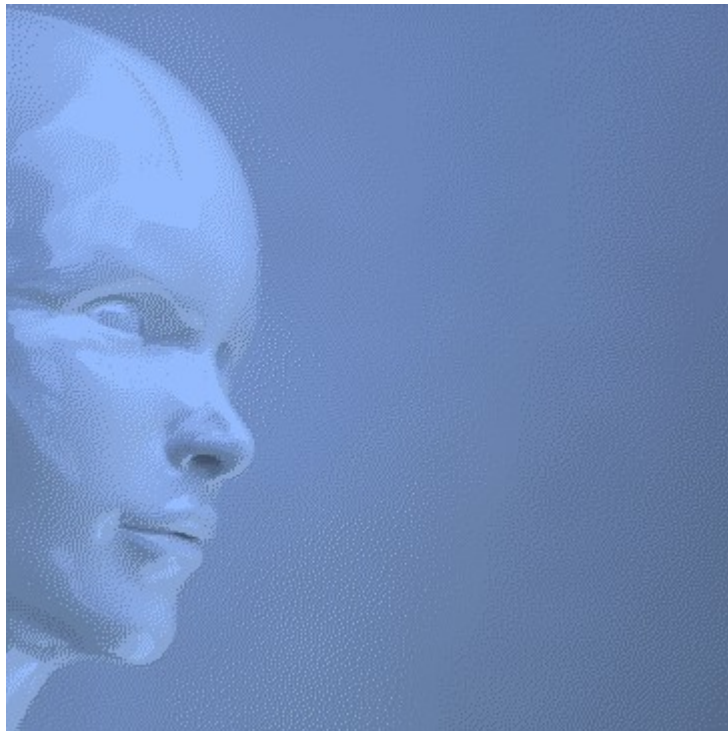


# Wetenschappelijke uitdaging:

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



## Inleiding

Alles in de natuur is gebonden aan natuurwetten, en verloopt volgens natuurwetten. Maar wetenschappers zijn niet in staat – of zijn onwillig – om intelligentie te definiëren als een verzameling natuurwetten (terwijl het mij wel is gelukt). Kunstmatige Intelligentie (AI) is dus niet gebaseerd op natuurwetten, waardoor het geen kunstmatige implementatie is van natuurlijke intelligentie. AI is daardoor geen fundamentele wetenschap, maar een *field of engineering*.

Een [fundamentele wetenschap](#) levert generieke oplossingen, terwijl een *field of engineering* beperkt is tot het leveren van specifieke oplossingen voor specifieke problemen. En *geëngineerde* oplossingen zijn beperkt tot het uitvoeren van routinetaken. AI is dus beperkt tot het uitvoeren van routinetaken, omdat het een *field of engineering* is.

Maar het is mogelijk om dit vakgebied te verheffen tot een fundamentele wetenschap, vergelijkbaar met het vakgebied elektromagnetisme, dat op natuurwetten is gebaseerd. Dankzij onderzoek naar de natuurwetten van elektromagnetisme, zijn we in staat om de cirkel te voltooien voor elektriciteit, magnetisme, licht en beweging, waardoor we in staat zijn om deze conversies toe te passen in het dagelijks leven. We zijn in staat:

- om elektriciteit om te zetten naar magnetisme, en magnetisme weer naar elektriciteit;
- om elektriciteit om te zetten naar licht, en om licht weer om te zetten in elektriciteit;
- om elektromagnetisme om te zetten naar beweging, en beweging weer naar elektromagnetisme.

Ik gebruik [fundamentele wetenschap](#) (logica en natuurwetten) in plaats van [cognitieve wetenschap / gedragswetenschap](#) (simulatie van gedrag):

- Zelfstandig redeneren vereist zowel natuurlijke intelligentie als natuurlijke taal;
- Intelligentie en taal zijn natuurfenomenen;
- Natuurfenomenen zijn onderworpen aan natuurwetten;
- Natuurwetten en logica worden onderzocht d.m.v. fundamentele wetenschap.

Door [intelligentie](#) te definiëren als een verzameling natuurwetten – en door de natuurwetten te onderzoeken die te maken hebben met [intelligentie in de grammatica](#) – ben ik in staat om de cirkel te voltooien voor natuurlijke intelligentie en natuurlijke taal. Feitelijk heb ik de [Logica van Aristoteles](#) uitgebreid. Door deze uitgebreide logica te gebruiken, is mijn systeem in staat:

- om leesbare zinnen – met een beperkte grammatica – om te zetten naar een logica die wetenschappers nog niet hebben beschreven;
- om – d.m.v. mijn uitgebreide logica – zelfstandig nieuwe kennis af te leiden van daarvoor nog onbekende kennis;
- en om deze afgeleide kennis uit te drukken in leesbare en – woord voor woord – zelfstandig geconstrueerde zinnen, met een beperkte grammatica.

De logicaregels van mijn zelfstandig redeneersysteem zijn (bijna) taalonafhankelijk. Daardoor kan ik elke taal toevoegen, gewoon door mijn redeneersysteem te configureren voor deze nieuwe taal, en door een beetje te programmeren. Mijn redeneersysteem is al in staat om te lezen, zelfstandig te redeneren en de afgeleide kennis zelfstandig te schrijven in het [Nederlands](#), [Engels](#), [Frans](#), [Spaans](#) en [Chinees](#), terwijl wetenschappers niet in staat zijn om op een generieke manier een meertalig redeneersysteem te ontwikkelen.

Via dit document daag ik iedereen uit om de eenvoudigste resultaten van mijn [Controlled Natural Language](#) (CNL) redeneersysteem op een generieke manier te verslaan: van natuurlijke taal, via algoritmes, terug naar natuurlijke taal, zonder voorgeprogrammeerde kennis, zonder uitvoerzinnen die door mensen zijn geschreven, en zonder gebruik te maken van uitgebreide woordenlijsten.

Natuurlijk is dit redeneersysteem gratis beschikbaar, en gepubliceerd als [open source software](#).

## Zelfstandig redeneren

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 in staat 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

Ook bezittelijk redeneren – redeneren met bezittelijk werkwoord “[heeft/hebben](#)” – wordt niet ondersteund door de predicaatenlogica (algebra):

> 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](#).”

---

Waarom ondersteunt predicaatenlogica (algebra) het bezittelijk redeneren niet op een natuurlijke manier? Waarom moet elk predicaat – dat niet kan worden uitgedrukt met het werkwoord “[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

Alhoewel de [Exclusieve OF](#) (XOR) functie voor algebra is gedefinieerd, is de taalkundige equivalent van deze functie – voegwoord “[of](#)” – niet geïmplementeerd in [CNL redeneersystemen](#). Ze 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](#)?”

---

Alles in de natuur is met elkaar verbonden. Maar het lijkt erop alsof taalkundigen en wiskundigen zich nooit hebben afgevraagd hoe hun beider vakgebieden aan elkaar verbonden zijn. Want wetenschappers zijn – bijna 2400 jaar na [Aristoteles](#) – nog geen stap verder in het begrijpen van de logica van taal.

Het is mijn ervaring dat taalkundigen bereid zijn om de logica van taal te onderzoeken, maar wiskundigen niet. Wiskundigen denken dat logica over getallen gaat. Ze hebben ongelijk. Logica gaat over natuurlijke intelligentie. En [natuurwetten van intelligentie](#) zijn onder andere te vinden in natuurlijke taal.

Net als een programmeertaal, heeft ook natuurlijke taal structuurwoorden en variabelen. De structuurwoorden van taal – die in dit document in blauw zijn weergegeven – hebben een natuurlijk-intelligente, logische, structuurgevende functie in taal. De volgende structuurwoorden van taal zullen in dit uitdagingsdocument worden toegelicht:

Bezittelijk werkwoord “[heeft/hebben](#)” (Blok 1, Blok 2 en Blok 3), werkwoorden in de verleden tijd “[was/waren](#)” en “[had/hadden](#)” (Blok 4), voegwoord “[of](#)” (Blok 5) en bepaald lidwoord “[de/het](#)” (Blok 6).

## 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 het 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, terug naar natuurlijke taal – om te zetten naar “**Jan heeft een zoon, genaamd Paul**”, waarbij zelfstandig naamwoord “**zoon**” en eigennamen “**Paul**” en “**Jan**” niet in het redeneer-systeem 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 [1.6.2. 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 twee zoons, genaamd Paul en Johan**”. Deze implementatie is dus niet flexibel. Het is daarom 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 een *field of engineering* tot een [fundamentele wetenschap](#), dat generieke oplossingen levert, gebaseerd op een natuurlijk fundament. Dat is wat 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 predicaatlogica. 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 journal);
- In het geval dat je resultaten iets anders zijn, dien je uit te leggen waarom je daarvoor hebt gekozen;
- Deze wedstrijd gaat door totdat alle blokken wetenschappelijk zijn geaccepteerd;
- 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

### Definitie 1:

---

“{eigenaam 1} is een/de {enkelvoudig zelfstandig naamwoord} van {eigenaam 2}”

is gelijkwaardig aan

“{eigenaam 2} heeft een {enkelvoudig zelfstandig naamwoord}, genaamd {eigenaam 1}”

---

### Voorbeelden:

---

Variabelen: eigenaam 1 = “Paul”, eigenaam 2 = “Jan”, enkelvoudig zelfstandig naamwoord = “zoon”

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

•

• Gegenereerde conclusie:

< “Jan heeft een zoon, genaamd Paul.”

---

Variabelen: eigenaam 1 = “Anna”, eigenaam 2 = “Laura”, enkelvoudig zelfstandig naamwoord = “dochter”

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

•

• Gegenereerde conclusie:

< “Laura is een dochter, genaamd Anna.”

---

## Definitie 2:

---

“Ieder(e) {enkelvoudig zelfstandig naamwoord 1} heeft een {enkelvoudig zelfstandig naamwoord 2}”

is gelijkwaardig aan

“Een {enkelvoudig zelfstandig naamwoord 2} is deel van ieder(e) {enkelvoudig zelfstandig naamwoord 1}”

---

## Voorbeelden:

---

Variabelen: enkelvoudig zelfstandig naamwoord 1 = “auto”, enkelvoudig zelfstandig naamwoord 2 = “motor”

> Gegeven: “Iedere auto heeft een motor.”

•

• Gegenerateerde conclusie:

< “Een motor is deel van iedere auto.”

---

Variabelen: enkelvoudig zelfstandig naamwoord 1 = “zeil”, enkelvoudig zelfstandig naamwoord 2 = “zeilboot”

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

•

• Gegenerateerde conclusie:

< “Iedere zeilboot heeft een zeil.”

---

## Blok 2: Indirecte conversies

### Definitie 3a:

---

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

waaruit kan worden geconcludeerd

“Een {enkelvoudig zelfstandig naamwoord 2} en een {enkelvoudig zelfstandig naamwoord 3} zijn deel van ieder(e) {enkelvoudig zelfstandig naamwoord 1}”

---

### Voorbeeld:

---

Variabelen: enkelvoudig zelfstandig naamwoord 1 = “gezin”, enkelvoudig zelfstandig naamwoord 2 = “ouder”, enkelvoudig zelfstandig naamwoord 3 = “kind”

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

•

• Gegenereerde conclusie:

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

---

### Definitie 3b:

---

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

en

“{eigenaam} is een {enkelvoudig zelfstandig naamwoord 2 of 3}”

waaruit kan worden geconcludeerd

“{eigenaam} is deel van een {enkelvoudig zelfstandig naamwoord 1}”

---

### Definitie 3c:

---

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

en

“{eigenaam} is een {enkelvoudig zelfstandig naamwoord 2}”

waaruit kan worden verondersteld

“{eigenaam} heeft waarschijnlijk een {enkelvoudig zelfstandig naamwoord 3}”

---

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

en

“{eigenaam} is een {enkelvoudig zelfstandig naamwoord 3}”

waaruit kan worden verondersteld

“{eigenaam} heeft waarschijnlijk een {enkelvoudig zelfstandig naamwoord 2}”

---

## Voorbeelden:

---

Variabelen: eigennaam = “Sjors”, enkelvoudig zelfstandig naamwoord 1 = “gezin”, enkelvoudig zelfstandig naamwoord 2 = “ouder”, enkelvoudig zelfstandig naamwoord 3 = “kind”

> Gegeven: “Sjors is een ouder.”

•

• Gegenerateerde conclusie:

< “Sjors is deel van een gezin.”

(gegenereerd door Definitie 3b)

•

• Gegenerateerde aanname:

< “Sjors heeft waarschijnlijk een kind.”

(gegenereerd door Definitie 3c)

---

Variabelen: eigennaam = “André”, enkelvoudig zelfstandig naamwoord 1 = “gezin”, enkelvoudig zelfstandig naamwoord 2 = “ouder”, enkelvoudig zelfstandig naamwoord 3 = “kind”

> Gegeven: “André is een kind.”

•

• Gegenerateerde conclusie:

< “André is deel van een gezin.”

(gegenereerd door Definitie 3b)

•

• Gegenerateerde aanname:

< “André heeft waarschijnlijk een ouder.”

(gegenereerd door Definitie 3c)

### Definitie 3d:

---

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

en

“{eigenaam} heeft een {enkelvoudig zelfstandig naamwoord 2 of 3}”

waaruit kan worden verondersteld

“{eigenaam} is waarschijnlijk deel van een {enkelvoudig zelfstandig naamwoord 1}”

---

### Definitie 3e:

---

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

en

“{eigenaam} heeft een {enkelvoudig zelfstandig naamwoord 2}”

waaruit kan worden verondersteld

“{eigenaam} is waarschijnlijk een {enkelvoudig zelfstandig naamwoord 3}”

---

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

en

“{eigenaam} heeft een {enkelvoudig zelfstandig naamwoord 3}”

waaruit kan worden verondersteld

“{eigenaam} is waarschijnlijk een {enkelvoudig zelfstandig naamwoord 2}”

---

## Voorbeelden:

---

Variabelen: eigennaam = “Peter”, enkelvoudig zelfstandig naamwoord 1 = “gezin”, enkelvoudig zelfstandig naamwoord 2 = “ouder”, enkelvoudig zelfstandig naamwoord 3 = “kind”

> Gegeven: “Peter heeft een ouder.”

•

• Gegeneerde aannames:

< “Peter is waarschijnlijk een kind.” (gegenereerd door Definitie 3e)

< “Peter is waarschijnlijk deel van een gezin.” (gegenereerd door Definitie 3d)

---

Variabelen: eigennaam = “Willem”, enkelvoudig zelfstandig naamwoord 1 = “gezin”, enkelvoudig zelfstandig naamwoord 2 = “ouder”, enkelvoudig zelfstandig naamwoord 3 = “kind”

> Gegeven: “Willem heeft een kind.”

•

• Gegeneerde aannames:

< “Willem is waarschijnlijk een ouder.” (gegenereerd door Definitie 3e)

< “Willem is waarschijnlijk deel van een gezin.” (gegenereerd door Definitie 3d)

---

## Blok 3: Het groeperen van kennis

### Definitie 4:

---

“{eigenaam 1} heeft een {enkelvoudig zelfstandig naamwoord}, genaamd {eigenaam 2}”  
en  
“{eigenaam 1} heeft een {enkelvoudig zelfstandig naamwoord}, genaamd {eigenaam 3}”  
is gelijkwaardig aan  
“{eigenaam 1} heeft {aantal: 2} {meervoudsvorm van enkelvoudig zelfstandig naamwoord},  
genaamd {eigenaam 2} en {eigenaam 3}”

---

### Voorbeeld:

---

Variabelen: eigenaam 1 = “Jan”, eigenaam 2 = “Paul”, eigenaam 3 = “Anna”, enkelvoudig zelfstandig naamwoord = “ouder”

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

•

• Gegenerateerde conclusie:

< “Paul heeft een ouder, genaamd Jan.” (gegenereerd door Definitie 1)

>

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

•

• Gegenerateerde 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 hebt het meervoudig zelfstandig naamwoord “ouders” ingevoerd, welke mij onbekend was.

## Blok 4: Redeneren in de verleden tijd

### Definitie 5:

---

“{eigennaam 1} was een/de/het {enkelvoudig zelfstandig naamwoord} van {eigennaam 2}”

waaruit kan worden geconcludeerd

“{eigennaam 2} heeft geen {enkelvoudig zelfstandig naamwoord} meer”

---

“{eigennaam 1} was een/de/het {enkelvoudig zelfstandig naamwoord} van {eigennaam 2}”

waaruit kan worden geconcludeerd

“{eigennaam 2} had een {enkelvoudig zelfstandig naamwoord}, genaamd {eigennaam 1}”.

---

### Voorbeeld:

---

Variabelen: eigennaam 1 = “Johan”, eigennaam 2 = “Peter”, enkelvoudig zelfstandig naamwoord = “vader”

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

•

• Gegenereerde conclusies:

< “Peter heeft geen vader meer.”

< “Peter had een vader, genaamd Johan.”

---

### Definitie 6:

---

“Ieder(e) {enkelvoudig zelfstandig naamwoord 1} is een {enkelvoudig zelfstandig naamwoord 2}”

en

“{eigenaam} was een {enkelvoudig zelfstandig naamwoord 1}”<sup>1</sup>

waaruit kan worden geconcludeerd

“{eigenaam} was een {enkelvoudig zelfstandig naamwoord 2}”

---

### Voorbeeld:

---

Variabelen: enkelvoudig zelfstandig naamwoord 1 = “vader”, enkelvoudig zelfstandig naamwoord 2 = “man”, eigenaam = “Johan”

> Gegeven: “Iedere vader is een man.”

•

• Gegenerateerde conclusie:

< “Johan was een man.”

---

1 De zin “Johan was de vader van Peter” uit het vorige voorbeeld moet herkend worden als “Johan was een vader”.

## Blok 5: Conflictdetectie en het genereren van een vraag

### Definitie 7:

---

“Ieder(e) {enkelvoudig zelfstandig naamwoord 1} is een {enkelvoudig zelfstandig naamwoord 2} of een {enkelvoudig zelfstandig naamwoord 3}”

is in conflict met

“{eigenaam} is een {enkelvoudig zelfstandig naamwoord 2} en een {enkelvoudig zelfstandig naamwoord 3}”

---

“Ieder(e) {enkelvoudig zelfstandig naamwoord 1} is een {enkelvoudig zelfstandig naamwoord 2} of een {enkelvoudig zelfstandig naamwoord 3}”

en

“{eigenaam} is een {enkelvoudig zelfstandig naamwoord 1}”

waaruit kan worden geconcludeerd

“{eigenaam} is een {enkelvoudig zelfstandig naamwoord 2} of een {enkelvoudig zelfstandig naamwoord 3}”

---

“{eigenaam} is een {enkelvoudig zelfstandig naamwoord 2} of een {enkelvoudig zelfstandig naamwoord 3}”

is gelijkwaardig aan

“Is {eigenaam} een {enkelvoudig zelfstandig naamwoord 2} of een {enkelvoudig zelfstandig naamwoord 3}?”

### Voorbeeld:

---

Variabelen: enkelvoudig zelfstandig naamwoord 1 = “persoon”, enkelvoudig zelfstandig naamwoord 2 = “man”, enkelvoudig zelfstandig naamwoord 3 = “vrouw”, eigenaam = “Anne”

> Gegeven: “Iedere 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:

< “Iedere persoon is een man of een vrouw.”

<

> Gegeven: “Anne is een persoon.”

•

• Gegenerateerde vraag:

< “Is Anne een man of een vrouw?”

---

### Definitie 8:

---

“Is {eigenaam} een {enkelvoudig zelfstandig naamwoord 1} of een {enkelvoudig zelfstandig naamwoord 2}?”

en

“{eigenaam} is geen {enkelvoudig zelfstandig naamwoord 1}”

waaruit kan worden geconcludeerd

“{eigenaam} is een {enkelvoudig zelfstandig naamwoord 2}”

---

“Is {eigenaam} een {enkelvoudig zelfstandig naamwoord 1} of een {enkelvoudig zelfstandig naamwoord 2}?”

en

“{eigenaam} is geen {enkelvoudig zelfstandig naamwoord 2}”

waaruit kan worden geconcludeerd

“{eigenaam} is een {enkelvoudig zelfstandig naamwoord 1}”

### Voorbeeld:

---

Variabelen: eigenaam = “Anne”, singular noun 1 = “man”, singular noun 2 = “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

### Definitie 9:

---

“{eigenaam 1} is **de** {enkelvoudig zelfstandig naamwoord} **van** {eigenaam 2}”

en

“{eigenaam 3} is **de** {enkelvoudig zelfstandig naamwoord} **van** {eigenaam 2}”

waaruit kan worden geconcludeerd

“{eigenaam 2} heeft een nieuw(e) {enkelvoudig zelfstandig naamwoord}, genaamd {eigenaam 3}”

en

“{eigenaam 2} heeft een vorig(e) {enkelvoudig zelfstandig naamwoord}, genaamd {eigenaam 1}”

---

### Voorbeeld:

---

Variabelen: eigenaam 1 = “**Barack Obama**”, eigenaam 2 = “**de Verenigde Staten**”, eigenaam 3 = “**Donald Trump**”, enkelvoudig zelfstandig naamwoord = “**president**”

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

•

• Gegeneerde conclusie:

< “**De Verenigde Staten heeft een president, genaamd Barack Obama.**” (gegeneerd door Definitie 1)

>

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

•

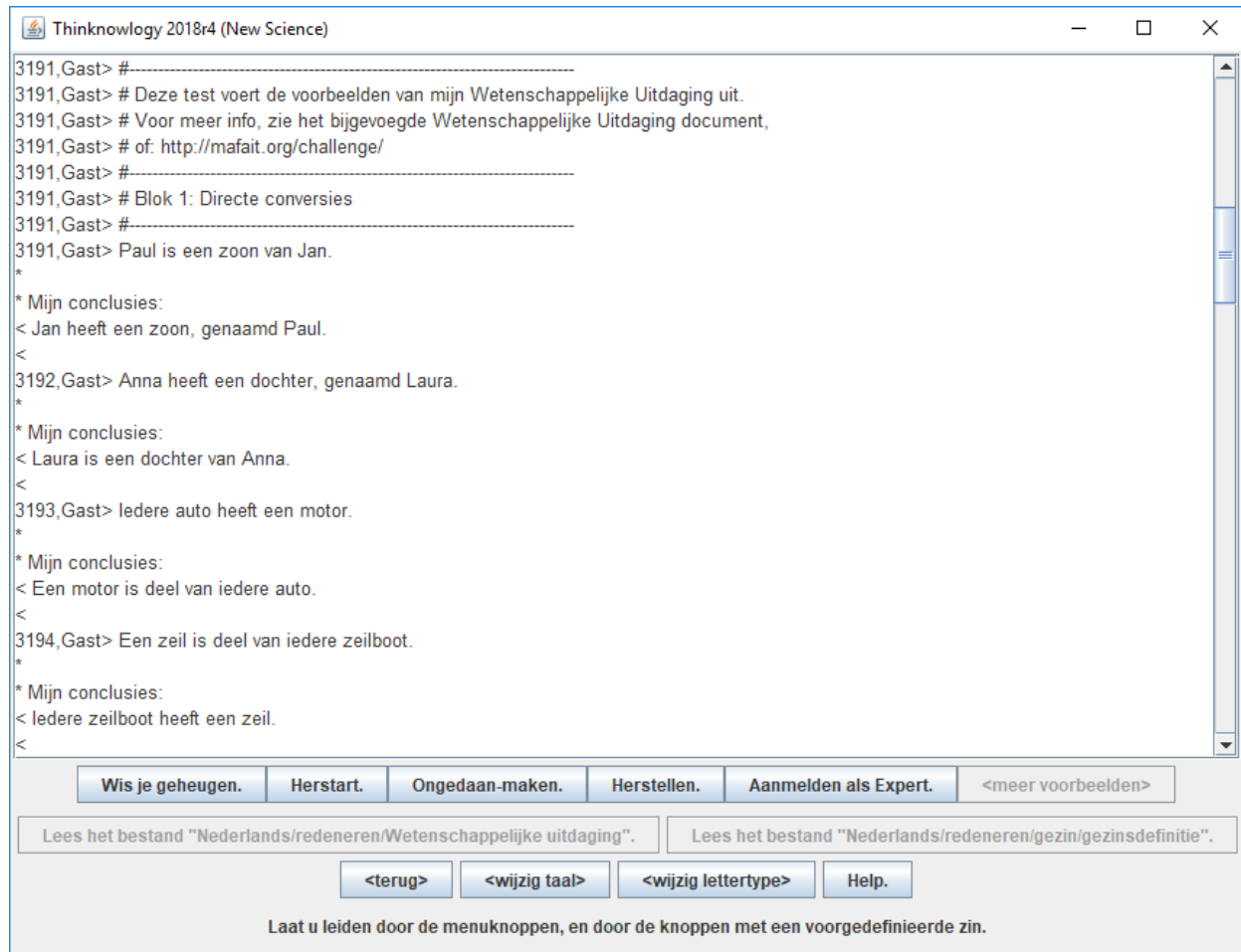
• Gegeneerde conclusies:

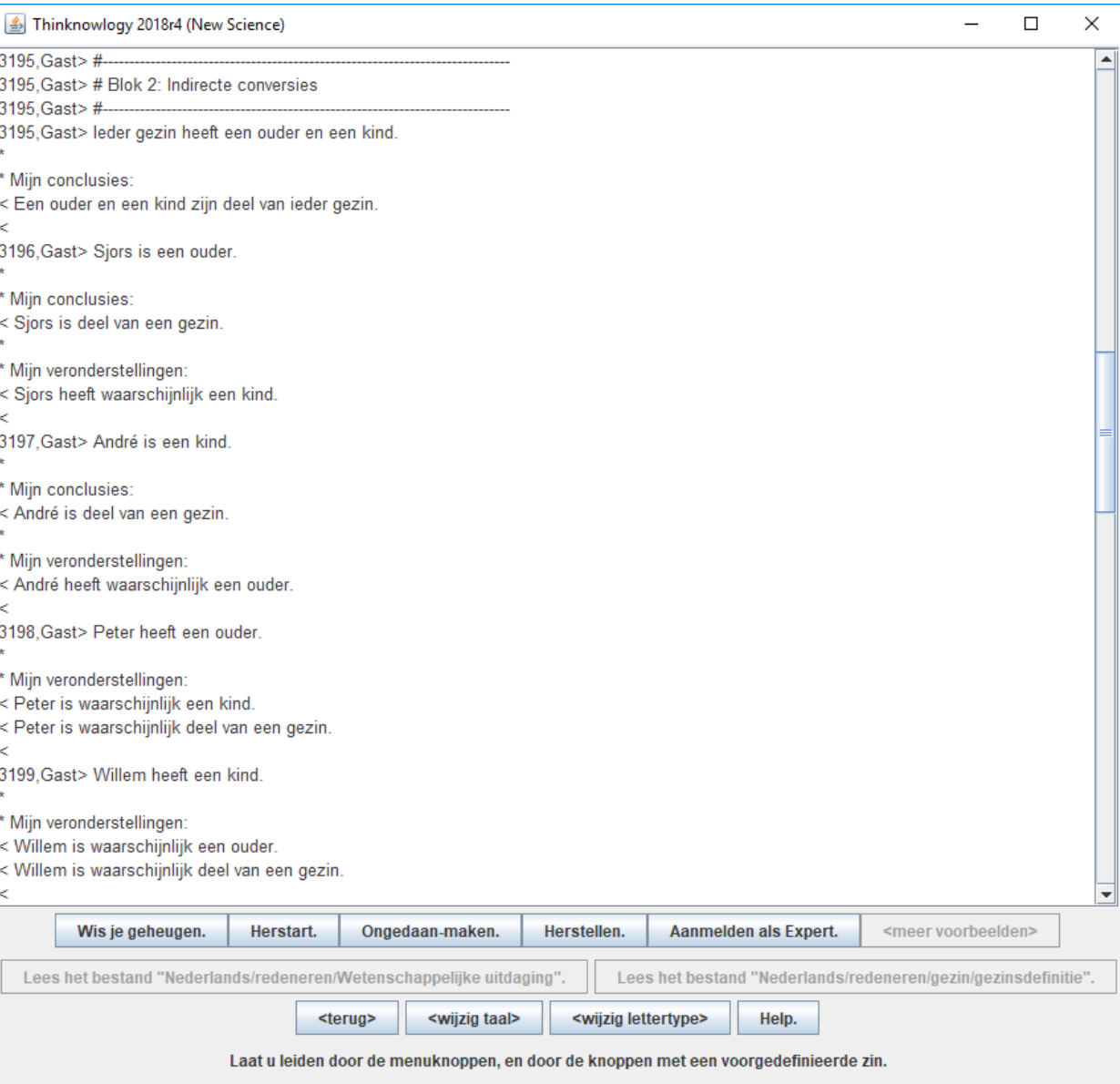
< “**De Verenigde Staten heeft een nieuwe president, genaamd Donald Trump.**”

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

---

## Schermafdrucken van deze uitdaging





Thinknowlogy 2018r4 (New Science)

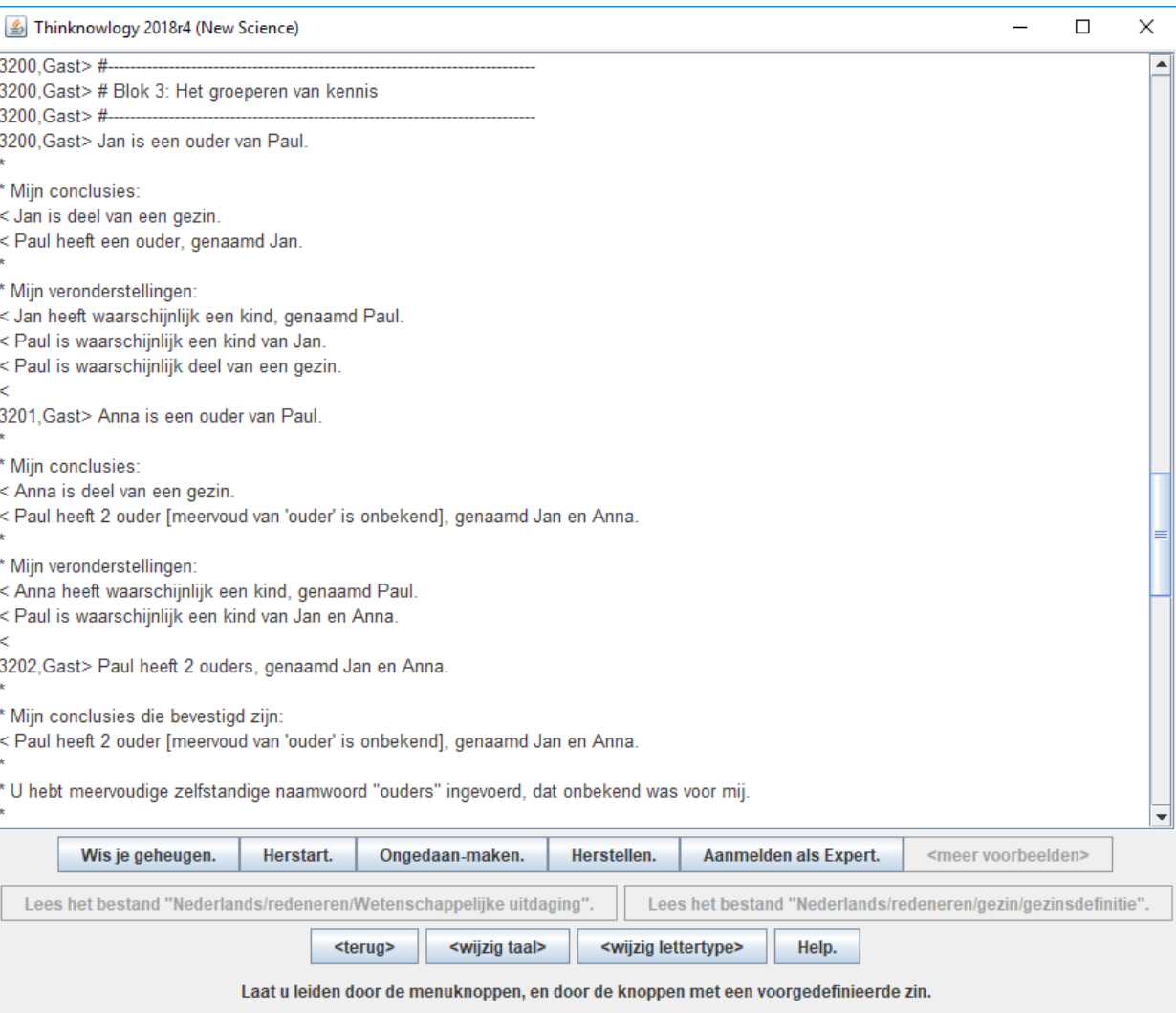
3195,Gast> #-----  
3195,Gast> # Blok 2: Indirecte conversies  
3195,Gast> #-----  
3195,Gast> Ieder gezin heeft een ouder en een kind.  
\*  
\* Mijn conclusies:  
< Een ouder en een kind zijn deel van ieder gezin.  
<  
3196,Gast> Sjors is een ouder.  
\*  
\* Mijn conclusies:  
< Sjors is deel van een gezin.  
\*  
\* Mijn veronderstellingen:  
< Sjors heeft waarschijnlijk een kind.  
<  
3197,Gast> André is een kind.  
\*  
\* Mijn conclusies:  
< André is deel van een gezin.  
\*  
\* Mijn veronderstellingen:  
< André heeft waarschijnlijk een ouder.  
<  
3198,Gast> Peter heeft een ouder.  
\*  
\* Mijn veronderstellingen:  
< Peter is waarschijnlijk een kind.  
< Peter is waarschijnlijk deel van een gezin.  
<  
3199,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/redeneren/Wetenschappelijke uitdaging".](#)
[Lees het bestand "Nederlands/redeneren/gezin/gezinsdefinitie".](#)

[<terug>](#)
[<wijzig taal>](#)
[<wijzig lettertype>](#)
[Help.](#)

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



Thinknowlogy 2018r4 (New Science)

3200,Gast> #-----

3200,Gast> # Blok 3: Het groeperen van kennis

3200,Gast> #-----

3200,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.

<

3201,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.

<

3202,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 hebt meervoudige zelfstandige naamwoord "ouders" ingevoerd, dat onbekend was voor mij.

\*

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

Thinknowlogy 2018r4 (New Science)

3203,Gast> #-----  
 3203,Gast> # Blok 4: Redeneren in de verleden tijd  
 3203,Gast> #-----  
 3203,Gast> Johan was de vader van Peter.  
 \*  
 \* Mijn conclusies:  
 < Peter heeft geen vader meer.  
 < Peter had een vader, genaamd Johan.  
 <  
 3204,Gast> Iedere vader is een man.  
 \*  
 \* Mijn conclusies:  
 < Johan was een man.  
 <

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

Thinknowlogy 2018r4 (New Science)

3205,Gast> #-----  
 3205,Gast> # Blok 5: Conflictdetectie en het genereren van een vraag  
 3205,Gast> #-----  
 3205,Gast> Ieder persoon is een man of een vrouw.  
 3206,Gast> Anne is een man en een vrouw.  
 !  
 ! Deze zin is niet geaccepteerd, omdat het in conflict is met:  
 < Ieder persoon is een man of een vrouw.  
 <  
 3206,Gast> Anne is een persoon.  
 \*  
 \* Mijn vragen:  
 < Is Anne een man of een vrouw?  
 <  
 3207,Gast> Anne is geen vrouw.  
 \*  
 \* Mijn vragen die beantwoord zijn:  
 < Is Anne een man of een vrouw?  
 \*  
 \* Mijn conclusies:  
 < Anne is een man.  
 <

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

