

729G78 Artificiell intelligens

Kunskapsrepresentation - Satslogik

Robin Keskisärkkä

HCS/IDA

Kunskapsrepresentation

- Introduktion
 - Wumpus-världen
- Logik
 - Satslogik
 - Predikatlogik, FOPL
- FOPL, Inferens
 - Resolution, Unifiering
- Representation av kunskap
 - Ontologier
 - Strukturerad representation/semantiska nät

Kunskapsbaserade agenter

- Generell kunskap om världen
 - Deklarativ beskrivning
- Härleda ny kunskap
 - Inferens
- Flexibla
 - Explicita mål
- Partiellt observerbara världar (jfr sökning)

Kunskapsbaserad agent

- Har en kunskapsbas som innehåller fakta och mekanismer för att dra slutsatser baserat på dessa fakta
- Agenten
 - tolkar percept vid en tid t
 - uppdaterar KB med perceptet
 - väljer handling baserat på KB
 - uppdaterar KB med handling vid t
 - utför handling (och uppdaterar tiden)

Kunskapsbaserad agent

```
KB = [ ]
```

```
t = 0
```

```
def KBagent(percept):
```

```
    tell(KB, makePerceptSentence(percept, t))
```

```
    action = ask(KB, makeActionQuery(t))
```

```
    tell(KB, makeActionSentence(action, t))
```

```
    t = t + 1
```

```
    return action
```

Modelleringsparadigm

- Tillståndsbaserade modeller
 - Tillstånd, handlingar, kostnader
 - Sökning
- Variabelbaserade modeller
 - Variabler, tilldelningar
 - Constraint Satisfaction Problem (CSP)
- Logikbaserade modeller
 - Satslogik, predikatlogik
 - Logiska formler och inferensregler

Wumpus-världen

Stank		Bris	HÅL
	Stank Bris Glitter 	HÅL	Bris
Stank		Bris	
	Bris	HÅL	Bris

Wumpus-världen

- **Percept** (Stank, Bris, Glitter, Bump, Skrik)
 - Wumpusen stinker en ruta vertikalt och horisontellt
 - Hål ger bris en ruta vertikalt och horisontellt
 - Guld glittrar
 - Bump vid krock med vägg
 - Döende Wumpus skriker

Wumpus-världen

- **Handlingar** (framåt, sväng höger, sväng vänster, ta, skjut, klättra)
 - Om agenten är vid skatten innebär “ta” att agenten har guldet
 - Om agenten står riktad mot wumpusen och skjuter dör wumpusen
 - Om agenten är vid startplatsen och klättrar kommer den ut
 - Om agenten hamnar i hål eller hos wumpusen dör den

Wumpus-världen resonemang

- Om det blåser på ruta x och ruta a, b, c och d är angränsande rutor så kan det finnas hål på a, b, c och/eller d
- Om det **inte** blåser på ruta x och a, b, c och d är angränsande rutor så kan det **inte** finnas hål på a, b, c eller d
- Om det finns ett hål på ruta x eller y och det inte finns ett hål på x så finns det ett hål på y
- ...

Wumpus-världen

Stank		Bris	HÅL
	Stank Bris Glitter 	HÅL	Bris
Stank		Bris	
	Bris	HÅL	Bris

Wumpus-världen

Stank		Bris	HÅL
	Stank Bris Glitter 	HÅL	Bris
 Stank		Bris	
	Bris	HÅL	Bris

Wumpus-världen

Stank		Bris	HÅL
	Stank Bris Glitter 	HÅL	Bris
Stank		Bris	
	Bris	HÅL	Bris

Wumpus-världen

Stank		Bris	HÅL
	Stank Bris Glitter 	HÅL	Bris
Stank		Bris	
	 Bris	HÅL	Bris

Wumpus-världen

Stank		Bris	HÅL
	Stank Bris Glitter 	HÅL	Bris
Stank		Bris	
	Bris	HÅL	Bris

Wumpus-världen

Stank		Bris	HÅL
	Stank Bris Glitter 	HÅL	Bris
Stank			
	Bris	HÅL	Bris

Wumpus-världen

Stank		Bris	HÅL
	Stank Bris Glitter 	HÅL	Bris
Stank		Bris	
	Bris	HÅL	Bris

Wumpus-världen

Stank		Bris	HÅL
	Stank Bris Glitter	HÅL	Bris
Stank		Bris	
	Bris	HÅL	Bris

Wumpus-världen

Stank		Bris	HÅL
	Stank Bris 	HÅL	Bris
Stank		Bris	
	Bris	HÅL	Bris

Wumpus-världen

Stank		Bris	HÅL
	Stank Bris	HÅL	Bris
Stank		Bris	
	Bris	HÅL	Bris

Wumpus-världen

Stank		Bris	HÅL
	Stank Bris	HÅL	Bris
Stank		Bris	
		HÅL	Bris

Wumpus-världen

Stank		Bris	HÅL
	Stank Bris	HÅL	Bris
Stank		Bris	
	Bris	HÅL	Bris

Logik

- Syntax
 - Uttryck som beskriver världen
- Semantik
 - Uttryckens betydelse
 - Utgörs av en mängd modeller
- Inferensregler
 - Ger nya beskrivningar av världen

Syntax vs semantik

- Syntax
 - Vilka formler (uttryck) är tillåtna
 - Ex matematik:
 - $x + y = 4$
 - $x \ 4 \ y = +$
- Semantik
 - Vad betyder formlerna. När är de sanna?
 - Ex matematik:
 - $x + y = 4$ är sann för $x = 1$ och $y = 3$ eller $x = 2$ och $y = 2$ etc
 - Flera olika **modeller**

Satslogik

Syntax för satslogik

- Atomära satser som kan vara sanna eller falska
 - Ofta representerade med bokstäver, t.ex. "AI är kul" kan representeras med K
- Logiska konnektiv
 - ¬ Negation (inte)
 - ∧ Konjunktion (och)
 - ∨ Disjunktion (eller)
 - ⇒ Implikation (medför)
 - ⇔ Ekvivalens
- Parenteser ()

Syntaxexempel

Formler:

A

$A \wedge B$

$\neg\neg A$

$A \vee \neg B \implies \neg A \wedge B$

$\neg (A \wedge B) \vee (A \wedge \neg B) \wedge A$

Inte formler:

$\wedge A$

$A \neg B$

Konjunktion

Syntax, exempel

K AI är kul

L Folk är lyckliga

$K \wedge L$ AI är kul **och** folk är lyckliga

Semantik

$K \wedge L$ är sant om både K och L är sanna

Disjunktion

Syntax, exempel

K AI är kul

L Folk är lyckliga

K V L AI kul **eller** folk är lyckliga.

Semantik

K V L är sant om någon av K eller L är sann (eller båda)

Negation

Syntax, exempel

L Folk är lyckliga

$\neg L$ Det är **inte** så folk är lyckliga

$L \vee \neg L$ Folk är lyckliga eller så är inte folk lyckliga

Semantik

$\neg L$ är falsk när L är sann. Annars sann

$L \vee \neg L$ är alltid sant, oavsett sanningsvärde på för L (en tautologi)

Implikation

Syntax, exempel

K AI är kul

L Folk är lyckliga

$K \Rightarrow L$ **Om** AI är kul **så** folk är lyckliga

OBS: Inget vi vet inget om vad som händer om AI inte är kul.

OBS: Implikation är inte samma som kausalitet.

Semantik

$K \Rightarrow L$ är enbart falsk om K är sann och L är falsk. Annars sann.

Ekvivalens

Syntax, exempel

K: AI är kul

L: Folk är lyckliga

$K \Leftrightarrow L$ – Folk är lyckliga **om och endast om (omm)** AI är kul. Är AI kul är folk lyckliga. Är folk lyckliga är AI kul.

Semantik

$K \Leftrightarrow L$ är sann om L och K har samma sanningsvärde. Annars falsk.

Sanningstabell

Kan lista alla sanningsvärden

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \Rightarrow B$	$A \Leftrightarrow B$
F	F	S	F	F	S	S
F	S	S	F	S	S	F
S	F	F	F	S	F	F
S	S	F	S	S	S	S

Implikation

- $A \Rightarrow B$ om båda sanna måste satsen vara sann och om A är sann och B falsk måste satsen vara falsk

A	B	$A \Rightarrow B$ Alt 1	$A \Rightarrow B$ Alt 2	$A \Rightarrow B$ Alt 3	$A \Rightarrow B$ Alt 4
F	F	F	F	S	S
F	S	F	S	F	S
S	F	F	F	F	F
S	S	S	S	S	S
		$\wedge$	B	$\Leftrightarrow$	OK

- Dessutom måste $A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$
 - $\equiv$ betyder logisk ekvivalens, sanna i samma modeller (metaspråk)

XOR

- Exklusivt eller (antingen eller), dvs bara sant då en av satserna sann
 $A \oplus B \equiv (A \vee B) \wedge \neg(A \wedge B)$

A	B	$A \oplus B$
F	F	F
F	S	S
S	F	S
S	S	F

Sanningstabell, exempel

$$P \wedge \neg Q \Rightarrow R$$

P	Q	R	$P \wedge \neg Q$	$P \wedge \neg Q \Rightarrow R$
F	F	F	F	S
F	F	S	F	S
F	S	F	F	S
F	S	S	F	S
S	F	F	S	F
S	F	S	S	S
S	S	F	F	S
S	S	S	F	S

Syntax för satslogik

Syntax byggs rekursivt

- Låt satser betecknas med stora bokstäver och formler (som består av satser) med små grekiska
- Om α och β är formler så är följande också formler:
 - $\neg\alpha$
 - $\alpha \wedge \beta$
 - $\alpha \vee \beta$
 - $\alpha \Rightarrow \beta$
 - $\alpha \Leftrightarrow \beta$
- Ex. Om $\alpha = A \wedge B$, $\beta = C \vee D$ då är $\alpha \wedge \beta = A \wedge B \wedge C \vee D$

Prioritesordning och parenteser

$\alpha \wedge \beta \vee \delta$ kan tolkas som

$$(\alpha \wedge \beta) \vee \delta \text{ eller } \alpha \wedge (\beta \vee \delta)$$

Formellt krävs parenteser men ofta skippar man dem, ex

$$\alpha \wedge \beta \vee \delta \text{ tolkas som } (\alpha \wedge \beta) \vee \delta$$

$$\alpha \vee \beta \Rightarrow \delta \text{ tolkas som } (\alpha \vee \beta) \Rightarrow \delta$$

$$\neg \alpha \wedge \beta \text{ tolkas som } (\neg \alpha) \wedge \beta$$

Prioritetsordning

$$\neg \wedge \vee \Rightarrow \Leftrightarrow$$

Några viktiga logiska ekvivalenser

- | | |
|---|----------------------------|
| 1. $\alpha \wedge \beta \equiv \beta \wedge \alpha$ | kommutativ lag |
| 2. $\alpha \vee \beta \equiv \beta \vee \alpha$ | kommutativ lag |
| 3. $\alpha \wedge (\beta \wedge \delta) \equiv (\alpha \wedge \beta) \wedge \delta$ | associativ lag |
| 4. $\alpha \vee (\beta \vee \delta) \equiv (\alpha \vee \beta) \vee \delta$ | associativ lag |
| 5. $\neg\neg\alpha \equiv \alpha$ | dubbelnegationseliminering |
| 6. $\alpha \Rightarrow \beta \equiv \neg\beta \Rightarrow \neg\alpha$ | kontraposition |
| 7. $\alpha \Rightarrow \beta \equiv \neg\alpha \vee \beta$ | eliminering av implikation |
| 8. $\alpha \wedge (\beta \vee \delta) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \delta)$ | distributiv lag |
| 9. $\alpha \vee (\beta \wedge \delta) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \delta)$ | distributiv lag |
| 10. $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$ | De Morgan |
| 11. $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$ | De Morgan |

”Sanningstabellsbevis”

		Eliminering av implikation		De Morgan		De Morgan	
A	B	$A \Rightarrow B$	$\neg A \vee B$	$\neg(A \wedge B)$	$\neg A \vee \neg B$	$\neg(A \vee B)$	$\neg A \wedge \neg B$
F	F	S	S	S	S	S	S
F	S	S	S	S	S	F	F
S	F	F	F	S	S	F	F
S	S	S	S	F	F	F	F

Översättningar, 1

Per och Pål är starka

egentligen

Per är stark och Pål är stark

E: Per är stark

Å: Pål är stark

$E \wedge Å$

Översättningar, 2

Steken ska kryddas med svartpeppar och vitpeppar eller kryddpeppar

S: Steken ska kryddas med svartpeppar

V: Steken ska kryddas med vitpeppar

K: Steken ska kryddas med kryddpeppar

$(S \wedge V) \vee K$

$S \wedge (V \vee K)$

Två olika tolkningar

Översättningar, 3

Du ska ha registrerat dig annars får du inte tenta

(annars signalerar att det är nödvändigt att registrera sig för att få tenta)

Om du inte registrerat dig får du inte tenta

T: Du får tenta

R: Du har registrerat dig

$\neg R \Rightarrow \neg T$

$T \Rightarrow R$

Konjunktiv normalform

- Standardiserat sätt att skriva formler
- En literal är en sats eller negerad sats
- En formel består bara av en konjunktion av disjunktiva literaler
 - $(a_1 \vee a_2 \vee \dots) \wedge (b_1 \vee b_3 \vee \dots) \wedge \dots$
- $\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \dots \wedge \alpha_n$ där α_i är $\beta_i \vee \delta_i$ och β eller δ är literaler, eller tomma
- T.ex. $(A \vee B) \wedge (C \vee \neg D) \wedge E$
- Alla uttryck går att skrivas om till konjunktiv normalform (eller disjunktiv normalform)

Konvertering till konjunktiv normalform

För satslogik i princip tre enkla samband

1. Eliminera implikation

$$\alpha \Rightarrow \beta \equiv \neg \alpha \vee \beta$$

2. Reducera negationernas räckvidd
de Morgan

$$\neg(\alpha \vee \beta) \equiv \neg \alpha \wedge \neg \beta \text{ och } \neg(\alpha \wedge \beta) \equiv \neg \alpha \vee \neg \beta$$

3. Konvertera till konjunktion av disjunktioner
Distributiva lagen

$$\alpha \wedge (\beta \vee \delta) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \delta)$$

$$\alpha \vee (\beta \wedge \delta) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \delta)$$

Exempel

$$A \Rightarrow (B \wedge \neg(C \wedge D))$$

1. Eliminera implikation
2. Reducera negationernas räckvidd
3. Konvertera till konjunktion av disjunktioner

$$\neg A \vee (B \wedge \neg(C \wedge D))$$

$$\neg A \vee (B \wedge (\neg C \vee \neg D))$$

$$(\neg A \vee B) \wedge (\neg A \vee \neg C \vee \neg D)$$