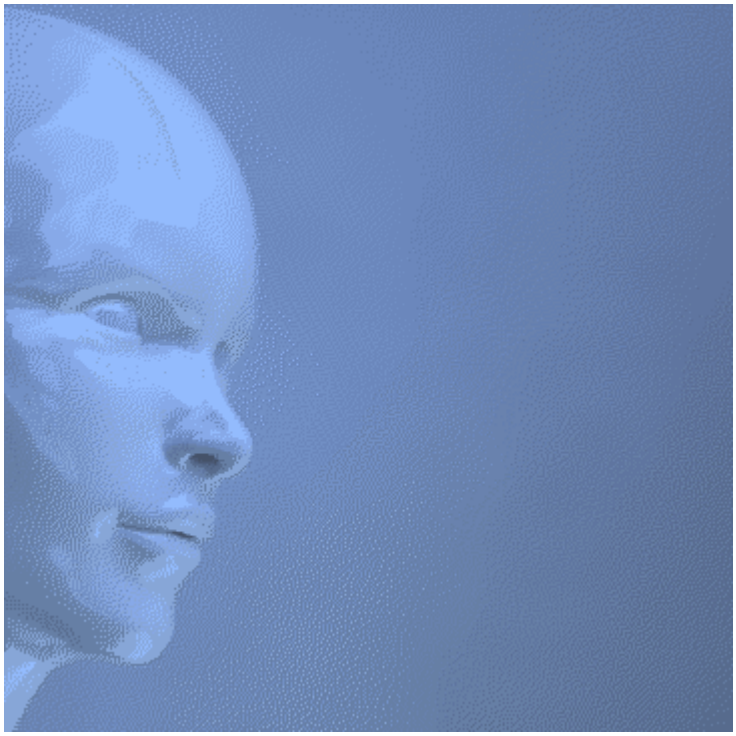
Scientific challenge:
Beat my natural language reasoner



Problem description

Knowledge technology is based on applying smart algorithms to keywords, by which the natural meaning of non-keywords is ignored.

Non-keywords (structure words) provide information to our brain about the structure of the sentence. But by ignoring the natural meaning (natural structure) of non-keywords (structure words), the field of knowledge technology got stuck with “bags of keywords” and unstructured texts. Even IBM's Watson is unable to organize its knowledge base automatically. Instead, Watson uses brute force “*to find a needle in the haystack of unstructured texts*” (quote of IBM). However, if Watson was able to organize its knowledge base automatically, a search would take significantly less processing power.

The core of the problem

We only know very little of the logic of language. Actually, our understanding of the logic of language is limited to verb “is/are”, like in:

> Given: “Every father **is** a man.”
> Given: “John **is** a father.”
•
• Logical conclusion:
< “John **is** a man.”.

As a consequence of this limited understanding, there is no algebra defined to describe for example past tense logic (see Block 3), or to describe a sentence containing possessive verb “has/have”, like: “John **has** a son, called Paul” (see Block 1). So, there is also no algebra defined to draw the following logical conclusion in a generic way. Let alone, expressed in natural language:

> Given: “John **has** three apples.”
> Given: “Paul **has** four apples.”
•
• Logical conclusion:
< “John and Paul **have** seven apples (together).”.

When we encounter a logical problem containing possessive verb “has/have”, like in the previous example, we have have been unconsciously taught to convert it into an “is/are” problem, like: “Three apples and four apples **are** seven apples (together)”.

Besides possessive verb “has/have”, scientists don't understand the logical (intelligent, self-organizing) function of for example definite article “the”, and past tense verbs, like “were” and “had”. And they only partly understand the logical (intelligent, self-organizing) function of conjunction “or”. (They don't understand that conjunction “or” represents a logical *Exclusive OR* function in natural language, and that a logical *Inclusive OR* function is expressed in natural language as “… and/or …”.)

Generally accepted workaround

The generally accepted workaround in the field of Artificial Intelligence (AI) and knowledge technology, 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 you need 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” in a generic way to “John **has** a son, called Paul”, 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).

Dietrich Hansköttter, student at the Stuttgart University, made an attempt to solve this problem. With his permission, below 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. And the logic needs to be copied with each new language, while a generic solution has one logical implementation. Moreover, this implementation isn't flexible – and therefore not generic – because it can't be expanded to process multiple specifications words, like in: “Paul is a son of John and Anna” or “John has 2 sons, called Paul and Joe”.

The field of AI and knowledge technology has no foundation. It is therefore only able to produce specific solutions to specific problems (engineering), like this solution of Dietrich. 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” is complementary to verb “is”. So, verb “has” can also be used in predicate logic. In order to utilize the natural meaning (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 5 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, the system should derive its knowledge directly from the input sentences listed below. Preferable: The nouns and proper nouns of the listed examples are unknown upfront;
• 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 screenshots of my natural language reasoner illustrate how multiple reasoning constructions reinforce each other. And the Screenshots of this challenge – included 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 also open source;
• 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. (If you don’t like me being the jury, feel free to have your solution scientifically accepted without my approval.)

Your rewards

• A small gesture from me: € 200 for each block;
• You will be the first one to have these results implemented in a scientifically accepted way, because I didn't use any technique known in this field.

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

Block 1

Required *Linguistic algebra*, which 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}”

Implementation:

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

Block 2

Required *Linguistic algebra*, which 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}”

Implementation:

> Entered: “John is a parent of Paul.”
•
• Generated conclusion:
< “Paul has a parent, called John.”
>
> Entered: “Anna is a parent of Paul.”
•
• Generated conclusion:
< “Paul has 2 parent [plural of 'parent' is unknown], called John and Anna.”
>
> Entered: “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 entered plural noun “parents”, which was unknown to me.

Block 3

Required *Linguistic algebra*, which applies structure words “was”, “has”, “had”, “of”, “called” and “no […] anymore”:

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

equals to

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

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

equals to

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

Implementation:

> Entered: “James was the father of Peter.”

•

• Generated conclusions:

< “Peter has no father anymore.”

< “Peter had a father, called James.”

Block 4

Required *Linguistic algebra*, which 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}”

Implementation:

> Entered: “Every person is a man or a woman.”
> Entered: “Addison is a man and a woman.”
!
! Detected conflict: This sentence is not accepted, because it is in conflict with itself.
!
> Entered: “Addison is a person.”
•
• Generated question:
< “Is Addison a man or a woman?”
>
> Entered: “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 5

Required *Linguistic algebra*, which applies structure words “is”, “has”, “of”, “the”, “called”, “previous” and “new”:

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

equals to

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

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

and

“{proper noun 3} is {**definite** article + singular noun} of {proper noun 2}”

equals to

“{proper noun 2} has {indefinite article} new {singular noun} called {proper noun 3}”

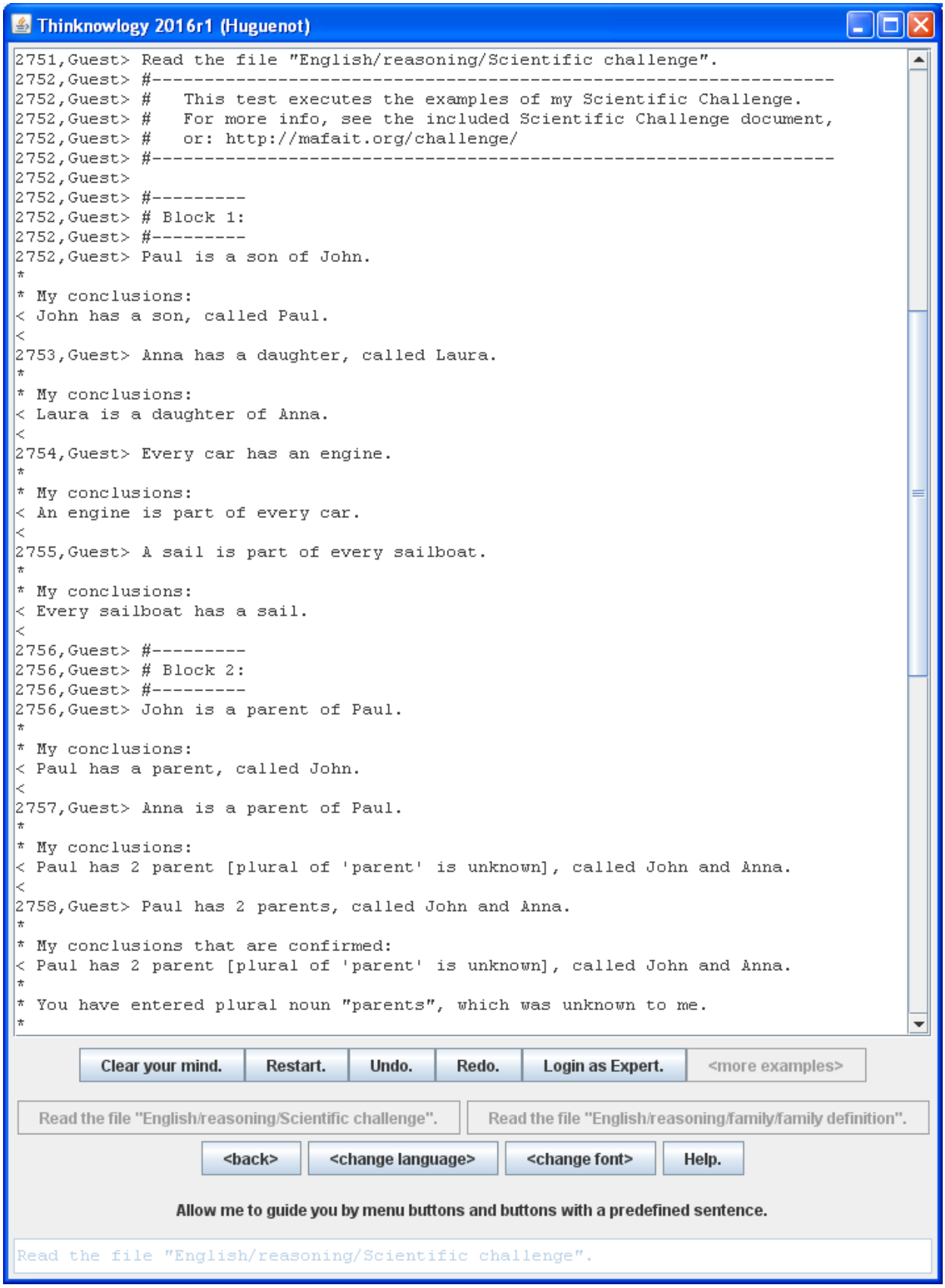
and

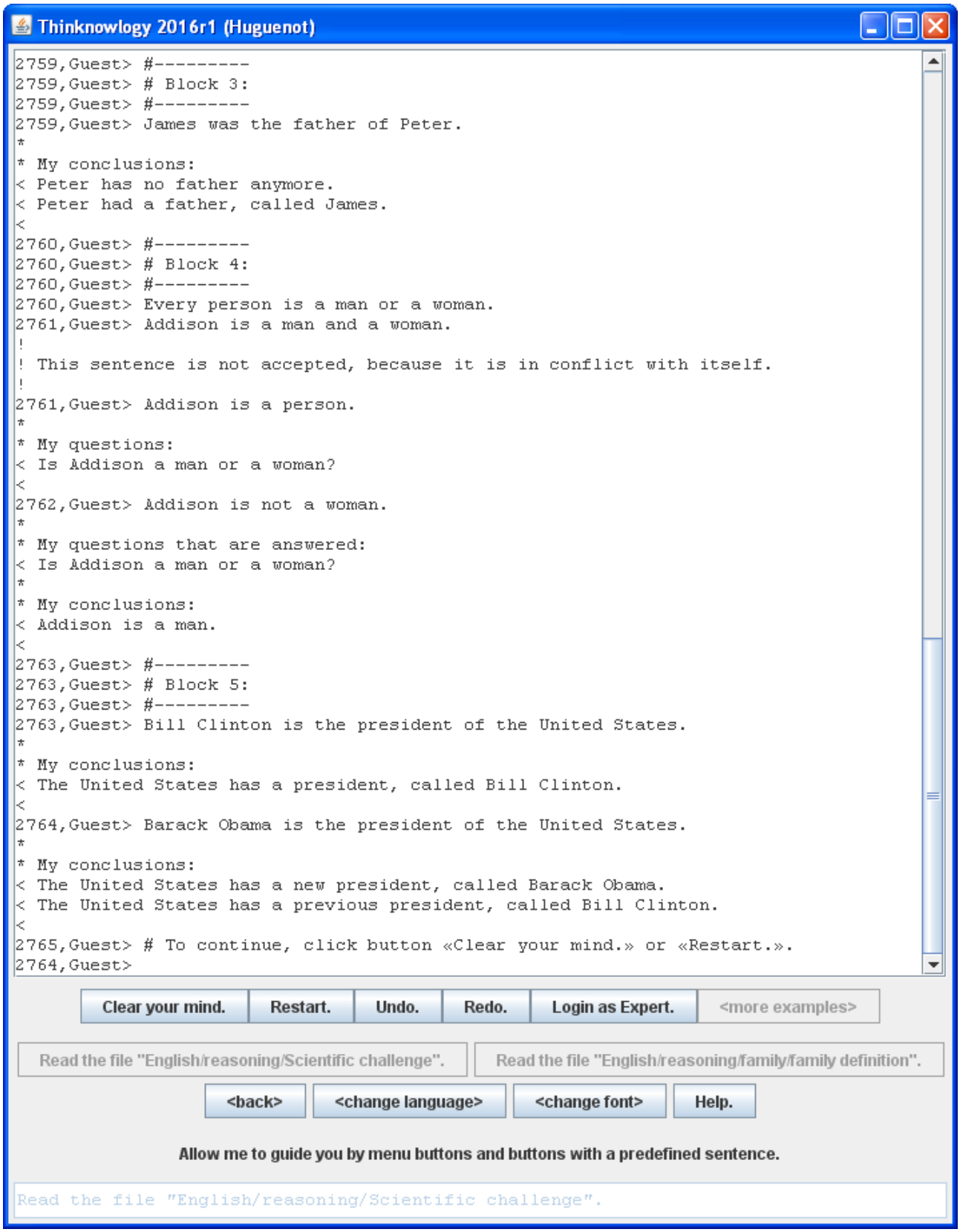
“{proper noun 2} has {indefinite article} previous {singular noun} called {proper noun 1}”

Implementation:

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

Screenshots of this challenge




Thinknowlogy 2016r1 (Huguenot)
2759,Guest> #---------
2759,Guest> # Block 3:
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> #---------
2760,Guest> # Block 4:
2760,Guest> #---------
2760,Guest> Every person is a man or a woman.
2761,Guest> Addison is a man and a woman.
!
! This sentence is not accepted, because it is in conflict with itself.
!
2761,Guest> Addison is a person.
*
* My questions:
< Is Addison a man or a woman?
<
2762,Guest> Addison is not a woman.
*
* My questions that are answered:
< Is Addison a man or a woman?
*
* My conclusions:
< Addison is a man.
<
2763,Guest> #---------
2763,Guest> # Block 5:
2763,Guest> #---------
2763,Guest> Bill Clinton is the president of the United States.
*
* My conclusions:
< The United States has a president, called Bill Clinton.
<
2764,Guest> Barack Obama is the president of the United States.
*
* My conclusions:
< The United States has a new president, called Barack Obama.
< The United States has a previous president, called Bill Clinton.
<
2765,Guest> # To continue, click button «Clear your mind.» or «Restart.».
2764,Guest>
Clear your mind.
Restart.
Undo.
Redo.
Login as Expert.
<more examples>
Read the file "English/reasoning/Scientific challenge".
Read the file "English/reasoning/family/family definition".
<back>
<change language>
<change font>
Help.
Allow me to guide you by menu buttons and buttons with a predefined sentence.
Read the file "English/reasoning/Scientific challenge".