Prolog - Rejsesystem

Carl Benjamin S. Dreyer

10 marts. 2024

1 Indledning

Ofte når man skal vælge et rejsemål bliver man overvældet i alle de muligheder der findes. Men hvor stor en del af de præsenterede muligheder opfylder den individuelle persons rejsekriterer? Hvad hvis personen har et stramt budget, eller påkræver at rejsestedet tilbyder en specifik aktivitet?
Dette projekt vil arbejde med problemstilling om at man bliver præsenteret med for mange urelevante rejsemål, som ikke påtaler individets rejseønsker.

2 Kravspecifikation

Der skal udvikles et rejseprogram, som kan hjælpe med at finde på rejseforslag målrettet persons rejseønsker. For dette projekt skal programmet kunne:

- Filtrere efter budget; billige og dyre byer.
- Filtrere efter temperatur; varme og kolde byer.
- Filtrere efter mindst en aktivitet som landet tilbyder.
- Man skal kunne kombinere de forskellige kriterier.

3 Design

3.1 Hvad er logik?

Logik er læren om at ræsonnere - dvs. At anvende logik og fornuft. Logikkens formål er at bevise noget eller komme frem til en endelige konklusion ved hjælp af korrekt ræsonnering. Dette er i lighed med matematiske beviser og love idét at det er universelt anvendelige og gyldige[1]. Når logikken anvendes på computere, kaldes det for datalogi.

3.2 Datalogi - logik og computerberegning

Udsagn og ræsonneringslove er fundamentet for logik. Logikken behandler og manipulerer udsagn på en meningsfuld måde, uden at vide meningen bag de konkrete udsagn. Logikken manipulerer udsagn vha. Slutningsregler (også kaldt inferensregler). Slutningsreglernes formål er at bevise noget ud fra nogle regler og udsagn – de står for ræsonneringen[2]. Den vigtigste af disse skrives

$$\frac{p \leftarrow q \quad q}{p} \tag{1}$$

Hvor der er to unikke udsagn over stregen, og et resulterende udsagn under stregen. Det første udsagn, $p \leftarrow q$, læses "p hvis q". $\leftarrow$ kaldes omvendt implikation operatøren. Denne regel siger at hvis vi har to udsagn p og q og et

[1] C. Hansen and P. Hansen 2009, s. 220.
[2] C. Hansen and P. Hansen 2009, s. 221.

logisk fakta for q, så slutter vi det nye udsagn p. Det betyder altså at hvis udsagnet $p \leftarrow q$ evalueres til at være sandt, så må p også være et sandt udsagn. For eksempelvis, givet "det er tirsdag i morgen", hvis "det er mandag i dag". Derudover er udsagnet "det er mandag i dag" oplyst. Ved indsættelse i reglen for oven, kan det ses at vi får følgende udtryk:

$$\frac{\text{Det er tirsdag i morgen} \leftarrow \text{Det er mandag i dag} \quad \text{Det er mandag i dag}}{\text{Det er tirsdag i morgen}} \tag{2}$$

Ved at anvende omvendte implikations operatøren, kan det ses at udtrykket $p \leftarrow q$ evalueres til at være sand, hvilket betyder at vi slutter med udsagnet p, "Det er tirsdag i morgen". Det er essentielt at observere at slutningen ikke afhænger af sandhedsværdien for det logiske fakta for udsagnet q. Denne regel er universel og udsagnet "Det er mandag i dag" er ligegyldig for slutningens logiske korrekthed.

3.3 Rejse ekspertsystemet

Det simpleste arbejde ekspertsystemet skal kunne er differentiere dyre og billige byer fra hinanden. Dvs. at der skal være en slutningsregel for om en by er dyr eller billig. Det kan observeres at denne simple logiske ræsonnering kan udformes ud fra den generelle slutningsregel (Se formel (1)).
Eksempelvis: har vi en rejser hvis rejsemål er alle byer som er billige. Derudover givet det logiske fakta "Istanbul er en billig by", kan slutningsregelen inferere om byen er et rejsemål eller ej:

$$\frac{\text{Byen er et rejsemål} \leftarrow \text{Istanbul er billig} \quad \text{Istanbul er billig}}{\text{Byen er et rejsemål}} \tag{3}$$

Altså ved anvendelse af omvendt implikationsoperatøren, kan det ses at slutningsreglen siger at Istanbul er et rejsemål for denne rejser. Denne samme slutningsregel kan anvendes til at determinere om byen er et rejsemål hvis temperaturen i byen f.eks. skal være kold eller varm. Dette betyder at slutningsregel kan udvides hvis både økonomien og temperaturen er en faktor for rejsemålet:

$$\frac{p \leftarrow q \hat{} z \quad q \quad z}{p} \tag{4}$$

Hvor q er udsagn for økonomien (billig eller dyr) og z er udsagn for temperaturen (varm eller kold).

3.4 Databaserelationer - aktivitetsudvidelse

For at udvide på logikken, er der en rejsende som afgør sine rejsemål baseret på om landet byen befinder sig i, facilitere forskellige aktiviteter, som f.eks. at stå på ski. Indtil videre er der kun blevet arbejdet med logiske fakta omkring byer, men dette kræver logiske fakta for de lande byerne befinder sig i. Der skal altså være en række logiske fakta omkring hvilke lande man kan stå på ski i.

En logisk sætning for om den rejsende betragter byen som et rejsemål ud fra udsagnene om at man skal kunne stå på ski landet som byen befinder sig i, kan formuleres som: "Byen er et rejsemål hvis byen befinder sig i et land hvor man kan stå på ski". Dette kan i formel logisk symbolik skrives som:

$$\forall B\,(ski(B) \leftarrow \exists(land(B, L)\,\hat{}\,ski(L))) \tag{5}$$

Her repræsenterer udsagnet B by, og L land. Der anvendes alkvantoren, eksistenskvantoren og ˆoperatøren som siges "og". Prædikatet "ski" betegner om man kan stå på ski i parameteren, og "land" prædikatet betegner om B er i L - altså byen B, befinder sig i landet L. Den formelle logiske symbolik kan siges som "For alle byer man kan ski i, eksistere der et land som byen befinder sig i og det land kan man stå på ski i". Dette formaliserede logiske udsagns korrekthed er uafhængig af det konkrete fakta suppleret, men er universel for enhver fakta suppleret.

4 Implementering

Prolog er et programmeringssprog specifikt designet til at løse formaliserede logiske problemer. Prologs egenskaber er perfekt designet til at implementere det ekspertsystem defineret i designprocessen, fordi den udelukkende består af formaliseret logik med tilhørende fakta og udsagn. Prolog programmets fundament består af en database af logiske fakta.

```
% City Country relationship
country(singapore, singapore).
country(zurich, switzerland).
country(paris, france).
country(dublin, ireland).
...

% Expensive cities
expensive(singapore).
expensive(zurich).
expensive(paris).
expensive(dublin).
...

% Cheap cities
cheap(bangkok).
cheap(delhi).
cheap(cusco).
...

% Skiing
skiing(switzerland).
skiing(france).
```

Figure 1: Rejsesystems programets logiske fakta

Med følgende fakta (Se figur 1), kan man query prologs database med `?-` prefixet. Hvis man for eksempel ville finde alle byer der er dyre, kan man query prolog `?- expensive(City)`, som genererer følgende:
For at implementere aktivitetsudvidelsen som beskrevet tidligere, har programmet brug for flere fakta:
Med dette kan der opstilles regler for landets temperatur, samt de aktiviteter som landet tilbyder:

Figure 2: Query efter alle dyre byer

```
temperature(singapore, 27).
temperature(zurich, 0.3).
...

skiing(switzerland).
skiing(france).
...
```

Figure 3: Flere fakta til programmet

De første 2 regler (se ll. 1-2 i figur 4) i programmet er for at kunne filtrere byer efter temperatur. Herefter opstilles en regel for om man kan stå på ski i det land som byen befinder sig i (l. 4) ud fra den formelle symbolik beskrevet tidligere i ligning(5).
En regel i prolog starter med et navn efterfulgt af parameter og `:-`. For at ski reglen kan være sand, kræver det at forholdet mellem byen og landet er opfyldt (l. 5). Herefter bruges `,` operatøren i prolog, som betyder "og". Og operatøren i prolog korresponderer til $\hat{}$ i den formelle symbolik. Det kræver udover by og land forholdet også at man kan stå på ski i det land byen befinder sig i (l. 6). Til sidst skrives kun byen til konsollen (l. 7), fordi vi kun er interesseret i hvilke byer som der er rejsemål.

```
1. warm(C) :- temperature(C, T), T >= 10.
2. cold(C) :- temperature(C, T), T < 10.
3.
4. canSki(City, Country):-
5.     country(City, Country),
6.     skiing(Country),
7.     write(City), nl.
```

Figure 4: Flere fakta til programmet

5 Test

For at teste rejsesystemet, kan man prøve at query prolog med forskellige rejsekriterier. Givet en person som gerne vil rejse til en by der er billig og varm, kan følgende query opstilles:

```
?- cheap(City),warm(City).
```

Som generere følgende output:

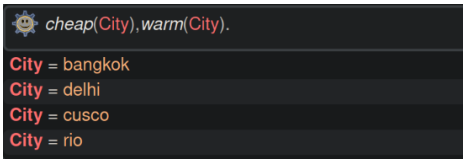

Figure 5: Første test

En mere kompleks query som anvender aktivitetsudvidelsen af programmet, kunne være:
Denne query skulle gerne outputte byer som er dyre, kold og er befinder sig i et land man kan stå på ski i:

```
?- expensive(City),cold(City),canSki(City,Country).
```

Figure 6: Anden test

En person vil rejse til et sted hvor man kan stå på ski, eller en by som er billig og varm, så vil man kunne opsætte en query vha. eller `;` operatøren:

```
?- canSki(City, Country);(warm(City),cheap(City)).
```

Som genererer:

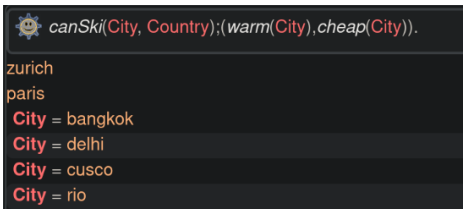

Figure 7: Tredje Test

6 Konklusion

Der er blevet undersøgt hvordan logik og computere relatere til hinanden i form af datalogi. Den datalogiske teori bag programmet er først blevet dokumenteret vha. formel symbolik og slutningsregler. Herefter er den viden blevet anvendt til at implementere slutningsregelerne i et konkret prolog program. Efterfølgende test bekræfter integriteten af programmets ræsonering som beskrevet i design afsnittet. Til at konkludere, er det blevet udarbejdet et program som kan give rejseforslag, baseret på en kombination af rejsekriterer som budget, temperatur og aktiviteter.

Kildeliste

Hansen, Carsten Broder and Per Christian Hansen, eds. (2009). *Matematiske horisonter*. Dansk. Technical University of Denmark. ISBN: 978-87-643-0453-4.