Hornklausuler i satslogiken

Hornklausuler (efter logikern Alfred Horn) är ett viktigt specialfall som tillåter effektiva algoritmer och ligger till grund för regelbaserade expertsystem och logiska programspråk som Prolog[a].

- Regelbaserade expertsystem
- Terminologi och exempel
- Minsta modeller
- Algoritmiska anmärkningar

[a]Prolog använder sig av hornklausuler i första ordningens predikatlogik.

Regelbaserade expertsystem

- Regelbaserade expertsystem som representerar kunskapen om ett område mha hornklausuler för att automatiskt kunna dra slutsatser utifrån givna fakta var mest populära på 70- och 80-talet.
- Två berömda exempel är
  - MYCIN, som skulle stödja läkare vid diagnos och ordination i samband med infektiösa blodsjukdomar. MYCIN användes aldrig i praktiken men var det första riktiga regelbaserade expertsystemet och förebilden för alla som kom efter.
  - R1, som skapades på slutet av 70-talet av DEC för att konfigurera VAX-datorsystem. R1 var en kommersiell framgång, delvis pga att användningsområdet var mindre stort och komplext men också mer tekniskt än MYCINS. Matchningsalgoritmen RETE som användes i R1 är en viktig metod för att öka effektiviteten.
- Moderna expertsystem kombinerar den regelbaserade metoden med andra.

Exempel

Ett expertsystem kan sägas bestå av tre komponenter; fakta, regler och restriktioner. I det här exemplet tittar vi på ett expertsystem resonerar om kubvärlden.

- Fakta:
  - Vi vet att båda pyramiderna ligger på varsin kub.
  - Vi vet att ena pyramiden inte ligger på bordet.
- Regler:
  - Om den ena kuben ligger på den andra och ena pyramiden ligger på en kub, så ligger den andra pyramiden på bordet.
  - Om ett objekt inte är en pyramid, så är det en kub.
  - Om båda pyramiderna ligger på varsin kub, så ligger inte den första kuben på bordet.
- Restriktioner:
  - Det är omöjligt att två pyramider ligger på samma kub.
  - Det är omöjligt att ett objekt varken är en kub eller en pyramid.

Exempel

Expertsystemet utgår från den fakta den initialt har och bygger på med nya fakta genom att iterativt använda sina regler för att dra slutsatser. När till sist inga nya fakta kan härledas, kontrollerar man mängden man åstadkommit mot restriktionerna.

- Fakta 1:
  - Vi vet att den ena kuben ligger på den andra.
  - Vi vet att ena pyramiden ligger på en kub.
- Regler:
  - Om den ena kuben ligger på den andra och ena pyramiden ligger på en kub, så ligger den andra pyramiden på bordet.
- Fakta 2:
  - Vi vet att den ena kuben ligger på den andra.
  - Vi vet att ena pyramiden ligger på en kub.
  - Vi vet att den andra pyramiden ligger på bordet .

Terminologi

Definition En litteral är positiv om den är en atom och negativ om den är en negerad atom. En hornklausul är en klausul som maximalt innehåller en positiv litteral. Den har alltså formen $\neg A_1 \vee \cdots \vee \neg A_n \vee B$ (där $B = \perp$ ifall klausulen inte innehåller någon positiv litteral alls), vilket alternativt kan skrivas $(A_1 \wedge \cdots \wedge A_n) \rightarrow B$.

Ytterligare terminologi: Klausulen sägs vara

- ett faktum om den endast består av en positiv litteral B (dvs $n = 0$ och därmed $A_1 \wedge \cdots \wedge A_n \equiv \top$),
- en regel om den innehåller en positiv samt minst en negativ litteral och
- en sammansatt negation om den endast består av negativa litteraler.

En speciell sammansatt negation är $\perp$ (som har ringa praktisk betydelse).

Uppgift 1

Vilka av nedstående klausuler är hornklausuler?

$\varphi_1 = (\perp)$	$\varphi_5 = (A \vee \neg B \vee \neg C)$
$\varphi_2 = (A \vee B)$	$\varphi_6 = (A)$
$\varphi_3 = (\neg A \vee B)$	$\varphi_7 = (\neg B)$
$\varphi_4 = (\neg A \wedge B)$	$\varphi_8 = (\neg A \wedge \neg B)$

Lösning 1

Av klausulerna på föregående oh-bild var alla utom φ_2, φ_4 och φ_8 hornklausuler.

Exempel ur blockvärldarna

- Positiva litteraler som t.ex. $\mathsf{Is\_cube}(K_1)$ och $\mathsf{Is\_pyramid}(P_1)$ är fakta. (Intuition: I en databas som beskriver tillåtna världar skulle de representera fakta som alltid gäller.)
- Implikationen $\mathsf{On}(K_2, K_1) \land \mathsf{On}(P_1, K_2) \to \mathsf{On\_table}(P_2)$ är en regel. (Intuition: I databasen skulle den representera en regel som gör det möjligt att dra slutsatsen $\mathsf{On\_table}(P_2)$ om man redan vet att $\mathsf{On}(K_2, K_1)$ och $\mathsf{On}(P_1, K_2)$ gäller.)
- En sammansatt negation är t ex $(\mathsf{On\_table}(P_1) \land \mathsf{On}(P_1, K_1)) \to \bot$ vilket är ekvivalent med $\neg(\mathsf{On\_table}(P_1) \land \mathsf{On}(P_1, K_1))$.
(Intuition: I databasen skulle den användas för att beskriva en förbjuden situation.)

Minsta modell

Definition För en formelmängd Φ låt

$$\mathsf{atCons}(\Phi) = \{A \in \mathbf{P} \mid \Phi \models A\} = \{A \in \mathbf{P} \mid v(A) = 1 \text{ för alla } v \in \mathrm{Mod}(\Phi)\}$$

vara mängden av Φs atomära konsekvenser. Tolkningen $v_\bot^\Phi$ definieras

$$v_\bot^\Phi(A) = \begin{cases} 1 & \text{om } A \in \mathsf{atCons}(\Phi) \\ 0 & \text{annars.} \end{cases}$$

Observation: Om $v_\bot^\Phi$ är en modell av Φ så är den dess minsta modell i den bemärkelsen att alla övriga modeller v uppfyller $v(A) \geq v_\bot^\Phi(A)$ för alla $A \in \mathbf{P}$ (och om $v_\bot^\Phi$ inte är en modell av Φ så har Φ ingen minsta modell).

Teorem Om en mängd Φ av hornklausuler har en modell, dvs om Φ inte är motsägelsefullt, så är $v_\bot^\Phi$ en modell av Φ (och således dess minsta modell).

Att beräkna den minsta modellen

Låt Φ vara en mängd av hornklausuler.

- Att beräkna atCons(Φ) och således $v_{\perp}^{\Phi}$ är enkelt: Börja med Φ som databas. Så länge databasen innehåller en regel $(A_1 \wedge \cdots \wedge A_n) \to B$ och fakta $A_1, \ldots, A_n$ men inte B, lägg till B som ett nytt faktum. Mängden av alla fakta som till slut finns i databasen är lika med atCons(Φ).
- Undantag: Om det finns en sammansatt negation $\neg(A_1 \wedge \cdots \wedge A_n)$ samt fakta $A_1, \ldots, A_n$ i databasen så är atCons$(\Phi) = \emptyset$ och Φ motsägelsefullt.
- Exempel: $\Phi = \{A_1, A_1 \to A_2, (A_1 \wedge A_2) \to A_3, (A_3 \wedge A_4) \to A_5\}$ ger atCons$(\Phi) = \{A_1, A_2, A_3\}$.
- Om den sammansatta negationen $\neg(A_1 \wedge A_5)$ läggs till så ändras inget i exemplet ovan men $\neg(A_1 \wedge A_3)$ har till följd att Φ blir motsägelsefullt.
- Utifrån den minsta modellen kan man lätt beräkna alla modeller. (Hur?)

Algoritmiska anmärkningar

- Enhetsresolution är fullständig för mängder Φ av hornklausuler: Om Φ' är mängden av alla klausuler som kan fås mha upprepad enhetsresolution så gäller antingen $\perp \in \Phi'$ (Φ är motsägelsefullt) eller $\Phi' \cap \mathbf{P} =$ atCons(Φ).
- Som vi redan vet kan enhetsresolution implementeras på så sätt att regler som har används ersätts med resolutionsresultatet (som innehåller en litteral mindre). Det behövs alltså maximalt length(Φ) resolutionssteg för att beräkna Φ'.