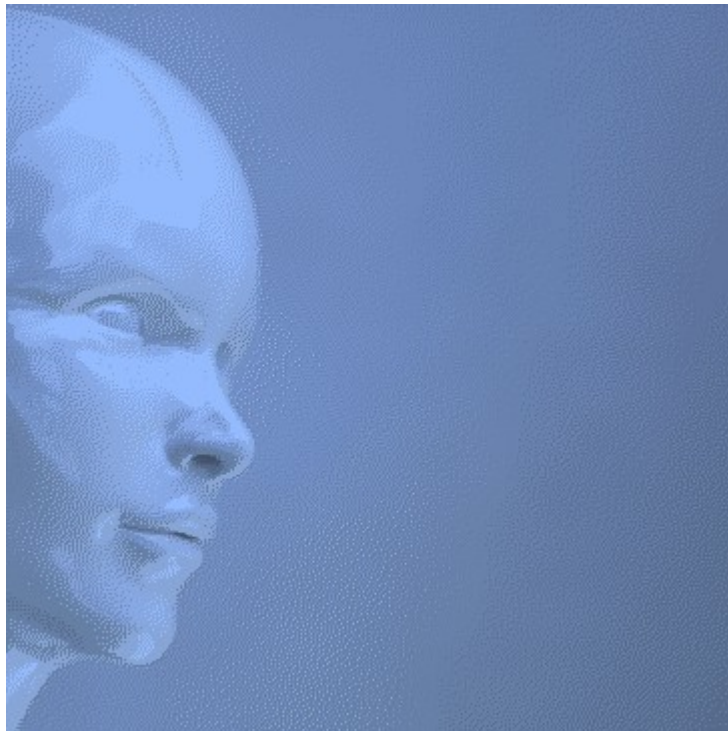


Wetenschappelijke uitdaging:

Versla de eenvoudigste resultaten van mijn CNL reasoner
(Controlled Natural Language redeneersysteem)



Inleiding

Zelfstandig redeneren vereist zowel [natuurlijke intelligentie](#) als natuurlijke taal. Zonder het te weten, paste [Aristoteles](#) grofweg 2.400 jaar geleden natuurlijke intelligentie toe op natuurlijke taal:

- > Gegeven: “[Alle mensen zijn sterfelijk.](#)”
- > Gegeven: “[Socrates is een mens.](#)”
-
- Logische conclusie:
- < “[Socrates is sterfelijk.](#)”

Grofweg 200 jaar geleden zijn dergelijk redeneerconstructies geformaliseerd in de [Predicaten-logica](#). En sinds het begin van deze eeuw zijn deze redeneerconstructies in software geïmplementeerd als [Controlled Natural Language](#) (CNL) redeneersystemen. CNL redeneersystemen zijn instaat om zelfstandig nieuwe kennis af te leiden van tot dan toe onbekende kennis, en om de afgeleide kennis uit te drukken in leesbare zinnen (met een beperkte grammatica).

Probleembeschrijving 1: Redeneren in de verleden tijd

Het bovengenoemde redeneervoorbeeld was waar gedurende het leven van [Socrates](#). Maar nu, na het ultieme bewijs van zijn sterfelijkheid – zijn dood in het jaar 399 v.Chr. – dienen we de verleden-tijdsvorm te gebruiken:

- > Gegeven: “[Alle mensen zijn sterfelijk.](#)”
- > Gegeven: “[Socrates was een mens.](#)”
-
- Logische conclusie:
- < “[Socrates was sterfelijk.](#)”

De tijdsvorm waarin een werkwoord staat, vertelt ons de status van de betrokken bewering:

- “[Socrates is een mens](#)” vertelt ons dat Socrates leeft;
- “[Socrates was een mens](#)” vertelt ons dat Socrates niet meer onder de levenden is.

Met betrekking tot de conclusie:

- “[Socrates is sterfelijk](#)” vertelt ons dat de dood van Socrates onvermijdelijk is, maar dat er nog geen hard bewijs voor zijn sterfelijkheid is;
- “[Socrates was sterfelijk](#)” vertelt ons dat zijn sterfelijkheid bewezen is.

In de afgelopen 2.400 jaar zijn wetenschappers 'vergeten' om algebra te definiëren voor de verleden tijd. Redeneren in de verleden tijd is dus in geen enkel wetenschappelijk artikel beschreven, terwijl het wel geïmplementeerd is in mijn CNL redeneersysteem.

Probleembeschrijving 2: Bezittelijk redeneren

Naast redeneren in de verleden tijd, wordt ook bezittelijk redeneren – redeneren met bezittelijk werkwoord “**heeft / hebben**” – niet ondersteund door de predicaatenlogica (algebra). Voorbeeld:

> Gegeven: “**Paul is een zoon van Jan.**”

•

• Logische conclusie:

< “**Jan heeft een zoon, genaamd Paul.**”

Of andersom:

> Gegeven: “**Jan heeft een zoon, genaamd Paul.**”

•

• Logische conclusie:

< “**Paul is een zoon van Jan.**”

Dus, waarom ondersteunt predicaatenlogica (algebra) niet op een natuurlijke manier het redeneren in de verleden tijd, noch het bezittelijk redeneren? Waarom moet elk predicaat met een ander werkwoord dan “**is/zijn**” in de tegenwoordige tijd, op een kunstmatige manier beschreven worden, zoals **heeft_zoon(jan,paul)**? Waarom is algebra nog steeds niet uitgerust voor natuurlijke taal, na eeuwen van wetenschappelijk onderzoek?

Probleembeschrijving 3: Genereren van vragen

Er is nog meer: Alhoewel algebra de **Exclusieve OF** (XOR) functie op een natuurlijke wijze beschrijft, implementeren CNL redeneersystemen nog steeds niet de taalkundige equivalent van deze functie, nl. voegwoord “**of**”. CNL redeneersystemen zijn daardoor niet in staat om de volgende vraag te genereren:

> Gegeven: “**Ieder persoon is een man of een vrouw.**”

> Gegeven: “**Anne is een persoon.**”

•

• Logische vraag:

< “**Is Anne een man of een vrouw?**”

Dus 2.400 jaar na [Aristoteles](#), begrijpen wetenschappers nog steeds niet de basisprincipes van natuurlijke intelligentie en natuurlijke taal:

Woorden zoals bepaald lidwoord “**de/het**” (zie Blok 6), voegwoord “**of**” (zie Blok 5), bezittelijk werkwoord “**heeft/hebben**” (zie Blok 1, Blok 2 en Blok 3) en werkwoorden in de verleden tijd “**was/waren**” en “**had/hadden**” (zie Blok 4) hebben een natuurlijk-intelligente functie in de taal.

Algemeen aanvaarde *workaround*

De algemeen aanvaarde *workaround* in het vakgebied Kunstmatige Intelligentie (AI) en kennistechnologie (NLP) is om kennis met het werkwoord “[hebben](#)” in te voeren, is om het direct in een redeneersysteem te programmeren, zoals: [heeft_zoon\(jan,paul\)](#). Dit is echter **geen** generieke oplossing (=wetenschap), maar een specifieke oplossing voor een specifiek probleem (=engineering), omdat elk afzonderlijk zelfstandig naamwoord direct in de redeneersysteem geprogrammeerd moet worden ([heeft_dochter](#), [heeft_vader](#), [heeft_moeder](#), et cetera), en voor iedere taal opnieuw. Met als gevolg dat er geen enkele techniek bestaat om een zin als “[Paul is een zoon van Jan](#)” op een generieke manier – van natuurlijke taal, via een algoritme, naar natuurlijke taal – om te zetten naar “[Jan heeft een zoon, genaamd Paul](#)”, waarbij zelfstandig naamwoord “[zoon](#)” en eigennamen “[Jan](#)” en “[Paul](#)” niet in het redeneersysteem geprogrammeerd hoeven te worden. Dit is slechts het eerste voorbeeld van deze uitdaging (zie Blok 1).

Ik ontving een bijdrage van een student, in een poging om dit probleem op te lossen. Met zijn toestemming, hieronder zijn Excel-implementatie voor de Engelse taal:

```
= 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))
```

Deze oplossing controleert niet op woordtype, zoals uitgelegd in paragraaf [2.3.4. The function of word types in reasoning](#) van [mijn fundamentele document](#). Daarnaast moet deze logica voor elke taal gekopieerd worden, terwijl een generieke oplossing slechts één logische implementatie heeft. Bovendien kan deze implementatie niet uitgebreid worden, bv. om meerdere specificatiewoorden te verwerken, zoals in: “[Paul is een zoon van Jan en Anna](#)” of “[Jan heeft 2 zoons, genaamd Paul en Johan](#)”. Deze implementatie is dus niet flexibel, en daarom ook niet generiek, en niet wetenschappelijk.

Het vakgebied AI en NLP is “geïnspireerd op de natuur”. Maar het heeft geen fundament in de natuur. Hierdoor is dit vakgebied beperkt tot het leveren van specifieke oplossingen voor specifieke problemen (=engineering), zoals de bovenstaande Excel-implementatie. Deze uitdaging gaat echter over het verheffen van dit vakgebied, van engineering tot wetenschap, dat generieke oplossingen biedt, gebaseerd op een natuurlijk fundament, zoals ik ontwikkel.

Mijn fundamentele benadering laat zien dat werkwoord “[heeft/hebben](#)” complementair is aan werkwoord “[is/zijn](#)”. Dus kan ook werkwoord “[heeft/hebben](#)” gebruikt worden in de predatenlogica. Om de natuurlijk-intelligente functie van structuurwoorden te kunnen benutten, heb ik eerst [natuurlijke intelligentie](#) gedefinieerd. Daarna heb ik enkele [\(Natuurwetten van\) Intelligentie ingebed in Grammatica](#) geïdentificeerd. En door deze natuurwetten als structureringsalgoritmen te implementeren, kan mijn systeem de kennis zelfstandig structureren.

De regels van deze uitdaging

- Er zijn 6 blokken waarin je de allereenvoudigste basistechnieken van mijn systeem kunt verslaan. Je implementatie dient de hieronder genoemde resultaten van tenminste één blok te leveren;
- Je implementatie mag vooraf geen kennis bevatten. In plaats daarvan dient het systeem de kennis te ontleen aan de invoerzinnen van de hieronder genoemde voorbeelden, van natuurlijke taal, via een algoritme, naar natuurlijke taal;
- Bij voorkeur moeten de gebruikte zelfstandige naamwoorden en eigennamen vooraf onbekend zijn. (Ik gebruik grammaticadefinities en een algoritme in plaats van een woordenlijst);
- Je implementatie dient zo generiek mogelijk te worden opgezet, zodat alle voorbeelden van deze uitdaging geïntegreerd kunnen worden tot één enkel systeem;
- De [redeneer-schermafdrucken](#) van mijn CNL redeneersysteem laten zien dat diverse redeneerconstructies elkaar versterken. De Schermafdrucken van deze uitdaging – die ook aan het einde van dit document zijn opgenomen – laten de uitvoer van mijn software zien van de onderstaande voorbeelden;
- Je implementatie dient te worden gepubliceerd als open source software, zodat de functionaliteit inzichtelijk is. [Ook mijn software is open source](#);
- Je implementatie moet aanvaard zijn door een wetenschappelijk comité (conferentie of journaal);
- In het geval dat je resultaten iets anders zijn, dien je uit te leggen waarom je daarvoor hebt gekozen;
- Deze wedstrijd gaat door totdat mijn systeem volledig is verslagen;
- Ik ben de jury.

Jouw beloning

- Een klein gebaar van mijn kant: € 250 per blok dat wetenschappelijk aanvaard is;
- Je bent de eerste die de logica van taal (die ik ontdekt heb) op een wetenschappelijk geaccepteerde manier heeft beschreven.

Je kunt me bereiken via de [contactpagina van mijn website](#), en via [LinkedIn](#).

Blok 1: Directe conversies

[De algebra van taal](#) die hieronder genoemd wordt, past de volgende structuurwoorden toe: “is”, “heeft”, “genaamd”, “ieder(e)” en “deel van”.

“{eigenaam 1} **is** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} **van** {eigenaam 2}”

is gelijkwaardig aan

“{eigenaam 2} **heeft** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} **genaamd** {eigenaam 1}”

“**Ieder(e)** {enkelvoudig zelfstandig naamwoord 1} **heeft** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}”

is gelijkwaardig aan

“{onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} **is deel van ieder(e)** {enkelvoudig zelfstandig naamwoord 1}”

Voorbeelden:

> Gegeven: “**Paul is een zoon van Jan.**”

•

• Gegenereerde conclusie:

< “**Jan heeft een zoon, genaamd Paul.**”

>

> Gegeven: “**Anna heeft een dochter, genaamd Laura.**”

•

• Gegenereerde conclusie:

< “**Laura is een dochter van Anna.**”

>

> Gegeven: “**Iedere auto heeft een motor.**”

•

• Gegenereerde conclusie:

< “**Een motor is deel van iedere auto.**”

>

> Gegeven: “**Een zeil is deel van iedere zeilboot.**”

•

• Gegenereerde conclusie:

< “**Iedere zeilboot heeft een zeil.**”

>

Blok 2: Indirecte conversies

[De algebra van taal](#) die hieronder genoemd wordt, past de volgende structuurwoorden toe: “is”, “zijn”, “heeft”, “en”, “ieder(e)” en “deel van”.

“[Ieder\(e\)](#) {enkelvoudig zelfstandig naamwoord 1} [heeft](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} [en](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

[waaruit kan worden geconcludeerd](#)

“{onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} [en](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3} [zijn deel van ieder\(e\)](#) {enkelvoudig zelfstandig naamwoord 1}”

“[Ieder\(e\)](#) {enkelvoudig zelfstandig naamwoord 1} [heeft](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} [en](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

[en](#)

“{eigenaam} [is](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2 or 3}”

[waaruit kan worden geconcludeerd](#)

“{eigenaam} [is deel van](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1}”

“[Ieder\(e\)](#) {enkelvoudig zelfstandig naamwoord 1} [heeft](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} [en](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

[en](#)

“{eigenaam} [is](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}”

[waaruit kan worden verondersteld](#)

“{eigenaam} [heeft waarschijnlijk](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

“[Ieder\(e\)](#) {enkelvoudig zelfstandig naamwoord 1} [heeft](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} [en](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

[en](#)

“{eigenaam} [is](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

[waaruit kan worden verondersteld](#)

“{eigenaam} [heeft waarschijnlijk](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}”

“Ieder(e) {enkelvoudig zelfstandig naamwoord 1} heeft {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} en {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

en

“{eigenaam} heeft {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2 or 3}”

waaruit kan worden verondersteld

“{eigenaam} is waarschijnlijk deel van {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1}”

“Ieder(e) {enkelvoudig zelfstandig naamwoord 1} heeft {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} en {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

en

“{eigenaam} heeft {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}”

waaruit kan worden verondersteld

“{eigenaam} is waarschijnlijk {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

“Ieder(e) {enkelvoudig zelfstandig naamwoord 1} heeft {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} en {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

and

“{eigenaam} heeft {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

waaruit kan worden verondersteld

“{eigenaam} is waarschijnlijk {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}”

Voorbeelden:

> Gegeven: “Ieder gezin heeft een ouder en een kind.”

•

• Gegenerateerde conclusie:

< “Een ouder en een kind zijn deel van ieder gezin.”

>

> Gegeven: “Sjors is een ouder.”

•

• Gegenerateerde conclusie:

< “Sjors is deel van een gezin.”

•

• Gegenerateerde aanname:

< “Sjors heeft waarschijnlijk een kind.”

>

> Gegeven: “André is een kind.”

•

• Gegenerateerde conclusie:

< “André is deel van een gezin.”

•

• Gegenerateerde aanname:

< “André heeft waarschijnlijk een ouder.”

>

> Gegeven: “Peter heeft een ouder.”

•

• Gegenerateerde aannames:

< “Peter is waarschijnlijk een kind.”

< “Peter is waarschijnlijk deel van een gezin.”

>

> Gegeven: “Willem heeft een kind.”

•

• Gegenerateerde aannames:

< “Willem is waarschijnlijk een ouder.”

< “Willem is waarschijnlijk deel van een gezin.”

>

Blok 3: Het groeperen van kennis

[De algebra van taal](#) die hieronder genoemd wordt, past de volgende structuurwoorden toe: “is”, “heeft”, “van”, “en” en “genaamd”.

“{eigenaam 1} **is** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} **van** {eigenaam 2}”

is gelijkwaardig aan

“{eigenaam 2} **heeft** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} **genaamd** {eigenaam 1}”

“{eigenaam 1} **heeft** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1} **genaamd** {eigenaam 2}”

en

“{eigenaam 1} **heeft** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1} **genaamd** {eigenaam 3}”

is gelijkwaardig aan

“{eigenaam 1} **heeft** {aantal: 2} {meervoudsvorm van enkelvoudig zelfstandig naamwoord 1} **genaamd** {eigenaam 2} **en** {eigenaam 3}”

Voorbeelden:

> Gegeven: “**Jan is een ouder van Paul.**”

•

• Gegenereerde conclusie:

< “**Paul heeft een ouder, genaamd Jan.**”

>

> Gegeven: “**Anna is een ouder van Paul.**”

•

• Gegenereerde conclusie:

< “**Paul heeft 2 ouder [meervoud van 'ouder' is onbekend], genaamd Jan en Anna.**”

>

> Gegeven: “**Paul heeft 2 ouders, genaamd Jan en Anna.**”

•

• Gedetecteerd dat de gegenereerde conclusie bevestigd is:

< “**Paul heeft 2 ouder [meervoud van 'ouder' is onbekend], genaamd Jan en Anna.**”

•

• Gedetecteerd: **U heeft meervoudig zelfstandig naamwoord “ouders” ingevoerd, welke mij onbekend was.**

Blok 4: Redeneren in de verleden tijd

[De algebra van taal](#) die hieronder genoemd wordt, past de volgende structuurwoorden toe: “was”, “heeft”, “had”, “van”, “genaamd” en “geen [...] meer”.

“{eigenaam 1} **was** {bepaald lidwoord + enkelvoudig zelfstandig naamwoord} **van** {eigenaam 2}”

waaruit kan worden geconcludeerd

“{eigenaam 2} **heeft geen** {enkelvoudig zelfstandig naamwoord} **meer**”

“{eigenaam 1} **was** {bepaald of onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} **van** {eigenaam 2}”

waaruit kan worden geconcludeerd

“{eigenaam 2} **had** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} **genaamd** {eigenaam 1}”

“**Ieder(e)** {enkelvoudig zelfstandig naamwoord 1} **is** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}”

en

“{eigenaam} **was** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1}”¹

waaruit kan worden geconcludeerd

“{eigenaam} **was** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}”

Voorbeelden:

> Gegeven: “**Johan was de vader van Peter.**”

•

• Gegenereerde conclusies:

< “**Peter heeft geen vader meer.**”

< “**Peter had een vader, genaamd Johan.**”

>

> Gegeven: “**Iedere vader is een man.**”

•

• Gegenereerde conclusie:

< “**Johan was een man.**”

1 De zin “**Johan was de vader van Peter**” moet herkend worden als “**Johan was een vader**”.

Blok 5: Conflictdetectie en het genereren van een vraag

[De algebra van taal](#) die hieronder genoemd wordt, past de volgende structuurwoorden toe: “is”, “of”, “en”, “ieder(e)” en “geen”.

“[Ieder\(e\)](#) {enkelvoudig zelfstandig naamwoord 1} [is](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} [of](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”
[is in conflict met](#)

“{eigenaam 1} [is](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} [en](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

“[Ieder\(e\)](#) {enkelvoudig zelfstandig naamwoord 1} [is](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} [of](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”
[en](#)

“{eigenaam 1} [is](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1}”

[waaruit kan worden geconcludeerd](#)

“{eigenaam 1} [is](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} [of](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

“{eigenaam 1} [is](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} [of](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

[is gelijkwaardig aan](#)

“[Is](#) {eigenaam 1} {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} [of](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}?”

“[Is](#) {eigenaam 1} {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1} [of](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}?”

[en](#)

“{eigenaam 1} [is geen](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1}”

[waaruit kan worden geconcludeerd](#)

“{eigenaam 1} [is](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}”

“[Is](#) {eigenaam 1} {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1} [of](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}?”

[en](#)

“{eigenaam 1} [is geen](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}”

[waaruit kan worden geconcludeerd](#)

“{eigenaam 1} [is](#) {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1}”

Voorbeelden:

> Gegeven: “Ieder persoon is een man of een vrouw.”

> Gegeven: “Anne is een man en een vrouw.”

!

! Gedetecteerd conflict: Deze zin is niet geaccepteerd, omdat het in conflict is met zichzelf.

!

> Gegeven: “Anne is een persoon.”

•

• Gegenereerde vraag:

< “Is Anne een man of een vrouw?”

>

> Gegeven: “Anne is geen vrouw.”

•

• Gedetecteerd dat de gegenereerde vraag beantwoord is:

< “Is Anne een man of een vrouw?”

•

• Gegenereerde conclusie:

< “Anne is een man.”

Blok 6: Het archiveren van kennis

[De algebra van taal](#) die hieronder genoemd wordt, past de volgende structuurwoorden toe: “de/het”, “is”, “heeft”, “van”, “genaamd”, “vorig(e)” en “nieuw(e)”.

“{eigenaam 1} **is** {bepaald lidwoord + enkelvoudig zelfstandig naamwoord} **van** {eigenaam 2}”

waaruit kan worden geconcludeerd

“{eigenaam 2} **heeft** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} **genaamd** {eigenaam 1}”

“{eigenaam 1} **is** {bepaald lidwoord + enkelvoudig zelfstandig naamwoord} **van** {eigenaam 2}”

gevolgd door

“{eigenaam 3} **is** {bepaald lidwoord + enkelvoudig zelfstandig naamwoord} **van** {eigenaam 2}”

waaruit kan worden geconcludeerd

“{eigenaam 2} **heeft** {onbepaald lidwoord} **nieuw(e)** {enkelvoudig zelfstandig naamwoord} **genaamd** {eigenaam 3}”

en

“{eigenaam 2} **heeft** {onbepaald lidwoord} **vorig(e)** {enkelvoudig zelfstandig naamwoord} **genaamd** {eigenaam 1}”

Voorbeelden:

> Gegeven: “Barack Obama is de president van de Verenigde Staten.”

•

• Gegenereerde conclusie:

< “De Verenigde Staten heeft een president, genaamd Barack Obama.”

>

> Gegeven: “Donald Trump is de president van de Verenigde Staten.”

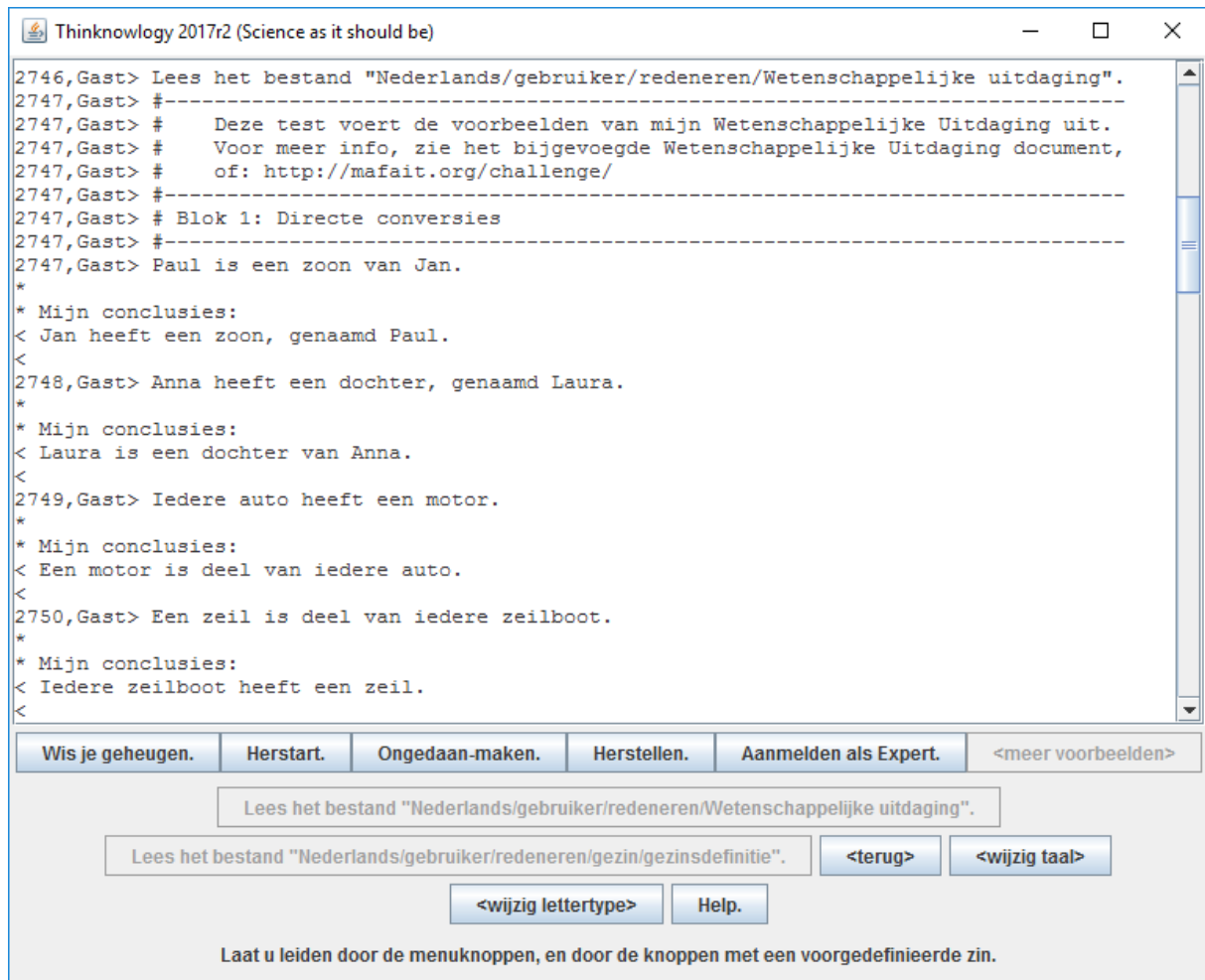
•

• Gegenereerde conclusies:

< “De Verenigde Staten heeft een nieuwe president, genaamd Donald Trump.”

< “De Verenigde Staten heeft een vorige president, genaamd Barack Obama.”

Schermafdrucken van deze uitdaging



Thinknowlogy 2017r2 (Science as it should be)

2751,Gast> #-----
2751,Gast> # Blok 2: Indirecte conversies
2751,Gast> #-----
2751,Gast> Ieder gezin heeft een ouder en een kind.
*
* Mijn conclusies:
< Een ouder en een kind zijn deel van ieder gezin.
<
2752,Gast> Sjors is een ouder.
*
* Mijn conclusies:
< Sjors is deel van een gezin.
*
* Mijn veronderstellingen:
< Sjors heeft waarschijnlijk een kind.
<
2753,Gast> André is een kind.
*
* Mijn conclusies:
< André is deel van een gezin.
*
* Mijn veronderstellingen:
< André heeft waarschijnlijk een ouder.
<
2754,Gast> Peter heeft een ouder.
*
* Mijn veronderstellingen:
< Peter is waarschijnlijk een kind.
< Peter is waarschijnlijk deel van een gezin.
<
2755,Gast> Willem heeft een kind.
*
* Mijn veronderstellingen:
< Willem is waarschijnlijk een ouder.
< Willem is waarschijnlijk deel van een gezin.
<

Wis je geheugen.
Herstart.
Ongedaan-maken.
Herstellen.
Aanmelden als Expert.
<meer voorbeelden>

Lees het bestand "Nederlands/gebruiker/redeneren/Wetenschappelijke uitdaging".

Lees het bestand "Nederlands/gebruiker/redeneren/gezin/gezinsdefinitie".

<terug>
<wijzig taal>

<wijzig lettertype>
Help.

Laat u leiden door de menuknoppen, en door de knoppen met een voorgedefinieerde zin.

© 2015-2017

Menno Mafait (<http://mafait.org/nl/challenge/>)

pagina 16 van 19


```

2756,Gast> #-----
2756,Gast> # Blok 3: Het groeperen van kennis
2756,Gast> #-----
2756,Gast> Jan is een ouder van Paul.
*
* Mijn conclusies:
< Jan is deel van een gezin.
< Paul heeft een ouder, genaamd Jan.
*
* Mijn veronderstellingen:
< Jan heeft waarschijnlijk een kind, genaamd Paul.
< Paul is waarschijnlijk een kind van Jan.
< Paul is waarschijnlijk deel van een gezin.
<
2757,Gast> Anna is een ouder van Paul.
*
* Mijn conclusies:
< Anna is deel van een gezin.
< Paul heeft 2 ouder [meervoud van 'ouder' is onbekend], genaamd Jan en Anna.
*
* Mijn veronderstellingen:
< Anna heeft waarschijnlijk een kind, genaamd Paul.
< Paul is waarschijnlijk een kind van Jan en Anna.
<
2758,Gast> Paul heeft 2 ouders, genaamd Jan en Anna.
*
* Mijn conclusies die bevestigd zijn:
< Paul heeft 2 ouder [meervoud van 'ouder' is onbekend], genaamd Jan en Anna.
*
* U heeft meervoudige zelfstandige naamwoord "ouders" ingevoerd, dat onbekend was voor mij.
*

```

Wis je geheugen.

Herstart.

Ongedaan-maken.

Herstellen.

Aanmelden als Expert.

<meer voorbeelden>

Lees het bestand "Nederlands/gebruiker/redeneren/Wetenschappelijke uitdaging".

Lees het bestand "Nederlands/gebruiker/redeneren/gezin/gezinsdefinitie".

<terug>

<wijzig taal>

<wijzig lettertype>

Help.

Laat u leiden door de menuknoppen, en door de knoppen met een voorgedefinieerde zin.

