Prolog alapú tételbizonyító

Gados Dániel
gdani@hexaco.axelero.net
2005

Tételbizonyítók

- automatikus
- nincs felhasználói vezérlés
- elsőrendű logika
- bizonyítás megadása

Alkalmazások

- bizonyítások (kombinatorika, Gödel nemteljességi tétele)
- tanácsadó (felhasználói beavatkozás)
- bizonyítás-ellenőrzés
- software és hardware verifikáció (RSA, Boyer-Moore szövegillesztő, 16-bites összeadó, processor)
- deduktív szintézis (algoritmus szintetizálása)

Ítéletlogika

mondat	$\rightarrow$	Atom | Igaz | Hamis | össz_mondat
össz_mondat	$\rightarrow$	(mondat) | $\neg$ mondat | mondat operátor mondat
operátor	$\rightarrow$	$\wedge$ | $\vee$ | $\Rightarrow$ | $\Leftrightarrow$

Következtetés ítéletlogikában

- ÉS-kiküszöbölés

$$\frac{p \wedge q}{p\ ,\ q}$$

- ÉS-bevezetés

$$\frac{p\ ,\ q}{p \wedge q}$$

- VAGY-bevezetés

$$\frac{p}{p \vee q}$$

- Dupla negáció

$$\frac{\neg \neg p}{p}$$

Következtetés ítéletlogikában

- Modus Ponens

$$\frac{p,\ p \Rightarrow q}{q}$$

- Egységrezolúció

$$\frac{p \vee q,\ \neg q}{p}$$

- Rezolúció

$$\frac{p \vee q,\ \neg q \vee r}{p \vee r}$$

$$\frac{p \Rightarrow q,\ \ q \Rightarrow r}{p \Rightarrow r}$$

Ítéletlogikai bizonyítás

- igazságtáblával mindig ellenőrizhető
- költsége: 2^n
- van-e polinomiális algoritmus?
- Cook: a kielégíthetőség (SAT) NP-teljes

Horn-klózok

- $p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q$
- $\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n \vee q$
- létezik polinomiális idejű bizonyítás

Elsőrendű logika

- az ítéletkalkulusban a világ csak tényekből állhat
- az elsőrendű logikában objektumok

- predikátumok
- függvények $\rightarrow$ $=$
- kvantorok $\rightarrow$ $\exists$, $\forall$

Elsőrendű logika

mondat	$\rightarrow$	össz_mondat | term = term | Predikátum(term, …) | kvantor Változó, … mondat
term	$\rightarrow$	Függvény(term, …) | Változó
össz_mondat	$\rightarrow$	(mondat) | $\neg$ mondat | mondat operátor mondat
operátor	$\rightarrow$	$\wedge$ | $\vee$ | $\Rightarrow$ | $\Leftrightarrow$
kvantor	$\rightarrow$	$\forall$ | $\exists$

Kvantifikált következtetés

- Univerzális elimináció

$$\frac{\forall x \ \text{Szereti}(x, \text{Fagyi})}{\text{Szereti(Béla, Fagyi)}}$$

- Egzisztenciális elimináció *(Skolemizáció)*

$$\frac{\exists x \ \text{Szereti}(x, \text{Fagyi})}{\text{Szereti}(\textbf{\textit{Valaki}}, \text{Fagyi})}$$

- Egzisztenciális bevezetés

$$\frac{\text{Szereti(Béla, Fagyi)}}{\exists x \ \text{Szereti}(x, \text{Fagyi})}$$

Átalakítás klóz-formára

Konjunktív normál forma

$\neg p_1 \vee \ldots \vee \neg p_n \vee q_1 \vee \ldots \vee q_m$

$\neg r_1 \vee \ldots \vee \neg r_k \vee s_1 \vee \ldots \vee s_l$

$t_1 \vee \ldots \vee t_i$

Implikatív normál forma

$p_1 \wedge \ldots \wedge p_n \Rightarrow q_1 \vee \ldots \vee q_m$

$r_1 \wedge \ldots \wedge r_k \Rightarrow s_1 \vee \ldots \vee s_l$

$\mathit{Igaz} \Rightarrow t_1 \vee \ldots \vee t_i$

Átalakítás klóz-formára

$\forall x\,(\text{Anya}(x) \vee \text{Apa}(x)) \Rightarrow (\exists y\,\text{Utódja}(y, x)) \vee (\exists y\,\text{Mostohája}(x, y))$

1. **implikációk megszüntetése**

   $\forall x\,\neg(\text{Anya}(x) \vee \text{Apa}(x)) \vee (\exists y\,\text{Utódja}(y, x)) \vee (\exists y\,\text{Mostohája}(x, y))$

2. **negálás csak atomokon**

   $\forall x\,(\neg\text{Anya}(x) \wedge \neg\text{Apa}(x)) \vee (\exists y\,\text{Utódja}(y, x)) \vee (\exists y\,\text{Mostohája}(x, y))$

3. **változók standardizálása**

   ($\rightarrow$ többszöriek egyedivé tétele)

   $\forall x\,(\neg\text{Anya}(x) \wedge \neg\text{Apa}(x)) \vee (\exists y\,\text{Utódja}(y, x)) \vee (\exists z\,\text{Mostohája}(x, z))$

4. **kvantorok kimozgatása a mondat elé**

   $\forall x\,\exists y\,\exists z\,(\neg\text{Anya}(x) \wedge \neg\text{Apa}(x)) \vee \text{Utódja}(y, x) \vee \text{Mostohája}(x, z)$

Átalakítás klóz-formára

5. **skolemizáció:** egzisztenciális kvantor kiküszöbölése (ha univerzális kvantorba van beágyazva)

   $\forall x\, \exists y\, (\neg \text{Anya}(x) \wedge \neg \text{Apa}(x)) \vee \text{Utódja}(y, x) \vee \text{Mostohája}(x, \text{Mgy}(x))$

   $\forall x\, (\neg \text{Anya}(x) \wedge \neg \text{Apa}(x)) \vee \text{Utódja}(\text{Egy}(x), x) \vee \text{Mostohája}(x, \text{Mgy}(x))$

   $(\neg \text{Anya}(x) \wedge \neg \text{Apa}(x)) \vee \text{Utódja}(\text{Egy}(x), x) \vee \text{Mostohája}(x, \text{Mgy}(x))$

6. **hozzuk konjunktív normál formára**

   $[\neg \text{Anya}(x) \vee \text{Utódja}(\text{Egy}(x), x) \vee \text{Mostohája}(x, \text{Mgy}(x))] \wedge$

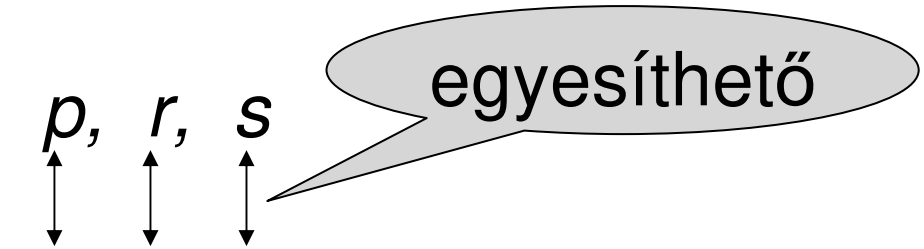
   $[\neg \text{Apa}(x) \vee \text{Utódja}(\text{Egy}(x), x) \vee \text{Mostohája}(x, \text{Mgy}(x))]$

7. **alakítsuk vissza implikációkká**

   $[\text{Anya}(x) \Rightarrow \text{Utódja}(\text{Egy}(x), x) \vee \text{Mostohája}(x, \text{Mgy}(x))] \wedge$

   $[\text{Apa}(x) \Rightarrow \text{Utódja}(\text{Egy}(x), x) \vee \text{Mostohája}(x, \text{Mgy}(x))]$

Általánosított Modus Ponens



$$\frac{p \quad p \Rightarrow q}{q}$$

→ előrefelé láncolás

→ hátrafelé láncolás

→ nem teljes: $\neg\, p \Rightarrow q$,

$p \Rightarrow q \vee r$ nem írhatók át

Általánosított rezolúció

$\boxed{p_1} \wedge p_2 \wedge \ldots \wedge p_i \Rightarrow q_1 \vee \ldots \vee q_j$

$r_1 \wedge \ldots \wedge r_k \Rightarrow \boxed{s_1} \vee s_2 \vee \ldots \vee s_l$

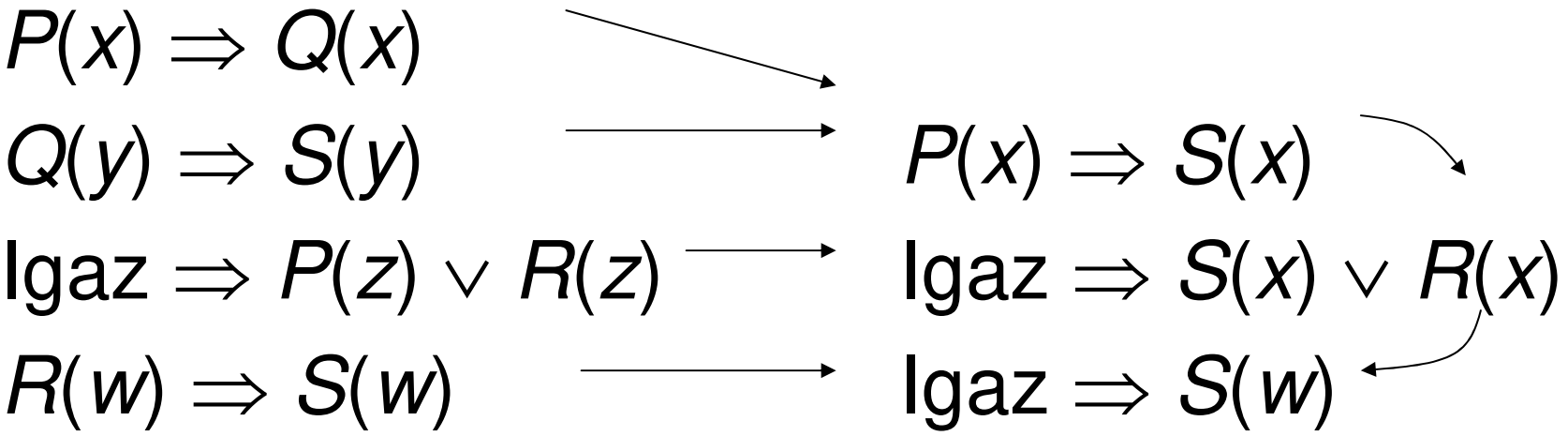
$p_2 \wedge \ldots \wedge p_i \wedge r_1 \wedge \ldots \wedge r_k \Rightarrow q_1 \vee \ldots \vee q_j \vee s_2 \vee \ldots \vee s_l$

$p \Rightarrow q$

$r \Rightarrow p$

$r \Rightarrow q$

Általánosított rezolúció



Általánosított rezolúció

- majdnem teljes az elsőrendű logikában
- de nem vezethető le pl. a

  $p \vee \neg p$

  egy üres tudásbázisból
- cáfolatteljes
- indirekt bizonyítás

Rezolúciós stratégiák

- egységpreferencia
- támogató halmaz
- bemeneti rezolúció
- lineáris rezolúció
- bennfoglalás

OTTER

- Organized Techniques for Theorem-proving and Effective Research

- támogató halmaz
- axiómák

} Tudásbázis

- demodulátorok [$x + 0 = x$]
- heurisztikus függvény

OTTER

- előrehaladó rezolúció
- best-first keresés
- legkönnyebb klóz a támogató halmazból
- az axiómákkal rezolválva
- a rezolvensek a támogató halmazba
- cáfolatnál leáll
- az egységcáfolatot rögtön ellenőrizzük

PTTP

- Prolog Technology Theorem Prover
- Prolog kiterjesztés:
  - módosított egyesítés
  - keresési stratégia
  - kiterjesztés a teljes elsőrendű logikára
- megtartja a Prolog hatékonyságát

PTTP

- a bemenet klózok konjunkciója
- implicit módon univerzálisan kvantifikált
- a következtetés egzisztenciálisan kvantifikált
- feltételezett az explicit kvantorok eliminálása

Előfordulás-ellenőrzés

- Egy olyan termhez kötjük-e hozzá a változót, amely tartalmazza azt?
- a Prologban átalában nincs
- költséges az ellenőrzés
- hibás vagy végtelen egyesítés

- 1995-ös szabvány:

`unify_with_occurs_check`

Előfordulás-ellenőrzés

```
k(X, (X+1)).
| ?- k(Y, Y).
Y = ... + ... +1+1+1+1+1+1+1+1+1 ?

p(A, B, f(A, B)).
| ?- p(a, Z, Z).
Z = f(a,f(a,f(a,f(a,f(a,f(a,
    f(a,f(a,f(a,f(...)))))))))) ?
```

Előfordulás-ellenőrzés

- a tételbizonyításnál nem jellemzők a komplex termek
- az előfordulás-ellenőrzés sem annyira költséges

- érzékeljük a végtelen ciklust és leállunk
- linearizáljuk a klózt

Linearizálás

- ha egy változó a fejben csak egyszer fordul elő, akkor biztosan nem lesz hibás kötés
- cseréljük le a fej változóit unikálisra

```
k(X, (X1+1)).
p(A, B, f(A1, B1)).
```

Helyes egyesítés

- az azonos változókat egyesíteni kell
- ezt a törzshöz hozzáadott saját egyesítővel oldhatjuk meg

```
k(X, (X1+1)) :-
    unify_with_occurs_check(X, X1).

p(A, B, f(A1, B1)) :-
    unify _with_occurs_check(A, A1),
    unify _with_occurs_check(B, B1).
```

Helyes egyesítés

- nem változik meg a Prolog
  – utasításkészlete
  – belső működése
- a diszkusszió körülményes
- alacsonyszintű megvalósítás gyorsít
- végtelen termek egyesítésére képes rendszernél elég az ismétlésellenőrzés

Keresési stratégia

- probléma: mélységi keresés
- szélességi keresés jobb lenne
- túl nagy a tárigény

- mélységkorlátolt keresés
- iteratívan mélyülő keresés
- párhuzamosíthatóság

Keresési stratégia – becslés

```
p(e,X,X).
p(U,Z,W) :- p(X,Y,U), p(Y,Z,V), p(X,V,W).

p(e,X,X, DepthIn,DepthOut) :-
  DepthIn >= 1, DepthOut = DepthIn - 1.
p(U,Z,W, DepthIn,DepthOut) :-
  DepthIn >= 1, Depth1 = DepthIn - 1,
  p(X,Y,U, Depth1,Depth2),
  p(Y,Z,V, Depth2,Depth3),
  p(X,V,W, Depth3,DepthOut).
```

Keresési stratégia – becslés

```
p(e,X,X, Depth,Depth).

p(U,Z,W, DepthIn,DepthOut) :-
  DepthIn >= 3, Depth1 = DepthIn - 3,
  p(X,Y,U, Depth1,Depth2),
  p(Y,Z,V, Depth2,Depth3),
  p(X,V,W, Depth3,DepthOut).
```

Következtető rendszer

- a Prolog (bemeneti rezolúció) nem teljes
- pl. nem bizonyítható

$$\frac{p \vee q \qquad \neg p \vee q}{q}$$

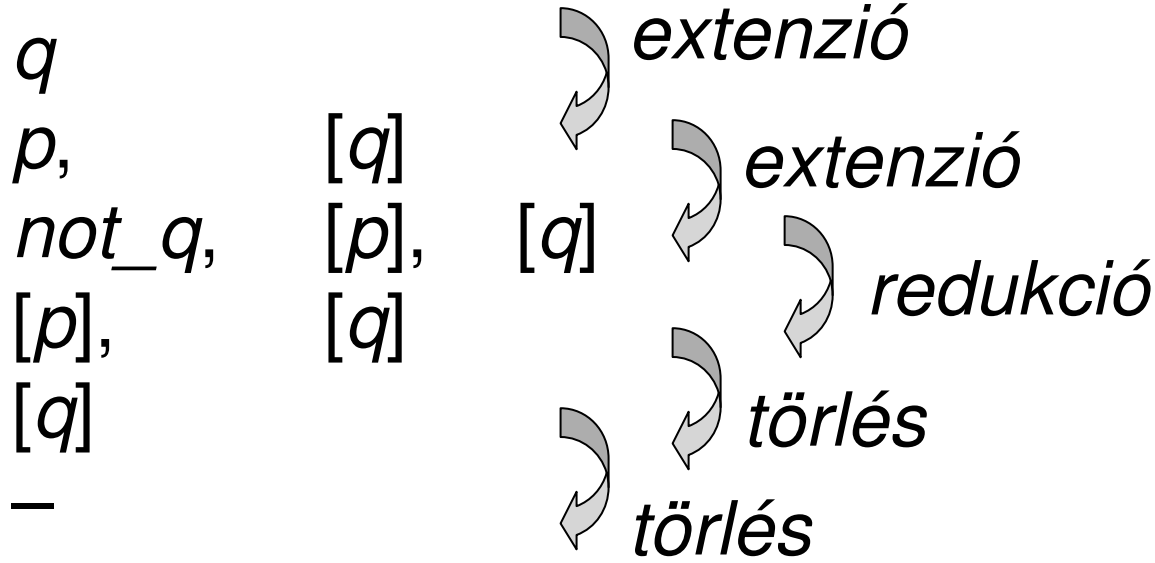
$(1)\ \neg p \Rightarrow q\ , \quad (2)\ \neg q \Rightarrow p$

$(3)\ p \Rightarrow q\ , \quad (4)\ \neg q \Rightarrow \neg p$

Kontrapozitív alakok
Zárolás

Modell eliminációó (ME)

- lineáris bemeneti rezolúció (teljes)



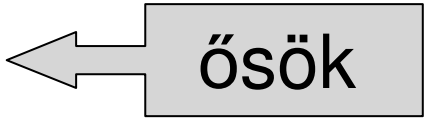
Modell eliminációs (ME)

- *extenzió*: a legbaloldalibb elemet helyettesítjük, és keretezzük
- *redukció*: ha egy elem valamely ősének az ellentettje, akkor az elemet bizonyítás nélkül elhagyjuk
- *törlés*: a legbaloldalibb keretezett elemet töröljük

Modell eliminációó (ME)

```
p(e,X,X).
p(U,Z,W) :- p(X,Y,U), p(Y,Z,V), p(X,V,W).
```

```
p(e,X,X, Ancestors).
p(U,Z,W, Ancestors) :-
  NewAncestors = [p(U,Z,W)|Ancestors],
  p(X,Y,U,NewAncestors),
  p(Y,Z,V,NewAncestors),
  p(X,V,W,NewAncestors).
```



Modell eliminációs (ME)

```
p(X,Y,Z,Ancestors) :-
  unifiable_member(not_p(X,Y,Z),Ancestors).

p(e,X,X, Ancestors).
p(U,Z,W, Ancestors) :-
  NewAncestors = [p(U,Z,W)|Ancestors],
  p(X,Y,U,NewAncestors),
  p(Y,Z,V,NewAncestors),
  p(X,V,W,NewAncestors).
```

Javítások

- ha egy elem megegyezik egy ősével, akkor nem bizonyítjuk
- ha egy elem egyesíthető egy egységklózzal, akkor bizonyítottnak tekintjük

- bizonytalan célnál célnegálás kell

Positive refinement

- csak a pozitív literálok redukálhatók
- csak a negatívokat kell számontartani
- gyorsabb a futás

$$\frac{p \vee q \qquad \neg p \vee q}{q}$$

(1) $\neg p \Rightarrow q$, (2) $\neg q \Rightarrow p$
(3) $p \Rightarrow q$, (4) $\neg q \Rightarrow \neg p$

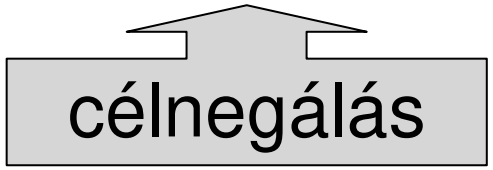
Positive refinement

q
p
not_q

q
not_p
not_q, [*not_p*]
[*not_p*]
–

```
q :- not_p.
p :- not_q.
q :- p.
not_p :- not_q

+
not_q.
```



Újraindított Modell Eliminációó

- RME = Restart Model Elimination
- pl. PROTEIN = A Prover with Theory Extension Interface
- a negált literálok nem fejthetők ki
  – redukció
  – újraindítás

Példa-1

```
go :- pttp(( (e(A,A)),(e(A,B) :- e(B,A)),
             (e(A,C) :- e(A,B), e(B,C)),
             (e(a,b)),(e(b,c)),(e(c,d)) )).
| ?- prove(e(a,d)).

Goal#  Wff#  Wff Instance
-----  ----  ------------
  [0]     0   query :- [1].
  [1]     3      e(a,d) :- [2] , [5].
  [2]     3         e(a,c) :- [3] , [4].
  [3]     4            e(a,b).
  [4]     5            e(b,c).
  [5]     6         e(c,d).
Proof end.
yes
```

Példa-2

```
go :- pttp((

% --- egyenloseg-relacio tulajdonsagai ---
(e(A,A)),
(e(A,B) :- e(B,A)),
(e(A,C) :- e(A,B), e(B,C)),

% --- helyettesitesi szabaly ---
%     -> a Prolog resze
```

Példa-2

```
% --- potlasi szabaly ---
( e(m(A,B), m(C,B)) :- e(A,C) ),
( e(m(B,A), m(B,C)) :- e(A,C) ),
( e(u(A,B), u(C,B)) :- e(A,C) ),
( e(u(B,A), u(B,C)) :- e(A,C) ),
( e(n(A),   n(C))   :- e(A,C) ),
```

Példa-2

```
% --- Bool-algebra axiomai ---
(e(u(A,A), A)),                          % A+A=A
(e(u(A,B), u(B,A))),                     % A+B=B+A
(e(u(u(A,B),C), u(A,u(B,C)))),           % (A+B)+C=A+(B+C)
(e(u(A,m(B,C)), m(u(A,B),u(A,C)))),      % A+BC=(A+B)(A+C)
(e(A, u(A,m(B,n(B))))),                  % A=A+B(!B)
(e(n(u(A,B)), m(n(A),n(B)))),            % !(A+B)=(!A)(!B)
(e(n(n(A)), A)),                         % !!A=A
(e(u(A,n(A)), 1)),                       % A+!A=1
(e(n(1), 0)),                            % !1=0

% --- de Morgan-szabaly ---
(query :- e(n(m(a,b)),u(n(a),n(b)))) %!(AB)=!A+!B
)),fail.
go :- prove(query).
```

Példa-2

```
Goal#  Wff Instance
-----  ------------
  [0] query :- [1].
  [1]  e(n(m(a,b)),u(n(a),n(b))) :- [2] , [11].
  [2]   e(n(m(a,b)),n(n(u(n(a),n(b))))) :- [3].
  [3]    e(n(n(u(n(a),n(b)))),n(m(a,b))) :- [4].
  [4]     e(n(u(n(a),n(b))),m(a,b)) :- [5],[9].
  [5]      e(n(u(n(a),n(b))),m(a,n(n(b)))) :- [6],[7].
  [6]       e(n(u(n(a),n(b))),m(n(n(a)),n(n(b)))).
  [7]       e(m(n(n(a)),n(n(b))),m(a,n(n(b)))) :- [8].
  [8]        e(n(n(a)),a).
  [9]      e(m(a,n(n(b))),m(a,b)) :- [10].
 [10]       e(n(n(b)),b).
 [11]   e(n(n(u(n(a),n(b)))),u(n(a),n(b))).
```