DD1350 Logik för dataloger

Fö 7 – Predikatlogikens semantik

Kryssprodukt av mängder

Om A och B är två mängder så är deras **kryssprodukt** $A \times B$ mängden av alla par (a,b), där $a \in A$ och $b \in B$.

Ex: $A=\{1,2\}$, $B=\{3,4\}$, $A \times B = \{ (1,3), (1,4), (2,3), (2,4) \}$

Ex: $\mathrm{N} \times \mathrm{N}$ är mängden av alla par av naturliga tal.

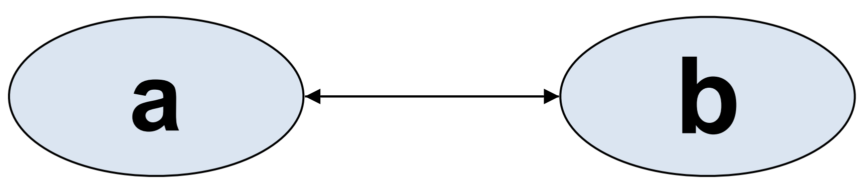
Binära relationer

En **binär relation** över A är en delmängd av $A \times A$.

Ex: ”<” (mindre än) är en binär relation över $\mathbb{N}$:

”<” = { (0,1), (0,2), (1,2), (0,3), (1,3), (2,3), ... }

Ex: Grafen nedan kan ses som relationen {(a,b), (b,a)}



n-ställiga relationer

En binär relation består av par.

Allmänt kan en mängd n-tupler, där alla komponenter är hämtade ur samma mängd A, ses som en relation över A^n.

Ex: $plus = \{ (x,y,z) \in \mathrm{N}^3 \mid x+y = z \} =$
$= \{ (0,0,0), (0,1,1), (1,0,1), (1,1,2), (0,2,2), \ldots \}$

Funktioner

En **funktion** kan ses som en specialfall av en relation.

Om F är en binär relation över A, där för varje $a \in A$ finns det max ett $b \in A$ sådant att $(a,b) \in F$, så är F en funktion.

Oftast skriver man då $F(a)=b$ (som ni är vana vid).

Mer generellt kan varje funktion $f : A^n \rightarrow A$ beskrivas som en (n+1)-ställig relation över A.

<u>Ex:</u> *plus* på föregående bild.

Unära relationer

Unära (1-ställiga) relationer kallas ibland för **egenskaper**.

I stället för att skriva

{ (a), (b), (c), ... }

så skriver vi helt enkelt

{ a, b, c, ... }

Dagens ämne

Vi har utökat **naturlig deduktion** till att även gälla predikatlogik (fö 5).

Vi vill nu definiera en semantik (modellteori) till predikatlogiken.

För **satslogik** kunde vi definiera en semantik med hjälp av sanningsvärdestabeller (fö 3).

- Naturlig deduktion för satslogik är **sund** och **fullständig** gentemot denna semantik.

Kan vi göra något liknande för predikatlogik?

Inledande exempel

Är följande formel sann?

$\forall x \forall y(x+y = y+x)$

Svar **ja** om vi tänker oss ”addition över heltal”.

Svar **nej** om vi tänker oss ”konkatenering av strängar”.

(Symbolen ’+’ används i Java i bägge betydelserna.)

dvs en formel kan vara **sann i en modell och falsk i en annan**.

Modeller

En **modell** till en predikatlogiska formel Φ specificerar:

- en mängd *A* (universum, alla objekt vi talar om)
- ett element i *A* för varje konstant i Φ
- en funktion $f : A^n \rightarrow A$ för varje funktionssymbol (med *n* argument) i Φ
- en *n*-ställig relation över *A* för varje predikatsymbol (med *n* argument) i Φ

Modeller, exempel

T.ex. $\Phi = \forall x \forall y \, (U(x) \wedge U(y) \rightarrow \neg U(x+y))$

Låt *M* vara följande modell:

- $A = \mathbb{N} = \{ 0, 1, 2, \ldots \}$
- '+' = addition
- $U(x)$ = "udda" = $\{1, 3, 5, \ldots \}$

Då är Φ sann i *M*, eftersom summan av två udda tal är jämn (icke udda).

Detta skrivs $\boxed{M \models \Phi}$

Modeller, exempel

T.ex. $\Phi = \forall x \forall y \, (\, U(x) \wedge U(y) \rightarrow \neg U(x+y) \,)$

Låt M_2 vara följande modell:

- A = { strängar över {a,b} }
- ’+’ = konkatenering av strängar
- $U(x)$ = ”strängen x har udda längd” = {*a*, *b*, *aaa*, *aab*, ... }

Då är Φ sann i M_2 , eftersom om man lägger två strängar av udda längd efter varandra får man en sträng av jämn längd.

Alltså: $M_2 \models \Phi$

Modeller, exempel

<u>T.ex.</u> $\Phi = \forall x \forall y\ (U(x) \wedge U(y) \rightarrow \neg U(x+y))$

Låt M_3 vara följande modell:

- $A = \mathbb{N} = \{ 0, 1, 2, \ldots \}$
- '+' = addition
- $U(x)$ = "x har ett udda antal siffror" = {0, 1, ...,9, 100, 101,... }

Då är Φ falsk i M_3 , eftersom t.ex. 1 och 2 har udda antal siffror, men 1+2=3 också har ett udda antal siffror.

Detta skrivs $\boxed{M_3 \not\models \Phi}$

En ändlig modell

<u>T.ex.</u> $\Phi = \forall x \forall y\ (\ U(x) \land U(y) \rightarrow \neg U(x+y)\ )$

Låt M_4 vara följande modell:

- $A = \{\ 0, 1\ \}$
- ’+’ = $\oplus$ (addition modulo 2 , ”xor”)
- $U(x)$ = ”udda” = {1}

Då är Φ sann i M_4 , eftersom $1 \oplus 1 = 0$, och 0 är jämnt (icke udda).

Alltså: $\boldsymbol{M_4} \models \Phi$

Rekursiv evaluering (def 2.18)

För att evaluera om en formel är sann eller ej i en viss modell kan man rekursivt evaluera alla dess delformler.

T.ex. för att avgöra om $M \models \Phi_1 \wedge \Phi_2$ ska vi avgöra om
$M \models \Phi_1$ och $M \models \Phi_2$.

För detaljer, se def 2.18 i boken.

Jämför satslogik

I satslogiken är en modell en **valuering**, dvs en rad i sanningstabellen, t.ex. { *p*:*F*, *q*:*T* }. (se fö 3).

Om det finns *n* variabler i den satslogiska formeln φ, så finns det 2^n rader i sanningstabellen för φ.

I predikatlogiken finns däremot **oändligt många modeller** till varje formel.

Slutna och öppna formler

En **sluten** formel (= en **sats**, eng. *sentence*) är en formel utan fria variabler, t.ex. $\forall x\,(U(x))$.

Given en modell M, så är en sats sann eller falsk i M.

En **öppen** formel innehåller fria variabler, t.ex. $U(x)$.

- För att kunna veta om $U(x)$ är sann eller falsk i M måste vi veta vad x är bundet till.
- Därför införs begreppet **omgivning** (def 2.17), vilket är en funktion från variabler till värden, t.ex. $[x \mapsto a]$.
- Detta är inte svårt men blir rätt tekniskt.

I denna kurs kommer vi fokusera på **slutna formler**.

Logisk konsekvens

Formeln ψ är en **logisk konsekvens** av $\varphi_1, \varphi_2, \ldots, \varphi_n$ om ψ är sann i alla modeller i vilka $\varphi_1, \varphi_2, \ldots, \varphi_n$ är sanna.

Detta skrivs

$$\boldsymbol{\varphi_1, \varphi_2, \ldots, \varphi_n \models \psi}$$

(Vi förutsätter att $\varphi_1, \varphi_2, \ldots, \varphi_n$, ψ är slutna formler).

Sundhet och fullständighet

Precis som för satslogik kan man visa:

Naturlig deduktion är ett **sunt** och **fullständigt** bevissystem för predikatlogik

dvs $\varphi_1, \varphi_2, \ldots, \varphi_n \vdash \psi$ omm $\varphi_1, \varphi_2, \ldots, \varphi_n \models \psi$

Detta visades 1929 av den österrikiske logikern **Kurt Gödel**.

Kurt Gödel (1906-1978)

Sundhet, följder

Predikatlogikens sundhet har några intressanta följder:

Om ψ **inte** är en logisk konsekvens av $\varphi_1, \varphi_2, \ldots, \varphi_n$ så **finns det heller inget bevis** för ψ utifrån premisserna $\varphi_1, \varphi_2, \ldots, \varphi_n$.

Dvs om vi kan hitta ett enda sätt att tolka symbolerna i formlerna så att $\varphi_1, \varphi_2, \ldots, \varphi_n$ blir sanna, men ψ falsk, så kan vi **inte bevisa** $\varphi_1, \varphi_2, \ldots, \varphi_n \vdash \psi$.

Ett (1) motexempel räcker!

Validitet och satisfierbarhet

En formel är **valid** om den är sann i alla modeller.

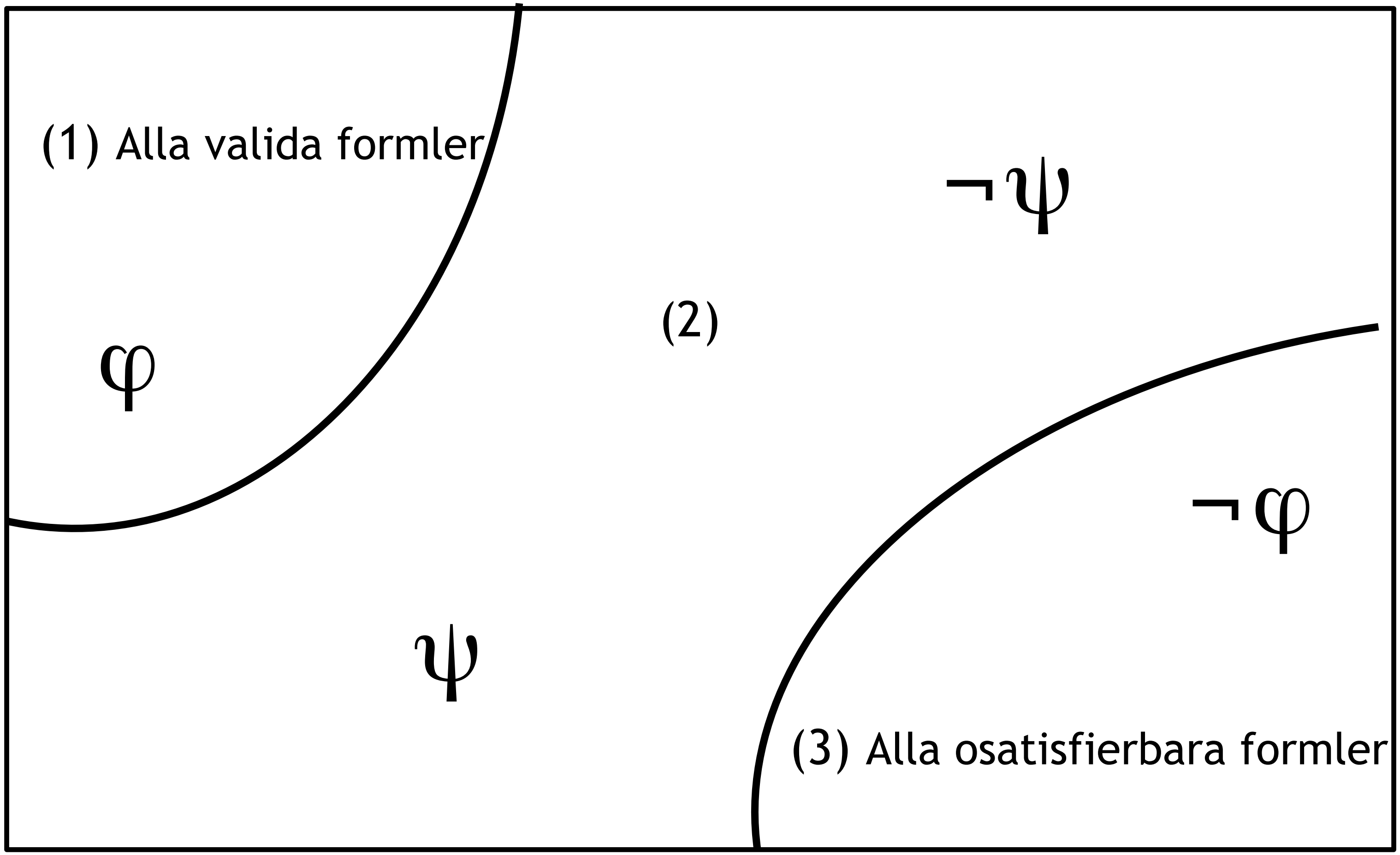
Exempel: $\forall x\ (P(x) \vee \neg P(x))$

En formel är **satisfierbar** om den är sann i någon modell.

Exempel: $\forall x\ (P(x))$

En formel är **osatisfierbar** om den är falsk i alla modeller.

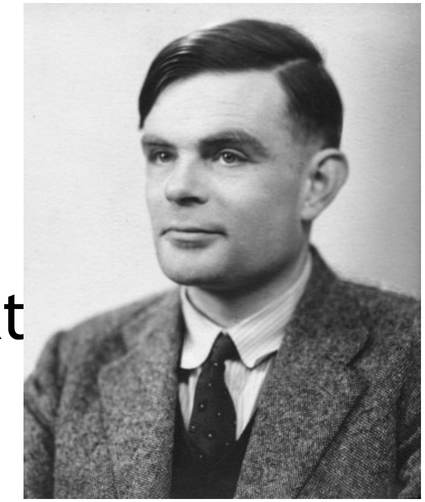
Exempel: $\forall x\ (P(x) \wedge \neg P(x))$



(1) + (2) + (3) = Alla predikatlogiska formler som finns
(1) + (2) = Alla satisfierbara formler
Om φ är valid så är $\neg\varphi$ osatisfierbar
Om ψ är varken valid eller osatisfierbar, så gäller samma sak även för $\neg\psi$

Oavgörbarhet

Predikatlogiken är **oavgörbar**. Detta visades av Alan Turing 1936.



Alan Turing (1912-1954)

dvs det **finns inget datorprogram** som **alltid terminerar och** svarar ”ja” exakt när Φ är valid, och ”nej” annars.

Man kan visa detta genom att, givet ett program P, generera en formel Φ som är valid omm P terminerar. Eftersom termineringsproblemet är oavgörbart (se fö 6) så måste då även validitetsproblemet vara oavgörbart.

Men kan vi alltid hitta ett bevis?

Om nu Φ är valid, kan vi alltid hitta ett bevis för Φ?

I princip **ja**: mängden av predikatlogiska bevis är en **rekursivt uppräkningsbar mängd** (se fö 6).

Vi kan helt enkelt (i teorin) generera (allt längre och längre) bevis. Till slut kommer vi hitta ett bevis för Φ.

Om däremot $\not\models \Phi$ så kommer bevissökningen att **aldrig terminera**.

Brist på bevis $\not\to$ motbevis

Om vi har försökt 1 timme (1 dag/1 år/...) att försöka hitta ett bevis men misslyckats, kan vi då fastslå att det inte finns något bevis?

Svar **nej**: Vissa teorem kan ha extremt långa och/eller svårfunna bevis., t.ex. "Fermats sista sats":

$$\forall n\ (\ n>2 \to \neg \exists x \exists y \exists z (x^n + y^n = z^n\ ))$$

Formuleringen är kort men beviset är på 109 sidor matematisk text (motsvarar säkert miljontals sidor bevis i naturlig deduktion).

Brist på motexempel $\not\to$ bevis

Om vi har försökt 1 timme (1 dag/1 år/...) att försöka hitta ett motexempel men misslyckats, kan vi då fastslå att det finns ett bevis?

Svar **nej**: t.ex. ”Eulers förmodan”:

$$\neg \exists x \exists y \exists z \exists w \, (x^4 + y^4 + z^4 = w^4)$$

vilket är **felaktigt**. Minsta motexemplet är dock stort:

$$95800^4 + 217519^4 + 414560^4 = 422481^4$$