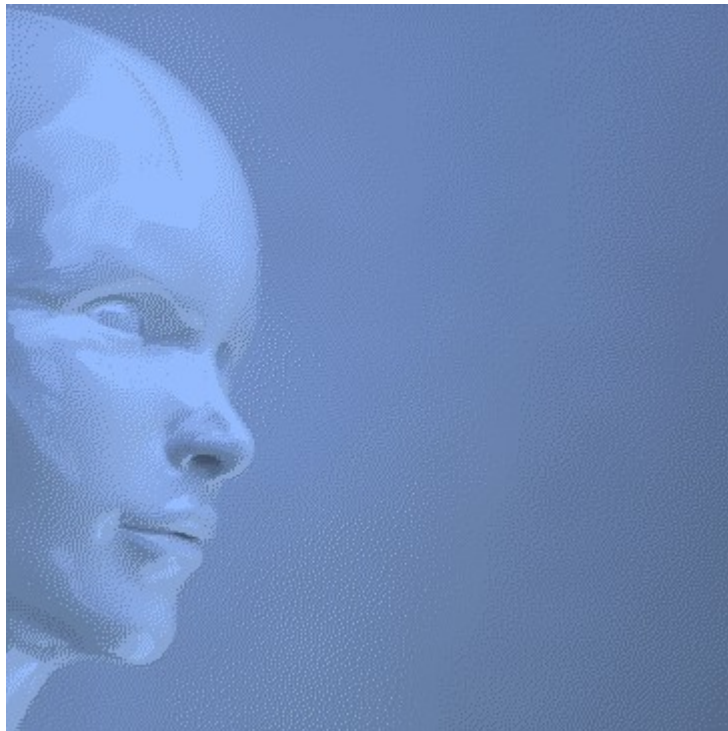


Wetenschappelijke uitdaging:

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



Probleembeschrijving

Zelfstandig redeneren vereist zowel [natuurlijke intelligentie](#) als natuurlijke taal. Zonder het te weten, hebben wetenschappers minstens 200 jaar geleden natuurlijke intelligentie toegepast op natuurlijke taal, door redeneerconstructies te beschrijven in [Predicatenlogica](#) (algebra). Later zijn deze redeneerconstructies geïmplementeerd in [Controlled Natural Language](#) (CNL) redeneersystemen. CNL redeneersystemen zijn instaat om nieuwe kennis af te leiden, van tot dan toe onbekende kennis, beide uitgedrukt in leesbare zinnen met een beperkte grammatica. Voorbeeld:

- > Gegeven: “Jan is een vader.”
- > Gegeven: “Iedere vader is een man.”
-
- Logische conclusie:
- < “Jan is een man.”

Echter, dezelfde redeneerconstructie - maar dan in de verleden tijd - is in geen enkel wetenschappelijk artikel beschreven, omdat wetenschappers "vergeten" zijn om predicatenlogica (algebra) voor de verleden tijd te definiëren:

- > Gegeven: “Jan was een vader.”
- > Gegeven: “Iedere vader is een man.”
-
- Logische conclusie:
- < “Jan was een man.”

Als het gaat om zelfstandig redeneren, is predicatenlogica (algebra) blijkbaar beperkt tot het ondersteunen van werkwoord “is/zijn” in de tegenwoordige tijd. Algebra ondersteunt ook geen bezittelijk redeneren – redeneren met bezittelijk werkwoord “heeft/hebben” – zoals:

- > Gegeven: “Paul is een zoon van Jan.”
-
- Logische conclusie:
- < “Jan heeft een zoon, genaamd Paul.”

Dus, waarom ondersteunt predicatenlogica (algebra) niet het redeneren in de verleden tijd op een natuurlijke manier, 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?

En 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 instaat 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, zelfs 60 jaar na het begin van dit vakgebied, heeft de kennistechnologie nog steeds een fundamenteel probleem: 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. Maar hun natuurlijk-intelligente functie is in geen enkel wetenschappelijk artikel beschreven. Blijkbaar begrijpen wetenschappers hun natuurlijk-intelligente functie in de taal niet.

Algemeen aanvaarde *workaround*

De algemeen aanvaarde *workaround* in het vakgebied Kunstmatige Intelligentie (AI) en kennistechnologie 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 specificatiwoorden 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 kennistechnologie heeft geen natuurlijk fundament, waardoor het vakgebied beperkt is tot het leveren van specifieke oplossingen voor specifieke problemen (=engineering), zoals deze bovenstaande Excel-implementatie. Deze uitdaging gaat echter over het verheffen van dit vakgebied, van engineering tot een 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 predicatenlogica. 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 ontfangen 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-schermafdrukken](#) van mijn CNL redeneersysteem laten zien dat diverse redeneerconstructies elkaar versterken. De Schermafdrukken 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: € 200 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}”

Implementatie:

> 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}”

Implementatie:

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

•

• Gegenereerde conclusie:

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

>

> Gegeven: “Sjors is een ouder.”

•

• Gegenereerde conclusie:

< “Sjors is deel van een gezin.”

•

• Gegenereerde aanname:

< “Sjors heeft waarschijnlijk een kind.”

>

> Gegeven: “André is een kind.”

•

• Gegenereerde conclusie:

< “André is deel van een gezin.”

•

• Gegenereerde aanname:

< “André heeft waarschijnlijk een ouder.”

>

> Gegeven: “Peter heeft een ouder.”

•

• Gegenereerde aannames:

< “Peter is waarschijnlijk een kind.”

< “Peter is waarschijnlijk deel van een gezin.”

>

> Gegeven: “Willem heeft een kind.”

•

• Gegenereerde 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}”

Implementatie:

> 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}”

Implementatie:

> 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}”

Implementatie:

> 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.”

•

• Gegenerateerde vraag:

< “Is Anne een man of een vrouw?”

>

> Gegeven: “Anne is geen vrouw.”

•

• Gedetecteerd dat de gegenerateerde vraag beantwoord is:

< “Is Anne een man of een vrouw?”

•

• Gegenerateerde 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}”

Implementatie:

> 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

