Paradigme de Programare

Ș.l. dr. ing. Andrei Olaru

Departamentul Calculatoare
slides: Andrei Olaru & Mihnea Muraru

2013 – 2014, semestrul 2

Cursul 9

Prolog și logica cu predicate de ordinul I

Cuprins

1. Introducere în Prolog
2. Logica propozițională
3. Evaluarea valorii de adevăr
4. Rezoluția

Introducere în Prolog

Prolog
Limbaj de programare logică

- introdus în anii 1970 ;
- programul $\rightarrow$ mulțime de propoziții logice în LPOI;
- mediul de execuție = demonstrator de teoreme care spune:
  - dacă un fapt este adevărat sau fals;
  - în ce condiții este un fapt adevărat.
- Resursă Prolog pe Wikibooks:

  [https://en.wikibooks.org/wiki/Prolog]

Prolog
Caracteristici

- fundamentare teoretică a procesului de raționament;
- motor de raționament ca unic mod de execuție;
  - → modalități limitate de control al execuției.
- căutare automată a valorilor pentru variabilele nelegate (dacă este necesar);
- posibilitatea demonstrațiilor și deducțiilor simbolice.

Logică

- formalism simbolic pentru reprezentarea faptelor și raționament.
- se bazează pe ideea de valoare de adevăr – e.g. *Adevărat* sau *Fals*.
- permite realizarea de argumente (argumentare) și demonstrații – deducție, inducție, rezoluție, etc.

Programare logică

- program scris folosind propoziții logice (clauze Horn pentru Prolog);
- mediul de execuție poate folosi propozițiile pentru a demonstra teoreme sau pentru a deduce fapte.
- pentru a înțelege cum funcționează programele scrise într-un limbaj de programare logică trebuie să înțelegem
  - ce sunt propozițiile, ce înseamnă și cum pot fi ele reprezentate;
  - cum funcționează procesele teoretice pe care se bazează mediul de execuție.

Logica propoziţională

Logica propoziţională
Context şi elemente principale

- Cadru pentru:
  - descrierea proprietăţilor obiectelor, prin intermediul unui limbaj, cu o semantică asociată;
  - deducerea de noi proprietăţi, pe baza celor existente.
- Expresia din limbaj: propoziţia, corespunzătoare unei afirmaţii, ce poate fi adevărată sau falsă.
- Exemplu: “Afară este frumos.”
- Accepţii asupra unei propoziţii:
  - secvenţa de simboluri utilizate sau
  - înţelesul propriu-zis al acesteia, într-o interpretare.

Logica propozițională
Sintaxă

- 2 categorii de propoziții
  - simple $\rightarrow$ fapte atomice: “Afară este frumos.”
  - compuse $\rightarrow$ relații între propoziții mai simple: “Telefonul sună şi câinele latră.”
- Propoziții simple: $p, q, r, \ldots$
- Negații: $\neg\alpha$
- Conjuncții: $(\alpha \wedge \beta)$
- Disjuncții: $(\alpha \vee \beta)$
- Implicații: $(\alpha \Rightarrow \beta)$
- Echivalențe: $(\alpha \Leftrightarrow \beta)$

Logica propozițională
Semantică

- Scop: dezvoltarea unor mecanisme de prelucrare, aplicabile independent de valoarea de adevăr a propozițiilor într-o situație particulară.
- Accent pe relațiile între propozițiile compuse şi cele constituente.
- Pentru explicitarea propozițiilor $\rightarrow$ utilizarea conceptului de interpretare.

Definiția 30.1 (Interpretare).

Mulțime de asocieri între fiecare propoziție simplă din limbaj şi o valoare de adevăr.

Exemplul 30.2.

Interpretarea I:

- $p^I = false$
- $q^I = true$
- $r^I = false$

Interpretarea J:

- $p^J = true$
- $q^J = true$
- $r^J = true$

- cum știu dacă p este adevărat sau fals? Pot ști dacă știu interpretarea – p este doar un *nume* pe care îl dau unei propoziții concrete.

Semantică
Propoziții compuse (1)

- Sub o interpretare *fixată* $\rightarrow$ dependența valorii de adevăr a unei propoziții compuse de valorile de adevăr ale celor constituente
- Negație: $(\neg\alpha)^I = \begin{cases} \mathit{true} & \text{dacă } \alpha^I = \mathit{false} \\ \mathit{false} & \text{altfel} \end{cases}$
- Conjuncție:
$(\alpha \wedge \beta)^I = \begin{cases} \mathit{true} & \text{dacă } \alpha^I = \mathit{true} \text{ şi } \beta^I = \mathit{true} \\ \mathit{false} & \text{altfel} \end{cases}$
- Disjuncție:
$(\alpha \vee \beta)^I = \begin{cases} \mathit{false} & \text{dacă } \alpha^I = \mathit{false} \text{ şi } \beta^I = \mathit{false} \\ \mathit{true} & \text{altfel} \end{cases}$

Semantică
Propoziții compuse (2)

- Implicație:

$$(\alpha \Rightarrow \beta)^I = \begin{cases} \mathit{false} & \text{dacă } \alpha^I = \mathit{true} \text{ şi } \beta^I = \mathit{false} \\ \mathit{true} & \text{altfel} \end{cases}$$

- Echivalență:

$$(\alpha \Leftrightarrow \beta)^I = \begin{cases} \mathit{true} & \text{dacă } \alpha \Rightarrow \beta \ \wedge \ \beta \Rightarrow \alpha \\ \mathit{false} & \text{altfel} \end{cases}$$

Evaluare
Cum determinăm valoarea de adevăr

Definiția 30.3 (Evaluare).

Determinarea valorii de adevăr a unei propoziții, sub o interpretare, prin aplicarea regulilor semantice anterioare.

Exemplul 30.4.

- Interpretarea I:
  - $p^I = false$
  - $q^I = true$
  - $r^I = false$
- Propoziția: $\phi = (p \wedge q) \vee (q \Rightarrow r)$
  $\phi^I = (false \wedge true) \vee (true \Rightarrow false) = false \vee false = false$

Evaluarea valorii de adevăr

Definiția 31.1 (Satisfiabilitate).

Proprietatea unei propoziții care este adevărată sub cel puțin o interpretare. Acea interpretare satisface propoziția.

Exemplul 31.2 (Metoda tabelei de adevăr).

p	q	r	$(p \wedge q) \vee (q \Rightarrow r)$
true	*true*	*true*	*true*
true	*true*	*false*	*true*
true	*false*	*true*	*true*
true	*false*	*false*	*true*
false	*true*	*true*	*true*
false	*true*	*false*	*false*
false	*false*	*true*	*false*
false	*false*	*false*	*false*

Definiția 31.3 (Validitate).

Proprietatea unei propoziții care este adevărată în toate interpretările. Propoziția se mai numeşte tautologie.

Exemplul 31.4 (Validitate).

Propoziția $p \vee \neg p$ este adevărată, indiferent de valoarea de adevăr a lui p, deci este validă.

- Verificabilă prin metoda tabelei de adevăr.

Definiția 31.5 (Nesatisfiabilitate).

Proprietatea unei propoziții care este falsă în toate interpretările. Propoziția se mai numeşte contradicție.

Exemplul 31.6 (Nesatisfiabilitate).

Propoziția $p \Leftrightarrow \neg p$ este falsă, indiferent de valoarea de adevăr a lui p, deci este nesatisfiabilă.

- Verificabilă prin metoda tabelei de adevăr.

Definiția 31.7 (Derivabilitate logică).

Proprietatea unei propoziții de a reprezenta consecința logică a unei mulțimi de alte propoziții, numite premise. Mulțimea de propoziții Δ derivă propoziția ϕ, fapt notat prin $\Delta \models \phi$, dacă şi numai dacă orice interpretare care satisface toate propozițiile din Δ satisface şi ϕ.

Exemplul 31.8.

- $\{p\} \models p \vee q$
- $\{p, q\} \models p \wedge q$
- $\{p\} \not\models p \wedge q$
- $\{p, p \Rightarrow q\} \models q$

- Verificabilă prin metoda tabelei de adevăr: toate intrările pentru care premisele sunt adevărate trebuie să inducă adevărul concluziei.

Exemplul 31.9.

Demonstrăm că $\{p, p \Rightarrow q\} \models q$.

p	q	$p \Rightarrow q$
true	*true*	*true*
true	*false*	*false*
false	*true*	*true*
false	*false*	*true*

Singura intrare în care ambele premise, p şi $p \Rightarrow q$, sunt adevărate, precizează şi adevărul concluziei, q.

Derivabilitate
Formulări echivalente

- $\{\phi_1, \ldots, \phi_n\} \models \phi$

  sau

- Propoziția $\phi_1 \wedge \ldots \wedge \phi_n \Rightarrow \phi$ este validă

  sau

- Propoziția $\phi_1 \wedge \ldots \wedge \phi_n \wedge \neg\phi$ este nesatisfiabilă

Inferența
Motivație

- Creşterea exponențială a numărului de interpretări în raport cu numărul de propoziții simple.
- De aici, diminuarea valorii practice a metodelor semantice, precum cea a tabelei de adevăr.
  - Derivabilitate logică $\rightarrow$ proprietate a propozițiilor, verificabilă prin tabela de adevăr.
- Alternativ, metode sintactice, care manipulează doar reprezentarea simbolică.
  - Inferență $\rightarrow$ Derivare mecanică $\rightarrow$ demers de calcul, în scopul verificării derivabilității logice.
  - folosind metodele de inferență, putem construi o mașină de calcul.

Definiția 31.10 (Inferență).

Derivarea mecanică a concluziilor unui set de premise.

Definiția 31.11 (Regulă de inferență).

Procedură de calcul capabilă să deriveze concluziile unui set de premise. Derivabilitatea mecanică a concluziei ϕ din mulțimea de premise Δ, utilizând regula de inferență *inf*, se notează $\Delta \vdash_{inf} \phi$.

Inferența
Reguli de inferență

- Şabloane parametrizate de raționament, formate dintr-o mulțime de premise şi o mulțime de concluzii.
- exemplu: *Modus Ponens* (MP):

$$\frac{\alpha \Rightarrow \beta \quad \alpha}{\beta}$$

- exemplu: *Modus Tollens*:

$$\frac{\alpha \Rightarrow \beta \quad \neg\beta}{\neg\alpha}$$

Definiția 31.12 (Consistență (*soundness*)).

Regula de inferență determină doar propoziții care sunt, într-adevăr, consecințe logice ale premiselor. Echivalent, $\Delta \vdash_{inf} \phi \Rightarrow \Delta \models \phi$.

Definiția 31.13 (Completitudine (*completeness*)).

Regula de inferență determină toate consecințele logice ale premiselor. Echivalent, $\Delta \models \phi \Rightarrow \Delta \vdash_{inf} \phi$.

- Ideal, ambele proprietăți – “nici în plus, nici în minus”.
- Incompletitudinea regulii *Modus Ponens*, din imposibilitatea scrierii oricărei propoziții ca implicație.

Rezoluția

Rezoluție
O regulă de inferență completă și consistentă

- Regulă de inferență foarte puternică.
- Baza unui demonstrator de teoreme consistent şi complet.
- Spațiul de căutare mai mic decât în alte sisteme.
- Se bazează pe lucrul cu propoziții în forma clauzală:
  - propoziție = mulțime de clauze (semnificație conjunctivă)
  - clauză = mulțime de literali (semnificație disjunctivă)
  - literal = atom sau atom negat
  - atom = propoziție simplă

Forma clauzală
Definiții

Definiția 32.1 (Literal).

Propoziție simplă sau negația ei. E.g. p şi $\neg p$.

Definiția 32.2 (Expresie clauzală).

Literal sau disjuncție de literali. E.g. $p \vee \neg q \vee r$.

Definiția 32.3 (Clauză).

Mulțime de literali dintr-o expresie clauzală. E.g. $\{p, \neg q, r\}$.

Definiția 32.4 (Forma clauzală – CNF).

Reprezentarea unei propoziții sub forma unei mulțimi de clauze, implicit legate prin conjuncții.

Forma clauzală
Exemplu

Exemplul 32.5 (FNC).

Forma clauzală a propoziției

$p \wedge (\neg q \vee r) \wedge (\neg p \vee \neg r)$

este

$\{p\}, \{\neg q, r\}, \{\neg p, \neg r\}$.

Forma clauzală
Obținere

- Orice propoziție convertibilă în această formă astfel:

1. Eliminarea implicațiilor:
$$\alpha \Rightarrow \beta \ \rightarrow \ \neg\alpha \vee \beta$$
2. Avansarea negațiilor până la literali:
$$\neg(\alpha \wedge \beta) \ \rightarrow \ \neg\alpha \vee \neg\beta, \neg(\alpha \vee \beta) \ \rightarrow \ \neg\alpha \wedge \neg\beta,$$
$$\neg(\neg\alpha) \ \rightarrow \ \alpha$$
3. Distribuirea lui $\vee$ față de $\wedge$:
$$\alpha \vee (\beta \wedge \gamma) \ \rightarrow \ (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$
4. Transformarea expresiilor în clauze:
$$\phi_1 \vee \ldots \vee \phi_n \ \rightarrow \ \{\phi_1, \ldots, \phi_n\}$$
$$\phi_1 \wedge \ldots \wedge \phi_n \ \rightarrow \ \{\phi_1\}, \ldots, \{\phi_n\}$$

Exemplul 32.6.

Transformăm propoziția $p \wedge (q \Rightarrow r)$ în formă clauzală.

1. $p \wedge (\neg q \vee r)$ Eliminare implicații
2. $\{p\}, \{\neg q, r\}$ Formare clauze

Exemplul 32.7.

Transformăm propoziția $\neg(p \wedge (q \Rightarrow r))$ în formă clauzală.

1. $\neg(p \wedge (\neg q \vee r))$ Eliminare implicații
2. $\neg p \vee \neg(\neg q \vee r)$ Împinge negații (1)
3. $\neg p \vee (q \wedge \neg r)$ Împinge negații (2,3)
4. $(\neg p \vee q) \wedge (\neg p \vee \neg r)$ Distribuire $\vee$
5. $\{\neg p, q\}, \{\neg p, \neg r\}$ Formare clauze

Rezoluție
Principiu de bază → pasul de rezoluție

- Ideea:

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

- "Anularea" lui p
- p falsă → $\neg p$ adevărată → r adevărată
- p adevărată → q adevărată
- $p \vee \neg p \Rightarrow$ Cel puțin una dintre q şi r adevărată $(q \vee r)$
- Forma generală a pasului de rezoluție:

$$\frac{\{p_1, \ldots, r, \ldots, p_m\} \quad \{q_1, \ldots, \neg r, \ldots, q_n\}}{\{p_1, \ldots, p_m, q_1, \ldots, q_n\}}$$

Rezoluție
Cazuri speciale

- Clauza vidă $\rightarrow$ indicator de contradicție între premise

$$\frac{\{\neg p\} \quad \{p\}}{\{\} = \emptyset}$$

- Mai mult de 2 rezolvenți posibili $\rightarrow$ se alege doar unul:

$$\frac{\{p, q\} \quad \{\neg p, \neg q\}}{\{p, \neg p\} \text{ sau } \{q, \neg q\}}$$

Rezoluție

Alte reguli de inferență $\rightarrow$ cazuri particulare ale rezoluției

- *Modus Ponens*:

$$\frac{p \Rightarrow q \quad p}{q} \quad \sim \quad \frac{\{\neg p, q\} \quad \{p\}}{\{q\}}$$

- *Modus Tollens*

$$\frac{p \Rightarrow q \quad \neg q}{\neg p} \quad \sim \quad \frac{\{\neg p, q\} \quad \{\neg q\}}{\{\neg p\}}$$

- Tranzitivitatea implicației:

$$\frac{p \Rightarrow q \quad q \Rightarrow r}{p \Rightarrow r} \quad \sim \quad \frac{\{\neg p, q\} \quad \{\neg q, r\}}{\{\neg p, r\}}$$

Rezoluție
Demonstrare

- Demonstrarea nesatisfiabilității $\rightarrow$ derivarea clauzei vide.
- Demonstrarea derivabilității concluziei ϕ din premisele $\phi_1, \ldots, \phi_n \rightarrow$ demonstrarea nesatisfiabilității propoziției $\phi_1 \wedge \ldots \wedge \phi_n \wedge \neg\phi$.
- Demonstrarea validității propoziției $\phi \rightarrow$ demonstrarea nesatisfiabilității propoziției $\neg\phi$.
- Rezoluția $\rightarrow$ incompletă generativ, i.e. concluziile nu pot fi derivate direct, răspunsul fiind dat în raport cu o "întrebare" fixată.

Rezoluție
Demonstrare – algoritm

0. Am premisele $\phi_1, \ldots, \phi_n$ și concluzia dorită ϕ
1. Transform $\phi_1, \ldots, \phi_n$ și $\neg\phi$ în FNC
   $\Rightarrow$ mulțime de clauze $\phi_1, \ldots, \phi_n, \neg\phi$
2. Aleg două clauze și aplic pasul de rezoluție
3. **Dacă** rezultatul pasului de rezoluție este clauza vidă ($\emptyset$)
4. **atunci** am terminat demonstrația cu succes
5. **altfel** merg la pasul 2

Exemplul 32.8.

Demonstrăm că $\{p \Rightarrow q, q \Rightarrow r\} \vdash p \Rightarrow r$, i.e. mulțimea $\{p \Rightarrow q, q \Rightarrow r, \neg(p \Rightarrow r)\}$ conține o contradicție.

Exemplul 32.8.

Demonstrăm că $\{p \Rightarrow q, q \Rightarrow r\} \vdash p \Rightarrow r$, i.e. mulțimea $\{p \Rightarrow q, q \Rightarrow r, \neg(p \Rightarrow r)\}$ conține o contradicție.

1.	$\{\neg p, q\}$	Premisă
2.	$\{\neg q, r\}$	Premisă
3.	$\{p\}$	Concluzie negată
4.	$\{\neg r\}$	Concluzie negată
5.	$\{q\}$	Rezoluție 1, 3
6.	$\{r\}$	Rezoluție 2, 5
7.	$\{\}$	Rezoluție 4, 6 $\rightarrow$ clauza vidă

Teorema 32.9 (Rezoluției).

Rezoluția propozițională este consistentă şi completă, i.e. $\Delta \models \phi \Leftrightarrow \Delta \vdash_{rez} \phi$.

- Terminare garantată a procedurii de aplicare a rezoluției: număr finit de clauze $\rightarrow$ număr finit de concluzii.

Sfârșitul cursului 9
Ce am învățat

- Bazele logicii propoziționale, Sintaxă și Semantică
- Inferență, Rezoluție, Forme normale