

BoekSamenvatting

2.

.1

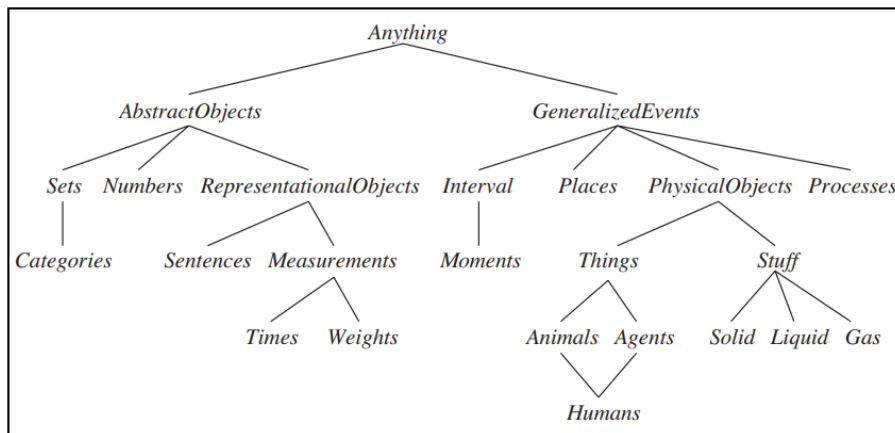
Een **Agent** is een object dat de omgeving kan herkennen met sensors en daarop reageren. Deze input wordt de **Percept** genoemd dat wordt verzameld in de **Percept sequence**. De manier dat de agent daarop gereageerd wordt bepaald en opgeslagen in de **Agent Function**. De **Agent Program** activeert deze functies.

De Agent Program kan op meerdere manieren een keuze maken. 1 daarvan is Rationeel. Een **Rational Agent** kiest bij elke stap de beste stap, waarbij niet gekeken wordt naar de volgende stap. Dat hangt af van: succesprestaties, omgeving voorkennis, acties, de input tot nu toe. Eenvoudige **reflexagents** reageren direct op waarnemingen. **Model-based reflexagents** behouden de interne toestand om aspecten van de wereld te volgen die niet duidelijk zijn in de huidige waarneming. **Goal-Based agents** handelen om hun doelen te bereiken. **Utility-based agents** proberen hun eigen verwachte 'geluk' te maximaliseren. **Learning agents** is het type agent dat kan leren van zijn ervaringen uit het verleden of leermogelijkheden heeft.

12.

.1

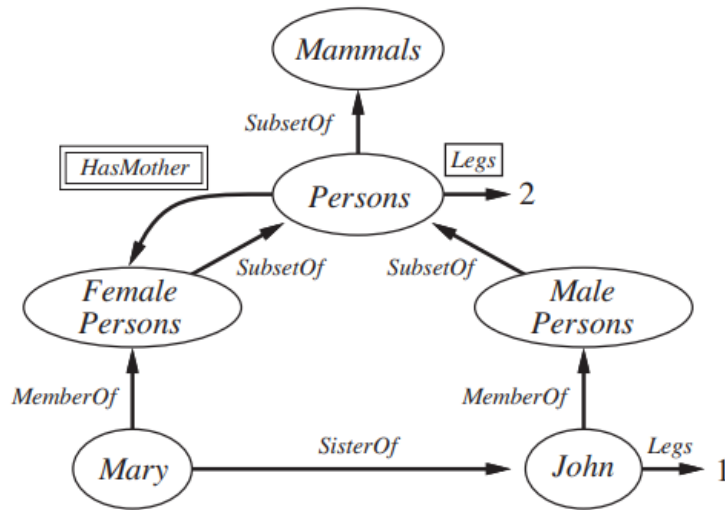
Een goeie manier om gebeurtenissen te presenteren is **Ontologie**. Zo'n ziet een schema eruit:



Bij Ontologie kan je een organisatie van objecten categoriseren. Je begint van boven naar beneden, wat betekent dat je bij de locatie begint en bedankt of er nog sublocaties zijn, anders verwerk je de objecten. Deze kunnen ook opgesplitst worden naar subobjecten/classes. Uiteindelijk kom je bij de eigenschappen van deze objecten en die eigenschappen kunnen ook subeigenschappen hebben.

Deze presentatie geeft een goed overzicht welke taken de agent moet doen, voordat je programmeert. Ook visualiseert het grote systeem door ze te categoriseren. Het is belangrijk om te onthouden dat meer kopjes een groter systeem overzichtelijker maakt dan weinig, toch kan het makkelijker door gebruikt te maken van andere netwerken zoals **Semantic**

Semantic maakt gebruik van boxes en pijlen. Elke box bestaat uit een groep/object en de pijl geven de relaties ertussen zoals MemberOf of SubsetOf, dit is een voorbeeld van een Semantic netwerk:



13.

.1 Theorieën

Agents kunnen de taak krijgen om keuzes te maken bij een **onzekerheid**. Ondanks het niet weet wat er echt daarna gebeurt. De keuze onderbouwt hij namelijk door naar alle mogelijke werelden te kijken en de kansen weergeven. Dit kost erg veel tijd, geheugen en fouten wegens het niet altijd 100% accuraat. Om deze fouten te herkennen moet je de volgende aspecten bekijken: Moeite (te veel moeite is niet goed), Theoretisch onwetend (Geen exacte uitkomsten) , Praktisch onwetend (Te veel kans op fouten). Dit is de **Probability theorie**.

Om deze fouten te voorkomen kan je een agent voorkeuren geven aan bepaalde keuzes en uitkomsten (**Utility theorie**). Het gebruik van kansen en van voorkeuren wordt de **decision theorie** genoemd. Wat bedoeld wordt dat de agent voor de keuze gaat met de hoogste succesratio (**Maximum expected utility (MEU)**).

.2 Opslaan en Formuleren

Tijdens het berekenen van alle mogelijkheden, slaat de agent alle werelden(ω) op in een **sample space**(Ω) met ieder een kans van voorkomen van 0 tot 1, waarbij elke kans bijelkaar opgeteld 1 is. Als men een gerichte uitslag wil dan wordt deze uitslag een **event** genoemd. De kans om deze event te voltooien wordt berekend door alle mogelijkheden met dezelfde uitkomst als het event, de kansen bijelkaar op te tellen. Je kan namelijk met 2 dobbelstenen op 1 manier 2 gooien, maar 2 (5+6 & 6+5) manieren 11 gooien.

Een event kan een conditie geven door te vragen wat de kans is bij een gooi van 11 ogen ($P(\text{Total}=11)$). Deze conditie kan je verbreden/verkleinen door nog meer condities te geven bijvoorbeeld door ook te vragen dat dobbelsteen 1 alleen de 5 mag hebben ($P(\text{Total}=11 \mid \text{Die1}=5)$). Dus dobbelsteen 1 moet eerst 5 gooien en dan pas maar conditie 1, Total = 11 bekeken worden. Om dus de kans te berekenen

moet je eerst de kans weten waarbij conditie 1 en 2 kloppen en dat gedeeld door de kans dat conditie 2 klopt:

$$P(a|b) = \frac{P(a \wedge b)}{P(b)}$$

Wat hetzelfde is Bayes' Rule maar dan a en b op andere volgorde:

$$P(b|a) = \frac{P(a|b)P(b)}{P(a)}.$$

Bij het formuleren van de formules is het van belang dat random condities ALTIJD met een hoofdletter moeten (P(Total=11)) en dat uitkomsten en variabelen met een kleine letter moeten: Weather {sunny, rain, cloudy, snow}. Als je voor elke mogelijkheid de kans opschrijft dan ziet het er zo uit:

“

P(Weather = sunny)=0.6

P(Weather = rain)=0.1

P(Weather = cloudy)=0.29

P(Weather = snow)=0.01

“

Dit kan je samenvatten met:

P(Weather) = <0.6, 0.1, 0.29, 0.01> (met dezelfde index als de condities <sunny, rain, cloudy, snow>)

Deze samenvatting is herkenbaar aan de dikgedrukte **P**.

Voor doorgaande variabelen (x) zoals tijd kost het opschrijven te veel moeite (een foutief aspect), dus wordt daarbij alleen de dichtheid van de variabelen opgeschreven:

P(TemperatuurInWinter = x) = Uniform[-10Celsius, 15Celsius] (x)

Bij het optellen van alle mogelijke werelden moet elke conditie keer de ander:

WilPatat, Snack (Kroket, Frikandel, Kaassoufflé, Kipstok), Dorst = 2 X 4 X 2 = 16

Ook als je meerdere malen de conditie moet gebruiken bij bijvoorbeeld een slotmachine met 4 keuzes (Cherry, Bell, Peach, Clown) en 3 rollen:

4 X 4 X 4 = 64 mogelijkheden

.3 Inzichten..

Kijkend naar kans nummers zijn er meerdere zichten:

- Een Frequentist zegt dat kans nummers correct zijn wanneer er bij genoeg testen en resultaten zijn gemaakt, de kansen het meest betrouwbaar zijn (Testen)
- Een Objectivist zal alleen kijken naar de object correct moet doen en negeert kleine uitschieters (Object kijkend)
- Een Bayesian of Subjectivist zal alleen kijken naar wat het meeste voor hem gebeurt en gaat daarvan altijd uit, de kansen van de rest maakt voor hen niet uit. (Eigen geluk)

Toch moeten ze allemaal een Subjective conclusie geven, want er zijn altijd nog kleine condities die het antwoord kunnen veranderen (“every event in life is unique, and every real-life probability that we estimate in practice is that of an event that has never occurred before.”).

Extra aantekeningen

13.

Chapter 15 deals with methods for the specific tasks of representing and updating the belief state over time and predicting the environment.

Chapter 16 covers utility theory in more depth

Chapter 17 develops algorithms for planning sequences of actions in uncertain environments.

Ontologie

Het overzichtelijk classificeren van objecten met Logica.

Stoel isA Meubel

Tafel isA Meubel