

Första ordningens predikatlogik – informell inledning

- Satslogikens brister
- Relationssymboler, variabler, konstanter och kvantifierare
- Funktionssymboler och termer

Klassikern

{ Alla människor är dödliga. Sokrates är^a en människa. } \models Sokrates är dödlig.

- Satslogik är inte adekvat för att uttrycka det här eftersom det handlar om objekt (t ex Sokrates) och deras egenskaper (vara människa, vara dödlig).
- Satslogik handlar bara om förhållanden mellan primitiva påståenden. De må prata om objekt och egenskaper men det berör inte semantiken: Satslogiskt är slutsatsen ovan lika med $\{A, B\} \models C$ vilket självklart är fel.
- Vad som egentligen ska uttryckas är parametriserade förhållanden:
 - (1) För alla objekt x gäller $\text{Människa}(x) \rightarrow \text{Dödlig}(x)$, vilket i synnerhet ger (1') $\text{Människa}(\text{Sokrates}) \rightarrow \text{Dödlig}(\text{Sokrates})$.
 - (2) Dessutom gäller $\text{Människa}(\text{Sokrates})$, alltså pga (1') och mha modus ponens $\text{Dödlig}(\text{Sokrates})$.

^aett tidlöst exempel i alla avseenden

Besvärligt att beskriva de tillåtna blockvärldarna satslogiskt

Ingen kub eller pyramid kan samtidigt ligga på bordet och på ett annat objekt.

- Med 2 kuber och 2 pyramider behövs 16 formler (eller deras konjunktion):
 $\neg(\text{On_table}(K_1) \wedge \text{On}(K_1, K_1)), \neg(\text{On_table}(K_1) \wedge \text{On}(K_1, K_2)), \dots$
- Mycket smidigare är det att säga att villkoret gäller för alla objekt x, y :

$$(\forall x)(\forall y)(\neg(\text{On_table}(x) \wedge \text{On}(x, y))).$$

Tecknet ' \forall ' uttalas "för alla".

- Om vi inte vill specificera hur många kuber/pyramider finns i en värld (eller de är oändligt många) är det inte ens möjligt att räkna upp alla "förbud".
- Man kan också säga att det inte känns adekvat att betrakta $\text{On_table}(K_1)$ och $\text{On_table}(K_2)$ som strukturlösa namn för skilda atomära påståenden som lika väl skulle kunna heta A och B .

Relationssymboler, variabler, konstanter och kvantifierare

- Predikatlogiska formler handlar om egenskaper hos objekt. Varje tolkning har en **domän** som definierar vilka objekt som finns.
- Satslogikens atomer blir n -ställiga **relationssymboler** ($n \in \mathbb{N}$) som tar n argument. T ex betraktas **On_table** och **Is_pyramid** lämpligtvis som enställiga relationssymboler medan **On** borde vara en tvåställig symbol.
- Semantiskt står varje relationssymbol för en relation. Om den appliceras på n objekt (tillhörande domänen) fås ett sanningsvärde beroende på om argumenten tillhör relationen eller inte. T ex skulle **On**(x, y) svara på om x ligger på y eller inte.
- **Variabler** x, y, z, \dots kan beteckna godtyckliga objekt medan **konstanter** som K_1, K_2, P_1, P_2 betecknar specifika objekt.
- **Kvantifierarna** \forall och \exists används i samband med variabler för att uttrycka att något är sant för alla objekt eller för minst ett objekt.

Funktionssymboler och termer

- Utöver variabler och konstanter kan n -ställiga **funktionssymboler** förekomma. Semantiskt är de funktioner på tolkningens domän. De tar alltså n objekt som argument och returnerar något annat objekt.
- **Termer** är uttryck som är sammansatta av variabler, konstanter och funktionssymboler. Semantiskt (dvs under en given tolkning) beskriver alltså varje term ett objekt.

Semantisk skillnad mellan relations- och funktionssymboler:

Både relationer och funktioner appliceras på objekt men i första fall är resultatet ett sanningsvärde medan det i andra fall är ett objekt tillhörande domänen.

Blockvärldsexemplet med en funktionssymbol

- Blockvärldsexemplet innehåller inget som motsvarar funktionssymboler. De enda termer som finns är således konstanterna K_1, K_2, P_1, P_2 och variabler.
- Vi kan dock t ex lägga till en enställig funktionssymbol **opposite** där **opposite**(K_i) = P_i och **opposite**(P_i) = K_i . Man kan då t ex bygga formler som

$$(\exists x)(\text{Is_pyramid}(x) \wedge \text{On}(x, \text{opposite}(x)))$$

(minst en pyramid ligger på motsvarande kub) eller

$$(\forall x)(\text{Is_cube}(x) \rightarrow \text{Is_cube}(\text{opposite}(\text{opposite}(x))))$$

(appliceras **opposite** två gånger på en kub så fås en kub).

- Termerna i formlerna ovan är x , **opposite**(x), och **opposite**(**opposite**(x)).