Tore Amble
Regnesentret
Trondheim

Å VÆRE ELLER Å HA, DET ER SPØRSMÅLET

1. MYKE SYSTEMER

Naturlig språk brukes vanligvis når mennesker kommuniserer med hverandre, og er velegnet til dette formål. I moderne tid kommer stadig flere mennesker i kontakt med datamaskiner, enten de liker det eller ikke, og måten kommunikasjonen foregår vil være viktig for effektiviteten og trivselen til de som har med dem å gjøre.

For spesialoppgaver som skal utføres av en fast stab som har fått opplæring, vil spesielle styrespråk være tilstrekkelig og i noen tilfeller å foretrekke. (Det har bl.a. vært hevdet at opplæring i slike styrespråk gir utøverne en slags yrkes-identitet.)

En utvikling i retning av naturlig språk vil være ønskelig i situasjoner der

a) bruken er sporadisk
b) brukerne har kontakt med flere datamaskiner, f.eks. via datanett
c) brukerne mangler opplæring i spesielle systemer

Hittil har situasjonen ofte vært den at et datasystem er utviklet for å utføre en oppgave. Til slutt har man laget et bruker-grensesnitt for å ta seg en kommunikasjon med brukerne. Et slikt brukergrensesnitt vil ofte utgjøre en mindre del av det totale system.

Etter som kravene til brukervennlighet stiger (f.eks. via arbeidsmiljøloven) vil vi kunne komme i den situasjon at en vesentlig del av et datasystem utgjøres av brukergrensesnitt.

Myke systemer

For å møte utfordringer fra brukerne og gjøre det lettere å lage brukergrensesnitt, har Regnesentret ved Universitetet i Trondheim (RUNIT) startet et forskningsprosjekt "Mjuke System" med støtte fra NTNF. Formålet er å lage et generelt bruker-grensesnitt som kan tilpasses EDB-systemer etter behov.

Elementer i et slikt mykt system vil være naturlig språk og menyer (dvs. ja/nei-spørsmål eller nummererte alternativer).

Som et første målsystem har vi tatt for oss operativsystemet SINTRAN (2), og har konsentrert oss om å stille spørsmål til dette operativsystemet på norsk.

Å være eller å ha, det er spørsmålet
Tore Amble
Proceedings of NODALIDA 1981, pages 74-90

2. FORUTSETNINGER FOR Å BRUKE NATURLIG SPRÅK I DIALOG MED DATAMASKINER

Hvorfor er naturlig språk vanskelig for en datamaskin?

- Mangel på presisjon, kompleksitet.

Det er flere grunner til at naturlig språk er vanskelig å behandle.

Datamskiner er ekspert på å utføre algoritmer, dvs. et begrenset antall entydige regne- eller beslutningsregler, men de har hittil ikke vært flinke når beslutningsreglene er upresise eller avhengige av ytre sammenhenger.

Som et lite eksempel kan vi ta setningen

HVILKEN X' HAR Y'

Her kan X' være subjekt og Y' objekt som f.eks.

HIVLKEN ANSATT HAR TELEFON?

eller X' har vært objekt, og Y' subjekt som i

HVILKEN TELEFON HAR HANSEN?

- Stort ordforråd

Mens datamaskinspråk hittil har et meget begrenset ordtilfang (f.eks. et kommando-språk kan ha et par hundre), så vil naturlige språk ha et ordforråd som er gigantisk i forhold. (Det ekstreme tilfelle er "Complete and unabridged" OXFORD Dictionary med 400.000 innslag).

- Stor kunnskapsmengde

Et voksent menneske har minst 18 års intens opplæring i bruk av naturlig språk i vid forstand, og besitter en enorm kunnskapsmengde som han bare delvis er bevisst.

Hvilke restriksjoner må vi lage?

Den viktigste forutsteningen for å lage dialog-system i naturlig språk er å innføre begrensninger for at systemet skal være praktisk mulig å implementere. Begrensningene må være slik at

en ny bruker ikke trenger mer opplæring enn det som kan presenteres i begynnelsen av en dialog. Reglene må være lette å formulere, og lette å huske.

Avvisning og feilmeldinger må dessuten alltid være konstruktive, og referere til alment vedtatte normer som f.eks. norsk grammatikk eller vanlig betydning av ord.

Brukeren er den endelige dommer over et system. I praksis blir det en forutsetning for å lykkes at brukeren er takknelig over å få kunne uttrykke seg på sitt morsmål, selv med en del restriksjoner. Det vil si at naturlig språk er mest velegnet for uerfarne brukere.

Restriksjoner faller i 2 hovedgrupper:

- begrensninger i emnet for dialogen
- begrensninger i språket

Begrensninger i emnet vil være en grei restriksjon, som er ikke bare nødvendig men også ønskelig. Vi må i alle fall lage spesialanvendelser først, og la generaliseringene komme etterpå.

Begrensninger i språket er mer problematisk.

Det er flere typer regler for naturlig språk:

1) ordforråd (leksikalske regler)
2) bøyningsregler (morfologiske regler)
3) ordstillingsregler (syntaks)
4) betydningsregler (semantikk)

1) Ordforråd kan begrenses til opplisting av godkjente ord, eller eventuelt eksklusjon av forbudte ord. For eksempel kan ordene AT og Å forbys fordi de gir opphav til for abstrakte setninger. En annen restriksjon er å begrense bruk av verb. Dette kommer vi tilbake til.

   Restriksjoner på ordforrådet kan formildes noe ved å innlede en dialog når et ukjent ord påtreffes.

2) Når det gjelder bøynigsregler, så bør strategien være at dersom systemet finner en bøyningsfeil som ikke forandrer meningen i setningen, så skal den godtas. Vår erfaring hittil er at vi kommer langt ved bare å analysere ordstammene, og finne meningen ved hjelp av konteksten.

3) Ordstillingsregler
Det bør være akseptabelt at man i utgangspunktet forlanger grammatikalsk riktig ordstilling, siden alle brukere bør forutsettes å kunne det.

4) Betydningsregler
Betydningsregler eller semantikk er vanskelig å formalisere. Gode feilmeldinger kan alikevel gi en pekepinn om hvor misforståelsen er.

3. VERBFRITT SPRÅK

Datamaskiner som brukes til datalagring har tradisjonelt store mengder ensartede data. Situasjoner vil endre seg etter hvert som vi er i stand til å mestre mer komplekse informasjonsstrukturer. Vi kan alikevel forutsi noe om arten av informasjon som vil bli lagret:

- informasjon som er uavhengig av tid og sted og hvem som fortalte det, altså ikke

  "Han er 14 uker gammel"

  men

  "Halvard er født 13/7-1981".

- opprinnelige informasjon i motsetning til avledet informasjon

  Altså, ikke

  "Per er eldre enn Pål"

  men

  "Per er født i 1975"
  "Pål er født i 1976"

Mange fenomener kan beskrives ved tilstander og endringer i tilstander. Vi kommuniserer vanligvis ikke alt vi ser og hører, bare det som er nødvendig for at tilhøreren skal kunne oppdatere sin kunnskapsbase. Som oftest uttrykker vi informasjonen indirekte, slik at det som vi ønsker å meddele er en logisk følge av det som blir sagt, og det som er kjent allerede.

Meget grovt sagt bruker vi ofte verb til å uttrykke endringer i tilstander, mens vi bruker substantiver og adjektiver til å beskrive tilstander. Om vi eliminerer bruk av andre verb enn HA og VÆRE, så vil vi fokusere på beskrivelsene av fakta og

tilstander, og dette er nettopp hensikten med restriksjonen. Den samme restriksjon møter de som skal forfatte skjema som noen skal fylle ut. De må ofte "avverbisere" informasjonen.

Eksempler på verbfrie omformuleringer (tatt fra innbydelses-skjema til denne kongressen):

Med verb	Uten verb
Hva heter du?	Hvilket navn har du?
Hvor bor du?	Hvilken adresse har du?
Hva skal foredraget ditt hete?	Hva er tittelen på foredraget ditt?
Hvor lang tid trenger foredraget?	Hva er tiden til foredraget?

Ut fra disse betraktninger har vi satt som et mål å først lage en språklig og betydningsmessig komplett system for å forstå samtaler, dvs. opplysninger og spørsmål i et verbfritt språk. Dette system skal være forutsetningsfritt, og kunne tilegne seg kunnskap på norsk. (Et blankt system vil ha den bisarre egenskap å kunne snakke norsk uten å vite noe.)

Et lite eksempel som er kjørt, vil illustrere hovedidéen. En liten gutt jeg kjenner ble meget imponert og fornøyd da vi (B) kjørte følgende sesjon på datamaskin (S).

B: ALLE GUTTER ER PERSONER
S: OK
B: NOEN GUTTER ER SNILLE
S: OK
B: NOEN GUTTER ER SLEMME
S: OK
B: JAN MAGNUS ER EN GUTT
S: OK
B: JAN MAGNUS ER SNILL
S: OK
B: HVEM ER SNILLE?
S: JAN MAGNUS
B: ER JAN MAGNUS SNILL?
S: JA

Idéen med verbfritt språk er ikke ny. Den er funnet i (1) som behandler gjennomførbarheten av å bruke engelsk som kommando og spørrespråk i industrielle applikasjoner. Ved protokollforsøk har de funnet at spørsmål og kommandoer ble formulert eller kunne lett reformuleres med setninger som bare benyttet verbene BE, HAVE og DO.

4. SOFTRAN - DIALOGSYSTEM FOR OPERATIVSYSTEMER

SOFTRAN er et dialogsystem basert på verbfritt norsk, med predefinert informasjon om operativsystemet SINTRAN (2). Dessuten er det laget tilkopling til selve operativsystemet.

SOFTRAN har en leksikalsk preprosessor (3) som er skrevet i PASCAL (4). Resten av systemet er skrevet i PROLOG (5).

Beskrivelsen av spørresystemet representerer tilstanden pr. dags dato, og vil gi en del informasjon om angrepsmåte og forventninger, men er ikke representativ for ytelse til det tiltenkt ferdige system. Resultatene må ses i relasjon til at prosjektet er i en startfase.

Alle ord som brukes må i prinsippet være kjent med stamme og bøyninger dersom de skal kunne brukes fritt. Hvis ikke kan man definere synonymer selv (på norsk).

Alle endelser blir fjernet innledningsvis. Dette virker kanskje brutalt, men virker i praksis.

Det har to gunstige effekter:

1) Systemet er ettergivende over uvesentlige grammatikalske feil, f.eks.

   HVILKE TERMINAL ER TILKOPLET

2) Implisitte påstander om antall svar blir ignorert, f.eks.

   HVILKEN FIL HAR JEG?

   når svaret er mer enn én fil.

En tredje forenkling er å slå sammen ord (f.eks. TIL, AV, FRA, PÅ, FOR, I . Jfr. det engelske ordet OF.)

Filosofien bak disse forenklingene er at den informasjon som derved forsvinner og som ikke lar seg rekonstruere av sammenhengen er unødvendig å bry seg om.

Ordklasser

Syntaksanalysen foregår etter en noe annen oppdeling enn den vanlige (substantiv, verb etc.). Vi opererer med følgende 6 klasser:

1. **Entitet** — Egennavn, individ
   Eksempel JEG, TA-EVA2

2. **Klasse** — Fellesnavn for entiteter, herunder også perfektum partisipp som brukes som klasse.
   Eksempel FIL, TERMINAL, ANSATT

3. **Attributt** — Substantiver som betegner en egenskap som kan ha en verdi, f.eks.
   FARGE, TYPE, STØRRELSE

4. **Egenskap** — Adjektiv, herunder også perfektum partisipp som brukes som adjektiv.
   Eksempel SYMBOLSK, GUL, STOR, TERMINERT

5. **Verb** — HA, VÆRE, UTFØRE

6. **Diverse** — Ordklassen er i praksis videre oppdelt. Lista er ikke komplett. Ord i parentes blir gjenkjent, men avvist.

   ALL, ALLE, ALT, (AT), AV
   BARE, BÅDE
   DE, DEN, DENNE, DET, DETTE, (DIN), (DINE), (DU)

   ELDRE, ELLER, ET, EN
   FOR, FORSKJELLIG
   HAN, HUN, HANS, HENNES
   HVA, HVEM, HVILKE, HVILKEN, HVILKET, HVOR
   (HVORFOR), (HVORDAN)

   I, IKKE, INGEN
   JEG

   LIK, LITEN
   MED, MELLOM, MEN, MEST, MIN, MINDRE, MINST
   NOE, NOEN
   OG, OGSÅ, OVER

   PÅ
   SAMME, SIN, SOM, STOR, STØRRE, STØRST
   TIL
   ULIK, UNDER
   (VI)
   YNGRE
   (Å)

5. SYNTAKSANALYSE

Metoden som blir brukt for syntaksanalyse er hentet fra Pereira og Warren (6), der PROLOG blir foreslått som et alternativ til ATN (7).

Den illustreres best ved et eksempel

En tekst i formatet ENTITET kan bestå av en tekst i formatet KLASSE fulgt av en tekst i formatet NAVN

```
    FIL TA-EVA2
X'   Y'      Z'
```

Fig. 1

Som antydet på figur 1 er mellomrommene mellom ordene markert med variable (X', Y', Z').

I PROLOG lar denne syntaksdefinisjonen uttrykke ved

```
ENTITET(X',Z'):KLASSE(X',Y'),NAVN(Y',Z').
```

og kan leses slik

For alle X', Y', Z', er det en tekst i formatet ENTITET mellom X' og Z' dersom det er en tekst i formatet KLASSE fra X' til Y' og en tekst i formatet NAVN fra Y' til Z'.

Siden vi ikke bare skal analysere, men også viderebehandle tekster, lager vi en produksjon med et resultatfelt (etter "="). Vi bygger opp en struktur som internt er en trestruktur, men som eksternt kan presenteres med et parentes-uttrykk.

Eksempel:

```
ENTITET(X',Z')= EN(M',N'):

        KLASSE(X',Y')=M',
        NAVN(Y',Z')=N'.
```

Resultatet av å analysere frasen

```
FIL SOFTRAN
```

blir en komponent i et semantisk tre

```
EN(FIL,SOFTRAN)
```

som blir analysert i den semantiske analysen.

Av de komponenter som inngår i et semantisk tre, vil vi nevne følgende:

EN (klasse, identitet)

BÅDE (egenskap, klasse)

VERDIAV (attributt, smlgn, verdi)

ANTALL (kvantifikator, klasse)

DETIL (attributt, entitet)

IKKE (egenskap)

HVA

HVILKE

ER

HAR

HAS-AV

6. SEMANTISK ANALYSE

Semantisk tre

Den syntaktiske analysen produserer et semantisk tre. Eksempel til setningen

HAR FIL SOFTRAN TYPE SYMB?

bli oversatt til en trestruktur

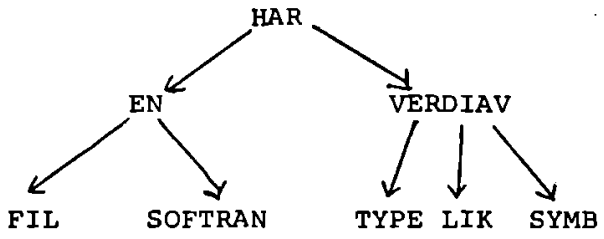


Fig. 2

som kan presenteres med parentesuttrykket

HAR (EN(FIL,SOFTRAN),VERDIAV(TYPE,LIK,SYMB))

Semantisk nett

Semantikk vil si betydningen av setningene, og vil alltid være avhengig av hvilken verden vi snakker om. (I forbindelse med datamaskiner er det ofte adekvat å snakke om mikroverden.) Det er flere filosofiske syn på hva egentlig mening er. Vi vil innta den pragmatiske holdning at et ord aldri har mening i og for seg, men bare i forbindelse med de relasjoner ordet har til andre ord. Disse forbindelser eller relasjoner kan vi ofte fremstilt grafisk i såkalte semantiske nett (9). Semantisk nett er ikke noen stringent formalisme, men en notasjonsform som kan ha mange varianter. Grunnelementene i et semantisk nett er navngitte noder og piler der nodene representerer entiteter og klasser, mens pilene representerer attributter og egenskaper.

Eksempel:

Fig. 3

som mekanisk oversatt kan leses

TORE har TELEFON 3016

Her er TORE og 3016 entiteter mens TELEFON er en relasjon.

Vi kan kople sammen opplysninger:

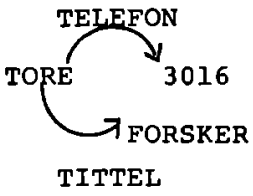


Fig. 4

I naturlig språk er det ikke bare nødvendig å vite de eksakte elementæropplysningene (hvilken telefon har Tore) men også opplysninger av mer generell art, som f.eks. hvilke klasser av ting har telefon.

Det er viktig for at spørsmål som

Hvilken tittel har telefon?

blir avvist som meningsløs, og ikke blir besvart med

"INGEN"

For å få til slike opplysninger, må vi utvide begrepsapparatet med noen spesielle relasjoner, som gis spesiell betydning.

1) ER-EN

uttrykker et element i en klasse

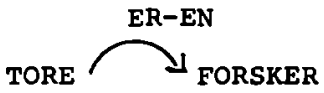


Fig. 5

Uttrykker at TORE er et medlem av klassen av forskere.

2) ER uttrykke subklasse - forhold mellom to klasser, f.eks.

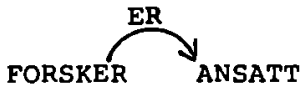


Fig. 6

som uttrykker at alle forskere er ansatt.

3) ER?

uttrykker at medlemmer av en klasse kan ha en egenskap (men uten verdi)
Eksempel:

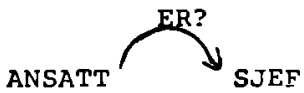


Fig. 7

4) HAR

uttrykk at alle medlemmer av en klasse har en verdi-avhengig egenskap (attributt)

Fig. 8

Uttrykker at alle ansatte har telefon, og at for hver ansatt har telefon er verdi (telefonnummer). Dersom en slik verdi mangler, er det adekvate svaret

"UKJENT".

5) HAR?

Uttrykke at noen medlemmer av en klasse kan ha en verdi-avhengig egenskap

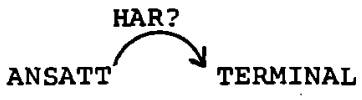


Fig. 9

HAR? uttrykker også ofte det vi forstår med et eier-forhold. Dersom en slik verdi mangler, er det adekvate svaret "INGEN".

Det semantiske nettet er på flere nivåer samtidig, både generelt og spesielt, og brukes både til å kontrollere semantikk og til å besvare spørsmål.

Følgende regler gjelder

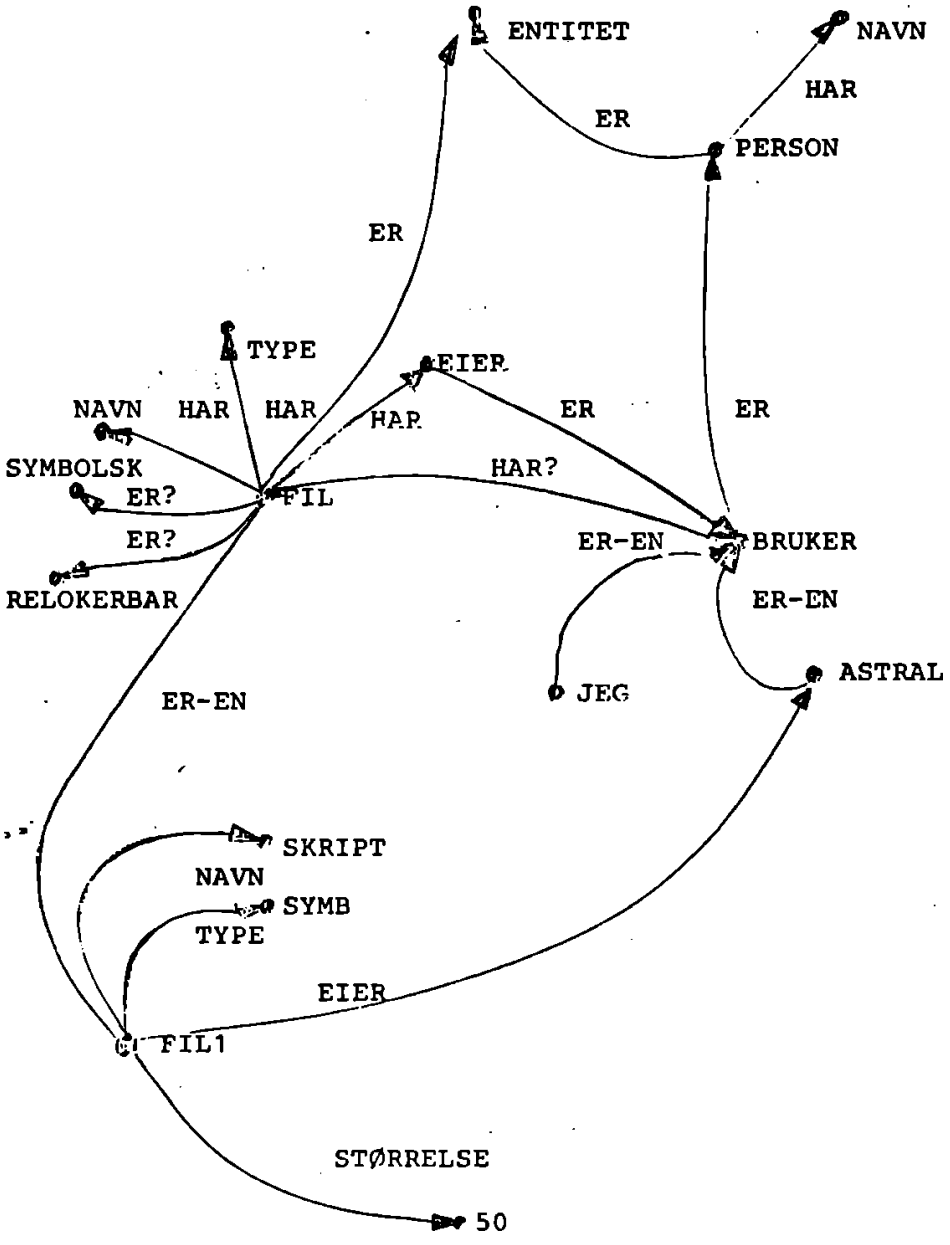
1) Dersom en entitetsklasse har et attributt eller en egenskap, så arves denne av alle entiteter som tilhører entitetsklassen.

2) En entitet (eller entitetsklasse) kan binde en verdi til et attributt dersom attributtet tilhører den omfattende klasse.

3) En egenskap kan bindes til sann eller usann dersom egenskapen er definert for klassen.

Hvordan defineres semantisk nett

Et semantisk nett kan dekomponeres i sine enkelte relasjoner, og hver av disse kan uttrykkes med en enkel setning i naturlig språk. Sammensetningen til et nett kan gjøres av et program-system.


ENTITET
NAVN
HAR
ER
PERSON
ER
TYPE
EIER
NAVN
HAR
HAR
HAR
ER
ER
SYMBOLSK
HAR?
ER?
FIL
ER?
ER-EN
BRUKER
RELOKERBAR
ER-EN
ASTRAL
JEG
ER-EN
SKRIPT
NAVN
SYMB
TYPE
EIER
FIL1
STØRRELSE
50

Figur 10

Skript

Informasjonen beskrevet i det semantiske nettet på figur 10 kan defineres ved en serie stiliserte setninger på norsk.

Alle personer er entiteter.

Alle personer har navn.

Alle brukere personer.

Alle eiere er brukere.

Jeg er en bruker.

ASTRAL er en bruker.

Alle filer er entiteter.

Noen brukere har filer.

Alle filer har navn.

Alle filer har type.

Alle filer har størrelse.

Alle filer har eier.

Noen filer er symbolske.

Noen filer er relokerbare.

FIL1 er en fil.

FIL1 har navn SKRIPT.

FIL1 har type SYMB.

FIL1 har eier ASTRAL.

Implementasjon

Implementasjon av semantiske nett blir gjort ved hjelp av PROLOG. Faktisk blir ikke nettet implementert internt som en nettstruktur, men blir lagret som tripler (tupler) i en relasjon. (Jfr. Realasjonsmodellen (8).)

For å demonstrere hvordan systemet virker, skal vi gå igjennom med et eksempel:

HVILKE SYMBOLSKE FILER HAR JEG?

som blir oversatt av leksikalanalysatoren til

(HVILKE SYMBOLSK FIL HA JEG ? NIL)

Syntaksanalysatoren vet at FIL er et klassenavn, mens SYMBOLSK er et adjektiv og JEG er en entitet. Den produserer derfor følgende semantiske tre

(HVILKE BÅDE (SYMBOLSK, FIL) HAS-AV JEG)

Den semantiske analysen går som følger:

1) BÅDE (SYMBOLSK, FIL)

   Det kontrolleres at adjektivet SYMBOLSK er definert for klassen FIL. Vi kan nå se bort fra adjektivet, og står igjen med

   HVILKE FIL HAS-AV JEG

2) Systemet undersøker hvilke klasser som har eller kan ha en FIL, og det finner én klasse BRUKER.

3) Det kontrolleres at entiteten JEG tilhører klassen BRUKER. (At JEG er synonym med en bruker ASTRAL finnes først etterpå.)

7. SVARFINNING

Svargenerering er basert på logikk og mengdelære. Eksemplet svarer til mengden

$$\{x' \mid \exists y' (x' \text{ er en fil } \& \; x' \text{ er symbolsk } \& \; y' \text{ er eier til } x' \; \& \; y' \text{ er brukernavn til JEG})\}$$

Svar på slike uttrykk lar seg naturlig formulere i PROLOG.

Man kan ikke uttrykke all informasjon i et semantisk nett, til det er formalismen for grov.

Derfor må vi knytte en rekke pragmatiske betingelser til det semantiske nettet i form av logikk-programmer (5).

8. KONKLUSJON

En komplett analyse av verbfrie setninger er en nødvendig del av et hvert program for naturlig språkprosessering. Restriksjoner på bruk av verb er derfor i verste fall bare uttrykk for en midlertidig begrensning.

Vi tar et viktig steg når og hvis denne restriksjonen blir opphevet, for da åpner vi for spørsmål av høyere kompleksitet som datamaskinen neppe vil klare å forstå fullt ut. Restriksjonene som omfattes av datamaskinens kapasitet vil da bli meget vanskelige å formulere.

En dag kan vi ha den situasjon at vi har muligheter for å oversette spørsmål med verb til verbfrie spørsmål, men vi lar det være av hensyn til konsekvensene for forventningene til systemet.

Vår målgruppe er i første omgang brukere som er takknemlige for å kunne uttrykke seg på sitt morsmål, selv om språket er pålagt restriksjoner. Vi håper å finne et språklig kompromiss mellom det vi er i stand til å implementere på dagens maskiner, og det brukerne er villig til å innrette seg etter.

Epilog

I sommer kom jeg over en avisoverskrift som jeg overlater til leserne å analysere:

Tre dommere og vekk med Norge

9 REFERANSER

(1): Microcomputer-Based Natural-Language Understanding, Phase I

Machine Intelligence Corp.
Mountain View, Ca 1980.

(2): SINTRAN II User's Guide
NORSK DATA Publ.

(3): Stålhane, T: Leksikalsk analyse i Mjuke System

Nordiske Lingvistikkdager 1981
Universitetet i Trondheim

(4): Wirth, N: The Programming Language Pascal
Eidgenössische Technische Hochshule
Zürich, Juli 1973.

(5): Amble, T: Introduction to Logic Programming
RUNIT notat, Universitetet i Trondheim

(6): Pereira & Warren:
Definite Clause Grammars Compared with Augmented Transition Networks
Department of Artificial Intelligence
University of Edinburth

(7): Bates, M: The Theory and Practice of Segmented Transition Network Grammars

Natural Language Communication with Computers

Lecture Notes in CS, no 63, Springer-Verlag

(8): Date, C.J: An Introduction to Database Systems
Part 2 The Relational Approach
Addison Wesley

(9): Findler, N.V.(ED): Associative Networks

Representation and use of Knowledge by Computers

Academic Press