5. tetel Mátrix fogalma, műveletek, determináns, rang. Speciális mátrixok, inverz. Mátrix, mint lineáris
transzformáció. Sajátérték, sajátvektor.

Mátrix fogalma:

Mátrixok

Definíció

Egy m sorral és n oszloppal rendelkező számtáblázatot $m \times n$-es mátrixnak nevezünk.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \qquad A \text{ elemei: } a_{ij} \qquad A = (a_{ij})$$

Az összes $m \times n$-es mátrix halmazát $\mathcal{M}_{m \times n}$-nel jelöljük.

Mátrixműveletek

1. Mátrixok összeadása: Csak azonos típusú mátrixokat tudunk összeadni. Legyenek A= (aij),B= (bij),C= (cij) m×n-es mátrixok. Ekkor C=A+B, ha cij=aij+bij; i= 1,...,m, j= 1,...,n.

2. Mátrixok skalárral való szorzása: Elemenként végezzük, azaz ha λ∈R, A= (aij)∈Mm×n, akkor λA= (λaij)∈Mm×n. Speciálisan: ha A és B sor-, vagy oszlopvektorok, akkor a fenti 2 művelet éppen a vektorok szokásos összeadása és skalárral való szorzása.

3. Mátrixszorzás: Legyen A= (aij) m×k, B= (bij) k×n típusú mátrix. Ekkor A és B szorzata az a C= (cij) m×n típusú mátrix, amelyre

$$c_{ij} = \sum_{r=1}^{k} a_{ir} b_{rj}.$$

A mátrixszorzás tulajdonságai:

Ha A m×n típusú, akkor Em·A=A és A·En=A.

Legyenek A,B mátrixok és tegyük fel, hogy létezik AB. Ha λ∈R tetszőleges, akkor λ(AB) = (λA)B=A(λB).

Ha A,B,C olyan mátrixok, hogy AB és BC létezik, akkor (AB)C=A(BC). Azaz a mátrixszorzás asszociatív. Ha A és B azonos típusú mátrixok és létezik AC, akkor BC is létezik és (A+B)C=AC+BC. Azaz teljesül a disztributivitás.

A mátrixszorzás nem kommutatív, azaz általában AB! = BA.

Legyen A egy m×n-es mátrix. Azt az AT-vel jelölt n×m-es mátrixot, amelynek sorai az A oszlopai A transzponáltjának nevezzük.

A transzponálás tulajdonságai

(AT)T=A (azaz a transzponálás involutív művelet)

A transzponálás és a mátrixszorzás kapcsolata: (AB)T=BT·AT.

Determinánsok:

Legyen n N és jelölje σ az{1,2,...,∈ n}halmaz egy permutációját, azaz legyen σ: {1,2,...,n}→{1,2,...,n},i→σ(i) bijektív függvény. (Itt σ(i) jelöli a permutációban az i. helyen álló elemet.) Azt mondjuk, hogy a σ permutációnál az i és j elem inverzióban áll, ha i<j és σ(i)> σ(j). Egy σ permutáció páros, ha benne az inverzióban álló párok száma páros és páratlan, ha ez a szám páratlan.

Legyen $A = (a_{ij})$ egy $n \times n$-es kvadratikus mátrix. Az A n^2 eleméből válasszunk ki úgy n elemet, hogy minden sorból és oszlopból pontosan egyet választunk. A kiválasztott elemek alakja:

$$a_{1\sigma(1)}, a_{2\sigma(2)}, \ldots, a_{n\sigma(n)}.$$

Az A mátrix determinánsa:

$$\det(A) = |A| = \sum_{\sigma} \varepsilon(\sigma) a_{1\sigma(1)} a_{2\sigma(2)} \cdots a_{n\sigma(n)}.$$

Ez az összeg $n!$ tagú. Itt: $\varepsilon(\sigma) = \begin{cases} 1, & \text{ha } \sigma \text{ páros,} \\ -1, & \text{ha } \sigma \text{ páratlan.} \end{cases}$

Determináns szemléletes jelentése

A determináns szemléletes jelentése

- másodrendű (2x2-es) determináns: a determináns sorai, mint vektorok által kifeszített parallelogramma előjeles területe

$$|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

- harmadrendű (3x3-as) determináns: a determináns sorai, mint vektorok által kifeszített paralelepipedon előjeles térfogata

Vektorrendszer rangja

Vektorrendszer rangja

Definíció

Legyen $\mathcal{A}$ a vektortér egy vektorrendszere. Az $\mathcal{A}$ vektorrendszer rangja az általa generált altér dimenziója:

$$\text{rang}(\mathcal{A}) = \dim(\mathcal{L}(\mathcal{A})).$$

Példa: $V = \mathbb{R}^3$, legyen $\mathcal{A} = \{u, v, w\}$, ahol

$$u = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad v = \begin{pmatrix} 1 \\ 3 \\ 0 \end{pmatrix}, \quad w = \begin{pmatrix} 3 \\ 5 \\ 2 \end{pmatrix}.$$

Mivel $w = 2u + v$, ezért w benne van a másik 2 vektor által generált altérben. Viszont u és v lineárisan független, ezért $\text{rang}(\mathcal{A}) = 2$.

Megjegyzés: Legyen V n-dimenziós vektortér, $\mathcal{A} = \{v_1, \ldots, v_m\} \subset V$. Ekkor $\text{rang}(\mathcal{A}) \le n$ és $\text{rang}(\mathcal{A}) \le m$.

Tétel

Egy vektorrendszer rangja nem változik, ha bármely eleméhez hozzáadjuk a többi elem tetszőleges lineáris kombinációját.

A V vektortér egy nemüres W részhalmazát V alterének nevezzük, ha W maga is vektortér, azaz zárt a vektor összeadásra és a skalárral való szorzásra.

Speciális mátrixok (szimmetrikus, ferdeszimmetrikus, kvadratikus)

Definíció

Legyen A egy n-edrendű kvadratikus mátrix.

- A szimmetrikus, ha $A^T = A$,
- A ferdeszimmetrikus, ha $A^T = -A$.

Példák:

$$A = \begin{pmatrix} 2 & -3 & 4 \\ -3 & -1 & 7 \\ 4 & 7 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 0 & 2 & 1 \\ -2 & 0 & -5 \\ -1 & 5 & 0 \end{pmatrix}$$

Állítás – szimmetrikus és ferdeszimmetrikus mátrixok tulajdonságai

- Ferdeszimmetrikus mátrix főátlójában 0-k állnak.
- Szimmetrikus mátrixok összege szimmetrikus.
- Ferdeszimmetrikus mátrixok összege ferdeszimmetrikus.
- Szimmetrikus mátrixok szorzata nem feltétlenül szimmetrikus. De ha A és B szimmetrikus és felcserélhetőek (azaz $AB = BA$), akkor AB is szimmetrikus.

Mátrixok inverze

Mátrixok inverze

Definíció

Azt mondjuk az A n-edrendű négyzetes mátrixról, hogy invertálható, vagy létezik az inverze, ha létezik olyan B n-edrendű kvadratikus mátrix, hogy

$$AB = BA = E_n.$$

Tétel

Ha A invertálható, akkor az inverze egyértelmű. Jele: A^{-1}.

Példa:

$$A = \begin{pmatrix} 4 & 3 \\ 7 & 5 \end{pmatrix} \qquad A^{-1} = \begin{pmatrix} -5 & 3 \\ 7 & -4 \end{pmatrix}$$

Állítás – a mátrixinvertálás tulajdonságai

- Ha A invertálható, akkor A^{-1} is az és $(A^{-1})^{-1} = A$.
- Ha A és B invertálható és létezik AB, akkor ez is invertálható és $(AB)^{-1} = B^{-1}A^{-1}$.
- Ha A invertálható, akkor A^T is az és $(A^{-1})^T = (A^T)^{-1}$.

Mátrix, mint lineáris transzformáció.

Legyen V vektortér R felett. φ:V→V lineáris transzformáció, ha additív, azaz ∀u,v∈V: φ(u+v) =φ(u) +φ(v); homogén, azaz ∀v∈V, λ∈R: φ(λv) =λφ(v).

Lineáris transzformációk esetén nullvektor képe nullvektor.

Példák:Forgatások, tükrözések,λ-nyújtások.

Vetítések, pl.R3 egy rögzített síkjára merőlegesen.

Identikus transzformáció: φ(v) =v,∀v∈V.

Egy lineáris transzformációt egyértelműen meghatároz egy bázison való hatása, azaz ha B= (b1,b2,...,bn) bázisa V-nek,w1,w2,...,wn pedig tetszőleges vektorai a vektortérnek, akkor egyértelműen létezik olyan φ lineáris transzformáció, hogy φ(bi) =wi. Továbbá ha v=λ1b1+λ2b2+···+λnbn, akkor ennek φ általi hatása:φ(v) =λ1w1+λ2w2+···+λnwn.

Legyen V egy n-dimenziós valós vektortér,B= (b1,b2,...,bn) bázisa V-nek, tekintsünk továbbá egy φ:V→V lineáris transzformációt. Ekkor φ-nek a B bázisra vonatkozó mátrixa az az n×n-es mátrix, amelynek i-edik oszlopában φ(bi)-nek a B bázisra vonatkozó koordinátái állnak.
Példa: Legyen φ:R2→R2, (x,y)→φ(x,y) = (2x−y,−12x+ 3y). φ mátrixa a természetes bázisban. φ(e1) =φ(1,0) = (2,−12), φ(e2) =φ(0,1) = (−1,3), ezért φ mátrixa ebben a bázisban

$$A_\varphi = \begin{pmatrix} 2 & -1 \\ -12 & 3 \end{pmatrix}$$

φ mátrixa ab1= (1,1),b2= (0,−1) bázisban. Ekkorφ(b1) = (1,−9)ésφ(b2) = (1,−3). Ezeket a vektorokat a (b1,b2) bázisban kell felírnunk:φ(b1) = (1,−9) = 1·b1+ 10·b2, φ(b2) = (1,−3) = 1·b1+ 4·b2.Ezért a mátrix:

$$[A_\varphi]_{(b_1,b_2)} = \begin{pmatrix} 1 & 1 \\ 10 & 4 \end{pmatrix}.$$

Lineáris transzformáció mátrixának alkalmazása

Állítás

Egy lineáris transzformáció különböző bázisokra vonatkozó mátrixainak megegyezik a rangja és a determinánsa.

Állítás

Ha φ mátrixa a $\mathcal{B}$ bázisban A, akkor φ hatása egy v vektoron: $\varphi(v) = Av$.

Példák: Forgatások és tükrözések mátrixa $\mathbb{R}^2$-ben a természetes bázisban:

$$\text{rot}_\alpha = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \qquad \text{refl}_\alpha = \begin{pmatrix} \cos(2\alpha) & \sin(2\alpha) \\ \sin(2\alpha) & -\cos(2\alpha) \end{pmatrix}$$

Így például a $v = \binom{2}{6}$ vektort az origó körül 60°-kal pozitív irányba elforgatva:

$$\begin{pmatrix} \cos 60^\circ & -\sin 60^\circ \\ \sin 60^\circ & \cos 60^\circ \end{pmatrix}\begin{pmatrix} 2 \\ 6 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{pmatrix}\begin{pmatrix} 2 \\ 6 \end{pmatrix} = \begin{pmatrix} 1-3\sqrt{3} \\ \sqrt{3}+3 \end{pmatrix}.$$

Sajátérték, sajátvektor.

Legyen φ:V→V lineáris transzformáció. Egy nem-nulla v∈V vektort φ sajátvektorának hívunk, ha ∃λ∈R: φ(v) =λv. Ekkor λ-t φ v-hez tartozó sajátértékének mondjuk.
Ha v sajátvektora φ-nek, akkor a hozzá tartozó sajátérték egyértelmű. Ha λ sajátérték, akkor a hozzá tartozó sajátvektorok halmaza altér: Lλ:={v∈V|φ(v) =λv} altér V-ben: a λ-φ(v) =λv} altér V-ben: a λ-hoz tartozó saját altér.
Egy φ lineáris transzformáció karakterisztikus polinomján a det(A−λEn) n-edfokú polinomot értjük, ahol n a tér dimenziója, A pedig φ mátrix a tetszőleges bázisban.
Ennek gyökei éppen φ sajátértékei.

A problémaredukciós reprezentáció és az ÉS/VAGY gráfok. Ismeretreprezentációs technikák, bizonytalanság-kezelés (fuzzy logika). A rezolúció kalkulus. A logikai program és az SLD rezolúció. A logikai programozás alapvető módszerei.

A problémaredukciós reprezentáció és az ÉS/VAGY gráfok

Problémamegoldás redukcióval

Gyakran előfordul, hogy egy problémát úgy próbálunk megoldani, hogy több külön-külön megoldandó részproblémára bontjuk. Ha a részproblémákat megoldjuk, az eredeti probléma megoldását is megkapjuk. A részproblémák megoldását további részek megoldására vezetjük vissza, egészen addig, amíg csupa olyan problémához nem jutunk, amelyeket egyszerűségüknél fogva már könnyedén meg tudunk oldani. A probléma megoldásnak ezt a módját problémaredukciónak nevezzük.

- Először is le kell írni az eredeti problémát, jelöljük ezt most p-vel.
- Egy probléma részproblémákra bontása során a nyert részek az eredeti problémához hasonló, de annál egyszerűbb problémák. Jelöljük az így nyert problémahalmazt $\mathcal{P}$-vel. Természetesen $p \in \mathcal{P}$.
- $\mathcal{P}$ problémáinak összegyűjtése során törekszünk arra, hogy legyenek közöttük olyanok, melyeket meg tudunk oldani, vagy ismerjük a megoldásukat. Ezek a problémák az ún. **egyszerű problémák**. Az egyszerű problémák halmazát $\mathcal{E}$-vel jelöljük. $\mathcal{E} \subset \mathcal{P}$, hiszen $p \notin \mathcal{E}$, különben nincs megoldandó feladat.

- Meg kell még adni a problémákat egyszerűsítő, illetve részekre bontó **redukciós operátorokat**. Egy redukciós operátor egy problémához azokat a (rész)problémákat rendeli hozzá, melyek egyenkénti megoldásával a probléma megoldása is előáll. Jelöljön a **redukciós operátorok** $\mathcal{R}$ véges **halmazából** r egy operátort. Ekkor

  $$\mathrm{Dom}(r) = \{\, q \mid q \in \mathcal{P} \setminus \mathcal{E} \text{ és } r\textit{-alkalmazásának-előfeltétele}(q) \,\}$$

  és

  $$\mathrm{Rng}(r) = \{\, r(q) = \{q_1, \ldots, q_m\} \mid q \in \mathrm{Dom}(r) \text{ és } q_1, \ldots, q_m \in \mathcal{P} \}.$$

  Tehát egy redukciós operátor egy-egy problémához $\mathcal{P}$ egy-egy részhalmazát rendeli, így értékkészlete $\mathcal{P}$ hatványhalmazának valamely részhalmaza.

4.1. DEFINÍCIÓ. Legyen p egy probléma. Azt mondjuk, hogy a p problémát **problémaredukciós reprezentációval** írtuk le, ha megadtuk a $\langle \mathcal{P}, p, \mathcal{E}, \mathcal{R} \rangle$ négyest, azaz

- a megoldandó $p \in \mathcal{P}$ problémát,
- a $\mathcal{P} \neq \emptyset$ halmazt, a p problémához hasonló problémák halmazát,
- az egyszerű problémák $\mathcal{E} \subset \mathcal{P}$ halmazát és
- a redukciós operátorok $\mathcal{R} \neq \emptyset$ véges halmazát.

Jelölése: $\langle \mathcal{P}, p, \mathcal{E}, \mathcal{R} \rangle$.

Az ÉS/VAGY gráfot olyan problémák reprezentálására alkalmazzuk, ahol a feladat felírható logikai állítások konjunkciójaként és diszjunkciójaként. Tehát a probléma végrehajtásánál a probléma szétválhat egy egy valamilyen formában eltérő problémára vagy alfeladatra (vagy kapcsolat), vagy egy adott probléma feladatit egyszerre kell megoldani (és kapcsolat). A gráfokkal további vizsgálatokat, átalakításokat vihetünk a feladatba, így könnyebben tudjuk elemezni a rendszerünket.

Ismeretreprezentációs technikák, bizonytalanság-kezelés (fuzzy logika)

Szemantikus háló

A szemantikus hálókat eredetileg Quillian fejlesztette ki a 70-es évek végén angol mondatok jelentésének leírására.

Az emberek a világ objektumait hierarchikus kapcsolataikkal együtt tárolják, az azokra jellemző tulajdonságokat pedig a lehető legmagasabb absztrakciós szintet társítják ebben a hierarchiában.

A szemantikus háló:

- irányított, címkékkel ellátott gráf, ahol
  - a csúcsok – objektumokat, tulajdonságaikat, a tulajdonságok értékeit tartalmazzák
  - az összekötő élek – a csúcsok közötti relációkat fejezik ki

- grafikusan ábrázolja az objektumokat, jellemzőiket, és a köztük fennálló függéseket
- azokon a területeken, ahol a következtetés az elemek valamilyen rendszerén alapul (taxonómia) a szemantikus háló természetes ábrázolási mód, hiszen segítségével kifejezhetjük a koncepciók közötti hierarchiát, az objektumok közötti távolságot.
- a csúcsok kettős természetűek: egyszerű predikátumok vagy struktúrák, osztályjellegűk vagy egyedek, koncepciók vagy individuumok
- általában olyan problémák reprezentálására alkalmas, amelyekben az univerzum szerkezetére valamilyen taxonómikus hierarchia jellemző, vagyis az egyedek egymásba ágyazott osztályok rendszerében helyezkednek el, az univerzum egyedei és osztályai között kiterjedt, de logikailag egyszerű kapcsolatrendszer áll fenn.
- A háló lehetővé teszi az osztály tulajdonságainak öröklését. Az osztályhoz tartozást az ez egy (is a) (részben rendezési) reláció jelöli, és a reláció révén az osztály tulajdonságait öröklik az egyedei.
- Az alkalmazott következtetés a mintaillesztés.

Keretalapú és szabály alapú rendszerek

Minsky a 70-es években fejlesztette ki a keretalapú reprezentációs módszert (keret=frame).

A frame vagy tudáskeret egy tipikus szituációt ábrázol; minden átélt szituáció a hozzátartozó viselkedéssel együtt egy gondolati egységet képez az amit agyunk keretként tárol.

Új szituációba kerülve a már átélt szituációkhoz illesztve alakítjuk ki viselkedésünket, elvárásainkat. Miközben a korábbi keretet esetleg módosítjuk vagy új keretekkel bővítjük.

A keret:

Valamely fogalom strukturált szimbolikus modellje; a fogalom számunkra fontos tulajdonságait egybefoglaló struktúra, amelyet a fogalom neve foglal egységbe.

Tulajdonságok (attribútumok):

A tulajdonságokat (attribútumokat) a keret bejegyzései (slot) nevezik meg, amelyek megadhatják az attribútum értékét, annak alapértelmezését, forrását, az érték változásakor végrehajtandó eljárásokat, ún. démonokat, különböző meta-leírásokat és további járulékos információkat. Attribútum lehet akár egy másik keret is. Kereteket relációkkal lehet összekapcsolni. Különös szerepe van a hierarchikus kapcsolatoknak, amelyek mentén bizonyos attribútumok és azok jellemzői öröklődhetnek.

Öröklődés:

- A példány rendelkezik mindazokkal az attribútumokkal, amelyek az örökítő osztályokban szerepelnek.
- Egy attribútum értékét az öröklési útvonalon mindig az első, a példányhoz legközelebbi bejegyzés határozza meg.
- Egy attribútumhoz több öröklési útvonal tartozik és a példány eltérő értékeket örököl.

Leíró logikák

- ismeretábrázolási nyelvcsalád
- fogalom, individuum (egyed), szerep
- fogalom: individuumok halmazának reprezentálására
- szerep: indiviuumok közötti bináris reprezentáció

Alapelvek 1, 2:

- A fogalom és a szerep strukturális leírásában konstruktorok vesznek részt. A fogalom és a szerep leírásához egy szemantika kapcsolódik az interpretáción keresztül. A különböző műveleteket ezen szemantikával összhangban hajtjuk végre.
- Az ismereteket különböző szinteken vesszük figyelembe. A fogalmak, szerepek ábrázolása és a műveleteik a terminológia szintjén, az individuumok leírása és műveleteik a tények és a hozzárendelések szintjén jelennek meg. A szakirodalomban a terminológia szintjét TBox-nak, a tények és hozzárendelések szintjét Abox-nak nevezik.

Bizonytalanság-kezelés (fuzzy logika)

- Szakértő rendszerek készítésekor a tárgyköri szakértők ismerete nehezen reprezentálható, nehezen formalizálható
- az ismeretek reprezentálása során használhatunk olyan adatokat, illetve tudást, amely csak bizonyos valószínűséggel biztos, az ilyen adatok kezelését nevezzük bizonytalanságkezelésnek.
- Bizonytalan adatok kezelése egy szakértői rendszerben azokban az esetekben indokolt, amikor a rendelkezésre álló információ
    - hiányos vagy
    - nem teljesen megbízható vagy
    - pontos lenne, de a reprezentáló nyelv nem elég precíz vagy
    - ellentmondásos

- Módszerek, modellek osztályozása
    1. numerikus modellek
        - klasszikus valószínűségszámítás
        - Fuzzy logika
    2. szimbolikus modellek
        - nem monoton logikák
    3. heurisztikus módszerek
- Felmerülő problémák
    - hogyan reprezentáljuk a bizonytalan információt?
    - hogyan kombináljunk több bizonytalan információt (and, or, not)?
    - a következtetés problémája

2. Fuzzy logika (kialakulása, tagsági függvény)

- 60-as évek közepén Zadeh dolgozta ki a fuzzy halmazelméletet a nyelvi fogalmakban rejlő pontatlanság matematikai kezelésére
- Zadeh bevezette a parciális tagság fogalmát, annak kifejezésére, hogy bizonyos objektumok jobban beletartoznak egy halmazba, mások kevésbé
- ezt a parciális tagságot egy [0,1] intervallumbeli számmal jellemezte, ahol az 1 azt jelenti, hogy az objektum benne van a halmazban, a 0 pedig hogy nincs benne, míg a kettő közötti érték azt a meggyőződésünket, hogy milyen mértékben tartozik az adott objektum a halmazhoz (nagyon, kissé, eléggé, meglehetősen)

Tagsági függvény: egy adott halmazhoz tartozás fokát fejezi ki egy [0,1] intervallumbeli számmal. A köznapi nyelvben kissé, eléggé, meglehetősen, nagyon, stb, módosítószavakkal fejezzük ki.

Előnyei:

- Szemlélete közel áll a napi valóságszemléletünkhöz.
- E rendszerleírás egyszerűbb, mint más numerikus modell esetén.
- Előnyösen alkalmazható hiányos adatokkal dolgozó, bonyolult feladatok esetén.
- A fuzzy bizonyosságokkal könnyű számolni.

Hátrányai:

- Elmélete még nem teljesen megalapozott.

- Kombinációs függvények(egymást kizáró halmazok együttes bizonyossága a két halmaz bizonyosságának minimuma és nem nulla).

A rezolúció kalkulus

Egy cáfolás teljes és helyes következtető algoritmus, tehát arra nem alkalmas, hogy felsoroljuk egy tudásbázis összes logikai következményét, de arra igen, hogy adott következtetés helyességét ellenőrizzük.

A rezolúció formulák konzisztenciájának az eldöntésére alkalmas, és az ellentmondásra vezetés módszerét használva használható logikai következtetésre. Itt is az alfa és nem béta formulát vizsgáljuk, ellentmondást szeretnénk levezetni.

Először hozzuk a formulát itt is konjunktív normálformára. Ezután alkalmazzuk bármely teljes és helyes gráfkereső algoritmust a következő keresési téren:

- keresési tér: alfa és nem béta konjunktív normálformájában szereplő literálokból álló klózokból álló halmazok
- kezdőállapot: alfa és nem béta konjunktív normálformájának klózaiból álló halmaz
- operátor: a rezolúciós szabály. Legyen l_1 vagy ... vagy l_k és m_1 vagy … v m_n olyan klózok amelyekre az l_i és m_j literálok éppen egymás tagadásai (pl. l_i = A, m_j = nem A). Ebből állítsuk elő a C = l_1 vagy … vagy l_{i-1} vagy l_{i+1} vagy … vagy l_k vagy m_1 vagy m_{j-1} v m_{j+1} vagy … vagy m_n klózt, majd C-ből állítsuk elő C'-t úgy, hogy az egyiket hagyjuk meg. Adjuk hozzá C'-t a klózok halmazához.
- célállapot: bármely állapot amely az üres klózt (kielégíthetetlen klóz, egy literál sincs benne) tartalmazza

Először is a rezolúciós keresés (teljes és helyes gráfkereséssel) mindig terminál, mert a keresési tér véges (csak véges literál van). Egy formula rezolúciós lezártján azoknak a klózoknak a halmazát értjük, amelyek a formulából levezethetők a rezolúciós szabállyal, azaz a keresési tér azon részhalmazát, amely a kezdőállapotból elérhető.

Ha a célállapotot elérjük, azaz ellentmondásra jutottunk, akkor ez igazolja az alfa |= béta logikai következményt. Ez abból következik, hogy a rezolúciós szabály helyes, hiszen, mivel l_i és az m_j egymás negáltjai, valamelyik biztosan hamis, tehát elhanyagolható. De a fenti C klóz éppen azt mondja, hogy vagy l_i vagy m_j elhagyható.

Másreszt a rezolúció cáfolás teljes, tehát ha alfa |= béta akkor levezeti az ellentmondást. Mivel az algoritmus mindig terminál, a cáfolás teljességét máshogyan fogalmazhatjuk meg, hogy ha nem értünk el célállapotot (azaz nem vezettünk le ellentmondást) termináláskor, akkor az alfa és nem béta formula kielégíthető.

Egy nulladrendű klózhalmaz kielégíthetetlen, ha bármely interpretáció mellett van a halmaz elemei között legalább egy h igazságértékű klóz. Mivel egy klóz pontosan akkor h igazságértékű egy interpretációban, ha minden literálja is h , így azok és csak azok az interpretációk teszik h igazságértékűvé, amelyekben a klóz negált ítéletváltozói i , a negálatlanok pedig h igazságértéket vesznek fel. Egy véges klózhalmaz kielégíthetetlenségének vizsgálatára alkalmas eszköz a klózhalmazban szereplő ítéletváltozók valamely rögzített sorrendje (bázisa) mellett felépített szemantikus fa, ami felsorolja az összes interpretációt. A szemantikus fa éleihez literál címkéket rendeltünk. Az X címke azt jelentette, hogy az X ítéletváltozót i igazságértékkel, a ¬X címke pedig azt, hogy X -et h igazságértékkel interpretáltuk.

Legyen S egy véges klózhalmaz, és rögzítsük S ítéletváltozóinak egy bázisát. Építsük fel a bázishoz a szemantikus fát. S egy C klózának illesztése a szemantikus fára azt jelenti, hogy megkeressük az(oka)t az ága(ka)t, amelyekben minden C -beli literálnak éppen a komplemense valamely él címkéje (egy literál komplemense a literállal azonos alapú, de ellentétesen negált literál, azaz X komplemense ¬X és ¬X komplemense X). Az ezeken az ágakon megadott interpretációkban a C klóz h igazságértékű. Illesszük most az S klózhalmaz minden klózát a szemantikus fa ágaira. Ekkor azt mondjuk, hogy elkészítettük az S klózhalmaz szemantikus fáját.

Egy klózhalmaz szemantikus fájának azt a csúcsát, amelybe a klózhalmaz valamely klózának illesztése során utoljára illesztett literál komplemensével címkézett él vezet, cáfoló csúcsnak nevezzük. Azt a csúcsot pedig, amelyet közvetlenül követő mindkét csúcs cáfoló csúcs, levezető csúcsnak hívjuk.

Egy klózhalmaz szemantikus fájának egy ága zárt, ha cáfoló csúcsban végződik. A szemantikus fa zárt, ha minden ága zárt.

Ha egy S klózhalmaz szemantikus fája zárt, akkor a klózhalmaz kielégíthetetlen, ugyanis a klózhalmazban szereplő ítéletváltozók minden interpretációjához van olyan C klóz, amely lezárja az illető interpretációt megadó ágat, vagyis C abban az interpretációban h igazságértékű.

Ha egy véges klózhalmaz szemantikus fája zárt, akkor a szemantikus fában van levezető csúcs. Legyen az S klózhalmaz ítéletváltozóinak száma n . Tegyük fel, hogy S szemantikus fájának nincs levezető csúcsa, azaz minden (nem levél) csúcsból kiinduló két él legalább egyike nem cáfoló csúcsba vezet. Megmutatjuk, hogy a fának van nyitott ága. A gyökér (a 0) nem levezető csúcs, ezért valamelyik gyereke nem cáfoló csúcs: legyen ez a 1 . Ha a 0 , a 1 , a 2 ,…, a j egy olyan út, amelyben egyik csúcs sem cáfoló csúcs, akkor legyen a j+1 az a j csúcsnak egy olyan gyermeke, amely nem cáfoló csúcs. Ilyen van, mert a feltevés szerint a j sem levezető csúcs. n lépés után megkapjuk a nyitott ágat.

Egy klózhalmaz kielégíthetetlenségének eldöntésére a továbbiakban nem a szemantikus fát fogjuk használni. A szemantikus fa azonban fontos segédeszköz marad az új kalkulus, az ún. rezolúciós kalkulus tulajdonságainak vizsgálatában. Egy ilyen fontos tulajdonság például, hogy egy levezető csúcsra illesztett két klóz pontosan egy komplemens literálpárt tartalmaz.

A logikai program és az SLD rezolúció

A logikai programozást arra használjuk, hogy deklaratív leírásokból következtetésekhez jussunk. Ezek a leírások, amelyeket ezentúl logikai programnak fogunk nevezni, logikai formulák véges halmazát tartalmazza. A logikai program formuláira néhány megkötést kell megadnunk, hogy majd alkalmazni tudjunk rájuk egy viszonylag egyszerű és hatékony automatikus tételbizonyítási eljárást, az ún. SLD-rezolúciót (amelyet majd a következő részben tárgyalunk).

Első lépésben megkülönböztetjük a természetes nyelv kijelentő mondatainak két speciális fajtáját, a tényeket és a szabályokat. A tények az egyes egyedek tulajdonságait és a köztük lévő kapcsolatokat írják le. A szabályok azt fejezik ki, hogy ha bizonyos kapcsolat(ok) fennáll(nak) az egyedek között, az milyen egyéb kapcsolt(ok) fennállását bizonyítja.

Az SLD-rezolúció

Ebben a fejezetben bemutatom azt a következtetési mechanizmust, amely a legtöbb logikai programozási rendszernek az alapja. A ’60-as évek közepén J. A. Robinson fejlesztett ki egy olyan következtetési eljárást -a rezolúciós elvet-, amely nemcsak logikai programoknál alkalmazható. Most ennek egy speciális változatát, az SLD rezolúciós elvet vizsgáljuk. Az

SLD betőszó arra utal, hogy az algoritmus:

(D) definit programklózokat használ (az előző részben ismertetett programklózok pontos elnevezése),

(L) lineáris inputstratégiával dolgozik,

(S) a célklózban a feldolgozandó literált és a literálnak megfelelő partícióból a rezolválás során felhasználandó programklózt egy rögzített (S) stratégia alapján választja ki.

SLD-rezolúció: Horn klózok körében (azaz amikben klózonként legfeljebb egy pozitív literál engedélyezett) olyan lineáris rezolúció, mely negatív klózból (azaz csupa negatív literált tartalmazó klózból) indul.

Tétel. Horn-klózok egy halmaza, akkor és csak akkor kielégíthetetlen ha SLD rezolúcióval levezethető belőle az üres klóz.

- Az SLD rezolúció minden olyan klózzal/negatív klózzal kezdhető, mely a klózhalmaz egy minimális kielégíthetetlen részhalmazának eleme.
- SLD rezolúció esetén, a levezetésbe nem írjuk be azt a klózt, amivel az előző klózt rezolváljuk.
- Viszont az indoklásban ezt meg kell említeni (vagy egalább vonallal oda kell húzni) a szokásos változóátnevezésekkel és az egyesítővel együtt.
- SLD rezolúciónál minden a levezetésben szereplő klóz negatív klóz lesz, mert az egyetlen pozitív literálok (hiszen Horn-klózokról van szó) a rezolúció során mindig kiesnek.

A logikai programozás alapvető módszerei

Ismertettünk az automatikus tételbizonyításra néhány kalkulust: a bizonyításelméleti kalkulust (Hilbert-rendszer), a szekventkalkulusokat, a rezolúciós kalkulusokat és a tablók módszerét. Ha egy problémát sikerül megfogalmazni mint feltételformulák és speciális axiómaformulák lehetséges következményét, azaz sikerül a problémát tételbizonyítási feladattá alakítani, akkor az ismert tételbizonyító kalkulusok bármelyikével megkísérelhető a feladat megoldása. Amennyiben egy tételbizonyító kalkulus számítógépes implementációja adott, akkor ez a megadott formulák alapján megpróbálja eldönteni, hogy a tételbizonyítási feladatra mi a válasz. Ennek megfelelően logikai programozási nyelvnek tekinthetünk minden elsőrendű logikai nyelvet. Kétféle módon szokás a tételbizonyítási feladatot leíró formulák együttesét minősíteni. Ha az implementált kalkulust mint számítógépes programot tekintjük, akkor a kalkulust megvalósító programnak a szóban forgó formulák ,,adatai”. Ebben az esetben a logikai nyelv adatdefiniáló, adatleíró nyelvnek minősül. Ha az implementált kalkulust mint speciális számítógépet tekintjük, akkor a logikai nyelven leírt formulák-e ,,logikai gép programjai”. Ebben az esetben a logikai nyelv egy programnyelv. Ezek a gondolatok az 1970-es évek kutatásaiban jelennek meg, és ez az az időszak, amikor a lineáris inputstratégiát alkalmazó rezolúciós kalkulus egy változatát számítógépen megvalósítják. Az implementált kalkulushoz a programozási nyelv a Prolog. Az első Prolog interpreter Marseille-ben 1972-ben Colmerauer, és röviddel utána Magyarországon Szeredi Péter nevéhez fűződik. Ezeket számos helyen követték újabb interpreterek és később fordítóprogramok. A Prologhoz készült fordítók eljárásorientált értelmezését adták a nyelvnek. A Prologot a számítástudományban mint specifikációs és mint deklaratív nyelvet is használják. Az újabb logikai programozási rendszerek az 1980-as évekre jelentek meg. Ugyanígy erre az időre tehető a logikai programozás lehetőségeit és korlátait kutató munkák eredményeinek összegzése. Az utóbbi évek összefoglaló műveiből képet kaphatunk a fejlődés irányáról és az újabb elképzelésekről.

A prolog nyelv építőkövei: változók, konstansok, függvényszimbólum, termek, atomi formulák. Speciális, és gyakran használta adatszerkezet a lista.

Egy logikai program futása során egy háttérben működő algoritmus ellenőrzi, hogy a célklóz a porgramklózok logikai következménye-e. Az ellenırzés során az SLD-rezolúció algoritmizált számítógépes megvalósítása működik. A programozó tud paraméterezni bizonyos beállításokat, ennek hatására némileg módosulhat a futtatókörnyezet által végrehajtott algoritmus. Ily módon tehát a Prolog az automatikus tételbizonyítás egy, a korlátai ellenére nagy számítóerőjű megvalósítása.

A Prologot gyakran használják mesterségesintelligencia-alkalmazások megvalósítására, illetve a számítógépes nyelvészet eszközeként (különös tekintettel a természetes nyelvfeldolgozásra, melyre eredetileg tervezték). A kutatások egyik fontos támogatója és gerjesztője volt az a tény, hogy a Prolog-variánsokat (pl. a Kernel Nyelvet) operációs rendszerként használó számítógépek előállítására, az ún. Ötödik Generációs

Számítógéprendszer-Projektre az 1980-s években elég nagy hangsúlyt fektettek (elsősorban japán kezdeményezésre), bár mára az eredeti kutatás leállt. Manapság a fő kutatási terület a nyelv felhasználása tudásalapú rendszerekre.