CAPITOLUL 1

NOŢIUNI INTRODUCTIVE

Tematica prezentei lucrări vizează aspectele formale ale gândirii umane, sintetizate sub denumirea de logică, precum şi structurile informaţionale derivate din gândirea logico-matematică aflate la baza conceptului de calculabilitate. Primul capitol îşi propune introducerea noţiunilor fundamentale din care s-au dezvoltat componentele ştiinţei calculatoarelor. Aceste noţiuni vor fi dezvoltate pe parcursul capitolelor următoare.

1.1. Limbajul logico-matematic

Matematica este o expresie a capacităţii gândirii umane de tip raţional. Ea studiază proprietăţile observabile ale obiectelor, atât extern în plan senzorial cât şi intern în plan mintal, în termenii de cantitate, structură, spaţiu şi eveniment. Obiectele studiate în matematică sunt obiecte abstracte, care nu există într-un timp şi/sau spaţiu particular, ci în principal există ca tip sau categorie. Matematica discretă se ocupă cu studiul obiectelor matematice care sunt discrete şi nu continue. Aceste obiecte, în general, nu prezintă o variaţie lină ci au valori distincte sau separate (de exemplu, numerele întregi, grafurile, sau declaraţiile logice). Matematica discretă a câştigat o mare popularitate în ultimul secol datorită aplicaţiilor sale în ştiinţa calculatoarelor. Conceptele şi notaţiile utilizate de matematica discretă se regăsesc în aspectele fundamentale ale structurilor de date, algoritmilor şi limbajelor de programare ale calculatoarelor. Matematica discretă acoperă în principal domenii precum teoria mulţimilor, combinatorica, teoria numerelor, logica, teoria informaţiei, teoria calculabilităţii, complexitate, cercetare operaţională, discretizare.

În procesul de comunicare, apare necesară introducerea unui limbaj matematic. Limbajul matematic este preferabil limbajului natural datorită capacităţii sale de a formaliza eficient declaraţiile informaţionale cu valoare de adevăr. Cuvintele utilizate în limbajul natural nu au întotdeauna o semnificaţie precisă şi pot conduce la ambiguităţi. În acest sens, vom încerca o primă definire a limbajului matematic ca fiind **un mijloc de exprimare precisă care are un conţinut semantic foarte clar definit.** De exemplu, „*Un număr prim este un număr întreg pozitiv, care nu are alţi factori în afara lui însuşi şi a lui 1*”, sau „*Un număr prim este un număr întreg pozitiv, care are doi factori, pe el însuşi şi pe 1*”. Un concept matematic poate fi legat de un altul definit corespunzător. Spre exemplu, conceptul de număr prim este legat de conceptul de număr compus. „*Un număr compus este orice număr care poate fi exprimat ca un produs de întregi pozitivi mai mici decât el însuşi*”. De exemplu: 4 (2 x 2), 6 (2 x 3), 8 (2 x 2 x 2), 9 (3 x 3), 10 (2 x 5), 12 (2 x 2 x 3) …. Evident că, numărul compus se poate scrie ca un produs de numere prime.

Vom vorbi, de asemenea, despre teoreme, leme, propoziţii şi corolare. Acestea sunt enunţuri prin care se arată adevărul unei propoziţii afirmative. De exemplu, ”*Toate numerele pare, cu excepţia lui 2 nu sunt numere prime*”, sau „*Există un număr infinit de numere prime*”, sau, teorema fundamentală a aritmeticii, „*Fiecare număr întreg pozitiv poate fi factorizat în mod unic ca un produs de primi*” (numărul 1 poate fi gândit ca un produs de 0 numere prime).

Legat de limbaj, introducem, de asemenea, conceptul de demonstraţie, care este o justificare prin descompunerea într-o succesiune de paşi a adevărului afirmaţiilor făcute. Înţelegem prin demonstraţie informală comunicarea în limbaj natural, folosind şi anumite simboluri, a paşilor care conduc la recunoaşterea adevărului afirmaţiilor făcute. Demonstraţia trebuie să fie însă acoperitoare pentru toate cazurile, fără să lase loc de îndoială. În decursul istoriei matematicii, s-au dezvoltat mai multe tehnici care să conducă la stabilirea adevărului afirmaţiei respective. Problematica stabilirii adevărului în general este de natură filosofică şi îşi are originile în gândirea umană. Întâlnim astfel, în filosofia antică, conceptul de *logos*, care se traduce prin cuvânt, limbaj, sunet, raţionament. Există în mod clar o legătură între limbaj şi raţionament, în sensul că orice raţionament care poate fi făcut necesită un limbaj. În momentul în care vorbim despre limbaj, vorbim despre o formalizare, în sensul conceperii unui set de simboluri prin intermediul cărora exprimăm sau împuternicim cuvinte cu conţinut semantic, simbolistica fiind valabilă atât pentru limbajul scris, cât şi pentru cel vorbit. Cu sensul de simboluri, stabilim un alfabet prin intermediul căruia, urmând nişte reguli sintactice, exprimăm nişte afirmaţii cu valoare de adevăr. Înlănţuirea după reguli precise a afirmaţiilor formează discursul logic.

Suntem interesaţi în a elabora un set complet de reguli sintactice, respectiv de manipulare a simbolurilor astfel încât, pornind de la premise să înlănţuim propoziţiile cu valoare de adevăr, ajungând la concluzii adevărate. Logica tradiţională a stat la baza formării raţionamentului matematic, prin aceea că regulile de înlănţuire logică, prin care conţinutul semantic al premiselor era păstrat în urma raţionamentului şi transferat concluziilor, formează în esenţă principiul raţionamentului. Deosebirea faţă de logica exprimată în limbaj natural este aceea că se utilizează limbajul matematic. Suntem însă oricând capabili de a parafraza o afirmaţie logică naturală, adică de a o exprima în limbaj matematic.

1.2. Afirmaţii logice

Afirmaţiile logice sunt cele care au valoare de adevăr fie pozitiv, fie negativ, adică sunt fie adevărate, fie false. . Există şi alte propoziţii care nu au valoare de adevăr, cum sunt de exemplu întrebările, comenzile, rugăciunile. S-a încetăţenit o modalitate familiară de a reprezenta afirmaţiile logice: fie ***S*** o anumită afirmaţie şi **¬*S*** (**nu *S***) negata ei. Spunem că o asemenea reprezentare este familiară în forma ce se numeşte *tabel de adevăr*. Acesta exprimă pe rânduri relaţiile valorilor de adevăr între ***S*** şi **¬*S***:

S	**¬*S***
Adevărat	Fals
Fals	Adevărat

Remarcăm că negaţia poate lega declaraţii despre *fiecare* caz cu declaraţii despre *anumite* cazuri. De exemplu, declaraţia „Nu fiecare femeie are copil” are aceeaşi semnificaţie cu „Anumite femei nu au copil”. Similar, declaraţia „Nu este cazul ca o anumită mamă să nu aibă copil” are acelaşi înţeles cu „Fiecare mamă are copil”.

Există posibilitatea de a executa operaţii logice cu enunţurile respective, şi care pot fi, de asemenea, descrise familiar prin tabelul de adevăr. Aceste operaţii sunt *conjuncţia* a două afirmaţii, „*A* şi *B*” ($A \wedge B$), respectiv *disjuncţia* între două afirmaţii „*A*” şi „*B*”, „*A* sau *B*” ($A \vee B$).

Fie tabelul următor:

A	***B***	***A* şi *B***	***A* sau *B***
Adev.	Adev.	Adev.	Adev.
Adev.	Fals	Fals	Adev.
Fals	Adev.	Fals	Adev.
Fals	Fals	Fals	Fals

Completarea acestui tabel de adevăr care implică operaţii logice elementare este un exerciţiu natural de contemplare logică.

Conjuncţiile şi disjuncţiile pot fi parafrazate. De exemplu, „Maria este femeie şi Lucia este femeie” poate fi înlocuită cu „Maria şi Lucia sunt femei”. În loc de „*a* este natural sau *b* este natural”, putem afirma „Ori *a* ori *b* este natural”.

Implementarea fizică a unei funcţii logice se numeşte poartă logică. Este interesant de a remarca aici că operaţia logică de negare îşi găseşte o implementare naturală sub forma inversorului generic. Acesta este format dintr-o sarcină potenţială şi un dispozitiv de comandă faţă de un nivel de referinţă, ca în Fig. 1.1. Comanda descărcării sarcinii către referinţă conduce la absenţa sarcinii la ieşire. Sarcina **S** poate fi de orice natură, electrică, hidraulică, mecanică etc.

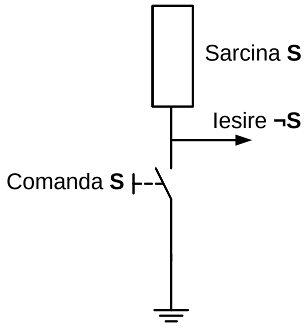

Fig. 1.1. Inversorul generic

Pornind de la inversor, prin *serializarea* respectiv *paralelizarea* funcţiei dispozitivului de comandă putem, în mod avantajos, forma operaţia logică de conjuncţie respectiv disjuncţie, în forma negată, ca în Fig. 1.2.

Porţile logice pot fi construite în multe tehnologii diferite. De exemplu, o sarcină electrică poate fi descărcată, prin intermediul a două comutatoare înseriate, pe un consumator de tipul unui bec. Se formează astfel o poartă logică de tip „ŞI” (Fig. 1.3). Prin diferite combinaţii serie-paralel de comutatoare se formează un circuit logic. De exemplu, in Fig. 1.4 se arată circuitul logic corespunzător funcţiei $A \wedge (B \vee C)$. Valoarea circuitului este „adevărat” atunci când sarcina electrică ajunge să aprindă becul. În caz contrar, valoarea este „fals”. Mai jos, se arată tabelul de adevăr pentru această funcţie logică cu trei variabile de intrare şi o ieşire. Intrările *A, B, C* au valoarea logică 1 atunci când comutatorul care le poartă numele este închis. În tehnologia actuală, structura logică a circuitelor este aceeaşi, dar se utilizează transistoare integrate în loc de circuite electrice.

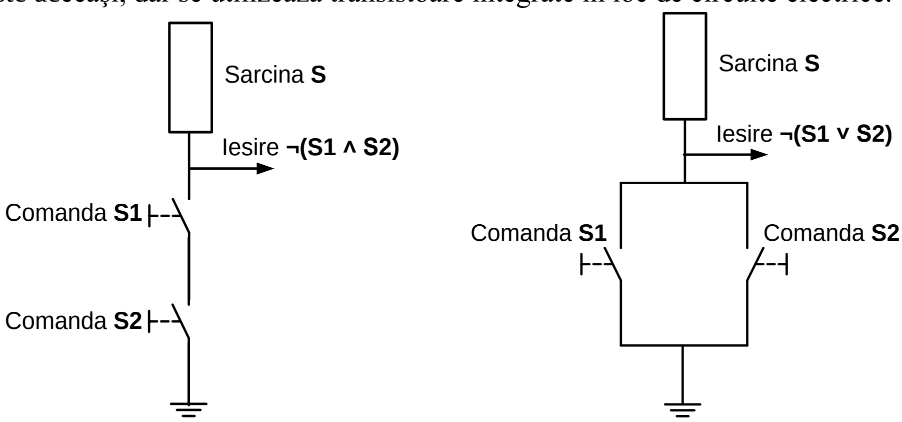

Fig. 1.2. Formarea conjuncţiei şi disjuncţiei negate pornind de la inversor

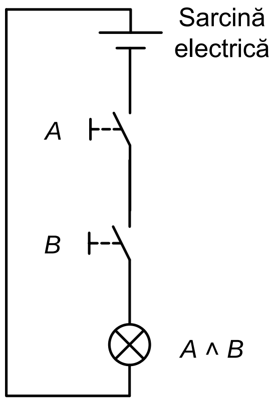

Fig. 1.3. Formarea porţii logice „ŞI” printr-un circuit de aprindere a unui bec electric.

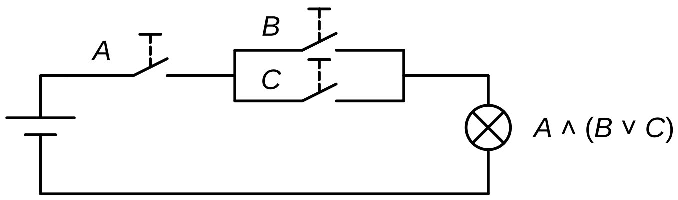

Fig. 1.4. Circuitul logic corespunzător funcţiei $A \wedge (B \vee C)$ în tehnologie electrică.

A	***B***	***C***	***A* ∧ (*B* ∨ *C*)**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

1.3. Implicaţia logică

Multe propoziţii sunt formulate în forma „Dacă *p* atunci *q*”, unde *p* şi *q* sunt de asemenea declaraţii logice. O asemenea formulare se numeşte implicaţie logică sau declaraţie condiţională, în care *p* este ipoteza (premisa) şi *q* este concluzia (consecinţa). Interpretarea acestei declaraţii se poate face în mai multe feluri. De exemplu: „*p* este o condiţie suficientă pentru *q*”, sau „*q* este o condiţie necesară pentru *p*”, sau „*p* implică *q*”.

Tabelul de adevăr al implicaţiei logice, notată $\boldsymbol{p \to q}$**,** este următorul:

p	***q***	***p* → *q***
Adev.	Adev.	Adev.
Adev.	Fals	Fals
Fals	Adev.	Adev.
Fals	Fals	Adev.

Se remarcă faptul că declaraţia condiţională este falsă doar când ipoteza este adevărată şi concluzia este falsă. În toate celelalte trei cazuri condiţionala este adevărată.

Există o tendinţă de a interpreta condiţionala astfel: „Dacă *p* atunci *q*” înseamnă că „*q* poate fi demonstrat din *p*”, ceea ce presupune că *p* şi *q* sunt într-o oarecare legătură. Astfel, o afirmaţie de genul „Dacă pământul este gol pe dinăuntru atunci 1=2” este acceptată ca adevărată chiar dacă nu este o legătură între ipoteză şi concluzie. În schimb, declaraţia „Dacă 1=1 atunci pământul este gol pe dinăuntru” nu poate fi acceptată ca adevărată. Când ipoteza unei condiţionale este falsă spunem că este în mod stupid adevărată. De exemplu, „Dacă 1=2 atunci 34=23” este stupid adevărată deoarece ipoteza este falsă. Dacă concluzia este adevărată, spunem că condiţionala este trivial adevărată. De exemplu, „Dacă 1=2 atunci 1+1 =2” este trivial adevărată deoarece concluzia este adevărată.

Condiţionala reciprocă sau contrară lui „dacă *p* atunci *q*” este „dacă *q* atunci *p*”. Reciproca nu are întotdeauna aceeaşi valoare de adevăr. De exemplu,

Dacă $x > 0$ şi $y > 0$ atunci $x + y > 0$.

Reciproca este

Dacă $x + y > 0$ atunci $x > 0$ şi $y > 0$

Reciproca este falsă. Aceasta se verifică printr-un contra-exemplu, de pildă pentru valorile $x = 4$ şi $y = -3$.

1.4. Echivalenţa logică

Tabelul de adevăr al echivalenţei logice, notată $\boldsymbol{p \leftrightarrow q}$**,** este următorul:

p	***q***	$\boldsymbol{p \leftrightarrow q}$
Adev.	Adev.	Adev.
Adev.	Fals	Fals
Fals	Adev.	Fals
Fals	Fals	Adev.

Astfel, două declaraţii sunt considerate echivalente dacă au aceeaşi valoare de adevăr pentru orice atribuire de valori adevărate variabilelor din expresie. În acest context, este interesant de studiat modul în care se formează declaraţiile logice echivalente. Putem combina negata cu disjuncţia sau conjuncţia pentru a obţine declaraţii echivalente.

„nu (*p* şi *q*) este echivalent cu „(nu *p*) sau (nu *q*)”
„nu (*p* sau *q*) este echivalent cu „(nu *p*) şi (nu *q*)”

De exemplu, „nu ($x > 0$ şi $y > 0$)” este echivalent cu „$x \leq 0$ sau $y \leq 0$” şi declaraţia „nu ($x > 0$ sau $y > 0$)” este echivalentă cu „$x \leq 0$ şi $y \leq 0$”.

Conjuncţiile şi disjuncţiile se distribuie una pe cealaltă astfel:

„ *p* şi (*q* sau *r*)” este echivalent cu „(*p* şi *q*) sau (*p* şi *r*)”.
„*p* sau (*q* şi *r*)” este echivalent cu „(*p* sau *q*) şi (*p* sau *r*)”.

Contrapoziţia declaraţiei condiţionale „daca *p* atunci *q*” este forma echivalentă „dacă nu *q* atunci nu *p*”. De exemplu, declaraţia

Dacă $x > 0$ şi $y > 0$ atunci $x + y > 0$

este echivalentă cu forma

Dacă $x + y = 0$ atunci ($x \leq 0$ sau $y \leq 0$)

Este important de reţinut că declaraţia condiţională „dacă p atunci q” se mai poate exprima sub forma echivalentă „nu p sau q”. De exemplu, declaraţia:

Dacă $x > 0$ şi $y > 0$ atunci $x + y > 0$

este echivalentă cu forma

($x \leq 0$ sau $y \leq 0$) sau $x + y > 0$.

Ţinând cont de forma echivalentă a implicaţiei în termenii negaţiei şi disjuncţiei, rezultă evident că declaraţiile „nu (dacă p atunci q)” şi „p şi (nu q)” sunt echivalente. De exemplu, declaraţia

„Nu este cazul că dacă Maria este mamă atunci Maria nu are copil”.

este echivalentă cu declaraţia

„Maria este mamă şi Maria are copil”.

În tabelul următor se dau expresiile echivalente concise conform celor discutate anterior.

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

1.5. Numere

Obiectul fundamental în matematica discretă este numărul natural. Conceptul de „număr” este unul dintre cele mai misterioase producţii ale minţii umane, în sensul că definirea sa necesită o profundă înţelegere a naturii minţii umane. Ne vom referi pe moment doar la un vestit aforism elaborat de matematicianul Kronecker, şi anume: ”Dumnezeu a creat numerele naturale: 0,1,2,....Toate celelalte sunt creaţia omului.”. Prin aceasta putem înţelege că numerele naturale sunt obiecte fireşti ale minţii sau gândirii umane. Gândim natural în termeni numerici când numărăm, începând cu numărul 1 (există „şase” mere în coş) sau ordonăm („al patrulea” măr este cel mai frumos). Pornind de aici, putem vorbi despre numere cardinale, folosite pentru a desemna o cantitate (de exemplu, zece persoane, trei biciclete) şi numere ordinale, folosite pentru a arăta ordinea (primul, al doilea, al treilea etc.) unor obiecte într-o mulţime (de exemplu, al cincilea din rând, al zecelea ca putere). Aspectele abstracte ale numerelor reies însă atunci când ne întrebăm: Ce este „şase” şi „al patrulea” în afară de mere? În afară de numerele cardinale şi ordinale mai putem vorbi despre numere nominale (de exemplu, un număr de casă, de telefon, sau codul numeric personal). Numerele nominale nu arată nici cantitatea şi nici rangul, ci sunt utilizate pentru a identifica un obiect.

Aşa cum am exemplificat deja, în introducerea conceptelor facem apel la diverse tipuri de definiţii. Rolul definiţiei este acela de a furniza cunoştinţe valide din punct vedere raţional asupra proprietăţilor caracteristice ale unui obiect respectiv a relaţiilor dintre elementele sale componente. În formarea definiţiei, apare deseori utilă încadrarea obiectului într-un *gen proxim* (clasă de obiecte) şi evidenţierea *diferenţei specifice*, adică a proprietăţilor caracteristice doar obiectelor desemnate de definit în raport cu celelalte obiecte aparţinând genului. De exemplu, „Un număr întreg este un număr natural pozitiv sau negativ, sau un număr care nu are o componentă fracţionară".

Bazat pe conceptul de numere întregi, definim noţiunea de număr par şi de număr impar, cu ajutorul operaţiei de divizare, şi spunem că un număr întreg n e un divizor al lui m dacă există un număr întreg k astfel încât $n \times k = m$. Divizarea stabileşte natura numerelor prime, care au o importanţă majoră în ştiinţa calculatoarelor. Adesea ne interesează să scriem numerele în termeni numiţi *modulo* faţă de alt număr. Dacă n şi m sunt numere întregi, atunci există numerele k şi l, astfel încât $n = m \times k + l$, unde $0 \leq l \leq m - 1$, $k \geq 0$. De exemplu, $25 = 3 \times 8 + 1$.

Uneori nu suntem interesaţi decât de valoarea lui l, astfel încât, în asemenea cazuri exprimăm $n = l \bmod m$. În ştiinţa calculatoarelor, aritmetica numerelor este interesantă pe de o parte pentru că ne permite păstrarea scrierii la dimensiuni rezonabile $(0, n)$ şi pe de altă parte este utilă în operaţii cu şiruri binare. Interesant de menţionat aici sunt anumite proprietăţi în reprezentarea modulo. Astfel, dacă $n_1 = l_1 \bmod m$, $n_2 = l_2 \bmod m$, atunci operaţia de adunare, respectiv înmulţire se transferă la nivelul resturilor: $n_1 + n_2 = (l_1 + l_2) \bmod m$, şi $n_1 \text{ x } n_2 = l_1 \text{ x } l_2 \bmod m$. Remarcăm că dacă n e un număr compus, se poate întâmpla să avem cazul $n_1 \text{ x } n_2 = 0 \bmod m$. Acest caz nu e adevărat dacă m e prim.

În limbajul matematic suntem interesaţi în a exprima, pe de-o parte, afirmaţii care să poată fi formalizabile utilizând cantităţi numerice şi, pe de altă parte, afirmaţii în care avem de a face cu valori logice. În toate cazurile însă sunt implicate valori discrete. Metodele matematice necesare pentru realizarea de calcule ştiinţifice şi inginereşti trebuie transformate din domeniul continuu în cel discret pentru a putea fi îndeplinite de un calculator.

1.6. Calculabilitate

Circuitele logice, ca cele din Fig. 1.4, pot fi privite ca executând un program în linie dreaptă. Este un program care conţine doar declaraţii de atribuire, fără bucle sau ramificări. Astfel, circuitele logice reprezintă cel mai simplu model de calculabilitate şi formează blocurile de bază în calculatoarele de astăzi. Aşa cum rezultă din tabelul de adevăr pentru Fig. 1.4, fiecare circuit logic are asociată o funcţie binară care mapează valorile variabilelor sale de intrare în valori ale variabilelor de ieşire. Funcţiile sunt astfel foarte importante pentru a avea o definire precisă a sarcinilor de îndeplinit. În descrierea funcţiilor, un loc central e deţinut de problema posibilităţii de a fi calculată, adică dacă o funcţie este efectiv sau mecanic calculabilă. Funcţiile care pot fi astfel calculate se numesc *recursive*. Una din definiţiile atribuite funcţiilor recursive este aceea că funcţiile recursive sunt calculabile de către o maşină matematică idealizată, numită „maşina Turing", după numele matematicianului britanic Alan Turing.

Maşina Turing este cel mai complex model de calculabilitate şi e considerat modelul standard. Aceasta din cauză că nu s-a descoperit ulterior nici un alt model de maşină care să execute operaţii pe care maşina Turing să nu le poată executa. Există însă şi alte modele de calculabilitate care pot fi dezvoltate pornind de la modelul cel mai simplu al circuitelor logice. Astfel, prin combinarea circuitelor logice cu celule de memorie binară se pot construi automate sau maşini cu memorie. Ele se numesc maşini cu stări finite (eng. *finite-state machine*) şi sunt prezente în practic toate aplicaţiile societăţii moderne. Pe baza maşinilor cu stări finite se pot dezvolta calculatoare de uz general prin modelul numit maşina cu acces aleator (eng. *random-access machine*). Modelul conţine o pereche de maşini cu stări finite interconectate, o unitate centrală de procesare, de o parte, şi de cealaltă parte o memorie cu acces aleator. Această maşină are memorie finită, spre deosebire de modelul maşinii Turing care este idealizat, admiţând o memorie oricât de mare. O formă restrânsă de maşină Turing este maşina cu stivă (eng. *pushdown automaton*). În concluzie, ideea de bază care se cuvine reţinută este importanţa modelelor de calculabilitate. Prin utilizarea unor modele adecvate se asigură un nivel de abstractizare care permite o viziune mai profundă asupra posibilităţii îndeplinirii sarcinilor şi dezvoltării performanţelor de calcul. Un model bun poate rămâne ca referinţă în timp şi poate de asemenea inspira înspre găsirea unor noi tehnologii de implementare mai avantajoasă a maşinilor de calcul.

Exerciții şi probleme

1. Fie p, q, r trei propoziţii (declaraţii logice). Arătaţi, utilizând tabelul de adevăr, că expresia logică $(p \wedge q) \Rightarrow r$ este echivalentă cu expresia $(p \Rightarrow (q \Rightarrow r))$.

2. Fie p, q, r trei propoziţii (declaraţii logice). Arătaţi utilizând tabelul de adevăr că expresia logică $(p \wedge q) \Rightarrow r$ este echivalentă cu expresia $(p \Rightarrow r) \vee (q \Rightarrow r)$, dar nu este echivalentă cu expresia $(p \Rightarrow r) \wedge (q \Rightarrow r)$.

3. Arătați că suma a doi întregi impari este un întreg par, adică dacă x și y sunt întregi impari atunci $x + y$ este un întreg par.

4. Arătați adevărul următoarei declarații privind divizibilitatea:

   Dacă $d \mid a$ și $a \mid b$, atunci $d \mid b$.