Logică pentru Informatică - Săptămâna 3
Semantica Logicii Propoziționale

1 Valoarea de adevăr a unei formule într-o atribuire

Reamintim că valoarea de adevăr a unei formule φ într-o atribuire τ este notată cu $\hat{\tau}\Big(\varphi\Big)$ și este definită astfel:

$$\hat{\tau}\Big(\varphi\Big) = \begin{cases} \tau(\varphi), & \text{dacă } \varphi \in A; \\ \overline{\hat{\tau}\Big(\varphi'\Big)}, & \text{dacă } \varphi = \neg\varphi' \text{ și } \varphi' \in LP; \\ \hat{\tau}\Big(\varphi_1\Big) \cdot \hat{\tau}\Big(\varphi_2\Big), & \text{dacă } \varphi = (\varphi_1 \land \varphi_2) \text{ și } \varphi_1, \varphi_2 \in LP; \\ \hat{\tau}\Big(\varphi_1\Big) + \hat{\tau}\Big(\varphi_2\Big), & \text{dacă } \varphi = (\varphi_1 \lor \varphi_2) \text{ și } \varphi_1, \varphi_2 \in LP. \end{cases}$$

În exemplele care urmează, vom folosi următoarele atribuiri:

Exemplul 1.1. *Fie* $\tau_1 : A \to B$ *o funcție definită după cum urmează:*
1. $\tau_1(\mathtt{p}) = 1$*; 2.* $\tau_1(\mathtt{q}) = 0$*; 3.* $\tau_1(\mathtt{r}) = 1$*; 4.* $\tau_1(a) = 0$ *pentru orice* $a \in A \setminus \{\mathtt{p}, \mathtt{q}, \mathtt{r}\}$.

Din moment ce este o funcție de la A *la* B, τ_1 *este o atribuire de valori de adevăr.*

Exemplul 1.2. *Fie* $\tau_2 : A \to B$ *o funcție definită după cum urmează:*
1. $\tau_2(\mathtt{p}) = 0$*; 2.* $\tau_2(\mathtt{q}) = 0$*; 3.* $\tau_2(\mathtt{r}) = 1$*; 4.* $\tau_2(a) = 1$ *pentru orice* $a \in A \setminus \{\mathtt{p}, \mathtt{q}, \mathtt{r}\}$.

Din moment ce este o funcție de la A *la* B, τ_2 *este de asemenea o atribuire.*

Exemplul 1.3. *Fie* $\tau_3 : A \to B$ *o funcție definită după cum urmează:*
1. $\tau_3(a) = 0$ *pentru orice* $a \in A$.

Din moment ce este o funcție de la A *la* B, τ_3 *este de asemenea o atribuire.*

Iată un exemplu de calcul a valorii de adevăr a formulei $(\mathtt{p} \lor \mathtt{q})$ în atribuirea τ_1:

$$\hat{\tau}_1\Big((\mathtt{p} \lor \mathtt{q})\Big) = \hat{\tau}_1\Big(\mathtt{p}\Big) + \hat{\tau}_1\Big(\mathtt{q}\Big) = \tau_1(\mathtt{p}) + \tau_1(\mathtt{q}) = 1 + 0 = 1.$$

Observație 1.1. *Important!*

Nu are sens să spunem "valoarea de adevăr a unei formule". Are sens să spunem "valoarea de adevăr a unei formule într-o atribuire".

De asemenea, nu are sens să spunem "formula este adevărată" sau "formula este falsă". În schimb, are sens să spunem "formula este adevărată în această atribuire" sau "formula este falsă în această atribuire".

Acest lucru deoarece formula ar putea fi adevărată într-o anumită atribuire, dar falsă în altă atribuire. De exemplu, formula $\neg\neg\mathtt{p}$ este adevărată în τ_1, dar falsă în τ_2.

O *atribuire* τ *satisface* φ dacă $\hat{\tau}\Big(\varphi\Big) = 1$. În loc de τ *satisface* φ, putem folosi oricare dintre următoarele forme echivalente:

1. τ este model al formulei φ;
2. φ ține în τ;
3. τ face φ adevărată;

Scriem $\tau \models \varphi$ (și citim: τ este model al formulei φ; sau: τ satisface φ etc.) ddacă $\hat{\tau}\Big(\varphi\Big) = 1$. Scriem $\tau \not\models \varphi$ (și citim: τ nu este model al formulei φ; sau: τ nu satisface φ) ddacă $\hat{\tau}\Big(\varphi\Big) = 0$.

Definiția 1.1. *Relația $\models$, dintre atribuiri și formule, se numește* relația de satisfacere. *Relația este definită astfel: $\tau \models \varphi$ ddacă $\hat{\tau}\Big(\varphi\Big) = 1$.*

Exemplul 1.4. *Atribuirea τ_1 definită mai sus este model pentru $\neg(\mathtt{p} \wedge \mathtt{q})$.*

Atribuirea τ_1 nu este model al formulei $(\neg\mathtt{p} \wedge \mathtt{q})$.

2 Satisfiabilitate

Definiția 2.1. *O formulă φ este* satisfiabilă *dacă, prin definiție, există cel puțin o atribuire τ astfel încât $\tau \models \varphi$ (i.e., dacă φ are cel puțin un model).*

Exemplul 2.1. *Formula $(\mathtt{p} \vee \mathtt{q})$ este satisfiabilă, din moment ce are un model (de exemplu, atribuirea τ_1, definită mai sus).*

Exemplul 2.2. *Formula $\neg\mathtt{p}$ este de asemenea satisfiabilă: de exemplu, atribuirea τ_3, definită mai sus, face formula adevărată.*

Exemplul 2.3. *Formula $(\mathtt{p} \wedge \neg\mathtt{p})$ nu este satisfiabilă, deoarece are valoarea de adevăr fals în orice atribuire.*

Proof. Considerăm o atribuire $\tau : A \to B$ oarecare.

Avem că $\hat{\tau}\Big((\mathtt{p} \wedge \neg\mathtt{p})\Big) = \hat{\tau}\Big(\mathtt{p}\Big) \cdot \hat{\tau}\Big(\neg\mathtt{p}\Big) = \tau(\mathtt{p}) \cdot \overline{\hat{\tau}\Big(\mathtt{p}\Big)} = \tau(\mathtt{p}) \cdot \overline{\tau(\mathtt{p})}$.

Dar $\tau(\mathtt{p})$ poate fi 0 sau 1:

1. în primul caz ($\tau(\mathtt{p}) = 0$), avem că $\hat{\tau}\Big((\mathtt{p} \wedge \neg\mathtt{p})\Big) = \ldots = \tau(\mathtt{p}) \cdot \overline{\tau(\mathtt{p})} = \mathtt{0} \cdot \overline{\mathtt{0}} = \mathtt{0} \cdot \mathtt{1} = \mathtt{0}$;

2. în al doilea caz ($\tau(\mathtt{p}) = 1$), avem că $\hat{\tau}\Big((\mathtt{p} \wedge \neg \mathtt{p})\Big) = \ldots = \tau(\mathtt{p}) \cdot \overline{\tau(\mathtt{p})} = \mathtt{1} \cdot \overline{\mathtt{1}} = \mathtt{1} \cdot \mathtt{0} = \mathtt{0}$.

Din acest motiv, in orice caz, avem că $\hat{\tau}\Big((\mathtt{p} \wedge \neg \mathtt{p})\Big) = \mathtt{0}$. Dar τ a fost o atribuire aleasă arbitrar, și din acest motiv înseamnă că $(\mathtt{p} \wedge \neg \mathtt{p})$ este falsă în orice atribuire τ. Dar acest lucru înseamnă că este nesatisfiabilă. $\square$

O formulă care este nesatisfiabilă se numește *contradicție*.

Exemplul 2.4. *După cum am văzut mai sus,* $(\mathtt{p} \wedge \neg \mathtt{p})$ *este o contradicție.*

3 Validitate

Definiția 3.1. *O formulă* φ *este* validă *dacă, prin definiție, orice atribuire* τ *are proprietatea că* $\tau \models \varphi$ *(orice atribuire este model pentru formulă).*

O formulă validă se mai numește și *tautologie*.

Exemplul 3.1. *Formula* $(\mathtt{p} \vee \neg \mathtt{p})$ *este validă, deoarece este adevărată în orice atribuire: fie* τ *o atribuire oarecare; avem că* $\hat{\tau}\Big((\mathtt{p} \vee \neg \mathtt{p})\Big) = \tau(\mathtt{p}) + \overline{\tau(\neg \mathtt{p})}$*, care este sau* $0+1$*, sau* $1+0$*, deci* 1 *în ambele cazuri.*

Notație 3.1. *Faptul că formula* φ *este validă se mai notează cu* $\models \varphi$.

Exemplul 3.2. *Formula* $\mathtt{p}$ *nu este validă (deoarece există o atribuire (de exemplu* τ_3*) care face formula falsă).*

4 Formule satisfiabile, dar care nu sunt valide

Definiția 4.1. *În engleză, o formulă care nu este nici contradicție și nici tautologie se numește* contingent formula. *În română, nu avem un cuvânt pentru acest concept, și vom folosi exprimarea* formulă satisfiabilă, nevalidă..

Exemplul 4.1. $(\mathtt{p} \wedge \neg \mathtt{p})$ *este o contradicție.*
$\mathtt{p}$ *este o* contingent formula.
$(\mathtt{p} \vee \neg \mathtt{p})$ *este o tautologie.*

5 Equivalențe

Definiția 5.1. *Spunem că două formule* $\varphi_1, \varphi_2 \in LP$ *sunt* equivalente, *și scriem* $\varphi_1 \equiv \varphi_2$*, dacă:*
pentru orice atribuire $\tau : A \to B$, $\hat{\tau}(\varphi_1) = \hat{\tau}(\varphi_2)$.

Intuitiv, formulele care sunt echivalente au același înțeles (adică exprimă același lucru).

Exemplul 5.1. *La începutul acestui curs, cineva a întrebat dacă formulele* $\mathtt{p}$ *și* $\neg\neg\mathtt{p}$ *sunt egale. Evident că nu, am spus, deoarece prima are* 1 *simbol și cealaltă* 3 *simboluri (dacă ar fi fost egale, ar fi avut același număr de simboluri).*

Totuși, acum suntem pregătiți să înțelegem relația dintre ele: $\mathtt{p} \equiv \neg\neg\mathtt{p}$*. Cu alte cuvinte, chiar dacă nu sunt egale, ele sunt echivalente: exprimă același lucru.*

Pentru a demonstra $\mathtt{p} \equiv \neg\neg\mathtt{p}$*, este suficient să arătăm că formulele au aceeași valoare de adevăr în orice atribuire. Fie* τ *o atribuire oarecare. Avem că* $\hat{\tau}\Big(\neg\neg\mathtt{p}\Big) = \overline{\overline{\tau(\mathtt{p})}} = \tau(\mathtt{p}) = \hat{\tau}\Big(\mathtt{p}\Big)$*. Pe scurt,* $\hat{\tau}\Big(\neg\neg\mathtt{p}\Big) = \hat{\tau}\Big(\mathtt{p}\Big)$*. Din moment ce* τ *a fost aleasă arbitrar, urmează că* $\hat{\tau}\Big(\neg\neg\mathtt{p}\Big) = \hat{\tau}\Big(\mathtt{p}\Big)$ *pentru orice atribuire* τ *și de aceeasi* $\mathtt{p}$ *este echivalent cu* $\neg\neg\mathtt{p}$*.*

Exemplul 5.2. *Următoarea echivalență are loc:* $\mathtt{p} \vee \mathtt{q} \equiv \neg(\neg\mathtt{p} \wedge \neg\mathtt{q})$ *(exercițiu: verificați că într-adevăr așa este).*

Iată încă două echivalențe importante, cunoscute sub denumirea de *legile lui De Morgan*:

Teorema 5.1. *Pentru orice formule* $\varphi_1, \varphi_2 \in LP$*, avem că:*

1. $\neg(\varphi_1 \vee \varphi_2) = (\neg\varphi_1 \wedge \neg\varphi_2)$*;*
2. $\neg(\varphi_1 \wedge \varphi_2) = (\neg\varphi_1 \vee \neg\varphi_2)$*.*

6 Consecință semantică

Definiția 6.1. *Fie* $\Gamma = \{\varphi_1, \ldots, \varphi_n, \ldots\}$ *o mulțime (posibil infinită) de formule. Spunem că* φ *este o* consecință semantică *a* Γ*, și scriem* $\Gamma \models \varphi$*, dacă orice atribuire care este model pentru toate formulele* $\varphi_1, \varphi_2, \ldots, \varphi_n$ *din* Γ*, este model și pentru formula* φ*.*

Spunem de asemenea că φ este o consecință logică a Γ sau că φ este o consecință tautologică a Γ în loc de φ *este o consecință semantică a* Γ.

Exemplul 6.1. *Fie* $\Gamma = \{\mathtt{p}, (\neg\mathtt{p} \vee \mathtt{q})\}$*. Avem că* $\Gamma \models \mathtt{q}$*.*

Într-adevăr, fie τ *un model al formulelor* $\mathtt{p}$ *și* $(\neg\mathtt{p} \vee \mathtt{q})$*. Din moment ce* τ *este model pentru* $\mathtt{p}$*, avem prin definiție că* $\tau(\mathtt{p}) = 1$*.*

Din moment ce τ *este model al* $(\neg\mathtt{p} \vee \mathtt{q})$*, avem că* $\hat{\tau}\Big((\neg\mathtt{p} \vee \mathtt{q})\Big) = 1$*. Dar* $\hat{\tau}\Big((\neg\mathtt{p} \vee \mathtt{q})\Big) = \overline{\tau(\mathtt{p})} + \tau(\mathtt{q})$*. Dar* $\tau(\mathtt{p}) = 1$*, și deci* $\hat{\tau}\Big((\neg\mathtt{p} \vee \mathtt{q})\Big) = 0 + \tau(\mathtt{q}) = \tau(\mathtt{q})$*. Înseamnă că* $\tau(\mathtt{q}) = 1$*.*

Deci τ *este model pentru* $\mathtt{q}$*. Am presupus că* τ *este model pentru* $\mathtt{p}$ *și* $(\neg\mathtt{p} \vee \mathtt{q})$ *și am arătat că în model necesar* τ *este și model pentru* $\mathtt{q}$*. Dar aceasta este fix definiția faptului că* $\{\mathtt{p}, (\neg\mathtt{p} \vee \mathtt{q})\} \models \mathtt{q}$*, ceea ce voiam să arătăm.*

Observație 6.1. *Câteodată scriem* $\varphi_1, \ldots, \varphi_n \models \varphi$ *în loc de* $\{\varphi_1, \ldots, \varphi_n\} \models \varphi$*.*

Observație 6.2. *Dacă* $n = 0$, $\varphi_1, \ldots, \varphi_n \models \varphi$ *se reduce la faptul că* φ *este consecință semantică a mulțimii vide. Arătați că acest lucru înseamnă că* φ *este validă, ceea ce justifică notația* $\models \varphi$ *pentru faptul că* φ *este validă.*

Exemplul 6.2. *Avem că* $\mathtt{p}, (\mathtt{p} \vee \mathtt{q}) \not\models \neg\mathtt{q}$, *adică că* $\neg\mathtt{q}$ *nu este consecință logică a formulelor* $\mathtt{p}, (\mathtt{p} \vee \mathtt{q})$. *Pentru a arăta această "neconsecință", este suficient să găsim un model pentru* $\mathtt{p}$ *și pentru* $(\mathtt{p} \vee \mathtt{q})$ *care să nu fie model al formulei* $\mathtt{q}$. *Orice atribuire* τ *cu* $\tau(\mathtt{p}) = 1$ *și* $\tau(\mathtt{q}) = 0$ *satisface cele două proprietăți.*

7 Aplicația 1

Ion scrie următorul cod:

```
if (((year % 4 == 0) && (year % 100!= 0)) || (year%400 == 0))
   printf("%d is a leap year", year);
else
   printf("%d is not a leap year", year);
```

Ioana simplifică codul:

```
if (((year % 4 != 0) || (year % 100 == 0)) && (year%400 != 0))
   printf("%d is not a leap year", year);
else
   printf("%d is a leap year", year);
```

Transformarea făcută de Ioana păstrează comportamentul programului? E dificil să spunem cu certitudine acest lucru dacă doar ne uităm la cod, dar putem să ne folosim de conceptele pe care le-am învățat pentru a modela problema de mai sus folosind uneltele logicii propoziționale și să determinăm dacă cele două programe au același comportament.

Înainte de toate, vom "traduce" condițiile din instrucțiunea if-else în logica propozițională. Vom identifica "propozițiile atomice" și le vom înlocui cu variabile propoziționale după cum urmează. Fie `year` un an fixat:

1. variabila propozițională `p` va ține locul propoziției `(year % 4 == 0)`;
2. variabila propozițională `q` va ține locul propoziției `(year % 100 == 0)`;
3. variabila propozițională `r` va ține locul propoziției `(year % 400 == 0)`.

Ținând cont de traducerea de mai sus, vedem că condiția din programul lui Ion este, în limbajul logicii propoziționale, $((\mathtt{p} \wedge \neg\mathtt{q}) \vee \mathtt{r})$.

Formula Ioanei este, în limbajul logicii propoziționale, $((\neg\mathtt{p} \vee \mathtt{q}) \wedge \neg\mathtt{r})$.

Observați de asemenea că ramurile celor două programe sunt inversate (ramura if a programului lui Ion corespunde ramurii else a programului Ioanei și invers). Pentru care cele două programe să aibă același comportament, este suficient ca negarea formulei lui Ion să fie echivalentă cu formula Ioanei. Este cazul? I.e., are loc echivalența

$$\neg((\mathtt{p} \land \mathtt{q}) \lor \mathtt{r}) \equiv ((\neg\mathtt{p} \lor \neg\mathtt{q}) \land \neg\mathtt{r})?$$

Aplicând legile lui De Morgan, vedem că echivalența are într-adevăr loc și deci transformarea propusă de Ioana este corectă.

8 Implicațiile și echivalențele

Mai există doi conectori logici importanți în logic propozițioanlă: implicațiile și echivalențele.

Folosim notația $(\varphi_1 \rightarrow \varphi_2)$ pentru formula $(\neg\varphi_1 \lor \varphi_2)$. Citim $(\varphi_1 \rightarrow \varphi_2)$ ca "φ_1 implică φ_2".

Similar, folosim $(\varphi_1 \leftrightarrow \varphi_2)$ pentru $((\varphi_1 \rightarrow \varphi_2) \land (\varphi_2 \rightarrow \varphi_1))$.

Exemplul 8.1. *Avem că* $(\mathtt{p} \rightarrow \mathtt{p})$ *este o formulă validă. De ce? Formula* $(\mathtt{p} \rightarrow \mathtt{p})$ *este doar o notație pentru* $(\neg\mathtt{p} \lor \mathtt{p})$*, despre care putem vedea imediat că este validă.*

Vă puteți gândi la conectorii $\rightarrow$ și $\leftrightarrow$ ca fiind macro-uri în limbajul C. De îndată ce vedem $(\varphi_1 \rightarrow \varphi_2)$ pe hârtie, mintea noastră imediat traduce notația și o citim ca $(\neg\varphi_1 \lor \varphi_2)$.

9 Legătura între implicație și consecință semantică

Există o legătură importantă între implicație și noțiunea de consecință semantică, dată de următoarea teoremă.

Teorema 9.1 (Legătura dintre implicație și consecință semantică)**.** *Pentru orice două formule* $\varphi_1, \varphi_2 \in LP$*, avem că* $\varphi_1 \models \varphi_2$ *dacă și numai dacă formula* $(\varphi_1 \rightarrow \varphi_2)$ *este validă.*

De asemenea, are loc următoarea teoremă mai generală:

Teorema 9.2 (Legătura generalizată dintre implicație și consecință semantică)**.** *Pentru orice formule* $\varphi_1, \varphi_2, \ldots, \varphi_n, \varphi \in LP$*, avem că* $\varphi_1, \varphi_2, \ldots, \varphi_n \models \varphi$ *dacă și numai dacă formula* $((((\varphi_1 \land \varphi_2) \land \ldots) \land \varphi_n) \rightarrow \varphi)$ *este validă.*

O legătură similară avem între conectorul logic echivalență și noțiunea de echivalență semantică:

Teorema 9.3 (Legătura dintre conectorul pentru echivalență și noțiunea semantică de echivalență)**.** *Pentru orice două formule* $\varphi_1, \varphi_2 \in LP$*, avem că* $\varphi_1 \equiv \varphi_2$ *dacă și numai dacă formula* $(\varphi_1 \leftrightarrow \varphi_2)$ *este validă.*

10 Aplicația 2

Următorul puzzle este preluat din cartea *Peter Smith. An Introduction to Formal Logic*: Fie majordomul, fie bucătarul a comis crima. Victima a murit otrăvită dacă bucătarul a comis crima. Majordomul a comis crima doar dacă victima a fost înjunghiată. Victima nu a murit otrăvită. Rezultă că victima a fost înjunghiată?

Asociem fiecărei propoziții atomice o variabilă propozițională, după cum urmează:

1. Pentru propoziția "majordomul a comis crima" variabila $\mathtt{p}$;
2. Pentru propoziția "bucătarul a comis crima" variabila $\mathtt{q}$;
3. Pentru propoziția "victima a murit otrăvită" variabila $\mathtt{r}_1$;
4. Pentru propoziția "victima a fost înjunghiată" variabila $\mathtt{r}_2$.

Ipotezele puzzle-ului se modelează în logica propozițională ca următoarele formule:

1. $((\mathtt{p} \vee \mathtt{q}) \wedge \neg(\mathtt{p} \wedge \mathtt{q}))$ (disjuncția exclusivă dintre $\mathtt{p}$ și $\mathtt{q}$);
2. $(\mathtt{q} \rightarrow \mathtt{r}_1)$;
3. $(\mathtt{p} \rightarrow \mathtt{r}_2)$ (atenție la sensul implicației);
4. $\neg\mathtt{r}_1$.

Întrebarea este modelată de formula $\mathtt{r}_2$.

Pentru a răspunde la puzzle cu da/nu, este suficient să verificăm dacă

$$\Big\{((\mathtt{p} \vee \mathtt{q}) \wedge \neg(\mathtt{p} \wedge \mathtt{q})), (\mathtt{q} \rightarrow \mathtt{r}_1), (\mathtt{p} \rightarrow \mathtt{r}_2), \neg\mathtt{r}_1\Big\} \models \mathtt{r}_2.$$

Într-adevăr consecința semantică are loc (exercițiu).