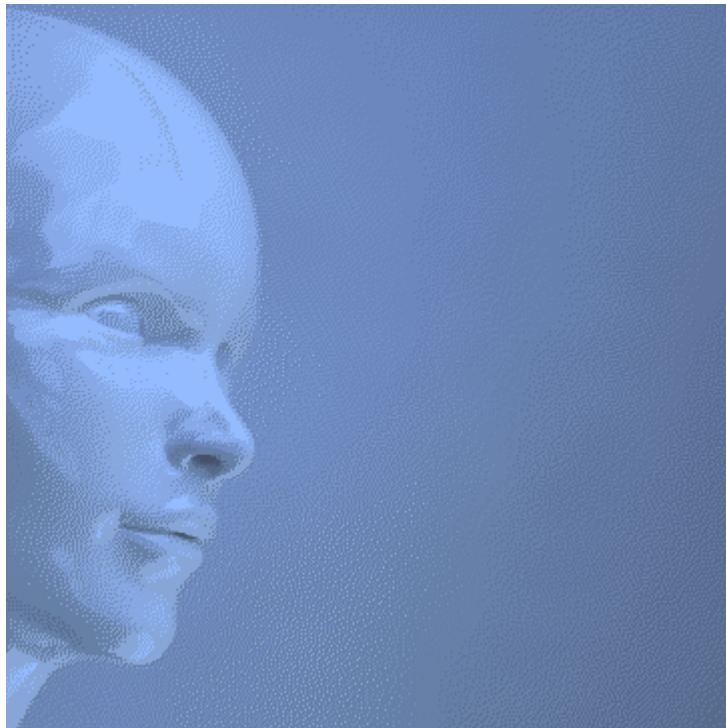


Wetenschappelijke uitdaging:

Versla mijn natuurlijke taal redeneersysteem



Probleembeschrijving

We weten maar heel weinig over de logica in taal. Sterker nog, al eeuwenlang, is algebra slechts beperkt tot het redeneren met het werkwoord “**is/zijn**” in de tegenwoordige tijd, zoals in:

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

Maar mensen zijn ook in staat tot bezittelijk redeneren – met bezittelijk werkwoord “**heeft**” – zoals:

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

En mensen zijn in staat tot redeneren in de verleden tijd, zoals:

- > Gegeven: “**Johan was de vader van Peter.**”
-
- Logische conclusies:
- < “**Peter heeft geen vader meer.**”
- < “**Peter had een vader, genaamd Johan.**”

Dus, waarom ondersteunt algebra het redeneren in de verleden tijd niet op een natuurlijke manier, noch bezittelijk redeneren? Waarom zou elk predicaat geavanceerder dan het werkwoord “**is/zijn**” in de tegenwoordige tijd moeten worden beschreven op een kunstmatige manier? Waarom is algebra nog niet uitgerust voor de taalkunde, na eeuwen van wetenschappelijk onderzoek?

En alhoewel algebra de **Exclusieve OF** (XOR) functie op een natuurlijke wijze beschrijft, implementeren automatische *reasoners* nog steeds niet de taalkundige equivalent van deze functie: voegwoord “**of**”. Automatische *reasoners* zijn dus 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, 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 5), voegwoord “[of](#)” (zie Blok 4), bezittelijk werkwoord “[heeft/hebben](#)” (zie Blok 1 and Blok 2) en werkwoorden in de verleden tijd “[was/waren](#)” en “[had/hadden](#)” (zie Blok 3) 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 *reasoner* te programmeren, zoals: [heeft_zoon\(jan,paul\)](#). Dit is echter **geen** generieke oplossing (=wetenschap), maar een specifieke oplossing voor een specifiek probleem (=engineering), omdat je elk afzonderlijk zelfstandig naamwoord direct in de *reasoner* moet programmeren ([heeft_dochter](#), [heeft_vader](#), [heeft_moeder](#), et cetera), en voor iedere nieuwe 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 (=d.m.v. een algoritme) om te zetten naar “[Jan heeft een zoon, genaamd Paul](#)”, waarbij zelfstandig naamwoord “[zoon](#)” en eigennamen “[Jan](#)” en “[Paul](#)” niet in de *reasoner* 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 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 aan het ontwikkelen ben.

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 natuurlijke semantiek (natuurlijk-intelligente functie) van niet-trefwoorden (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 5 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 van de hieronder genoemde invoerzinnen. Bij voorkeur moeten de gebruikte zelfstandige naamwoorden en eigennamen vooraf onbekend zijn;
- 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 natuurlijke-taal-redeneersysteem laten zien dat diverse redeneerconstructies elkaar versterken. En 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 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 mijn systeem volledig is verslagen;
- Ik ben de jury. (Als je het niet leuk vindt dat ik de jury bent, voel je dan vrij om je oplossing wetenschappelijk geaccepteerd te krijgen zonder mijn goedkeuring.)

Jouw beloning

- Een klein gebaar van mijn kant: € 200 per blok dat wetenschappelijk aanvaard is;
- Je bent de eerste die deze resultaten op een wetenschappelijk geaccepteerde manier heeft bereikt, omdat ik geen enkele techniek uit dit vakgebied gebruik.

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

Blok 1: Conversies

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

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

is gelijkwaardig aan

“{eigennaam 2} **heeft** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} **genaamd** {eigennaam 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: Het groeperen van kennis

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

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

is gelijkwaardig aan

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

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

en

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

is gelijkwaardig aan

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

Implementatie:

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

•

• Gegenerateerde conclusie:

< “Paul heeft een ouder, genaamd Jan.”

>

> 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 heeft meervoudig zelfstandig naamwoord “ouders” ingevoerd, welke mij onbekend was.

Blok 3: 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”.

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

waaruit kan worden geconcludeerd

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

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

waaruit kan worden geconcludeerd

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

Implementatie:

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

•

• Gegenereerde conclusies:

< “Peter heeft geen vader meer.”

< “Peter had een vader, genaamd Johan.”

Blok 4: 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 gegenereerde vraag beantwoord is:

< “Is Anne een man of een vrouw?”

•

• Gegenerateerde conclusie:

< “Anne is een man.”

Blok 5: 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: “**Bill Clinton is de president van de Verenigde Staten.**”

•

• Gegenereerde conclusie:

< “**De Verenigde Staten heeft een president, genaamd Bill Clinton.**”

>

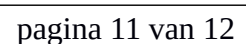
> Gegeven: “**Barack Obama is de president van de Verenigde Staten.**”

•

• Gegenereerde conclusies:

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

< “**De Verenigde Staten heeft een vorige president, genaamd Bill Clinton.**”



Thinkknowlogy 2016r2 (Restyle)

```

2782,Gast> #-----
2782,Gast> # Blok 3: Redeneren in de verleden tijd
2782,Gast> #-----
2782,Gast> Johan was de vader van Peter.
*
* Mijn conclusies:
< Peter heeft geen vader meer.
< Peter had een vader, genaamd Johan.
<
2783,Gast> #-----
2783,Gast> # Blok 4: Conflictdetectie en het genereren van een vraag
2783,Gast> #-----
2783,Gast> Ieder persoon is een man of een vrouw.
2784,Gast> Anne is een man en een vrouw.
!
! Deze zin is niet geaccepteerd, omdat het in conflict is met zichzelf.
!
2784,Gast> Anne is een persoon.
*
* Mijn vragen:
< Is Anne een man of een vrouw?
<
2785,Gast> Anne is geen vrouw.
*
* Mijn vragen die beantwoord zijn:
< Is Anne een man of een vrouw?
*
* Mijn conclusies:
< Anne is een man.
<
2786,Gast> #-----
2786,Gast> # Blok 5: Het archiveren van kennis
2786,Gast> #-----
2786,Gast> Bill Clinton is de president van de Verenigde Staten.
*
* Mijn conclusies:
< De Verenigde Staten heeft een president, genaamd Bill Clinton.
<
2787,Gast> Barack Obama is de president van de Verenigde Staten.
*
* Mijn conclusies:
< De Verenigde Staten heeft een nieuwe president, genaamd Barack Obama.
< De Verenigde Staten heeft een vorige president, genaamd Bill Clinton.
<
2788,Gast> # Om verder te gaan, klik op knop «Wis je geheugen.» of «Herstart.»
2787,Gast>

```

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.