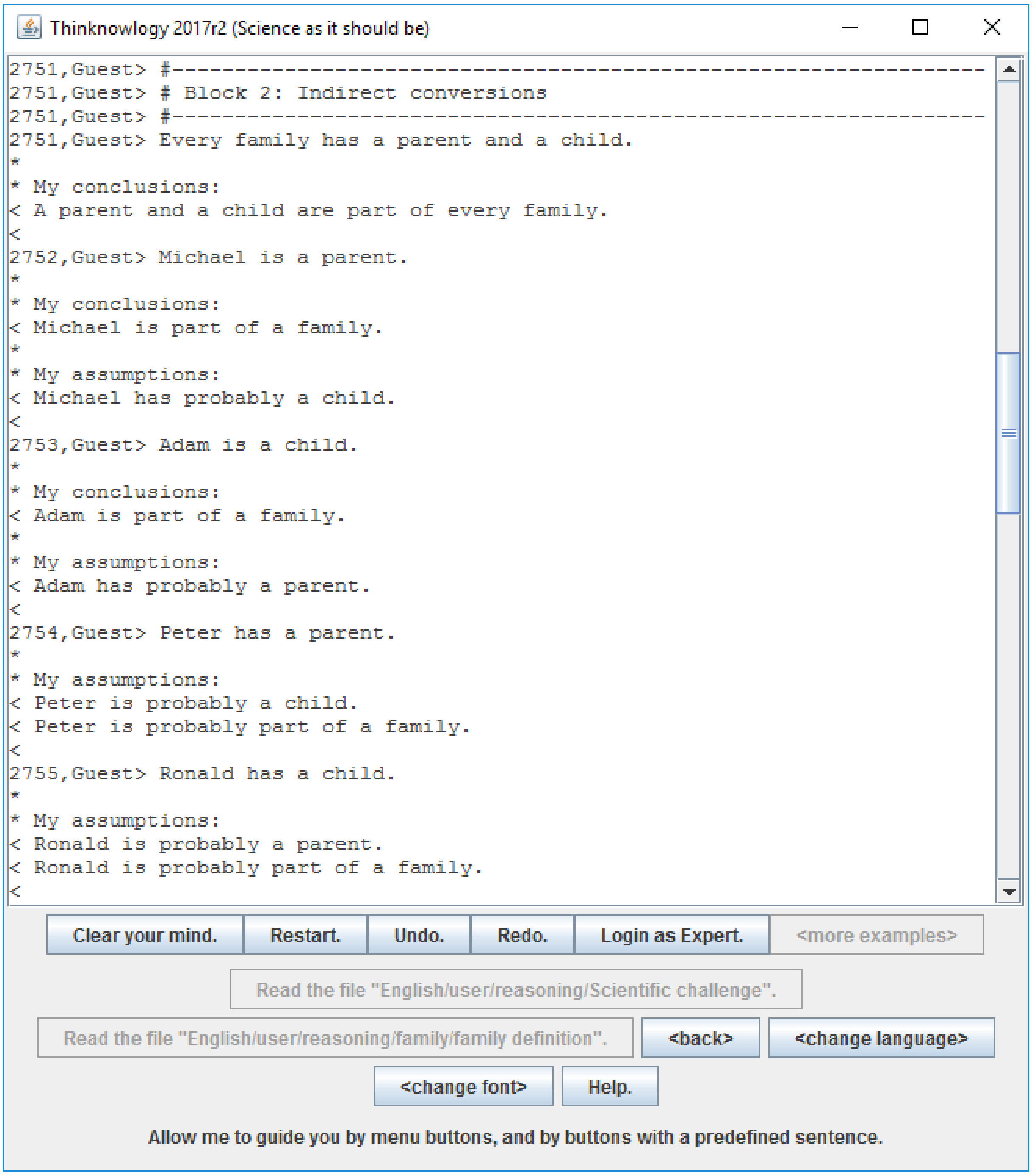
“{proper noun 1} is {**definite** article + singular noun} of {proper noun 2}”
followed by
“{proper noun 3} is {**definite** article + singular noun} of {proper noun 2}”
from which can be concluded
“{proper noun 2} has {indefinite article} new {singular noun} called {proper noun 3}”
and
“{proper noun 2} has {indefinite article} previous {singular noun} called {proper noun 1}”

Examples:

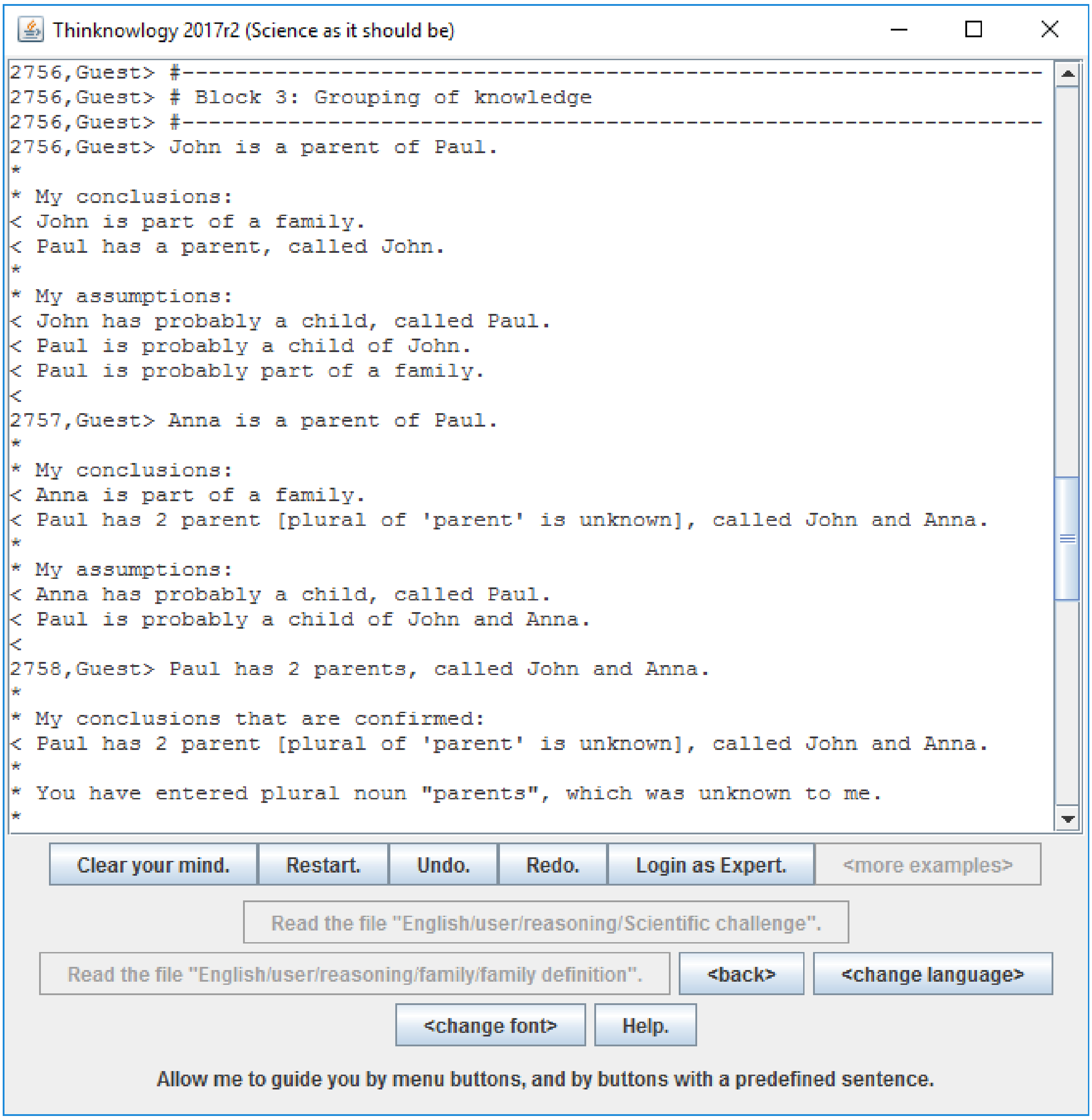
> Given: “Barack Obama is the president of the United States.”
•
• Generated conclusion:
< “The United States has a president, called Barack Obama.”
>
> Given: “Donald Trump is the president of the United States.”
•
• Generated conclusions:
< “The United States has a new president, called Donald Trump.”
< “The United States has a previous president, called Barack Obama.”

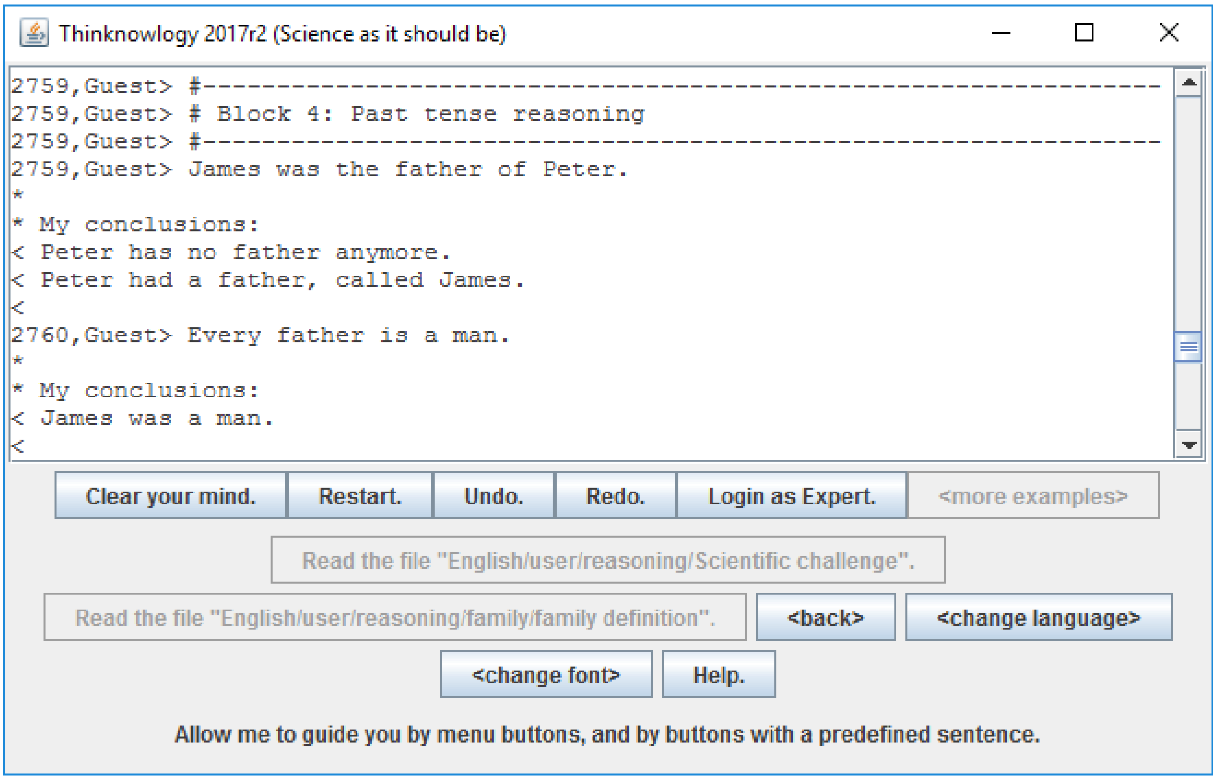
Screen shots of this challenge

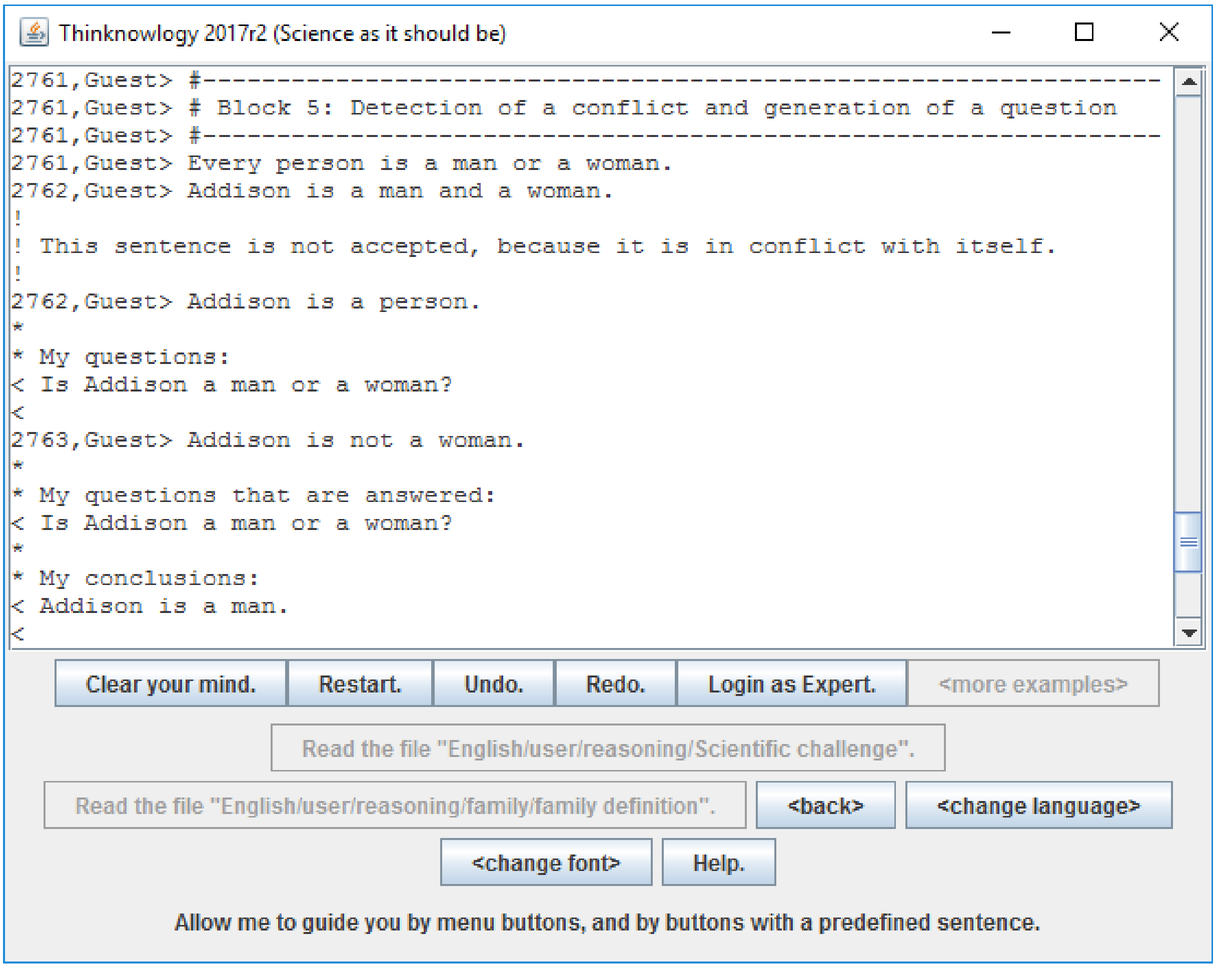


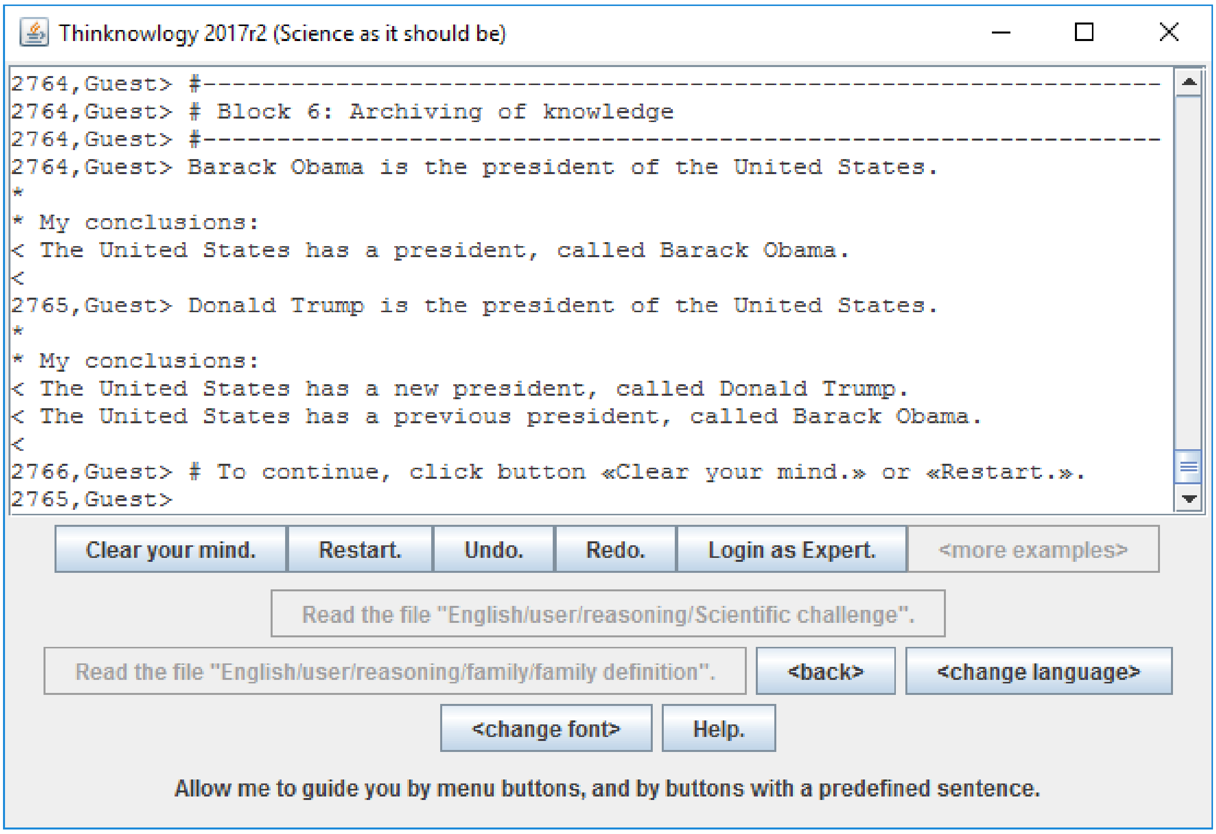


Thinknowlogy 2017r2 (Science as it should be)
2751,Guest> #---
2751,Guest> # Block 2: Indirect conversions
2751,Guest> #---
2751,Guest> Every family has a parent and a child.
*
* My conclusions:
< A parent and a child are part of every family.
<
2752,Guest> Michael is a parent.
*
* My conclusions:
< Michael is part of a family.
*
* My assumptions:
< Michael has probably a child.
<
2753,Guest> Adam is a child.
*
* My conclusions:
< Adam is part of a family.
*
* My assumptions:
< Adam has probably a parent.
<
2754,Guest> Peter has a parent.
*
* My assumptions:
< Peter is probably a child.
< Peter is probably part of a family.
<
2755,Guest> Ronald has a child.
*
* My assumptions:
< Ronald is probably a parent.
< Ronald is probably part of a family.
<
Clear your mind.
Restart.
Undo.
Redo.
Login as Expert.
<more examples>
Read the file "English/user/reasoning/Scientific challenge".
Read the file "English/user/reasoning/family/family definition".
<back>
<change language>
<change font>
Help.
Allow me to guide you by menu buttons, and by buttons with a predefined sentence.


Thinknowlogy 2017r2 (Science as it should be)
2756,Guest> #---
2756,Guest> # Block 3: Grouping of knowledge
2756,Guest> #---
2756,Guest> John is a parent of Paul.
*
* My conclusions:
< John is part of a family.
< Paul has a parent, called John.
*
* My assumptions:
< John has probably a child, called Paul.
< Paul is probably a child of John.
< Paul is probably part of a family.
<
2757,Guest> Anna is a parent of Paul.
*
* My conclusions:
< Anna is part of a family.
< Paul has 2 parent [plural of 'parent' is unknown], called John and Anna.
*
* My assumptions:
< Anna has probably a child, called Paul.
< Paul is probably a child of John and Anna.
<
2758,Guest> Paul has 2 parents, called John and Anna.
*
* My conclusions that are confirmed:
< Paul has 2 parent [plural of 'parent' is unknown], called John and Anna.
*
* You have entered plural noun "parents", which was unknown to me.
*
Clear your mind.
Restart.
Undo.
Redo.
Login as Expert.
<more examples>
Read the file "English/user/reasoning/Scientific challenge".
Read the file "English/user/reasoning/family/family definition".
<back>
<change language>
<change font>
Help.
Allow me to guide you by menu buttons, and by buttons with a predefined sentence.


Thinknowlogy 2017r2 (Science as it should be)
2759,Guest> #--
2759,Guest> # Block 4: Past tense reasoning
2759,Guest> #--
2759,Guest> James was the father of Peter.
*
* My conclusions:
< Peter has no father anymore.
< Peter had a father, called James.
<
2760,Guest> Every father is a man.
*
* My conclusions:
< James was a man.
<
Clear your mind.
Restart.
Undo.
Redo.
Login as Expert.
<more examples>
Read the file "English/user/reasoning/Scientific challenge".
Read the file "English/user/reasoning/family/family definition".
<back>
<change language>
<change font>
Help.
Allow me to guide you by menu buttons, and by buttons with a predefined sentence.


Thinknowlogy 2017r2 (Science as it should be)
2761,Guest> #--
2761,Guest> # Block 5: Detection of a conflict and generation of a question
2761,Guest> #--
2761,Guest> Every person is a man or a woman.
2762,Guest> Addison is a man and a woman.
!
! This sentence is not accepted, because it is in conflict with itself.
!
2762,Guest> Addison is a person.
*
* My questions:
< Is Addison a man or a woman?
<
2763,Guest> Addison is not a woman.
*
* My questions that are answered:
< Is Addison a man or a woman?
*
* My conclusions:
< Addison is a man.
<
Clear your mind.
Restart.
Undo.
Redo.
Login as Expert.
<more examples>
Read the file "English/user/reasoning/Scientific challenge".
Read the file "English/user/reasoning/family/family definition".
<back>
<change language>
<change font>
Help.
Allow me to guide you by menu buttons, and by buttons with a predefined sentence.


Thinknowlogy 2017r2 (Science as it should be)
2764,Guest> #---
2764,Guest> # Block 6: Archiving of knowledge
2764,Guest> #---
2764,Guest> Barack Obama is the president of the United States.
*
* My conclusions:
< The United States has a president, called Barack Obama.
<
2765,Guest> Donald Trump is the president of the United States.
*
* My conclusions:
< The United States has a new president, called Donald Trump.
< The United States has a previous president, called Barack Obama.
<
2766,Guest> # To continue, click button «Clear your mind.» or «Restart.».
2765,Guest>
Clear your mind.
Restart.
Undo.
Redo.
Login as Expert.
<more examples>
Read the file "English/user/reasoning/Scientific challenge".
Read the file "English/user/reasoning/family/family definition".
<back>
<change language>
<change font>
Help.
Allow me to guide you by menu buttons, and by buttons with a predefined sentence.