Logică și structuri discrete

Logica predicatelor

Marius Minea
marius@cs.upt.ro

http://cs.upt.ro/~marius/curs/lsd/

14 noiembrie 2016

Logică: recapitulare

Folosim logica pentru a exprima *riguros* (*formaliza*) raționamente.

Logica ne permite să facem *demonstrații (deducții)*
din *axiome* (totdeauna adevărate)
și *ipoteze* (considerate adevărate în problema dată)
folosind *reguli de inferență* (de deducție)

$$\frac{p \qquad p \rightarrow q}{q} \qquad \textit{modus ponens}$$

Modus ponens e suficient, dar sunt și alte reguli de deducție valide.

Logica propozițională e
consistentă: orice formulă demonstrată (teoremă) e validă
completă: orice formulă validă (tautologie) poate fi demonstrată

Avem nevoie de logică

în *specificații* pentru programe: de exemplu, sortare

```
/*@ ensures
  @   (\forall int i; 0<=i && i<a.length - 1;
  @     a[i] <= a[i+1])
  @*/
```

în condiții (*predicate*) pentru datele de prelucrat

```
M.filter (fun k v -> k < "M" && v >= 5) stud_dict
```

când exprimăm proprietăți: formalizând teoria mulțimilor

$$\exists empty\ \forall x\ \neg contains(empty, x)$$

sau structura fișierelor pe disc

$$\forall x\ ((file(x) \wedge x \neq root) \rightarrow contains(parent(x), x))$$

Logica propozițională nu poate exprima tot

Un exemplu clasic:
(1) Toți oamenii sunt muritori.
(2) Socrate e om.
Deci, (3) Socrate e muritor.

Acesta e un *silogism* (tipar de regulă de inferență)
logica clasică: Aristotel, stoici

Seamănă cu *modus ponens*
dar premisa din (1) ("toți oamenii")
nu e la fel cu (2) (Socrate, un anumit om)

Am putea reformula (1):
Dacă X e om, atunci X e muritor.
sau mai precis
Pentru orice X, dacă X e om, atunci X e muritor

Logica modernă: *logica predicatelor* (logica de ordinul I)
Gottlob Frege, Charles Peirce (sec. 19)

Avem nevoie de formule mai expresive

$$\forall x \; ((\mathit{file}(x) \wedge x \neq \mathit{root}) \rightarrow \mathit{contains}(\mathit{parent}(x), x))$$

În loc de *propoziții* (a, p, q) avem *predicate*: $\mathit{file}(x)$, $\mathit{contains}(x, y)$

Un *predicat* = o afirmație relativ la una sau mai multe variabile, care, dând valori variabilelor, poate lua valoarea adevărat sau fals.

Apar *cuantificatori*: $\forall$ (orice), $\exists$ (există)

Argumentele predicatelor pot fi *variabile* x sau *funcții*: $\mathit{parent}(x)$

$$\mathit{contains}(\mathit{parent}(x), x)$$

Logica predicatelor (logica de ordinul întâi): Sintaxa

Definim, structural recursiv, noțiunile de *termen* și *formulă*:

Termeni

variabilă v sau *constantă* c

$f(t_1, \cdots, t_n)$ cu f *funcție* n-ară și $t_1, \cdots, t_n$ *termeni*

Formule (well-formed formulas, formule bine formate):

$P(t_1, \cdots, t_n)$	cu P *predicat* n-ar; $t_1, \cdots, t_n$ *termeni*
$\neg\alpha$	unde α este o formulă
$\alpha \to \beta$	unde α, β sunt formule
$\forall v\, \alpha$	cu v *variabilă*, α formulă: *cuantificare universală*
$t_1 = t_2$	cu t_1, t_2 termeni (în logica de ordinul I cu egalitate)

În loc de propoziții avem *predicate* (peste *termeni*).
În logica *de ordinul I* se pot cuantifica ($\forall$, $\exists$) doar variabile.
În logici *de ordin superior*, se poate cuantifica și peste predicate.

Reprezentare în ML

Termenii și formulele se pot traduce direct în *tipuri recursive*

```
type term = V of string
          | F of string * term list

type predform = Pr of string * term list
              | Neg of predform
              | And of predform * predform
              | Or of predform * predform
              | Forall of string * predform
```

O formulă poate conține termeni. Termenii nu conțin formule!

Am ales să reprezentăm constantele ca funcții cu zero argumente.

Atât termenii cât și predicatele au argumente: listă de termeni.

Exemplu: $\forall x \neg\forall y\, P(x, f(y))$

```
Forall("x", Neg(Forall("y", Pr("P",[V "x"; F("f", [V "y"])])))))
```

Despre cuantificatori. Cuantificatorul existențial $\exists$

Notăm: $\exists x\varphi \stackrel{def}{=} \neg\forall x(\neg\varphi)$
Există x pentru care ϕ e adevărată $\leftrightarrow$ nu pentru orice x ϕ e falsă.

Cei doi cuantificatori sunt *duali*. Putem scrie și $\forall x\varphi = \neg\exists x(\neg\varphi)$

Cuantificatorii au *precedență mai mare* decât conectorii $\neg$, $\wedge$, $\rightarrow$ ...

punct . înseamnă: cuantificatorul se aplică la tot restul formulei
(până la sfârșit sau paranteză închisă)

$(\exists x.P(x) \rightarrow Q(x)) \wedge R(x)$ înseamnă $\exists x(P(x) \rightarrow Q(x)) \wedge R(x)$
$P(x) \vee \forall y.Q(y) \wedge R(x,y)$ înseamnă $P(x) \vee \forall y(Q(y) \wedge R(x,y))$

În formula $\forall v\varphi$ (sau $\exists v\varphi$) variabila v se numește *legată*
Variabilele care nu sunt legate se numesc *libere*

O variabilă poate fi liberă și legată în aceeași formulă.
Mai sus, x e *legată* în $\exists x.P(x) \rightarrow Q(x)$ și e *liberă* în $R(x)$
(e în afara cuantificatorului)

Variabile libere și legate

Înțelesul unei formule *nu depinde* de variabilele legate

- înțelesul lor e "*legat*" de cuantificator ("pentru orice", "există")
- pot fi redenumite, fără a schimba înțelesul formulei

O formulă *fără variabile libere* are înțeles de sine stătător.

- engl. *closed formula*

Rol similar: parametrii formali la funcții în limbaje de programare

- putem să îi redenumim fără a schimba efectul funcției
- `fun x -> x + 3` și `fun y -> y + 3` sunt aceeași funcție

Interpretarea unei formule *depinde* de variabilele sale libere

- (ce valoare primesc; discutăm la semantica formulelor)

La fel și `fun x -> x + y`

- nu are înțeles de sine stătător (`y` se presupune declarat anterior)
- înțelesul/efectul depinde de definiția lui `y`

Formalizarea limbajului natural

1. Fiecare investitor a cumpărat acțiuni sau obligațiuni.
2. Dacă indicele Dow Jones scade, toate acțiunile mai puțin aurul scad.
3. Dacă trezoreria crește dobânda, toate obligațiunile scad.
4. Orice investitor care a cumpărat ceva care scade nu e bucuros.
5. Dacă indicele Dow Jones scade și trezoreria crește dobânda,
toți investitorii bucuroși au cumpărat ceva acțiuni de aur.

Exemplu: `http://www.cs.utexas.edu/users/novak/reso.html`

Verbele devin *predicate* (ca în limbajul natural):
cumpără, scade, crește, ...

Subiectul și *complementele* (in)directe: *argumentele* predicatului
investitor, ceea ce cumpără (acțiuni, obligațiuni)

Atributele (proprietăți) sunt *predicate* despre entități (*argumente*)
bucuros (investitor), de aur (acțiune)

Categoriile devin predicate, cu argument obiectul din categorie
e acțiune, e obligațiune (ce se cumpără)

O *frază* e o *propoziție* (0 argumente) dacă verbul apare doar în ea
trezoreria crește dobânda (“crește” apare doar aici)

Exemplu de formalizare

1. Fiecare investitor a cumpărat acțiuni sau obligațiuni.

Două entități: investitorul, ce cumpără (cu două categorii)
Introducem un predicat $inv(X)$ (X e investitor)

$$inv(X) \rightarrow cumpără(X, C) \wedge (acțiune(C) \vee oblig(C))$$

Vrem formule *fără variabile libere* (independente de context)
X e cuantificat *universal* (*fiecare* investitor)
C e cuantificat *existențial* (investitorul cumpără *ceva*)

$$\forall X.inv(X) \rightarrow \exists\, C.cumpără(X, C) \wedge (acțiune(C) \vee oblig(C))$$

2. Dacă indicele Dow Jones scade, toate acțiunile mai puțin aurul scad.

$$scade(dj) \rightarrow \forall\, X.acțiune(X) \wedge \neg aur(X) \rightarrow scade(X)$$

Indicele Dow Jones e o noțiune unică $\Rightarrow$ folosim o *constantă dj*
Putem folosi și o *propoziție scadedj*
(celelalte lucruri care scad sunt acțiuni, diferite de indicele general)

Exemplu de formalizare (cont.)

3. Dacă trezoreria crește dobânda, toate obligațiunile scad.

$$\mathit{creștedob} \to \forall X.\mathit{oblig}(X) \to \mathit{scade}(X)$$

Dobânda e unicul lucru care crește $\Rightarrow$ predicat fără parametri

Alternativ: o constantă *dobânda*

4. Orice investitor care a cumpărat ceva care scade nu e bucuros.

$$\forall X.\mathit{inv}(X) \to (\exists C.\mathit{cumpără}(X, C) \land \mathit{scade}(C)) \to \neg \mathit{bucuros}(X)$$

$\to$ asociază la dreapta, $p \to q \to r = p \to (q \to r) = p \land q \to r$, echivalent:

$$\forall X.\mathit{inv}(X) \land (\exists C.\mathit{cumpără}(X, C) \land \mathit{scade}(C)) \to \neg \mathit{bucuros}(X)$$

5. Dacă indicele Dow Jones scade și trezoreria crește dobânda, toți investitorii bucuroși au cumpărat ceva acțiuni de aur.

$\mathit{scade}(dj) \land \mathit{creștedob} \to$

$$\forall X.\mathit{inv}(X) \land \mathit{bucuros}(X) \to \exists C.\mathit{cumpără}(X, C) \land \mathit{acțiune}(C) \land \mathit{aur}(C)$$

Atenție la cuantificatori!

Cuantificatorul *universal* ("toți") cuantifică o *implicație*:

Toți studenții sunt tineri

$Studenți \subseteq Tineri$

$$\forall x.student(x) \to tânăr(x)$$

Eroare frecventă: $\wedge$ în loc de $\to$: $\cancel{\forall x.student(x) \wedge tânăr(x)}$

Oricine/orice din univers e și student și tânăr!!!

Cuantificatorul *existențial* ("unii","există") cuantifică o *conjuncție*.

Există premianți studenți.

$Premianți \cap Studenți \neq \emptyset$

$$\exists x.premiant(x) \wedge student(x)$$

Eroare frecventă: $\to$ în loc de $\wedge$: $\cancel{\exists x.premiant(x) \to student(x)}$

E adevărată dacă există un ne-premiant! (fals implică orice)

Distributivitatea cuantificatorilor față de $\wedge$ și $\vee$

Cuantificatorul *universal* e *distributiv față de conjuncție*:

$$\forall x(P(x) \wedge Q(x)) \leftrightarrow \forall x\, P(x) \wedge \forall x\, Q(x)$$

dar cuantificatorul *existențial* NU ~~e distributiv~~ față de conjuncție:

$$\exists x(P(x) \wedge Q(x)) \not\leftrightarrow (\exists x\, P(x) \wedge \exists x\, Q(x))$$

avem implicație $\rightarrow$, dar nu și invers, poate să nu fie același x !

Dual, $\exists$ e distributiv față de disjuncție:

$$\exists x\, P(x) \vee \exists x\, Q(x) \leftrightarrow \exists x. P(x) \vee Q(x)$$

$\forall$ nu e distributiv. Avem doar:

$$\forall x\, P(x) \vee \forall x\, Q(x) \rightarrow \forall x. P(x) \vee Q(x)$$

Consistență și completitudine în logica predicatelor

Ca și în logica propozițională:
demonstrația se face pur sintactic
determinarea *adevărului*: semantic, considerând *interpretări*
(care e universul valorilor, ce înseamnă fiecare funcție/predicat)

Dar: avem o infinitate de interpretări $\Rightarrow$ nu putem verifica toate
$\Rightarrow$ e important să putem *demonstra*

Calculul predicatelor de ordinul întâi este *consistent* și *complet*
(la fel ca și logica propozițională):

Orice teoremă e validă (adevărată în toate interpretările/atribuirile).
Orice formulă validă (tautologie) poate fi demonstrată (e teoremă).

Dar: relația de implicație logică e doar *semidecidabilă*
dacă o formulă e o tautologie, ea poate fi demonstrată
dar dacă nu e validă, încercarea de a o demonstra (sau o refuta)
poate să continue la nesfârșit

Demonstrația prin metoda rezoluției

Putem demonstra o teoremă prin *reducere la absurd* arătând că *negația ei e o contradicție* (nerealizabilă).

Fie teorema $A_1 \wedge A_2 \ldots \wedge A_n \to C$
adică: ipotezele $A_1, A_2, \ldots A_n$ implică împreună concluzia C

Arătăm că *negația* ei $\neg(A_1 \wedge A_2 \ldots \wedge A_n \to C)$ e o *contradicție*
adică $A_1 \wedge A_2 \ldots \wedge A_n \wedge \neg C$ e contradicție
am rescris $\neg(H \to C) = \neg(\neg H \vee C) = H \wedge \neg C$

Reducere la absurd: ipoteze adevărate+concluzia falsă e imposibil.

Dacă o formulă e contradicție, putem determina asta prin *rezoluție*.

Rezoluția în calculul propozițional

Rezoluția e o *regulă de inferență* care produce *o nouă clauză* din două clauze cu *literali complementari* (L și $\neg L$).

$$\frac{L \vee A \qquad \neg L \vee B}{A \vee B} \qquad \textit{rezoluție}$$

Clauza obtinută = *rezolventul* celor două clauze în raport cu L

Exemplu: $rez_p(p \vee q \vee \neg r, \neg p \vee s) = q \vee \neg r \vee s$

Modus ponens poate fi privit ca un *caz particular de rezoluție*:

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

Rezoluția e o regulă de inferență *validă*: $\{p \vee A, \neg p \vee B\} \models A \vee B$

orice atribuire care face premisele adevărate face și concluzia adevărată

pentru $p = T$, obținem $B \models A \vee B$: dacă $B = T$ și $A \vee B = T$

simetric pentru $p = F$, deci regula e validă

Corolar: dacă $A \vee B$ e contradicție, la fel și $(p \vee A) \wedge (\neg p \vee B)$

dacă ajungem la contradicție, și formula inițială era contradicție

Exemplu de rezoluție (1)

$$\begin{array}{ll} \quad(a \vee \neg b \vee \neg d) & b \text{ negat} \\ \wedge\, (\neg a \vee \neg b) & b \text{ negat} \\ \wedge\, (\neg a \vee c \vee \neg d) & \\ \wedge\, (\neg a \vee b \vee c) & b \text{ pozitiv} \end{array}$$

Luăm o propoziție cu ambele polarități (b) și construim rezolvenții
$rez_b(a \vee \neg b \vee \neg d,\ \neg a \vee b \vee c) = a \vee \neg d \vee \neg a \vee c = \mathrm{T}$
$rez_b(\neg a \vee \neg b,\ \neg a \vee b \vee c) = \neg a \vee \neg a \vee c = \neg a \vee c$

Adăugăm noii rezolvenți (ignorăm T); eliminăm vechile clauze cu b

$$\begin{array}{l} \quad(\neg a \vee c \vee \neg d) \\ \wedge\, (\neg a \vee c) \end{array}$$

Nu mai putem crea rezolvenți. Nu avem clauza vidă.
$\Rightarrow$ formula e realizabilă, de exemplu cu $a = \mathrm{F}$. Sau cu $c = \mathrm{T}$.

Pentru o atribuire suficientă ca să facă formula realizabilă, revenim la formula inițială, și dăm valori și lui b și/sau d.

Exemplu de rezoluție (2)

$$\begin{array}{ll} \quad a & \\ \wedge (\neg a \vee b) & \\ \wedge (\neg b \vee c) & \qquad c \text{ pozitiv} \\ \wedge (\neg a \vee \neg b \vee \neg c) & \qquad c \text{ negat} \end{array}$$

Aplicăm rezoluția după c, avem o singură pereche de clauze:
$rez_c(\neg b \vee c,\ \neg a \vee \neg b \vee \neg c) = \neg b \vee \neg a \vee \neg b = \neg a \vee \neg b$

Eliminăm cele două clauze cu c și adăugăm clauza nouă:

$$\begin{array}{l} \quad a \\ \wedge (\neg a \vee b) \\ \wedge (\neg a \vee \neg b) \end{array}$$

Aplicăm rezoluția după b:
$rez_b(\neg a \vee b, \neg a \vee \neg b) = \neg a \vee \neg a = \neg a$

Eliminăm cele două clauze cu b, adăugăm clauza nouă: $\begin{array}{l} \quad a \\ \wedge \neg a \end{array}$

Aplicăm rezoluția după a: $rez_a(a, \neg a) = \mathrm{F}$ (clauza vidă)
Deci formula inițială e o contradicție (e nerealizabilă).

Aplicarea rezoluției în calculul propozițional

Pornind de la o formulă în formă normală conjunctivă (CNF), *adăugăm rezolvenți*, încercând să *obținem clauza vidă:*

Alegem o propoziție p și adăugăm toți rezolvenții în raport cu p:
  din m clauze cu p și n clauze cu $\neg p$, creăm $m \cdot n$ rezolvenți
  am eliminat $p \Rightarrow$ ștergem cele $m+n$ clauze inițiale

Dacă vreun rezolvent e *clauza vidă*, formula e *nerealizabilă*
Dacă nu mai putem crea rezolvenți (literalii au polaritate unică), formula e *realizabilă* (facem T toți literalii rămași)

Numărul de clauze poate crește exponențial (problematic!)
Algoritmul DPLL aplică rezoluția doar la clauze cu un literal
$\Rightarrow$ formula nu crește, dar poate încerca nr. exponențial de cazuri

Rezoluția: de la propoziții la predicate

În logica predicatelor, un *literal* nu e o propoziție, ci un *predicat*
nu doar p și $\neg p$, ci $P(\ldots)$ și $\neg P(\ldots)$
$\Rightarrow$ trebuie să ținem cont de argumentele predicatului

Fie două formule în care un predicat apare pozitiv și negativ:
$\forall x.\forall y.P(x, g(y))$ și $\forall z.\neg P(z, a)$ sau
$\forall x.\forall y.P(x, g(y))$ și $\forall z.\neg P(a, z)$ Se contrazic ?

Cuantificarea universală înseană că variabila poate lua *orice* valoare
$\Rightarrow$ o putem *substitui* cu un *termen*

În exemplul 2, substituind $x \mapsto a$, $z \mapsto g(y)$ obținem $P(a, g(y))$ și $\neg P(a, g(y))$, *contradicție*

În ex. 1 nu putem substitui *constanta* a cu $g(y)$ (a nu e variabilă)
g e funcție arbitrară, nu știm că există un y cu $g(y) = a$

Substituții și unificări de termeni

O *substituție* e o *funcție* care asociază unor *variabile* niște *termeni*:

$$\{x_1 \mapsto t_1, \ \ldots, x_n \mapsto t_n\}$$

Doi termeni se pot *unifica* dacă există o substituție care îi face egali
$f(x, g(y, z), t)\{x \mapsto h(z), y \mapsto h(b), t \mapsto u\} = f(h(z), g(h(b), z), u)$

Reguli de unificare

O *variabilă* x poate fi unificată cu orice *termen* t (substituție)
dacă x *nu apare* în t deci nu: x cu $f(h(y), g(x, z))$
pentru că altfel, substituția ar duce la un termen infinit
(Dar: putem unifica în cazul trivial, x cu x)

Doi *termeni* $f(\ldots)$ pot fi unificați doar dacă au funcții identice, și *argumentele* (termeni) pot fi unificate unul câte unul

Două *constante* pot fi unificate doar dacă sunt *identice*
(caz particular, o constantă e o funcție cu zero argumente)

Vom discuta ulterior detalii și un algoritm de unificare.

Rezoluția în calculul predicatelor

Fie două clauze A și B, și un predicat P (apare pozitiv și negat).

Redenumim variabilele comune (pot însemna altceva în A și în B)

Alegem literali $P_1, ..., P_j \in A$ și $\neg P_{j+1}, ... \neg P_{j+k} \in B$ cu pred. P

Unificăm termenii $\{P_1, \ldots, P_j, P_{j+1}, \ldots P_{j+k}\}$

La clauza $A \cup B \setminus \{P_1, \ldots P_k, P_{k+1}, \ldots P_{k+l}\}$ aplicăm substituția rezultată din unificare. Adăugăm noua clauză la lista clauzelor.

Dacă repetând obținem clauza vidă, formula inițială nu e realizabilă.
Dacă nu mai putem crea rezolvenți noi, formula inițială e realizabilă.

Metoda rezoluției e *completă* relativ la refutație
pentru orice formulă nerealizabilă, va ajunge la clauza vidă
dar nu poate *determina* realizabilitatea *oricărei formule*
(există formule pentru care rulează la infinit)

Exemplu de aplicare a rezoluției

Reluăm exercițiul formalizat anterior.
Folosim () și nu . pentru a nu greși la ce se aplică cuantificatorii.

A_1: $\forall X(inv(X) \rightarrow \exists\, C(cump(X, C) \wedge (act(C) \vee oblig(C))))$

A_2: $scadedj \rightarrow \forall\, X(act(X) \wedge \neg aur(X) \rightarrow scade(X))$

A_3: $creștedob \rightarrow \forall\, X(oblig(X) \rightarrow scade(X))$

A_4: $\forall X(inv(X) \rightarrow (\exists\, C(cump(X, C) \wedge scade(C)) \rightarrow \neg bucur(X)))$

C: $scadedj \wedge creștedob \rightarrow$
$\forall\, X(inv(X) \wedge bucur(X) \rightarrow \exists\, C(cump(X, C) \wedge act(C) \wedge aur(C)))$

Pentru a demonstra $A_1 \wedge A_2 \wedge A_3 \wedge A_4 \rightarrow C$ prin *reducere la absurd negăm concluzia*, arătăm că $A_1 \wedge A_2 \wedge A_3 \wedge A_4 \wedge \neg C$ e *contradicție*

$\neg C$: $scadedj \wedge creștedob \wedge$
$\neg\forall\, X(inv(X) \wedge bucur(X) \rightarrow \exists\, C(cump(X, C) \wedge act(C) \wedge aur(C)))$

Negăm concluzia *la început*, înainte de a transforma cuantificatorii!

Eliminăm implicația, ducem negația până la predicate

1. *Eliminăm implicația*: $A \rightarrow B = \neg A \vee B$, $\neg(A \rightarrow B) = A \wedge \neg B$
2. Ducem $\neg$ *înăuntru*: $\neg \forall x P(x) = \exists x \neg P(x)$ $\quad \neg \exists x P(x) = \forall x \neg P(x)$

A_1: $\forall X(inv(X) \rightarrow \exists\, C(cump(X, C) \wedge (act(C) \vee oblig(C))))$
$\quad \forall X(\neg inv(X) \vee \exists\, C(cump(X, C) \wedge (act(C) \vee oblig(C))))$

A_2: $scadedj \rightarrow \forall\, X(act(X) \wedge \neg aur(X) \rightarrow scade(X))$
$\quad \neg scadedj \vee \forall\, X(\neg act(X) \vee aur(X) \vee scade(X))$

A_3: $creștedob \rightarrow \forall\, X(oblig(X) \rightarrow scade(X))$
$\quad \neg creștedob \vee \forall\, X(\neg oblig(X) \vee scade(X))$

A_4: $\forall X(inv(X) \rightarrow (\exists\, C(cump(X, C) \wedge scade(C)) \rightarrow \neg bucur(X)))$
$\quad \forall X(\neg inv(X) \vee \neg \exists\, C(cump(X, C) \wedge scade(C)) \vee \neg bucur(X))$
$\quad \forall X(\neg inv(X) \vee \forall\, C(\neg cump(X, C) \vee \neg scade(C)) \vee \neg bucur(X))$

ATENȚIE! În $\forall x$(*formula*), când transformăm *în formula* ($\rightarrow$, $\neg$, ..)
NU se schimbă cuantificatorul care e *în afara* ei (nici la $\exists x$)

Eliminăm implicația, ducem negația înăuntru (cont.)

$\neg C$: $scadedj \wedge creștedob \wedge$
$\neg \forall X(inv(X) \wedge bucur(X) \rightarrow \exists C(cump(X, C) \wedge act(C) \wedge aur(C)))$

$scadedj \wedge creștedob \wedge$
$\exists X(inv(X) \wedge bucur(X) \wedge \neg \exists C(cump(X, C) \wedge act(C) \wedge aur(C)))$

$scadedj \wedge creștedob \wedge$
$\exists X(inv(X) \wedge bucur(X) \wedge \forall C(\neg cump(X, C) \vee \neg act(C) \vee \neg aur(C)))$

Redenumim: nume unice la variabile cuantificate

3. *Redenumim* variabilele cuantificate: *nume unic* în fiecare clauză, pentru a putea elimina ulterior cuantificatorii. De exemplu:
$\forall x\, P(x) \vee \exists x\, \forall y\, Q(x, y)$ devine $\quad \forall x\, P(x) \vee \exists z\, \forall y\, Q(z, y)$

Nu e nevoie în exemplul nostru:

A_1: $\forall X(\neg inv(X) \vee \exists\, C(cump(X, C) \wedge (act(C) \vee oblig(C))))$

A_2: $\neg scadedj \vee \forall\, X(\neg act(X) \vee aur(X) \vee scade(X))$

A_3: $\neg creștedob \vee \forall\, X(\neg oblig(X) \vee scade(X))$

A_4: $\forall X(\neg inv(X) \vee \forall\, C(\neg cump(X, C) \vee \neg scade(C)) \vee \neg bucur(X))$

$\neg C$: $scadedj \wedge creștedob \wedge$
$\exists\, X(inv(X) \wedge bucur(X) \wedge \forall\, C(\neg cump(X, C) \vee \neg act(C) \vee \neg aur(C)))$

Skolemizare: eliminăm cuantificatorii existențiali

4. Pentru $\exists y$ în *interiorul* lui $\forall x_1 ... \forall x_n$, introducem o *funcție Skolem* $y = g(x_1, \ldots, x_n)$: valoarea lui y depinde de $x_1, \ldots x_n$

A_1: $\forall X(\neg inv(X) \vee \exists\, C(cump(X, C) \wedge (act(C) \vee oblig(C))))$
devine

$\forall X(\neg inv(X) \vee (cump(X, f(X)) \wedge (act(f(X)) \vee oblig(f(X)))))$
acel C care există depinde de X $\Rightarrow$ alegem o nouă funcție $f(X)$

Atenție! fiecare cuantificator primește o *nouă funcție* Skolem!

Pentru $\exists y$ în *exterior*, se alege o nouă *constantă Skolem*

$\neg C$: $scadedj \wedge creștedob \wedge \exists\, X(inv(X) \wedge bucur(X)$
$\qquad \wedge \forall\, C(\neg cump(X, C) \vee \neg act(C) \vee \neg aur(C)))$
în loc de $\exists X$ alegem pentru X o nouă constantă b:
$scadedj \wedge creștedob \wedge inv(b) \wedge bucur(b)$
$\qquad \wedge \forall C(\neg cump(b, C) \vee \neg act(C) \vee \neg aur(C))$

Forma normală prenex. Eliminăm cuantificatorii universali

5. Ducem cuantificatorii universali în față: *forma normală prenex*

A_4: $\forall X(\neg inv(X) \vee \forall C(\neg cump(X, C) \vee \neg scade(C)) \vee \neg bucur(X))$
$\forall X \forall C(\neg inv(X) \vee \neg cump(X, C) \vee \neg scade(C) \vee \neg bucur(X))$

6. Eliminăm cuantificatorii universali
(devin impliciți, o variabilă poate fi înlocuită cu orice termen).

A_1: $\neg inv(X) \vee (cump(X, f(X)) \wedge (act(f(X)) \vee oblig(f(X))))$

A_2: $\neg scadedj \vee \neg act(X) \vee aur(X) \vee scade(X)$

A_3: $\neg creștedob \vee \neg oblig(X) \vee scade(X)$

A_4: $\neg inv(X) \vee \neg cump(X, C) \vee \neg scade(C) \vee \neg bucur(X)$

$\neg C$: $scadedj \wedge creștedob \wedge inv(b) \wedge bucur(b)$
$\wedge(\neg cump(b, C) \vee \neg act(C) \vee \neg aur(C))$

Forma clauzală

7. Ducem conjuncția în exterior (prin distributivitate) și scriem fiecare clauză separat (*formă clauzală*)

(1) $\neg inv(X) \vee cump(X, f(X))$

(2) $\neg inv(X) \vee act(f(X)) \vee oblig(f(X)))$

(3) $\neg scadedj \vee \neg act(X) \vee aur(X) \vee scade(X)$

(4) $\neg creștedob \vee \neg oblig(X) \vee scade(X)$

(5) $\neg inv(X) \vee \neg cump(X, C) \vee \neg scade(C) \vee \neg bucur(X)$

(6) $scadedj$

(7) $creștedob$

(8) $inv(b)$

(9) $bucur(b)$

(10) $\neg cump(b, C) \vee \neg act(C) \vee \neg aur(C)$

Generăm rezolvenți până la clauza vidă

Căutăm predicate $P(...)$ și $\neg P(...)$ și unificăm, obținând rezolvenții:

(11) $\neg act(X) \vee aur(X) \vee scade(X)$ (3, 6)

(12) $\neg cump(b, C) \vee \neg act(C) \vee scade(C)$ (10, 11, $X = C$)

(13) $\neg oblig(X) \vee scade(X)$ (4, 7)

Când unificăm, redenumim clauzele să nu aibe variabile comune:

(13) $\neg oblig(Y) \vee scade(Y)$ vom unifica cu (2), redenumim X

(14) $\neg inv(X) \vee act(f(X)) \vee scade(f(X))$ (2, 13, $Y = X$)

(15) $\neg cump(b, f(X)) \vee \neg inv(X) \vee scade(f(X))$ (12, 14, $C = f(X)$)

(16) $\neg cump(b, C) \vee \neg scade(C) \vee \neg bucur(b)$ (5, 8, $X = b$)

(17) $\neg cump(b, C) \vee \neg scade(C)$ (9, 16)

(18) $\neg cump(b, f(X)) \vee \neg inv(X)$ (15, 17, $C = f(X)$)

(19) $\neg inv(b)$ (1, 18, $X = b$)

(20) $\emptyset$ (contradicție = succes în reducerea la absurd) (8, 19)

Rezumat

Putem traduce (*formaliza*) afirmații din limbaj natural în logica predicatelor

Putem *demonstra teoreme* prin reducere la absurd:

- negăm concluzia
- transformăm în *formă clauzală* (conjuncție $\wedge$ de disjuncții $\vee$)
- prin metoda *rezoluției* găsim o *contradicție* (clauza vidă)