

# 5. Inteligență Artificială

## 5. Inteligență Artificială

### 5.1 ISTORIC

### 5.2 BAZELE LOGICE ALE A.I.

### 5.3 AGENȚI INTELIGENȚI

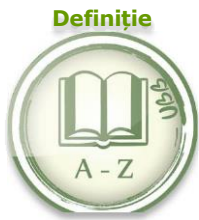
Întrebări recapitulative

## 5.1 ISTORIC

*„Nici un lanț nu este mai puternic decât cea mai slabă verigă a lui.”*

### 5.1.1 Descriere Generală

#### 5.1.1.1 Definiții ale Inteligenței Artificiale



**(D.W.Patterson, 1990)** – ramură a informaticii care se ocupă de studierea și realizarea SC cu elemente de inteligență:

- învață concepte și sarcini noi;
- analizează, trage concluzii și reacționează la mediul înconjurător;
- înțelege limbajul natural;
  - efectuează activități care necesită elemente de inteligență umană.

**(N. Nilson, 1998)** – „comportament inteligent în artificial; comportament inteligent – percepție, raționament, instruire, comunicare și acțiune într-un mediu complex”

**Luger - 1992** „ramura informaticii destinată automatizării comportamentului inteligent” (Microsoft1999)

**Larouse** - „Ansamblu de tehnici utilizate pentru realizarea de automate apropiate de gândirea și modul de acțiune umană”.

**(Russell&Norvig – 2003)** rezumă definiția dată în 8 cărți – într-un tabel bidimensional:

- orizontal legat de:
  - stânga – apropierea față de performanța umană;
  - dreapta – apropierea de conceptul ideal de inteligență – raționalitate;
- vertical legat de:
  - sus – procesul de gândire;
  - jos – comportament;

Sistem ce raționează uman	Sistem care raționează
<b>Haugeland 1985</b> – „efortul de a face mașinile să gândească... mașini care gândesc în sens complet ad-literam” <b>Bellman 1978</b> – „automatizarea activităților asociate gândirii umane, activități cum ar fi: luarea deciziilor, rezolvarea problemelor, învățarea...”	<b>Cahrniak &amp; McDermott 1985</b> – „studiul modelului mental prin utilizarea modelelor computationale” <b>Winson 1992</b> – „studiul calculului care face posibilă percepția, raționamentul și acțiunea”
Sisteme care se comportă ca oamenii	Sisteme care acționează rațional
<b>Kurzweil 1992</b> - „arta de a crea mașini care realizează funcții ce necesită inteligență analogă oamenilor” <b>Rich &amp; Knight 1991</b> – „studiul de a face calculatoarele să efectueze lucruri în care deocamdată oamenii sunt mai buni”	<b>Poole &amp; alții 1998</b> – „inteligența computațională este studiul proiectării agenților inteligenți” <b>Nilson 1998</b> – „comportament inteligent în artificial”

**Concluzie.** Domeniu recent al informaticii – dezideratul primilor constructori de calculatoare – emulează gândirea umană.

**Acțiunea umană** – Testul Turing (Alan Turing 1950) – definiția operațională a inteligenței umane – punând în scris un număr de întrebări, calculatorul răspunde ca un om. Caracteristici pentru calculatoare:

- prelucrarea limbajului natural;
- reprezentarea cunoștințelor;
- raționarea automată – memorează informații pentru a răspunde la întrebări și trage concluzii noi;
- machine learning: extrage și extrapolează modele;

Pentru test Turing:

- vedere artificială – să recepționeze obiecte;
- robotică – să manipuleze obiecte;

AI nu trebuie să copieze cea umană – exemplul cu avionul, care nu copiază zborul păsărilor sau mașina de spălat, care nu copiază spălatul manual.

**Gândire umană** – abordarea cognitivă -> trebuie cunoscut modul de raționare uman – 2 căi:

- introspecție – analiza gândirii;
- experiențe psihologice;

Studiul comportamentului uman prin – analiza I/O & comportamentul în timp. Exemplu: Allen Newel & Herbert Simon 1961 – GPS „General Program Solver” – nu au urmărit rezolvarea unor categorii de probleme concrete și nici dacă programul rezolvă corect problemele, ci trasarea raționamentului sistemului comparativ cu cel uman. Diferențele AI față de științe cognitive: științele cognitive încearcă să înțeleagă raționamentul și limbajul uman; AI încercă să transpună aceste raționamente pe sisteme de calcul.

**Gândire rațională** – „legea gândirii” – Aristotel – logica formală– „gândirea corectă” – silogismul (modus ponens) „Socrate este un om – Orice om este muritor – Socrate este muritor”.

Premize corecte -> concluzii corecte – prima abordare în logică. Notăția matematică din secolul 19 (logica formală, matematică) -> 1965 program care poate rezolva orice problemă descriptibilă prin FOPL.

Probleme:

- formalizarea unor probleme neformale sau parțial formalizabile;
- trecerea de la „în principiu” la realizare ~ complexitatea calcului la câteva sute de fapte.

**Acțiunea rațională** – abordarea agentului rațional; definiție agent – agent rațional (acțiune autonomă, percepe mediul, persistă un timp îndelungat, adaptare la schimbări, alegerea țintei) – caută cea mai bună soluție sau în condiție de incertitudine – o soluție bună.

**Artificial intelligence (AI)** is defined as intelligence exhibited by an artificial entity. Such a system is generally assumed to be a computer. Although AI has a strong science fiction connotation, it forms a vital branch of computer science, dealing with intelligent behavior, learning and adaptation in machines. Research in AI is concerned with producing machines to automate tasks requiring intelligent behavior. Examples include control, planning and scheduling, the ability to answer diagnostic and consumer questions, handwriting, speech, and facial recognition. As such, it has become a scientific discipline, focused on providing solutions to real life problems. AI systems are now in routine use in economics, medicine, engineering and the military, as well as being built into many

common home computer software applications, traditional strategy games like computer chess and other video games<sup>1</sup>.



Ce este inteligența artificială? (John McCharty, 24 nov. 2004)<sup>2</sup>

### 5.1.1.2 Noțiuni legate de domeniul discursului

Ca în alte domenii complexe, în AI se presupun înțelese o serie de noțiuni și categorii: inteligență, cunoaștere, raționare, învățare, gândire, etc. – legate de informatică. Înainte de o descriere riguroasă – una semantică.

**Oxford Dictionary:** *inteligența* – activitatea de a achiziționa, înțelege și aplica cunoștințe, de a executa raționamente și judecăți. Mai mult – inteligența înseamnă înglobarea unor cunoștințe și fapte achiziționate experimental sau prin raționament, conștient sau inconștient.

**E. Feigebaum** – definește *inteligența* ca: o percepție superioară a imaginilor și sunetelor, raționament, imaginație, abilitatea de a citi, a scrie, a conversa, a conduce mașina, a memora, a reaminti fapte petrecute cu mult timp în urmă, a exprima stări emoționale, etc.

Fundamentul inteligenței sunt cunoștințele.

### 5.1.1.3 Relația reală dintre AI și cea umană – specifică și generală

Există posibilitatea de a construi sisteme inteligente? – Da!

- sisteme care învață (case base reasoning, reinforcement learning, machine learning, rețele neuronale);
- sintetizează experiențe anterioare (sisteme expert);
- raționează și explică raționamentele (sisteme expert, agenți inteligenți);
- rezolvă probleme complexe de matematică sau de joc (șah);
- planifică sarcini (sisteme suport de decizie);

---

<sup>1</sup> [http://en.wikipedia.org/wiki/Artificial\\_intelligence](http://en.wikipedia.org/wiki/Artificial_intelligence)

<sup>2</sup> <http://www-formal.stanford.edu/jmc/whatisai/>

- determină configurații optimale pentru sisteme complexe;
- elaborează strategii militare și macroeconomice complexe;
- diagnostichează boli;
- înțeleg limbajul natural;
- recunosc imagini și forme preluate prin camere video sau senzori (roboți inteligenți, robotul Honda [Wikipedia]), etc.

Poate AI rezolva probleme generale de inteligență? – Nu – nu depășește inteligența unui copil de 3 ani:

- recunoașterea unui număr mare de obiecte din mediu;
- învățarea unor sunete noi și asocierea cu obiecte și concepte;
- adaptarea la situații noi, neprevăzute.

#### 5.1.1.4 Scopul și domeniile



**Nu se ocupă:** sistemele de calcul convenționale, corpul uman, limbajul uman, comportamentul uman (psihologia, filozofia, lingvistica, etc.) – relația dintre acestea și AI.

**Scopul** – realizarea unor sisteme cu un grad înalt de inteligență, în anumite domenii depășește inteligența umană.

Domeniile de bază:

- robotica inteligentă și domeniile conexe;
- organizarea memoriei;
- reprezentarea cunoștințelor;
- memorare, căutare și regăsirea a informațiilor;
- modele de învățare;
- tehnici de interfață inteligentă;
- rezolvarea unor probleme logice complexe;
- recunoașterea formelor, vedere artificială și realitatea virtuală;
- recunoașterea și sinteza vocii;
- suport decizional în condiții de incertitudine;
- limbaj natural;
- varietate de instrumente (sisteme expert, rețele neouronale, sisteme fuzzy, algoritmi genetici, etc).

- AI distribuită și sisteme de agenți inteligenți, etc.
- teoria jocurilor și planificare strategică;
- creativitate artificială;
- sisteme inteligente hibride;
- control inteligent;
- Data Mining, etc.

#### **5.1.1.5 Discipline care au contribuit la dezvoltarea AI.**

Domeniu distinct în jurul anilor 50 – domenii discutate atunci au devenit ramuri ale AI. Detalii relativ la domenii [Norvig2003].

#### **Filosofia**

Probleme:

- Regulile formale pot duce la concluzii corecte?
- Cum se poate reprezenta raționamentul din creierul uman formal pe un creier fizic?
- Cum se creează cunoștințele și din ce provin?
- Cum guvernează cunoștințele acțiunile?

Începuturile – Aristotel (384-322 î.e.n) silogismul : „orice om este muritor – Socrate este om -> socrate este muritor” – modus ponens, se generează concluziile mecanic. A dezvoltat primul set de reguli relativ la partea rațională a gândirii.

Rene Descartes (1596-1615) – fondatorul ideii de raționare ca sistem fizic și a puterii raționamentului. Este autorul teoriei *dualismului*:

- o parte a gândirii (spiritului) uman este deasupra naturii și a legilor fizice – gândirea umană are un caracter dual;
- gândirea animalelor nu are caracter dual – pot fi tratate ca mecanisme.

Opusul dualismului – *materialismul* – creierul uman funcționează fizic conform structurii sale.

O altă problemă – sursa cunoștințelor:

- *Empirism* – începe cu Francis Bacon (1561-1626): „nimic nu se înțelege dacă nu a fost în prealabil perceput”.
- Curentul inducționist David Hume (1711-1770): regulile generale se obțin prin asocieri repetate ale elementelor lor.

- Cercul de la Viena – Bernard Russell (1872-1970), Rudolf Carnap (1891-1970) – doctrina pozitivismului logic: cunoștințele se pot lega de teorii logice legate de experiențe. Cartea lui Carnap *Logical Structure of the World* (1928) – teoria raționamentului prin procese computaționale.

Legătura dintre cunoștințe și acțiune este vitală pentru AI deoarece inteligența nu înseamnă numai raționament ci și acțiune. Stă la baza unor domenii de o importanță deosebită în IE: de ex., teoria agenților inteligenți sau a deciziilor bazate pe mecanismele AI.

## Matematica

Probleme:

- Care sunt regulile formale care duc la concluzii corecte?
- Ce poate fi calculat?
- Cum putem raționa cu informații incerte sau incomplete?

Domeniile principale ale formalizării: logica, calculul, probabilitățile, teoria mulțimilor fuzzy etc.

*Legătura cu logica simbolică:*

- George Bool (1815-1864) – logica propozițiilor.
- Gottlob Frege (1848-1925) – include obiecte și relații -> FOPL.
- '20 – Whitehead, Russell, Tarski, Chuma – bazele aplicării logicii formale (FOPL) în reprezentarea lumii reale;
- '30- 40 – Alonso Church, Kurt Goedel, Emile Post, Alan Turing (părintele AI, 1912-1954). Mașinile Turing – (1936) – limba engleză poate fi descrisă cu un automat și un singur procesor de poate prelucra orice informație simbolică sau numerică.

*Legătura cu teoria algoritmilor:*

- David Hilbert (1862-1943): 23 de probleme fundamentale ale matematicii: ultima este decidabilitatea algoritmilor, prin care a vrut să stabilească limitele demonstrabilității.
- Kurt Godel (1906-1978) – există o metodă numerică pentru orice problemă decidabilă din FOPL, dar FOPL nu acoperă principiul inducției complete.
- Church-Turing – există funcții care nu pot fi reprezentate printr-un algoritm -> nu pot fi calculate; care sunt limitele calculabilității. De exemplu, nu se poate determina pe cale mecanică dacă un program va da un rezultat pentru un input dat sau va intra în ciclu infinit.



- Problema intractabilității: timpul de rezolvare a unei instanțe a problemei crește exponențial cu dimensiunea instanței – problema complexității algoritmilor & NP-completitudinii (Cook – 1971 & Karp – 1972).

*Legătura cu teoria probabilității:*

- Gerolamo Cardano (1501-1576) – probabilitățile legate de jocuri.
- Pierre Fermat (1601-1660), Blaise Pascal (1623-1790), James Bernoulli (1654-1705), Pierre Laplace (1749-1827), teoria probabilităților & statistică.
- Thomas Bayes (1702-1761) determinarea probabilităților unor evenimente viitoare pe baza unor cunoscute și a unor probabilități de trecere între evenimente -> baza raționamentelor în condiții de incertitudine în AI.

*Teoria informațiilor: Claude Shanon.*

Mulțimile Fuzzy și logica fuzzy a fost introdusă de Lotfy Zadeh, Universitatea Berkley în 1965, incertitudinea lingvistică – dezvoltată de școala japoneză, dar și de alții, prin includerea în aplicații de electronică casnică (mașini de spălat, televizoare, instalații de aer condiționat etc.). Aplicații diverse în rețele neuronale, în reprezentarea cunoștințelor și sisteme decizionale.

## Științe Economice

Probleme:

- Cum trebuie luate decizii pentru a obține un profit maxim?
- Cum procedăm dacă beneficiul se va obține după o perioadă lungă de timp?
- Cum procedăm dacă o afacere merge prost?

1776 Adam Smith – gândirea economică tratată științific – colaborarea unor agenți individuali pentru a-și asigura bunăstarea.

Economie, nu înseamnă numai bani – modul în care se ajunge la un anumit rezultat economic. Reprezentarea matematică a utilității (rezultatului preferat) Leon Warlas (1834-1910), Frank Ramsey (1931) – John von Neumann & Oskar Morgenstern în *The Theory of Games and Economic Behaviour*, 1944.

Teoria deciziilor: combinarea teoriei utilității cu teoria probabilităților și altele în vederea creării unui mediu decizional, nu numai economic. Se bazează pe teoria jocurilor (Neumann & Morgenstern), teoria deciziilor secvențiale de tip Markov (formalizate de Richard Bellman – 1957)

Simularea euristică: Herbert Simon (1916-2001) laureat al premiului Nobel în economie în 1978 – înlocuirea soluțiilor optime cu „soluții suficient de bune”.

- **Cibernetica** - Wiener, Odobeja (^40-^50) – studiul reglajului automat și a comunicării om-mașină; combină elemente de teoria informațiilor cu elemente de control bazate pe feed-back utilizate la sisteme de calcul.
- **Gramaticile formale** – 1900 extindere a logicii formale – gramatici comparative -> lingvistică, lingvistică computațională.

## **Ingineria calculatoarelor**

- Leonardo de Vinci (1452-1519) – proiecte, dar nu a realizat calculatorul mecanic.
- Wilhem Ashickard (1592-1635) în 1623 primul calculator mecanic, continuate cu Blaise Pascal (1623-1662) în 1642, Wilhelm Leibnitz (1646-1716).
- Holleritz (1912);
- Charles Babbage (1842);
- 1944 Mark I (Harvard 1944);
- ENIAC (University of Pensylvania 1947) și UNIVAC (Sperry-Land).

## **5.1.2 Cronologia AI**

### **5.1.2.1 Faza pre-incipientă (1943-1955)**

Primele cercetări AI de Waren McCulloch & Walter Pitts (1943) cele 3 surse:

- cunoștințe despre funcționarea neuronilor și a creierului;
- analiza formală a logicii propozițiilor al lui Rusell & Whitehead;
- teoria computațională a lui Turig.

Teoria rețelelor neuronale care pot fi instruite.

*Donald Hebb (1949)* – regulile de modificare a conexiunilor între neuroni -> regula de învățare a lui Hebb utilizată și azi.

*Marvin Minsky & Dean Edmonds, absolvenți de Princeton* – prima rețea neuronală (SNARC) în 1951 cu 40 de noduri. Teza a fost primită cu reținere dar von Neumann a spus „chiar dacă nu funcționează va fi operațional”. Minsky a elaborat ulterior o teoremă care a stopat pentru mult timp cercetările în rețele neuronale.

*Alan Turing – 1950*, „*Computing Machinery and Intelligence*” – a introdus o serie de concepte moderne: testul Turing, machine learning, algoritmi genetici și reinforcement learning.

*Jocuri* – caracter euristic cu strategii de rezolvare a problemelor.

- 1950 – jocurile de șah, go etc.
- 1952, 1955 Programele Claude Shanon la MIT de șah.
- Allen Newel – jocuri complexe (GO) – RAND Corporation (logica binară de ordinul I - Newel & Simon - 1972).
- 1955 – Warr – traducere automată cu dicționare mari – abordare simplistă.

### 5.1.2.2 Perioada inițială (1956)

*John McCarty* – Princeton după doctorat trece la Dartmouth College și convinge pe Shanon, Minsky și Rochester să organizeze în iunie 1956 – **IBM (Dartmouth College)** – Workshop de 2 luni cu 10 participanți). Problemele discutate cum ar fi: rețele neuronale, reprezentarea cunoștințelor, demonstrarea teoremelor, etc. au devenit domenii în AI.-> s-a introdus numele propus de McCarthy „**Artificial Intelligence**” – mai corect „raționalitate computațională” -> explozia AI, dominată peste 20 de ani de MIT, CMU, Stanford și IBM.

1956-1957 - programul Logic Theorist (demonstrarea automată a teoremelor) Newel, Shaw, Simon (Newel & Simon - 1972) de la Carnegie Mellon University, în limbajul IPL (Information Processing Language) predecesorul LISP, ALGOL. Simon: „am construit un program care poate rezolva probleme nenumerate”. Ei au dezvoltat limbajul IPL și au compilat manual programele. IPL a influențat apariția limbajului Fortran (1954) și teoria gramaticilor generative a lui Naom Chomsky (1955-1957) - > a influențat domenii ale AI ca de ex., lingvistica computațională și tratarea limbajului natural.

### 5.1.2.3 Epoca entuziastă (1956-1969)

1958 – *Rosenbloom* – teoria percepției – perceptroni, recunoașterea, învățarea liniară.

1958 – *John McCarthy* – LISP limbaj în care s-au dezvoltat multe aplicații ale AI.

1958 MIT – primul laborator specializat sub conducerea lui McCarthy mutat de la Dartmouth.

1958 GPS (*General Problem Solver dezvoltat de Newell, Shaw & Simon*) – simulează modul de gândire umană – printre probleme rezolvate a fost: misionar-canibal => ipoteza sistemelor fizice simbolice (Newell & Simon, 1976) „condiția necesară și suficientă pentru a rezolva o problemă care necesită inteligență este să se dispună de un sistem fizic uman sau mașină, care prelucrează simboluri”.

*Echipa lui Nathaniel Rochester IBM – mai multe programe de AI.*

- ⇒ Geometry Theorem Prover - Herbert Galetner (1959) probleme de geometrie sintetică de nivel mediu din liceu -> puzzle & jocuri, roboți, integrarea simbolică etc. 60-70.
- ⇒ Arthur Samuel (MIT) 1956-1965 - joc de șah care învață și ajunge la nivel de maestru, demonstrat la televizor inițial în februarie 1956.

1958 EPAM (*Elementary Perceiver and Monitorizer, Feigenbaum & Simon*) – în IPL primul program care învață -> Machine Learning.

1958 McCarthy – Advice Taker, descrie un program care poate rezolva orice problemă care poate fi descrisă prin FOPL, de exemplu, organizarea primirii automate a avioanelor pe un aeroport, reorientarea automată a unui curs, etc.

1963 McCarthy - trece la Stanford și dezvoltă pe baza Advice Taker, teoria roboților inteligenți.

Minsky dezvoltă la MIT programe cu studenți, formând domeniul microworlds, care rezolvă problemele de colegiu de anul 1, de teste IQ etc. (1961-1976), programe de tratare a limbajului natural etc.

1965 Robinson – program care rezolvă orice Problemă a Logicii inferențiale – reprezentabilă actual în Prolog.

#### 5.1.2.4 Epoca de maturitate

*Probleme de traducere automată* – finanțată de National Research Council, din rusă în americană: Proiect Sputnik (1957).

*Rezolvarea problemelor din microworld nu se putea aplica la probleme de dimensiuni mari* -> *algoritmii genetici* (Friedberg 1958, 1959) – ideea de bază: aplicând mici mutații asupra unui program se poate obține un program suficient de bun pentru anumite sarcini simple -> sute de ore de rulare fără succes.

*A doua problemă*: raportul Lighthill, 1973, critică cercetarea AI și Guvernul Britanic suspendă subvențiile.

*A treia problemă*: apar limitările sistemelor inteligente, de exemplu Minsky & Papert în cartea *Perceptrons* (1969) demonstrează că sistemele au o putere de reprezentare foarte limitată -> abandonarea rețelelor neuronale până în '80 multilayer.

S-au dezvoltat sistemele bazate pe cunoaștere:

- ⇒ 1965-1971- DEDRAL (Feigenbaum – elev al lui Simon, Buchanan – filosof devenit informatician, Lederberger, laureat al premiului Nobel în genetică - Stanford) – primul SE, structura moleculară (masa și compoziția moleculară);
- ⇒ Feigenbaum – HPP (Heuristic Programming Project) extinderea SE la alte activități umane;
- ⇒ Feigenbaum, Buchanan, Shortliffe -> MYCIN - 450 de reguli pentru infecția sangvină; actualmente peste 2000 de reguli; multe alte sisteme de diagnostică [P.Korduk 1979].

1968 – MACSYMA (MIT, Carl Engleman, William Martin, Sell Moses) – orice problemă de logică formală bazată pe axiome.

Explozia de după 1965.

Roberts 1963 – analiza vizuală a mediului – vedere artificială 1993 Nawal.

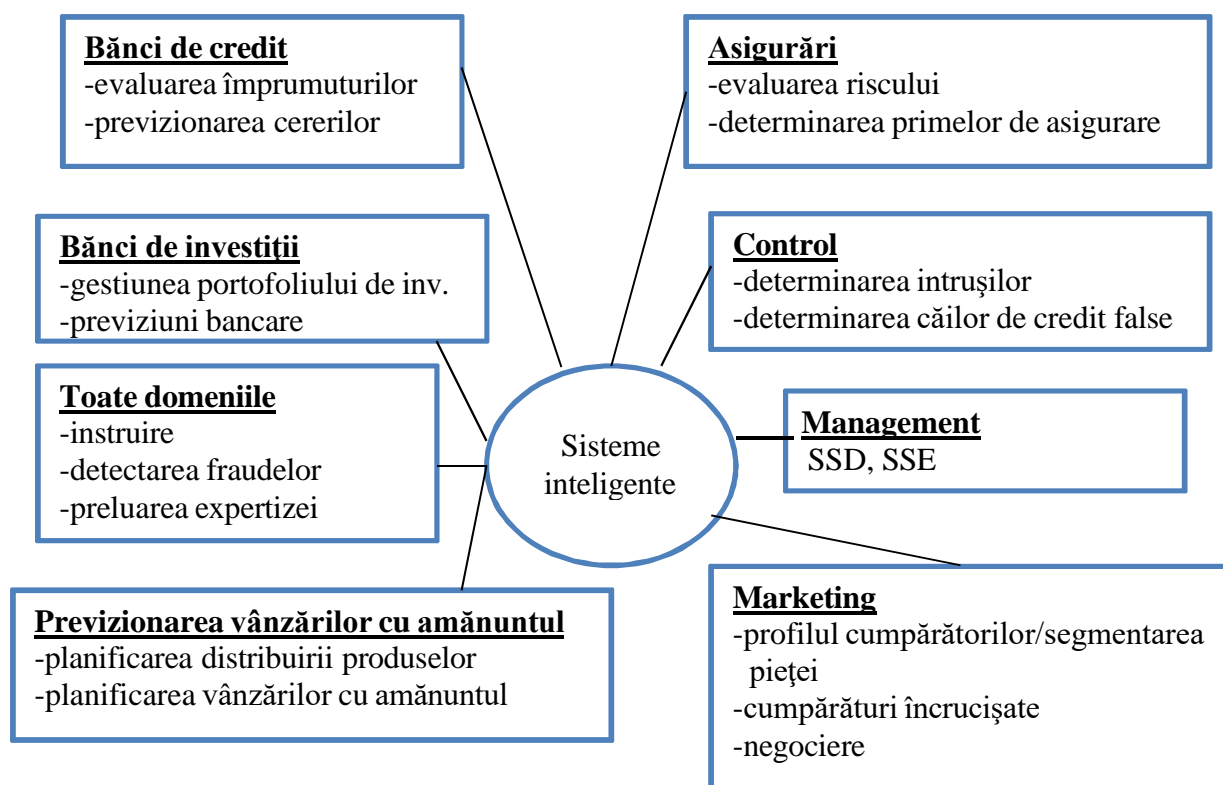
*Limbaaj natural* Winograd 1972 (*SHRDLU*) – recunoașterea limbajului natural, referențierea pronumelor, blocuri de cuvinte. Elevul său de la MIT, Eugen Charniak – pentru tratarea limbajului natural trebuie cunoștințe generale despre lume.

Yale – Roger Schank dezvoltă cu studenții o serie de programe de înțelegere a limbajului natural.

1973 LUNAR System (Woods) – NASA.

11 mai 1997, DEEP BLUE, IBM, bate pe Garry Kasparov 3,5 la 2,5 în 6 partide:  
algorithm performant, calculator de viteză mare, hard specializat

### 5.1.3 Legătura cu IE



#### 5.1.3.1

utilizarea SI de principalele bănci și instituții financiare;

#### 5.1.3.2

Country Wide Founding – SE adaptive;

#### 5.1.3.3

Fuji Bank – sisteme fuzzy – creșterea serviciilor, reducerea costurilor;

#### 5.1.3.4

American Express de la 15\$/tranzacție la 1,44;

#### 5.1.3.5

Visa Internațional – 6 luni 40.000.000 USD – rețea neuronală;

Avantaje:

#### 5.1.3.6

lucrează non-stop;

#### 5.1.3.7

lucrează repede;

#### 5.1.3.8

nu sunt influențabile ca și experții umani;

#### 5.1.3.9

lucrează în orice condiții.

Caracteristici care le recomandă în afaceri:

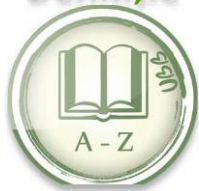
- 5.1.3.10      Capacitate de învățare – burse, bănci, asigurări – rețele neuronale, algoritmi genetici;
- 5.1.3.11      Flexibilitate, adaptabilitate – date incomplete și incorecte – rețele neuronale; imprecizie lingvistică – sistemele fuzzy;
- 5.1.3.12      Explicație – domeniul bancar – SE;
- 5.1.3.13      Descoperire de modele, Data Mining – rețele neuronale.
- 5.1.3.14      Extragerea de informații distribuite – agenți inteligenți, mobili, sisteme colaborative, multi-expert etc.

## 5.2 BAZELE LOGICE ALE A.I.

*„Erorile nedetectabile sunt infinite în varietate, spre deosebire de erorile detectabile care sunt limitate prin definiție.”*

### 5.2.1 Logica și logica simbolică

#### Definiție



[DEX03] **LOGICA** este știința demonstrației al cărui obiect este stabilirea condițiilor corectitudinii gândirii, a formelor și legilor generale ale raționării corecte, conforme prin ordinea ideilor cu organizarea logică a realității obiective”.

[Larouse1998] „Teorie științifică a raționamentului ce exclude procesele fiziologice”.

Dintre principalele aplicații ale logicii din domeniul IE:

- baze de date și de cunoștințe, a căror interogare se bazează pe logică.
- SE, SSD, SSE, dar și deciziile de zi cu zi ale managerului sau omului de afaceri se bazează pe raționamente logice.
- Practic nu există domeniu de afaceri în care logica să nu joace un rol major.

Logica are două ramuri esențiale:

- **logica clasică**, sau **logica aristoteliană**<sup>3</sup> - categoriile logice fundamentale: noțiunea, judecata sau raționamentul;
- **logica matematică, logica formală sau simbolică**<sup>4</sup>.

Logica simbolică fundamentul raționamentului uman<sup>5</sup> - avantaje:

- asigură expresivitatea și rigurozitatea în reprezentarea cunoștințelor;
- asigură deducerea unor cunoștințe noi pe baze altora deja existente.

Utilizează simboluri - reprezentarea obiectelor și a operațiilor executate asupra simbolurilor.

Pentru informatică și pentru IE în special, logica formală prezintă un interes aparte. Dicționarele de informatică se mărginesc în general numai la acestea:



- [Colin90]: "**logica** = substantiv, știința care se ocupă cu gândirea și raționamentele; **logica formală** = tratarea formei și structurii, ignorând conținutul";  
[Oxford91] "Logica este un formalism de reprezentare a cunoștințelor și a raționamentelor, dezvoltat inițial de către matematicieni pentru a formaliza raționamentele matematice. În logica matematică, investigația cuprinde metode matematice împrumutate din algebră și teoria algoritmilor. Sistemele cele mai uzuale sunt calcul propozițiilor și cel al predicatelor."
- [Boden87] Calculatorul și deci, informatica prelucrează simboluri. Din această cauză logica formală, în tratatele de informatică [Patterson90], apare în general sub denumirea de *logică simbolică*.

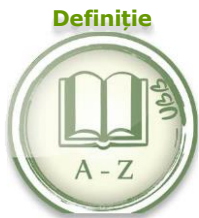
---

<sup>3</sup> Aristotel (384-322 î.e.n) a fost cel care a definit pentru prima dată precis o serie de reguli care guvernează partea rațională a gândirii. El a dezvoltat un sistem informal al silogismelor proprii raționamentelor, în baza cărora pornind de la premise se pot genera mecanic concluzii [Russell&Norvig03]

<sup>4</sup>Exprimarea ideilor logicii formale așa cum a fost ea definită de grecii antici sub o formă matematică a început cu lucrările lui George Bool (1815-1864), care a dezvoltat în 1847 logica propozițiilor. Gottlob Frege (1848-1925) a dezvoltat teoria lui Bool incluzând obiecte și relații creând astfel logica predicatelor de ordinul I utilizată astăzi în reprezentarea cunoștințelor. Alfred Tarski (1902-1983) a introdus o teorie referențială prin care leagă obiectele logicii de cele din lumea reală [Russell&Norvig03, Luger02].

5 [Malița&Malița87]

### 5.2.2 Logica propozițiilor (PL)



Conform [DEX03], noțiunea de **propoziție** are atât semnificația de CEA MAI MICĂ UNITATE SEMANTICĂ care exprimă o idee, o judecată etc., utilizată în gramatică, cât și cea de enunț a cărui valoare de adevăr este întemeiată pe bază de reguli explicit exprimate, utilizată în logica simbolică.

Exemplu: simbolul **p** - atașat propoziției “Grivei latră” ia valoarea **adevărat** (True - T) dacă într-adevăr latră, respectiv **fals** (False-F) dacă nu latră.

**Diferența dintre gramatică și logica simbolică.** În PL, valoarea de adevăr a propoziției este calitatea acesteia de a fi adevărată sau falsă în întregul ei și nu interesează obiectele constitutive ale sale. Exemplu: propoziția „Ionescu este managerul societății comerciale” sau „Ionescu manager”.

Teoria logică este în esență un limbaj de reprezentare a cunoștințelor. Ca orice limbaj, are două aspecte esențiale:

- aspectul *semantic* sau *abordarea semantică*;
- aspectul *sintactic* sau *abordarea sintactică*.

Alți autori [Nilson98] consideră logica sub 3 aspecte:

- *limbajul* (cu sintaxă care specifică expresiile corecte în acest limbaj);

---

<sup>6</sup> Bibliografie foarte vastă în domeniu chiar și în limba română [Florea&Boangiu94, Malița&Malița87, Mihăiescu, 66] sau [D.vanDalen84, Gray85, Graham88, Nilson98, Turner84]

- *regulile inferențiale*, prin care se manevrează propozițiile limbajului;
- *semantica* pentru asocierea elementelor limbajului de semnificația lor.

**Semantica** - aspectele intime (interne) ale universului problemei - obiectivele:



- notarea propozițiilor atașate universului problemei cu ajutorul unor simboluri și fixarea valorii de adevăr a acestor simboluri;
- stabilirea simbolurilor care joacă rolul de conectori, adică leagă simbolurile atașate propozițiilor;
  - stabilirea valorii de adevăr a noilor propoziții astfel obținute (compuse).

Conceptul central în abordarea semantică este cel de „*valoare de adevăr*”: „Este o formulă, o tautologie, adică este ea adevărată indiferent de faptul că are părțile componente adevărate sau false?”

Within the study of logic, a tautology is a statement that is true by its own definition.  
[<http://en.wikipedia.org/wiki/Tautology>]



Spre deosebire de abordarea semantică, cea *sintactică* are ca și concept central “*demonstrația logică*” și anume, trebuie să răspundă la întrebarea: “Este o formulă demonstrabilă în cadrul unui sistem logic, sau nu?”.

Din acest motiv, de regulă, semantica este asemănată cu studiul expresiilor din algebră, unde se demonstrează corectitudinea formulelor, în timp ce sintaxa se aseamănă cu rezolvarea sistemelor de ecuații prin metoda substituției.

### **5.2.2.1 Abordarea semantică**

În cadrul abordării semantice trebuie fixate **5 elemente de bază** [Patterson90]:

- limbajul de descriere a formulelor logice (alfabetul limbajului);
- valoarea de adevăr a simbolurilor atașate propozițiilor (respectiv predicatelor);
- funcțiile de evaluare;
- mecanismul de raționament reprezentat de consecințele logice;
- principiile teoriei logice.

PL- propoziții simple - tratate atomic, ca un tot unitar - se vor nota cu câte un simbol - **litere mari sau mici de la mijlocul alfabetului P,Q,R,...**, respectiv **p,q,r,...**  
*Propozițiile simple - nici o parte a lor nu este o propoziție.*

PL clasic - orice propoziție poate fi T sau F, dar nu amândouă deodată (**legea terțului exclus**) - **logică bivalentă**.

*Teoremele* în logică, sisteme în care se pleacă cu valoarea de adevăr a unor propoziții, numite **ipoteze (premise)**, și aplicând o serie de reguli de raționare (reguli inferențiale), operații și funcții logice, se ajunge la alte propoziții, numite **concluzii**.

### A. Limbajul de descriere a formulelor (sistemul notațional)



În PL - alfabetul este format din simboluri propoziționale definite astfel:

- litere mici, p,q,r,...sau mari P,Q,R,... atașate propozițiilor;
- conectori logici:
  - $\sim$  sau  $\neg$  - negația
  - $\wedge$  - conjuncția
  - $\vee$  - disjuncția
  - $\rightarrow$  - implicația
  - $\leftrightarrow$  - echivalența
- alte simboluri, cum ar fi de exemplu: (, ).

PL - *formulele corecte, corect formulate sau bine formulate*, notate în literatură cu *wff* – *well formatted formulas* - definesc recursiv:

- un atom (simbol atașat unei propoziții simple) este un *wff*; un *wff* în paranteză;
- dacă P este un *wff*, negatul său  $\sim P$  este un *wff*;
- dacă P și Q sunt *wff*, atunci  $P \wedge Q$ ,  $P \vee Q$ ,  $P \rightarrow Q$  și  $P \leftrightarrow Q$  au aceeași proprietate;
- mulțimea *wff*-urilor este generată de regulile (i)-(iii).

$(P \rightarrow (Q \wedge \sim R)); (P \rightarrow)$ .

*Propozițiile compuse* sunt *wff*-uri care se realizează pe baza unor conectori sau operatori logici. Principalii operatori logici, așa după cum s-a prezentat mai sus, sunt: **negația, conjuncția și disjuncția**.

**Negația** – unei propoziții, “**non P**” - [French91]: "not.P",  $\neg P$ ,  $\bar{P}$ , sau  $\sim P$ . Operațiile logice - cu ajutorul *tablelor de adevăr* (corespunzătoare tabelor operațiilor aritmetice elementare). Aceste table indică valoarea de adevăr a rezultatului în funcție de valorile de adevăr ale componentelor.

P	$\sim P$
F	T
T	F

Exemplu: P = “Bugetul trebuie aprobat anual”, care are valoarea T, negația  $\sim P$ , “Bugetul nu trebuie aprobat anual”, are valoarea F.

**Conjunția** – propozițiilor P și Q - “P și Q” - se notează cu  $P \wedge Q$ . Notății [French91]: P.Q, P .and. Q sau P&Q. Exemplu: P = “Pământul este rotund” și Q= “impozitul este o datorie față de stat”. P&Q va fi “Pământul este rotund” și „impozitul este o datorie față de stat”.

**Disjuncția** – “P sau Q” și se notează cu “ $P \vee Q$ ”. Notății și cu: P+Q, P.or.Q sau  $P \cup Q$ .  $P \vee Q = T$  dacă  $P=T$ , sau  $Q=T$ , sau ambele propoziții sunt T. Sau exclusiv “sau Cezar sau nimic”. “sau” logic trebuie interpretat în sensul limbajului curent astfel: dacă P este propoziția “Firma X practică comerț en-gros” și Q este propoziția “Firma X practică comerț en-detail”, propoziția “P sau Q” - “**sau exclusiv**” și se notează cu **xor**.

## B. Valoarea de adevăr



O teorie logică - cel puțin două valori de adevăr: true (T) și false (F) - exact două valori, avem de a face cu *logica bivalentă*. Aceasta are diferite forme, cum ar fi: logica propozițiilor, cea a predicatelor, modală sau temporală<sup>7</sup>.

Klee, Bochvar și Lukasiewicz - *logica trivalentă* “adevărat” și “fals”, poate exista și o a treia valoare, “nu știu” sau “necunoscut” sau „nedeterminat în momentul de față dar urmează să fie determinat în viitor”, „imposibil”, „absurd”. etc . Ulterior - alte logici trivalente sau cu 4 valori de adevăr, logici denumite *neclasice*, cum ar fi: logicile lui Lukasiewicz, Post, Klee sau Bochvar. -> [Turner85], -> logicile fuzzy în

<sup>7</sup> [Turner85, <http://plato.stanford.edu/entries>]

care pot exista o infinitate de valori de adevăr, logici bazate pe teoria introdusă de Lotfi Zadeh [Zadeh65].

### C. Funcția de evaluare



**Funcție de evaluare - valorizare - de adevăr** a unui atom, o funcție care atașează atomului respectiv o valoare T sau F în logica bivalentă.  $\rightarrow$  funcția de evaluare a unei propoziții atașează o valoare de adevăr propoziției în funcție de valorile de adevăr ale componentelor, pe baza regulilor de evaluare a conectorilor.

Regulile de evaluare a conectorilor sunt date prin tablele de adevăr. În general, valoarea de adevăr a unei propoziții  $p$  se notează cu  $p^V$ .

Interpretare a unei formule - atribuirea unei valori de adevăr fiecărei componente a formulei respective. Exemplu,  $p \vee q$  - 4 interpretări posibile, care formează domeniul de interpretare:

$$I_1 = \{T, T\}, \quad I_2 = \{T, F\}, \quad I_3 = \{F, T\}, \quad I_4 = \{F, F\}$$

Considerând, de exemplu, interpretarea  $I_2$ , valorile componentelor sunt:

$$p^V = T \text{ și } q^V = F$$

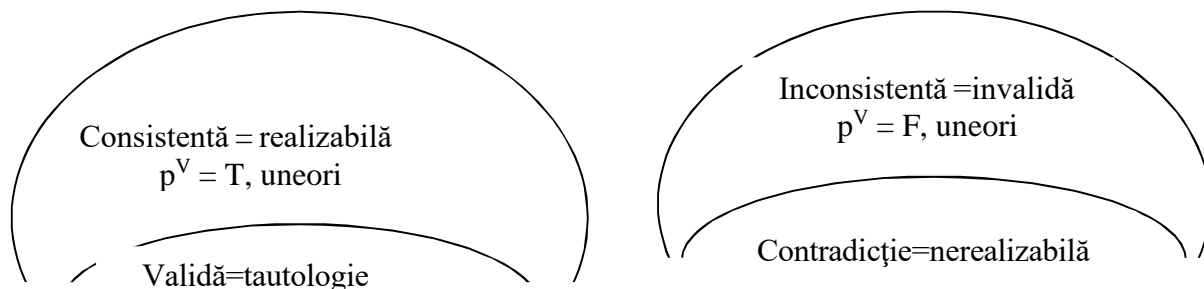
iar valoarea obținută prin funcția de evaluare este:

$$(p \vee q)^V = T$$

Pe baza domeniilor de interpretare și a funcțiilor de evaluare, *wff* – urile se clasifică în:

- *tautologii* sau *formule valide* – T indiferent de interpretare;
- *consistente* – care iau valoarea T pentru unele interpretări;
- *inconsistente (invalide)* – F pentru unele interpretări;
- *contradicții* - F pentru orice interpretare.

Relația dintre aceste tipuri de *wff* - schemele Wyne:



wff-uri - echivalente dacă au aceeași valoare de adevăr pentru orice interpretare.

*modelul unui wff* - o interpretare pentru care formula ia valoarea T (în exemplul nostru  $I_2$ ). Analog, se poate defini *modelul unui ansamblu de formule* ca fiind o interpretare pentru care toate formulele sunt adevărate.

#### D. Principiile teoriei logice



Orice teorie logică trebuie să respecte anumite principii fundamentale. Respectarea sau dimpotrivă, eliminarea unor restricții, stabilește tipul logicii.

##### a. Principiul fixării numărului de valori logice

Teoriile logice - un număr fixat,  $n$  de valori logice distincte cu  $2 \leq n \leq \infty$ .  $|V|$  cardinalul mulțimii  $V$  - de cele mai multe ori acest cardinal are o valoare finită. Există însă și logici infinite.

Logică este cea bivalentă, în care

$$V = \{T, F\} \text{ și deci } |V|=2$$

În general se consideră 3 conectori de bază și anume:  $\sim, \vee, \wedge$  cu tablele de adevăr:

P	$\sim P$
T	F
F	T

$P \wedge Q$	T	F
T	T	F
F	F	F

$P \vee Q$	T	F
T	T	T
F	T	F

*Funcție logică* - oricărei combinații de valori logice să-i corespundă o valoare logică. Funcțiile logice binare, adică acelea care au 2 argumente. Aceste funcții fac ca la combinațiile FF, FT, TF și TT să le corespundă valorile F sau T. Cum în domeniul de definiție există patru combinații posibile, iar în cel al valorilor, două  $\rightarrow$  [Gray85] există numai  $2^4=16$  funcții logice binare distincte. Se poate demonstra, de asemenea, că orice funcție binară se poate reprezenta cu ajutorul celor 3 operatori definiți anterior.

Dintre funcțiile logice, cele mai importante sunt considerate:

- **Implicația - inferența logică** - operația principală în domeniul bazelor de cunoștințe; notată cu  $\rightarrow$  și are semnificația **“dacă P, atunci Q”**. Propoziția  $P \rightarrow Q$  ia valoarea fals numai dacă din P „adevărat” rezultă Q „fals”. Avem, deci, următoarele situații posibile:
  - dacă P este adevărată, atunci și Q trebuie să fie adevărată;
  - dacă P este falsă, Q poate fi adevărată sau falsă;
- **Echivalența logică – echivalența** - notată cu  $\leftrightarrow$  - reprezentată de dubla implicație și deci, generează o **propoziție compusă care este adevărată dacă cele două propoziții P și Q sunt concomitent adevărate sau false**.

Pentru a reprezenta cele două funcții binare, vom recurge la tabla de adevăr:

P	Q	$P \rightarrow Q$	$Q \rightarrow P$	$\sim P$	$\sim P \vee Q$	$P \leftrightarrow Q$	$P \rightarrow Q \wedge Q \rightarrow P$
T	T	T	T	F	T	T	T
T	F	F	T	F	F	F	F
F	T	T	F	T	T	F	F
F	F	T	T	T	T	T	T

Din tabel se pot deduce o serie de relații utile, dintre care amintim:

$$P \rightarrow Q = \sim P \vee Q$$

$$P \rightarrow Q = \sim P \vee Q = \sim(\sim Q) \vee \sim P = (\sim Q) \rightarrow (\sim P) \quad (1)$$

$$P \leftrightarrow Q = P \rightarrow Q \wedge Q \rightarrow P$$

**Observație.** În logica formală relațiile de mai sus sunt deosebit de importante. (1) joacă un rol deosebit în raționamentele matematice. De exemplu, o funcție f este injectivă dacă:

$$\text{pentru orice } x \neq y \text{ are loc } f(x) \neq f(y)$$

$$P \rightarrow Q$$



Verificarea injectivității se realizează cu implicația echivalentă:

$$\sim Q \rightarrow \sim P, \text{ adică } f(x) = f(y) \rightarrow x = y.$$

**Observație.** Formula (1) trebuie tratată cu grijă - (1) nu asigură egalitatea  $P \rightarrow Q = Q \rightarrow P$ . Pentru ilustrare să considerăm următorul exemplu:

„avem o creștere a veniturilor”  $\rightarrow$  „avem o creștere a vânzărilor”  
aceasta este echivalentă cu „nu avem o creștere a vânzărilor”  $\rightarrow$  „nu avem o creștere a veniturilor” dar aceasta nu este echivalentă cu „creșterea vânzărilor”  $\rightarrow$  „creșterea veniturilor”, deoarece vânzările pot crește și din alte motive, cum ar fi, solduri, perioadă de sărbători, acțiuni promoționale, etc.

Algebra booleană nu este singurul sistem logic bivalent - sistemul logic al lui Hilbert-Ackermann are la bază 3 conectori  $\sim$ ,  $\vee$  și  $\rightarrow$ .

**Observație.** Pornind de la faptul că au loc relațiile:

$$P \wedge Q = \sim (P \rightarrow \sim Q)$$

$$P \vee Q = (\sim P) \rightarrow Q$$

Lukasiewicz a definit un sistem logic în care operațiile de bază sunt negația și implicația – sistemul logic Lukasiewicz.

În cazul logicii trivalente, deci în cazul  $|V|=3$ , avem mai multe sisteme.

Astfel, avem sistemul lui Kleene cu  $V=\{T,F,U\}$ , unde U este „necunoscut”, având 2 conectori de bază,  $\sim$  și  $\wedge$ , definiți astfel:

p	$\sim p$	$p \wedge q$	T	F	U
T	F	T	T	F	U
F	T	F	F	F	F
U	U	U	U	F	U

De unde se poate deduce:

$p \vee q$	T	F	U	$p \rightarrow q$	T	F	U	$p \leftrightarrow q$	T	F	U
T	T	T	T	T	T	F	U	T	T	F	U
F	T	F	U	F	T	T	T	F	F	T	U
U	T	U	U	U	T	U	U	U	U	U	U

Sistemul Lukasiewicz, are  $V = \{T, F, I\}$ , unde  $I$  indică „un eveniment din viitor nerealizabil în prezent, dar nu un necunoscut”. În acest caz conectorii sunt  $\sim$  și  $\rightarrow$ , cu tablele de adevăr:

p	$\sim p$	$p \rightarrow q$	T	F	I
T	F	T	T	F	I
F	T	T	T	T	T
I	I	T	T	I	T

de unde se deduce:

$p \vee q$	T	F	I	$p \wedge q$	T	F	I	$p \leftrightarrow q$	T	F	I
T	T	T	T	T	T	F	I	T	T	F	I
F	T	F	I	F	F	F	I	F	F	T	I
I	T	I	I	I	I	I	I	I	I	I	T

Sistemul Bochvar, creat pentru explicarea unor paradoxuri semantice, are  $V = \{T, F, M\}$ , cu  $M$  indicând „absurd” și cu conectorii  $\sim$  și  $\rightarrow$ , având tablele de adevăr:

p	$\sim p$	$p \rightarrow q$	T	F	M
T	F	T	T	F	M
F	T	T	T	T	M
M	M	M	M	M	T

de unde se deduce:

$p \vee q$	T	F	M	$p \wedge q$	T	F	M	$p \leftrightarrow q$	T	F	M
T	T	T	M	T	T	F	M	T	T	F	M
F	T	F	M	F	F	F	M	F	F	T	M
M	M	M	M	M	M	M	M	M	M	M	M

Logicile infinite, cele mai cunoscute, - logicile fuzzy, introduse de Lotfy Zadeh, pentru a reprezenta cunoștințele descrise cu ajutorul incertitudinii lingvistice.  $V$  este o submulțime mărginită a lui  $R_+$ . Această mulțime - normalizată luând în locul lui  $V$ , mulțimea  $V/|V|$ , înlocuind fiecare element  $v$  din  $V$  cu  $v/|V|$ .  $\rightarrow V=[0,1]$  în toate cazurile. De exemplu, considerând o firmă și notând cu  $x$  beneficiul obținut de firmă, putem defini măsurile fuzzy de exemplu astfel:

- dacă beneficiul este de  $x=1.000.000$ , Beneficiu  $(x)=0$
- dacă beneficiul este de  $x=50.000.000$ , Beneficiu  $(x)=0.2$
- dacă beneficiul este de  $x=1.000.000.000$ , Beneficiu  $(x)=0.6 \dots$

## b. Principiul consistenței și non-contradicției

Pentru un domeniu de interpretare dat, o propoziție - o valoare unică din  $V \rightarrow$  principiul valorizării *non-contradictorii* și *consistente*. Cu alte cuvinte, fie  $D$  mulțimea propozițiilor, considerând  $V=\{T,F\}$  și

$$D_T=\{p \in D \mid p^V=T\}, \text{ respectiv } D_F=\{p \in D \mid p^V=F\}; \text{ se obține } D_T \cap D_F=\Phi$$

Logicile polivalente, cu  $V=\{v_1, v_2, \dots, v_n\}$ ,  $D_i=\{p \in D \mid p^V=v_i\}$ , condiția de consistență se poate scrie:

$$D_i \cap D_j = \Phi, \text{ pentru } i \neq j.$$

Logicile care nu sunt consistente se numesc *paraconsistente* sau *logici suprasaturate*. În aceste logici propozițiile pot primi deodată două sau mai multe valori de adevăr, deoarece în general

$$D_i \cap D_j \neq \Phi, \text{ pentru } i \neq j.$$

## c. Principiul excluderii celei de a n+1-a valori

Unui atom  $i$  se poate asocia o valoare din mulțimea  $V$ , pe domeniul de interpretare  $\rightarrow$  funcția de evaluare este total definită pe domeniul de interpretare.

Principiul excluderii celei de a n+1-a valori - în logica bivalentă, legea terțului exclus:

$$D_T \cup D_F = D$$

$\rightarrow$  orice propoziție este ori adevărată, ori falsă, nu poate lua o a treia valoare.

În general, legea excluderii celei de a (n+1)-a valori logice are forma:

$$\bigcup_{i=1}^n D_i = D, \text{ pentru } V = \{v_1, v_2, \dots, v_n\}.$$

Pornind de la două valori logice, admiterea sau respingerea principiilor b. sau c generează diferite logici. Astfel,

- admiterea ambelor, duce la *logica clasică*;

- admiterea consistenței b și respingerea c duce la *logici lacunare* de tipul logicilor trivalente a lui Klee, Lukasiewicz, Bochvar sau la *logici intuitiviste*, unde se admite și a 3-a valoare;
- respingerea lui b și admiterea lui c duce la *logici paraconsistente*;
- respingerea ambelor principii, duce la *logici cu semantici aproximative*, numite și *logici relevante*, unde o propoziție poate admite mai multe valori sau rămâne neevaluată.

#### d. Principiul constanței de valorizare a unei propoziții elementare

Valoarea de adevăr a unei propoziții elementare date trebuie să rămână aceeași la toate aparițiile acesteia atâta timp cât se consideră o anumită interpretare. Renunțarea la acest principiu duce la *logici paraconsistente*.

#### E. Mecanismul raționamentului



Realizarea raționamentului într-un sistem logic → obținerea unor formule noi pe baza celor existente → o extindere consistentă a cunoștințelor despre universul problemei.

*O formulă Q - consecință logică a unei formule P, dacă Q ia valoarea T pentru toate interpretările pentru care P ia valoarea T. În general, Q - consecință logică a unei mulțimi de formule  $P_1, P_2, \dots, P_n$ , dacă Q primește valoarea T pentru orice interpretare pentru care  $P_1, P_2, \dots, P_n$  iau valoarea T. Consecința logică se notează în general cu simbolul implicației „ $\rightarrow$ ”.*

**(P1).** Q este consecința logică a lui  $P_1, P_2, \dots, P_n$  este echivalent cu faptul că  $P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow Q$  este validă.

Pe baza proprietăților implicației, propoziția de mai sus mai poate fi enunțată și astfel:

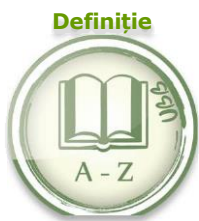
**(P2).** Q este consecința logică a lui  $P_1, P_2, \dots, P_n$  este echivalent cu faptul că  $P_1 \wedge P_2 \wedge \dots \wedge P_n \vee \sim Q$

este inconsistentă.

→ (P1), (P2) - constituie baza raționamentului logic și a demonstrării corectitudinii, deoarece reduce problema consecințelor logice la cea a demonstrării validității sau inconsistenței unor formule.

### 5.2.2.2 Abordarea sintactică a teoriei logice

**A. Sistemul formal** - concept de bază a abordării sintactice a teoriei limbajelor.



*Sistemul formal este un instrument de analiză, prelucrare și generare a simbolurilor de bază (semnelor limbajului) și care duce la structuri de simboluri, respectiv șiruri de simboluri (cuvinte, propoziții, fraze) acceptate în limbaj.*

Din cele de mai sus rezultă că un sistem formal se definește ca un cvadruplu:

$$S = \{ A, F, Y, \mathfrak{R} \}$$

unde:

*A* - alfabetul sistemului (mulțimea simbolurilor de bază)

*F* - mulțimea formulelor corecte (wff-urilor);  $F \subseteq A^*$ , adică *F* este o submulțime a șirurilor de caractere sau a șirurilor de simboluri din alfabetul *A*;

*Y* este mulțimea axiomelor, adică  $Y \subseteq A^*$ ; despre *Y* se poate demonstra că este decidabilă, adică pentru orice element *x* al lui  $A^*$  se poate demonstra că face parte sau nu din mulțimea *Y* a axiomelor;

$\mathfrak{R}$  - mulțimea regulilor de deducție sau inferențiale. O regulă de deducție este o relație de aritate  $n+1$  în mulțimea wff-urilor, - orice  $R \in \mathfrak{R}$  este o submulțime a produsului cartezian de ordinul  $n+1$  a lui *F*, adică  $R \subseteq F^{n+1}$ , astfel ca, pentru orice  $Y = \langle y_1, y_2, \dots, y_n \rangle$  corespunde prin *R* un  $x \in F$ , cu  $y_i, i=1,2,\dots,n$ . wff-urile  $y_i, i=1,2,\dots,n$  se numesc antecedente, iar *x* se numește consecința lui *R*.

$\Gamma = \{y_1, y_2, \dots, y_n\}$ , mulțimea premizelor. Se notează cu  $E_0 = \Gamma \cup Y$  reuniunea dintre mulțimea premizelor și cea a axiomelor. Se consideră:

$$E_1 = E_0 \cup \{x \mid \exists Y \in E_0, \text{ astfel încât } R: Y \rightarrow x, \text{ cu } R \in \mathfrak{R}\}$$

adică  $E_1$  este format din  $E_0$  și imaginile elementelor din  $E_0$  prin reguli de deducție din  $\mathfrak{R}$ . Având construit  $E_1$ , putem construi -  $E_2, E_3, \dots$

Dacă în mulțimea  $E_0, \Gamma = \Phi$ , adică  $E_0$  este format numai din axiome, atunci elementele lui  $E_i, i=1,2,\dots$ , - teoreme ale sistemului formal → un rezultat cunoscut din

matematică și anume, *teoremele unui sistem formal* sunt acele formule care pot fi demonstrate utilizând numai axiome.

Fie  $x \in E_i$  o teoremă - printr-o secvență de deducții  $D = \{ E_0, E_1, \dots, E_{i-1} \}$  - secvență de deducții  $D$  formează o *demonstrație a teoremei*.

*Un sistem  $S$  de formule este decidibil, dacă există o procedură efectivă prin care se poate decide dacă o formulă din  $S$  este sau nu o teoremă.*

Decidabilitatea - esențială în teoria sistemelor formale - baza teoriei demonstrabilității teoremelor, deci a extinderii cunoștințelor în cadrul sistemelor logice. Se poate demonstra:

- (PL) este decidabilă
- (FOPL) nu este.

## B. Algebra Booleană

**Algebra booleană** - cel mai important sistem logic bivalent George Boole (1815-1864).

### Semantica -Alfabetul și cuvintele (wff)

1. **A** - simbolurile propoziționale notate cu  $p, q, r, \dots$  sau  $P, Q, R, \dots$ , constante -  $A, B, \dots$  precum și pe cei trei conectori:  $\sim, \wedge$  și  $\vee$ .

2. **F** – wff

3. **Y** - **Axiomele de bază ale algebrei booleene sunt:**

$Y_1$  – comutativitatea:

$$P \wedge Q = Q \wedge P, \text{ respectiv } P \vee Q = Q \vee P$$

$Y_2$  – asociativitatea:

$$P \wedge (Q \wedge R) = (P \wedge Q) \wedge R, \text{ respectiv } P \vee (Q \vee R) = (P \vee Q) \vee R$$

$Y_3$  – proprietatea lui  $\wedge$  și  $\vee$ , adică:

$$P \wedge T = P \text{ și } P \wedge F = F, \text{ respectiv } P \vee T = T \text{ și } P \vee F = P$$

$Y_4$  – proprietatea negației, adică:  $P \wedge \sim P = F$  respectiv  $P \vee \sim P = T$

$Y_5$  – distributivitatea lui  $\wedge$  față de  $\vee$  și a lui  $\vee$  față de  $\wedge$ , adică:

$$P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R),$$

$$\text{respectiv } P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R).$$

Axiomele - 5 legi de bază ale algebrei booleene.

Din cele de mai sus se pot deduce o serie de alte legi. Dintre legile cele mai importante deductibile amintim:

- legea dublei negații sau a complementului  $\sim\sim P=P$ ;
- idempotența:  $P \wedge P=P$ , respectiv  $P \vee P=P$ ;
- legea absorbției,  $P \wedge (P \vee Q)=P$ .

Pentru ilustrarea celor de mai sus vom demonstra legea absorbției [Gray85];

$$\begin{aligned}
 P \wedge (P \vee Q) &= (P \vee F) \wedge (P \vee Q) && \{\text{proprietatea operatorului } \vee\} \\
 &= P \vee (F \wedge Q) && \{\text{distributivitatea lui } \vee \text{ față de } \wedge\} \\
 &= P \vee (Q \wedge F) && \{\text{comutativitatea}\} \\
 &= P \vee F && \{\text{proprietatea operatorului } \wedge\} \\
 &= P && \{\text{proprietatea operatorului } \vee\}
 \end{aligned}$$

Celelalte legi se demonstrează analog.

Legile booleene - caracter dual: *în orice teoremă din algebra booleană dacă se înlocuiește  $\vee$  cu  $\wedge$  și invers, teorema rămâne adevărată.*

Legile deductibile: **legile lui DeMorgan:**

$$\sim(P \wedge Q) = \sim P \vee \sim Q, \text{ respective dualul, } \sim(P \vee Q) = \sim P \wedge \sim Q$$

Prin inducție completă  $\rightarrow$  generalizarea legilor lui DeMorgan la  $n$  propoziții:

$$\begin{aligned}
 \sim(P_1 \wedge P_2 \wedge \dots \wedge P_n) &= \sim P_1 \vee \sim P_2 \vee \dots \vee \sim P_n, \text{ respectiv} \\
 \sim(P_1 \vee P_2 \vee \dots \vee P_n) &= \sim P_1 \wedge \sim P_2 \wedge \dots \wedge \sim P_n
 \end{aligned}$$

$\rightarrow$  ”**Pentru a nega o formulă, schimbă semnul și operatorul cu complementarul**”.

De exemplu:

$$\sim(P \vee \sim Q) = \sim P \wedge \sim(\sim Q) = \sim P \wedge Q.$$

**Observație.** Proprietățile lui  $\vee$  se aseamănă cu  $+$ , iar ale lui  $\wedge$  cu ale operatorului de înmulțire  $*$ . Proprietatea care le diferențiază este însă distributivitatea; distributivității lui  $\wedge$  față de  $\vee$  îi corespunde distributivitatea înmulțirii față de adunare  $A * (B + C) = A * B + A * C$ , invers nu este adevărat; adunarea nu este distributivă față de înmulțire, adică

$$(A * B) + C \neq (A + C) * (B + C)$$

în timp ce distributivitatea disjuncției  $\vee$  față de conjuncție  $\wedge$  are loc.

$\rightarrow$  5 legi + legile lui DeMorgan, permit aducerea oricărei expresii din algebra booleană la una din următoarele forme:

- **forma normală conjunctivă** - forma  $(C_1 \wedge C_2 \wedge C_3 \wedge \dots)$  unde  $C_i$  se numește clauză; fiecare clauză - din propoziții simple sau disjuncții de propoziții și eventual negații ale acestora. Forma normală conjunctivă - teoria demonstrației, deoarece propoziția scrisă sub formă clauzală este adevărată dacă și numai dacă toate propozițiile componente sunt adevărate și invers.
- **forma normală disjunctivă** – expresia sub forma unor disjuncții de expresii, expresiile din propoziții simple sau conjuncții de propoziții și eventual negații ale acestora. Această formă este utilă în teoria circuitelor.

### Observații.

- Orice propoziție logică poate fi adusă la una dintre formele normale, în următorii pași:
  - se reduc funcțiile logice din expresie la cei trei conectori de bază, la implicații și echivalențe;
  - se înlocuiesc echivalențele cu implicații duble;
  - se înlocuiesc implicațiile cu forma lor disjunctivă;
  - se aplică legile lui De Morgan și cele cinci legi fundamentale ale algebrei booleene.
- Orice formulă se poate demonstra fie cu ajutorul tablei de adevăr, fie utilizând cele cinci legi fundamentale ale algebrei booleene. Utilizarea tablei de adevăr este mai sigură dar mai lungă, în timp ce utilizarea celor cinci legi are un caracter euristic mai pronunțat.

### Regulile inferențiale în logica propozițiilor



Implicația - în **deducțiile logice** - din propoziții adevărate  $\rightarrow$  propoziții adevărate. Acest mod de utilizare a mecanismului logicii diferă de cel al algebrei booleene, deoarece aceasta, ca orice algebră, permite determinarea valorii de adevăr a unor formule pe baza valorii de adevăr a componentelor, dar nu și manipularea acestora indiferent de valoarea de adevăr a componentelor.

În deducția logică - două **reguli inferențiale** clasice, cunoscute de mult timp.

- „**modus-ponens**” sau „**modus-ponens**”, care se enunță astfel:



Fiind dat P adevărat  
din  $P \rightarrow Q$  \_\_\_\_\_  
rezultă Q adevărat

notat și cu

$\frac{P}{P \rightarrow Q}$   
Q

- b. *regula de înlănțuire a inferențelor* sau *închiderea tranzitivă a inferențelor*. Ea permite ca pe baza a două implicații să se deducă o a treia. Astfel, această regulă se poate scrie:  
dacă  $P \rightarrow Q$  și  $Q \rightarrow R$  atunci  $P \rightarrow R$

**Exemplu.** Considerăm următoarele formule:

- i.  $(P \rightarrow Q) \rightarrow ((P \vee Q) \rightarrow (R \vee Q))$
- ii.  $(R \vee Q) \rightarrow (R \vee S)$
- iii.  $P \rightarrow Q$

modus ponens la (3) și (1), rezultă

- iv.  $(P \vee Q) \rightarrow (R \vee Q)$

înlănțuirea inferențelor (4) și (2) rezultă

- v.  $(P \vee Q) \rightarrow (R \vee S)$ .

### Strategii de demonstrare automată

Constituie o ramură importantă a inteligenței artificiale. În demonstrație, în general, trebuie arătat că o formulă B este „T” folosind pentru aceasta mecanismul modus-ponens, adică presupunând că A este „T” și are loc implicația  $A \rightarrow B$ , adică

$$\left( \frac{A, A \rightarrow B}{B} \right).$$

Această regulă este dificil de aplicat direct și din această cauză se utilizează diferite strategii de demonstrație pe care se bazează demonstrarea automată.

**A. Strategia demonstrării prin adoptarea unei premise/ipoteze/asertiuni auxiliare sau suplimentare.** Această strategie se bazează pe cele 2 teoreme ale lui Stoll (1961) :

1. Demonstrarea concluziei B din premisa A este echivalentă cu demonstrarea lui B fără a presupune premise speciale asupra lui A.  
Adică  $A \vdash B \leftrightarrow A \rightarrow B$  (dacă A atunci  $B \leftrightarrow A \rightarrow B$ )
2. Dacă propoziția B depinde individual de  $A_1, A_2 \dots A_n$  ea depinde și de conjuncția acestor premise și invers.  
 $A_1, A_2 \dots A_n \vdash B \leftrightarrow A_1 \wedge A_2 \wedge \dots \wedge A_n \rightarrow B$

La cele două teoreme ale lui Stoll se mai pot adăuga o serie de tautologii auxiliare, mai importante:

$$(X \rightarrow (Y \rightarrow Z)) \leftrightarrow ((X \wedge Y) \rightarrow Z)$$

Demonstrație. Prin transcrierea implicației și aplicarea asociativității și a legii lui deMorgan

$$\sim X \vee (\sim Y \vee Z) \leftrightarrow (\sim X \vee \sim Y) \vee Z \leftrightarrow \sim (X \wedge Y) \vee Z$$

Demonstrarea prin adăugarea unei premise suplimentare constă în următoarele: *dacă prin adăugarea la ipotezele  $A_1...A_n$  a unei premise auxiliare  $P$ , are loc concluzia  $Q$ , adică  $(A_1...A_n, P \rightarrow Q)$  atunci  $A_1, A_2...A_n \rightarrow (P \rightarrow Q)$*

Demonstrație : aplicăm  $T_2$  a lui Stoll  $\rightarrow A_1 \wedge A_2 \wedge \dots \wedge A_n \wedge P \rightarrow Q$

Aplicăm  $T_1$  a lui Stoll  $\rightarrow A_1 \wedge A_2 \wedge \dots \wedge A_n \rightarrow (P \rightarrow Q)$

Aplicăm  $T_2$  a lui Stoll  $\rightarrow \vdash A_1, \dots A_n \rightarrow (P \rightarrow Q)$

Această metodă este foarte lungă și este dificilă de aplicat  $\rightarrow$  se pot folosi rar în cadrul demonstrației (exemplu : laturile egale ale triunghiului isoscel) – importanța istorică.

**B. Strategia reducerii la absurd** – se bazează pe faptul că se adoptă ipotezele  $B$  și  $\sim C$  și trebuie să ajungem la  $\sim(B \rightarrow C)$

$$\sim(B \rightarrow C) = \sim(C \vee \sim B) = B \wedge \sim C$$

3 metode de bază în strategia reducerii la absurd :

- i) se pornește de la  $B$  și se ajunge la  $C$
- ii) se pornește de la  $\sim C$  și se demonstrează că are loc  $\sim B$  ;  $B$ - premisă  $\rightarrow$  contradicție ;
- iii) se pornește de la  $B \wedge \sim C$  și se aplică regulile și axiomele SF până se ajunge la o propoziție  $p \wedge \sim p$  (se contrazic)

**Observație.** Această metodă se poate utiliza doar în cazurile în care implicația  $B \rightarrow C$  este 'T'. Dacă  $B \rightarrow C$  este 'F', metoda reducerii la absurd poate să ducă la raționamente infinite. Exemplul tipic de astfel de raționament - a 5-a axiomă a lui Euclid : printr-un punct exterior unei drepte se poate duce o singură paralelă la acea dreaptă. Această afirmație este adevărată doar în cadrul unui plan, într-un spațiu multidimensional nu este adevărat și deci demonstrarea pe baza axiomelor lui Euclid a dus la un raționament infinit ce a generat geometriile neeuclidiene de tip Lobacevski-Bolyai sau Hilbert.

**C. Strategia bazată pe rezoluție/rezolvare** – regula de rezoluție – fundamentală în demonstrarea automată, regula de bază alături de modus-ponens și înlănțuirea inferențelor (le include pe ambele)

Legea rezoluției		Înlănțuirea inferențelor		Modus-ponens	
din	$X \vee A$	din	$\sim X \rightarrow A$	din	$A$
și	$Y \vee \sim A$	și	$A \rightarrow Y$	și	$A \rightarrow Y$
rezultă	$X \vee Y$	rezultă	$\sim X \rightarrow Y$	Rezultă	$Y$

Regula rezolvării permite să combinăm 2 formule disjuncte în care apare atomul A și negatul lui, eliminând atomul respectiv.

Modus-ponens – caz particular al regulii rezoluției pentru că de considerăm  $X \in \Phi$  sau  $X$  = propoziție falsă în ipoteza că are loc premiza  $A = T$ . rezultă  $Y$ .

Demonstrația legii rezoluției:

$$(X \vee A), (Y \vee \sim A) \vdash (X \vee Y)$$

T<sub>2</sub> Stoll & definiția implicației  $\rightarrow \sim((X \vee A) \wedge (Y \vee \sim A)) \vee (X \vee Y)$

Din regula lui DeMorgan și dubla negație  $(\sim X \wedge \sim A) \vee (\sim Y \wedge A) \vee (X \vee Y)$  din asociativitatea lui  $\vee$  :  $(X \vee (\sim X \wedge \sim A)) \vee (Y \vee (\sim Y \wedge A))$   
din distributiv lui  $\wedge$  :

$$\begin{aligned} & ((X \vee \sim X) \wedge (X \vee A)) \vee (Y \vee \sim Y) \wedge (Y \vee A) = \\ & (X \vee \sim A) \vee (Y \vee A) = (X \vee Y) \vee (A \vee \sim A) = T \end{aligned}$$

Argumente pentru generalizarea demonstrației pe baza regulii rezoluției – automatismul și simplitatea.

Pași utilizați în demonstrarea prin metoda rezoluției:

- P<sub>1</sub> : se presupune prin absurd că avem concluzia falsă ; se transformă echivalențele în implicații duble și se transcriu implicațiile prin disjuncții;
- P<sub>2</sub> : pentru negarea din fața parantezelor se aplică regula lui De Morgan;
- P<sub>3</sub> : se aplică distributivitatea  $\vee$  față de  $\wedge$  și invers.
- P<sub>4</sub> : fiecare premisă se aduce la forma clauzală, adică la atomi, negați de atomi și de clauze, adică atomi negați de atomi, legați prin disjuncție.
- P<sub>5</sub> : Se aplică principiul rezoluției până se ajunge la o propoziție și negatul acesteia, adică la o contradicție.

**Exemplu.** Să se demonstreze  $[(P \vee Q) \wedge (P \rightarrow R) \wedge (Q \rightarrow S)] \vdash R \vee S$

Aceasta este echivalentă cu:  $P \vee Q, P \rightarrow R, Q \rightarrow S \vdash R \vee S$  – folosește metode bazate pe rezoluție pentru demonstrație:

P1. Se consideră ipotezele, negatul concluziei și definiția implicației se obține:

$$(1) P \vee Q$$

$$(2) \sim P \vee R$$

$$(3) \sim Q \vee S$$

$$\text{din } \sim(R \vee S) = \sim R \wedge \sim S \rightarrow (4) \sim R, \text{ și } (5) \sim S - \text{sunt } .T.$$

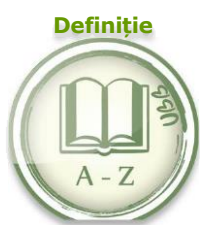
$$\text{din } (4), (2) \text{ \& perinc. rezoluției } \rightarrow (6) (\sim P \vee R), (\sim R) \rightarrow \sim P$$

$$\text{din } (6), (1) \rightarrow (7) (P \vee Q), (\sim P) = \dots\dots \left. \begin{array}{l} \rightarrow Q \\ \text{Contradiție} \\ \sim Q \end{array} \right\}$$

$$(3), (5) \rightarrow (8) (\sim Q \vee S), (\sim S) = \dots\dots \rightarrow \sim Q$$

$\rightarrow$  cond.de la care am plecat e falsă, rezultă formula inițială este „T”.

### 5.2.2.3 FORME NORMALE ȘI CONSECINȚE LOGICE



**Logica matematică** = logica simbolică în care afirmațiile (pe care le vom numi propoziții) sunt notate cu literele alfabetului.

**Logica propozițiilor** = un limbaj formal care conține un alfabet, reguli de sintaxă, axiome și o regulă de deducție. (studiază legăturile dintre propoziții. (!”Această propoziție este falsă”?).

**Propoziții** = *atom* = afirmații cărora li se atașează o valoare de adevăr.

**Logica simbolică** = utilizează simboluri pentru reprezentarea elementelor universului problemei și a operațiilor asupra acestora.

#### Operații logice:

Negatia	
p	~p
1	0
0	1

Disjuncția		
p	q	$p \vee q$
1	1	1
1	0	1
0	1	1
0	0	0

Disjuncția exclusivă		
p	q	$p \vee q$
1	1	0
1	0	1
0	1	1
0	0	0

Conjuncția		
p	q	$P \wedge q$
1	1	1
1	0	0
0	1	0
0	0	0

Implicația		
p	q	$p \rightarrow q$
1	1	1
1	0	0
0	1	1
0	0	1

Echivalența		
p	q	$p \leftrightarrow q$
1	1	1
1	0	0
0	1	0
0	0	1

ordinea efectuării operațiilor logice:  $\sim$ ,  $\wedge$ ,  $\vee$ ,  $\setminus$ ,  $\rightarrow$ ,  $\leftrightarrow$

### Calcul propozițional:

*atom* = propoziție cu valoare de adevăr

*formulă* = propoziție compusă

**Formulă bine formată** (sau **bine formulată - wff**): se construiește după următoarele reguli:

R1. Un atom este o formulă.

R2. Dacă F este o formulă, atunci  $\sim F$  este formulă bine formată.

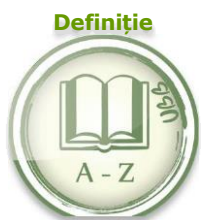
R3. Dacă F și G sunt formule, atunci :  $F \vee G$ ,  $F \wedge G$ ,  $F \rightarrow G$ ,  $F \leftrightarrow G$  sunt bine formate.

R4. Orice f.b.f. se obține prin aplicarea regulilor R1,R2,R3.

O formulă bine formată poate fi:

VALIDĂ		INVALIDĂ	
Totdeauna adevărată.	Nu totdeauna adevărată.	Totdeauna falsă.	
	Nu totdeauna falsă.		
CONSISTENTĂ		INCONSISTENTĂ	

## FORME NORMALE



**Echivalența** (se notează  $F=G$ ):  $F$  și  $G$  sunt echivalente atunci când  $F$  și  $G$  au aceeași valoare pentru orice interpretare  $I$ .

**Reguli de echivalență în utilizarea lui  $.T.$  și  $.F.$ :**

$$\begin{array}{ll} P \vee .F. = P & P \vee .T. = .T. \\ P \wedge .T. = P & P \wedge .F. = .F. \end{array}$$

**Forma normală conjunctivă (FNC):** Dacă  $F_1, F_2, \dots, F_n$  sunt formule bine formate care conțin doar literale (atomi sau negație de atomi) și sunt de forma  $L_1 \vee L_2 \vee \dots \vee L_m$  (disjuncție de literale), atunci  $F_1 \wedge F_2 \wedge \dots \wedge F_n$  este o **fnc**.

**Forma normală disjunctivă (FND):** Dacă  $F_1, \dots, F_n$  sunt formule bine formate care conțin doar literale (atomi sau negații de atomi) și sunt de forma  $L_1 \wedge L_2 \wedge \dots \wedge L_m$  (conjuncție de literale) atunci  $F_1 \vee F_2 \vee \dots \vee F_n$  este o **fnd**.

Exemple: - fnc:  $(P \vee Q \vee \sim R) \wedge (P \vee \sim Q \vee R) \wedge (\sim P \vee Q \vee R)$

- fnd:  $(P \wedge \sim Q) \vee (\sim P \wedge \sim Q) \vee (P \wedge Q)$ .

### Algoritm de transformare a unei formule bine formate într-o formă normală:

Pas 1: Se elimină echivalența și implicația:

$$F \leftrightarrow G = (F \rightarrow G) \wedge (G \rightarrow F)$$

$$F \rightarrow G = \sim F \vee G$$

Pas 2: Se elimină dubla negație și se distribuie negația (Legile lui de Morgan):

$$\sim(\sim F) = F$$

$$\sim(F \vee G) = \sim F \wedge \sim G$$

$$\sim(F \wedge G) = \sim F \vee \sim G$$

Pas 3: Se aplică distributivitatea pentru separarea operațiilor  $\wedge, \vee$ :

$$F \vee (G \wedge H) = (F \vee G) \wedge (F \vee H)$$

$$F \wedge (G \vee H) = (F \wedge G) \vee (F \wedge H)$$

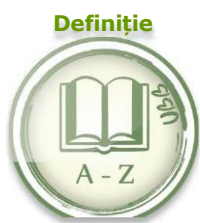
Exemple:

a) **fnd:**  $((P \rightarrow Q) \rightarrow (R \rightarrow S)) \wedge (Q \rightarrow \sim(P \wedge R)) \dots$

$$\dots (P \wedge \sim Q) \vee (P \wedge Q \wedge \sim R) \vee (Q \wedge \sim P) \vee (Q \wedge P \wedge \sim R) \vee (\sim R \wedge \sim P) \vee (\sim R \wedge P) \vee \sim R \vee (S \wedge \sim Q) \vee (S \wedge \sim P) \vee (S \wedge R)$$

b) **fnc:**  $P \wedge (Q \rightarrow R) \rightarrow S \dots (S \vee \sim P \vee Q) \wedge (S \vee \sim P \vee \sim R)$

## Consecințe logice:



Dacă  $F_1, \dots, F_n, G$  sunt f.b.f., și *dacă* pentru orice interpretare  $I$  pentru care  $F_1 \wedge F_2 \wedge \dots \wedge F_n$  este adevărată, atunci  $G$  este adevărată atunci se spune că  $G$  este o **consecință logică** a lui  $F_1, \dots, F_n$ , iar  $F_1, \dots, F_n$  sunt **axiome** pentru  $G$ .

**Teorema1:**  $G$  este o consecință logică a lui  $F_1, \dots, F_n$  dacă și *numai dacă* formula  $F_1 \wedge \dots \wedge F_n \rightarrow G$  este validă.

**Teorema2:**  $G$  este o consecință logică a lui  $F_1, \dots, F_n$  dacă și *numai dacă* formula  $F_1 \wedge \dots \wedge F_n \wedge \sim G$  este inconsistentă.

**Problema1:**

Fie  $F_1: P \rightarrow Q$

$F_2: \sim Q$

$G: \sim P$

Să se demonstreze că:  $G$  este o consecință logică a lui  $F_1$  și  $F_2$ .

Indicație: Din  $T1 \Rightarrow F_1 \wedge F_2 \rightarrow G$  - validă sau din  $T2 \Rightarrow F_1 \wedge F_2 \wedge \sim G$  - inconsistentă.

**Problema2:**

Se dau:

$P$  = Parlamentul refuză să acționeze.

$Q$  = Greva s-a terminat.

$R$  = Directorul firmei demisionează.

$S$  = Greva continuă de mai mult de un an.

$F_1$  : Dacă parlamentul refuză să acționeze atunci directorul firmei demisionează și greva continuă mai mult de un an.

$F_2$  : Parlamentul refuză să acționeze.

$F_3$  : Greva nu s-a terminat.

Se cere:

i) să se transpună în formule  $F_1, F_2$  și  $F_3$ ;

ii) să se demonstreze că  $F_3$  este o consecință logică a lui  $F_1 \wedge F_2 \wedge F_3$ .

## Tema 1

Să se construiască o problemă în stilul problemei 2, să se transpună apoi în formule și să se rezolve.

Ex de rezv cu tabele. Tema voastra este sa demonstrati cu ajutorul teoremelor.

Se cunosc:

N = se formează numărul;

T = telefonul sună;

L = linie ocupată;

A = abonatul răspunde;

M = linie ocupată pe timpul convorbirii.

$F_1$  : Se formează numărul, telefonul sună sau linie ocupată.

$F_2$  : Telefonul sună, abonatul răspunde, linia este ocupată pe timpul convorbirii.

$F_3$  : Dacă linia nu e ocupată telefonul sună.

$F_1$						$F_2$		$F_3$		
T	L	A	M	N	$N \rightarrow T$	$N \rightarrow T \vee L$	$T \rightarrow A$	$T \rightarrow A \rightarrow M$	$\sim L$	$\sim L \rightarrow T$
1	0	1	1	0	1	1	1	1	1	1
1	1	0	0	1	1	1	0	1	0	1
0	0	1	1	1	0	0	1	1	1	0
0	1	0	0	0	1	1	1	0	0	1



## 5.3 AGENȚI INTELIGENȚI

Un **agent** este o entitate care percepe mediul înconjurător prin senzori și acționează asupra lui prin efecotori. Un agent uman are ochi, urechi și alte organe ca senzori și mâini, picioare etc. ca și efecotori. Un agent robotic poate avea camere sau dispozitive cu infraroșu pe post de senzori și diferite motoare pe post de efecotori.

### Definiție



Folosim termenul **“percepție”** pentru a ne referi la inputul perceptual pe care îl are agentul la un moment dat. O secvență de percepții a unui agent reprezintă o istorie completă a tot ce a perceput agentul. Dacă specificăm acțiunile pe care poate să le facă agentul pentru fiecare secvență de percepții posibilă, precizăm, într-o mare măsură, tot ce este de zis despre agent. Din punct de vedere matematic, spunem că, comportamentul agentului este

descriș de către funcția agent care mapează fiecare secvență de percepții pe o acțiune. Putem să ne imaginăm funcția agent ca și un tabel. Tabelul este o caracterizare externă a agentului. Din punct de vedere intern, funcția agent pentru un agent artificial va fi implementată de un program agent.

Un **agent rațional** este acela care face lucrul potrivit -adică fiecare intrare în tabelul cu funcțiile agentului este completată în mod corect. Ce înseamnă să faci lucrul bun/corect? Considerăm consecințele comportamentului agentului. Când un agent este într-un mediu, generează o secvență de acțiuni în funcție de percepții pe care îi primește. Secvența aceasta determină mediul să treacă printr-o secvență de stări. Dacă secvența este cea dorită, atunci agentul a efectuat în mod corect ce avea de făcut. Noțiunea de “dorită” este dată de o măsură a performanței care evaluează fiecare secvență dată de stări ale mediului.

Nu există o măsură a performanței fixă pentru toți agenții și activitățile. În mod normal, un designer va decide o măsura potrivită în funcție de circumstanțe. Când vine vorba de un agent rațional, ceea ce ceri este ceea ce primești. Este mai bine să decizi o măsură de performanță în funcție de ceea ce vrea cineva în mediu, decât în funcție de cum crede cineva că trebuie să se comporte agentul.

Ceea ce este rațional la un moment dat, depinde de **4 lucruri**:

- Măsura de performanță care definește gradul de succes
- Secvența de percepții
- Cunoștințele pe care le-a dobândit agentul despre mediu până în momentul de față
- Acțiunile pe care agentul le poate face

### Definiția unui agent rațional :

Pentru fiecare secvență de percepții posibilă, un agent rațional ar trebui să selecteze o acțiune care se așteaptă să maximizeze măsura sa de performanță, având în vedere informațiile pe care i le dă secvența de percepții și orice alte cunoștințe pe care le are.

## Omnisciență vs raționalitate

Un agent atotștiutor cunoaște foarte bine ceea ce ar trebui să facă și acționează asemenea. Dar omnisciența este imposibilă în realitate. Spre exemplu, vreau să traversez strada, mă asigur, mă uit în stânga și în dreapta și traversez. În timp ce traversez, sunt lovit de către un meteorit. Pot spune că am acționat irațional când am traversat strada ?



**Raționalitatea** nu este același lucru cu perfecțiunea. Raționalitatea maximizează performanța care se așteaptă, în timp ce perfecțiunea maximizează performanța actuală.

Definiția raționalității nu implică omnisciența pentru că o decizie rațională depinde doar de secvența de percepții de până în prezent. Spre exemplu, dacă un agent nu se uită în ambele părți când vrea să traverseze, atunci secvența de percepții nu o să-i spună că este un camion care se apropie cu viteză mare, pentru că nu are de unde să știe. Din punct de vedere rațional este bine să treci strada în cazul de față ? Nu . În primul rând, nu este rațional pentru că eu știu din secvența de percepții că riscul de accident în cazul în care treci strada fără să te asiguri este mare. În al doilea rând, un agent rațional ar trebui să aleagă acțiunea de a se uita în stânga și în dreapta înainte să facă pasul pe stradă, pentru că asigurarea ajută maximizarea performanței așteptate. A face acțiuni pentru a modifica percepții viitoare este o parte importantă a raționalității și se numește adunarea de informații. O altă metodă de a aduna informații este explorarea.

**Raționalitatea** cere nu numai adunarea de informații, ci și învățarea pe cât de mult din ceea ce percepe agentul. Configurarea inițială a agentului ar putea reflecta niște cunoștințe anterioare despre eveniment, dar , pe parcurs ce agentul capătă experiență, asta s-ar putea modifica. Sunt puține cazuri în care evenimentul este cunoscut în totalitate dinainte.

În măsura în care un agent se bazează pe cunoștința anterioară a designerului său decât pe propriile percepții, putem spune că agentul duce lipsă de autonomie. Un agent rațional ar trebui să fie autonom, ar trebui să învețe cum să compenseze cunoștințele incorecte sau incomplete de până atunci. Spre exemplu, un aspirator care învață cum să anticipeze unde și când va apărea mai multă mizerie, se va descurca mai bine decât unul care nu va face asta . Bineînțeles, ar trebui ca designerul să îi asigure agentului niște cunoștințe inițiale precum și capacitatea de a învăța. După o experiență suficientă în mediul său înconjurător, comportamentul unui agent rațional poate deveni independent de cunoștințele anterioare. Așadar, dacă îi este oferită oportunitatea de a învăța, un singur agent poate avea succes într-o varietate de medii.

Dupa ce am înțeles ce înseamnă raționalitatea, putem să ne gândim să construim un agent rațional. Primul aspect care trebuie luat în considerare este mediul, care imi definește, în esență, problemele pentru care agenții raționali sunt soluțiile.

Cand am vorbit despre aspirator, a trebuit să specificăm măsura de performanță, mediul, senzorii și efectorii săi . Toate aceste lucruri adunate poartă denumirea de mediu de activitate sau descriere PEAS. Prima dată când concepem un agent, trebuie să specificăm cât mai concret mediul de activitate.



Să luăm exemplul unui șofer de taxi automat. Trebuie specificat că acest caz este într-un fel peste capacitățile tehnologiilor existente. Prima dată, vom preciza care sunt măsurile de performanță pe care dorim să le aibă șoferul nostru automat: să ne ducă la destinația corectă, să minimizeze consumul de combustibil, timpul călătoriei și costul, încălcarea legilor, să maximizeze siguranța pasagerilor și confortul, profitul. Următorul pas este reprezentat de mediu: drumuri, trafic, pietoni, clienți, mașini, animale etc. Vremea poate face parte, de asemenea, din mediu.

Efactorii unui șofer automat coincid în mare cu cei ai unui șofer uman. Controlul asupra accelerației, ambreiajului, franei. Pe lângă acestea, mai avem nevoie și de un ecran prin intermediul căruia să comunice cu pasagerii sau alte vehicule.

Senzorii vor include una sau mai multe camere video cu ajutorul cărora să poată vedea drumul. Mai pot avea nevoie de senzori de sunet sau infraroșu ca să detecteze distanța față de alte mașini. De asemenea, ca să evite amenzi de viteză, ar trebui să aibă un vitezometru și accelerometru. Evident, un motor, combustibil. Totodată, un GPS ca să nu se piardă și o tastatură sau un microfon ca pasagerii să introducă o destinație.

### 5.3.1 Proprietăți:

În ceea ce privește inteligența artificială, putem vorbi despre un număr mare de medii care pot să apară. Pentru acestea se poate determina un număr relativ minim de dimensiuni după care se poate face o categorizare a acestora.

Aceste dimensiuni determină designul potrivit și aplicabilitatea fiecărei grupe de tehnici pentru implementare.

*Exemplu:*

Tip	Performanță	Mediu	Actori	Senzori
Sistem de diagnosticare medical	Pacienți sănătoși, Costuri reduse	Pacient, spital, staff	Întrebări, teste, diagnostice	Tastatură, răspunsurile pacienților

#### 5.3.1.1 Complet observabil vs parțial observabil

**Un mediu este complet observabil** dacă senzorii agentului detectează toate aspectele mediului care sunt relevante și determină alegerea acțiunii. Practic senzorii au acces la întregul univers în fiecare moment de timp. Alegerea acțiunii este un aspect care este influențat de relevanța acesteia și de asemenea depinde de performanță. Un astfel de mediu este necesar pentru ca agenții nu trebuie să mențină starea internă pentru a ține pasul cu lumea.

Un mediu este considerat **parțial observabil** datorită inacurateții senzoriilor sau deoarece lipsesc părți din datele senzoriilor. De asemenea mai apare situația în care lipsesc senzorii și atunci mediul este considerat **neobservabil**

*Exemple:*

- Aspiratorul cu locație nu poate să prezică dacă există mizerie și în alte locuri.
- Șoferul de taxi automat nu poate să determine ce gândesc ceilalți șoferi.

- **Deterministic vs stochastic**

Un mediu este considerat **deterministic** atunci când următoarea stare a mediului este determinată de starea curentă și de acțiunile selectate de agenți. Dacă un mediu este complet observabil și deterministic rezultatul este inexistentă incertitudinii.

**Stochastic** – este un termen folosit în inteligența artificială atunci când vorbim despre programe care rezolvă problemele cu probabilități.

Un mediu **stochastic** poate să fie un mediu parțial observabil. De exemplu șofer de taxi automat care nu poate să prezică exact cum o să fie traficul pe parcursul cursei. Pentru situațiile complexe este convenabil să fie considerate stochastice.

De asemenea mai apar încă două notații:

-**incert** : mediu parțial observabil sau nedeterministic

-**nondeterministic** – acțiunile sunt caracterizate după rezultatele lor posibile, dar nicio probabilitate nu este atașată

- **Episodic vs secvențial**

În ceea ce privește un **mediu episodic**, întreaga experiență a agentului este împărțită în episoade atomice. În fiecare episod agentul primește o percepție (input) și realizează o singură acțiune. În acest caz este crucial ca următorul episod să nu depindă de cel anterior. Aceste medii sunt mai preferabile deoarece agenții nu trebuie să se gândească înainte la ce urmează.

În cadrul unui **mediu secvențial**, decizia curentă afectează toate deciziile viitoare. Se merge pe ideea " acțiunile pe termen scurt pot avea consecințe pe termen lung". Atât în ceea ce privește exemplul cu șoferul de taxi automat cât și un simplu joc de șah, putem vorbi despre medii secvențiale, deoarece orice decizie curentă va afecta deciziile viitoare.

#### **5.3.1.4 Static vs dinamic**

Ne putem referi la un mediu ca fiind **dinamic** dacă mediul se poate schimba în timp ce agentul hotărăște ce acțiune va face în continuare, iar în caz contrar mediul este **static**. Acesta din urmă este mai preferabil deoarece agenții nu trebuie să fie totuși atenți la mediu sau să își facă griji în privința timpului.

Un mediu dinamic întreabă constant agentii ce vor face în continuare, dacă aceștia nu acționează se presupune că nu vor face nimic. Aici apare noțiunea de mediu semidynamic atunci când mediul nu se schimbă însă performanța agentului se schimbă.

*Exemple:*

- Sofer de taxi- mediu dinamic
- Șah (cu ceas) –mediu semidynamic
- Rebus – mediu static

## • Discret vs continuu

Această distincție se aplică stărilor mediilor, în felul în care timpul este administrat și pe baza la imputurile și acțiunile agenților. Dacă timpul, numărul de percepții sau acțiuni diferite sunt finite atunci mediul este **discret**. Spre exemplu, șahul are un număr finit de stări diferite și un set discret de percepții și acțiuni.

Sper deosebire de jocul de șah, în cazul șoferului de taxi automat putem vorbi despre o stare **continuă** la care se adaugă și o problemă constantă cu timpul.

## • Cunoscut vs necunoscut

Această proprietate se referă la starea de cunoaștere ale agenților despre legile fizice ale mediilor:

- **Mediu cunoscut** – rezultatele pe toate acțiunile sunt date
- **Mediu necunoscut**- agenții trebuie să învețe cum funcționează pentru a lua decizii bune.  
Trebuie scoasă în evidență diferența dintre un mediu cunoscut- necunoscut și un mediu complet-partial observabil.
  - Un mediu cunoscut poate fi parțial observabil
  - Solitaire- cunoscutul joc de cărți pentru că știm regulile, însă nu putem ști ce carte o să vină în continuare.
  - Un mediu necunoscut poate fi complet observabil
  - Joc video- putem vedea întreaga suprafață de joc, însă nu știm ce funcționalități au toate butoanele.

## • Cu un singur agent vs multiagent

Fiecare mediu poate să aibă mai unul sau mai mulți agenți. De exemplu un rebus sau o integrăă reprezintă un mediu cu un singur agent, pe când un joc șah impune din start 2 agenți.

Aici apare noțiunea de entitate, care poate fi văzută ca și un agent. Însă problema se pune în felul următor: cum poate să fie văzută o entitate ca agent? Dacă avem doi agenți A și B, A poate să vadă agentul B fie ca agent fie ca un obiect care acționează după regulile fizicii.

Cheia constă în felul în care comportamentul lui B este cel mai bine descris ca maximizare a măsurătorii performanței lui A. De exemplu, în cadrul jocului de șah dacă B își maximizează performanțele are ca și consecință o minimizare a performanței lui A.

În cadrul mediilor multiagent apare o nouă clasificare:

- Medii competitive – jocul de șah
  - Putem vorbi aici de *Comportament random*- pentru evitarea predictibilității în mediile competitive
- Medii cooperative – evitarea unui accident în trafic , crește performanța tuturor soferilor.

*Exercițiu :*

Medii	Observabil	Deterministic	Episodic	Static	Discret
Șah cu ceas	Da	Da	Nu	Semi	Da
Șah fără ceas	Da	Da	Nu	Da	Da
Poker	Nu	Nu	Nu	Da	Da
Solitaire	Nu	Da	Nu	Da	Da

### 5.3.2 Structura

Sarcina AI este de a construi programe agent care implementează funcția agentului, practic toate percepțiile sunt mapate în acțiuni. Programele funcționează pe anumite dispozitive cu senzori și efectori fizici. Aceasta reprezintă arhitectura.

Toate programele agent au același schelet: acceptă percepțiile curente de la senzori ca și input și returnează o acțiune. Trebuie menționat faptul că există o diferență între programele agent și funcțiile agent: programele agent iau ca și input percepția curentă, în timp ce funcțiile agent iau ca și input întreaga istorie de percepții. Dacă acțiunea agentului depinde de o secvență de percepții atunci acesta trebuie să își amintească întreaga secvență de percepții.

În continuare sunt prezentate 4 tipuri de bază de programe agent :

#### 5.3.2.1 Agentul reflex simplu

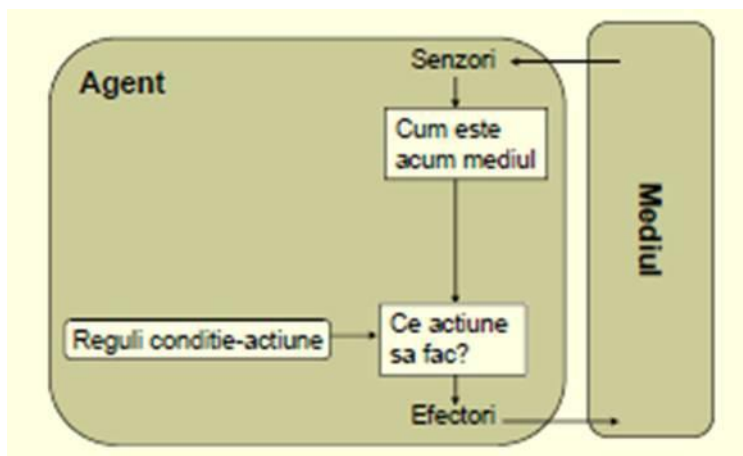
Acești agenți selectează acțiunea pe care o returnează, bazându-se doar pe percepția curentă, ignorând istoria de percepții.

În ceea ce privește exemplul cu aspiratorul este vorba despre un agent reflex simplu, deoarece decizia se bazează doar pe locația curentă și pe faptul dacă aceasta conține sau nu mizerie.

Un alt exemplu relevant are în vedere un taxi automat. Dacă mașina din față frânează , și lumina de la frână se aprinde, agentul observă și inițiază frânarea. Practic, se realizează o procesare pe baza inputului vizual pentru a stabili condiția pe baza căreia se stabilește condiția : ”mașina din față frânează”. Declanșatoarele stabilesc o conexiune la nivelul agentului program pentru acțiunea

de frânare. Astfel regula de acțiune pe baza condiției este: Dacă mașina din față frânează atunci inițializez frânarea.

Agentul percepe starea curentă a mediului prin intermediul senzorilor, iar pe baza unei reguli a cărei condiție se potrivește cu starea curentă, selectează acțiunea pe care o realizează prin intermediul efectorilor. Sunt simpli și cu inteligență limitată.



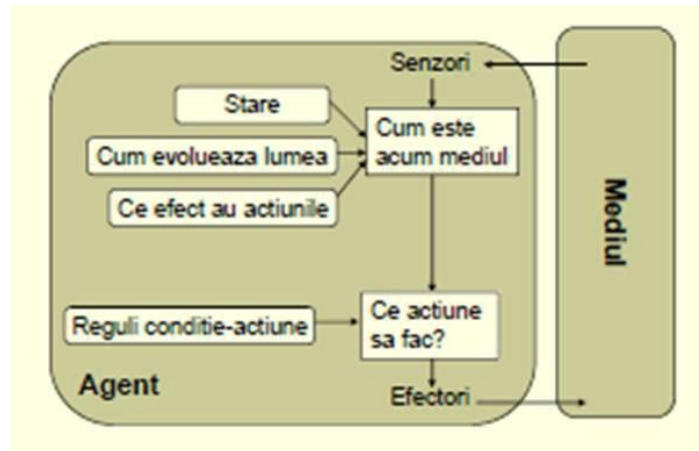
### 5.3.2.2 Agent reflex cu stare internă

Cel mai eficient mod al unui agent de a rezolva problemele referitoare la observabilitatea parțială este de a păstra urma părții mediului pe care nu o vede. Astfel agentul menține o stare internă care depinde de istoria de percepții, și care reflectă unele aspecte neobservate ale stării curente. Referitor la exemplul anterior, cu taxi-ul automat, în cazul în care se dorește schimbarea benzii, trebuie păstrată urma celorlalte mașini, în cazul în care nu are o vedere de ansamblu asupra celorlalte mașini.

Actualizarea stării interne se realizează ținând cont atât de modul în care mediul evoluează independent de agent, de exemplu dacă o mașină în depășire e mai aproape decât a fost cu un moment în urmă, cât și de informația privind modul în care acțiunea agentului afectează mediul: de exemplu, o mașină conduce 5 km spre nord, ea va fi cu 5 km mai în nord decât a fost în urma cu câteva minute.

În cazul de față, se observă modul în care percepția curentă este combinată cu starea internă veche pentru a genera o descriere actualizată a stării curente, bazându-se pe modul în care evoluează mediul. Cu toate acestea sunt foarte rare situațiile în care agentul determină starea curentă a unui mediu parțial observabil, cu exactitate. De exemplu: în cazul taxi-ului, dacă are în față un tir, și acesta oprește, nu știe motivul pentru care oprește. Știe doar că oprește și trebuie să reacționeze, trebuie să ia o decizie.



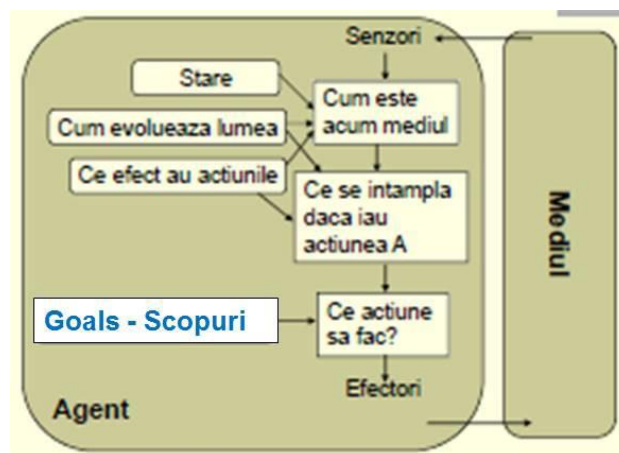


### 5.3.2.3 Agent cu scop exact

În anumite situații, nu este suficient să fie cunoscută doar starea curentă a mediului pentru a decide ce trebuie făcut. De exemplu, în cazul unei intersecții, taxi-ul poate să meargă la stânga, la dreapta sau în față. Decizia corectă depinde de destinația taxi-ului. Astfel pe lângă descrierea stării curente, agentul, are nevoie de informație în ceea ce privește scopul care să descrie situația care este dorită să se întâmple: destinația pasagerului.

Este păstrată atât urma stării mediului, cât și a unui set de scopuri, pe care încearcă să le atingă. Când o acțiune este aleasă, decizia este luată astfel încât să conducă în cele din urmă la realizarea scopurilor.

Deși pare mai puțin eficientă, furnizează mai multă flexibilitate, deoarece cunoașterea pe baza căreia se ia o decizie este reprezentată cu exactitate și poate fi modificată. De exemplu, dacă începe să plouă, agentul își actualizează cunoașterea referitor la cât de eficient vor funcționa frânele. Astfel toate comportamentele relevante se adaptează noilor condiții. În cazul unui agent reflex, trebuie rescrise regulile condițiilor de acțiune.





### 5.3.2.4 Agent bazat pe funcționalitate

În cele mai multe medii , doar scopurile nu sunt suficiente pentru a genera comportamente de înaltă calitate. De exemplu: mai multe secvențe de acțiuni vor duce taxi-ul la destinație (scopul este atins) , dar nu toate sunt la fel de rapide, sigure, mai rentabile, mai ieftine decât celelalte).

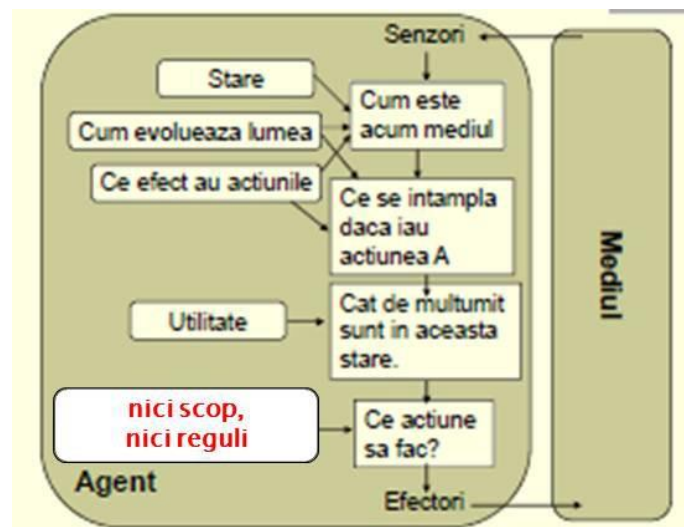
Scopul are rolul doar de a face distincția dintre o stare fericită și o stare nefericită.

O măsură de măsurare a performanței mai generală trebuie să permită să stabilim exact cât de fericit va face agentul, în urma acțiunilor alese .

Un agent bazat de funcționalitate, folosește un model al mediului împreună cu o funcție a utilității care îi măsoră preferința în ceea ce privește stările mediului. Ulterior este aleasă acțiunea care conduce la cea mai bună utilitate așteptată, unde utilitatea așteptată este calculată.

Funcția de utilitate a unui agent este o internalizare a măsurii de performanță ; dacă funcția de utilitate internă este în concordanță cu măsura de performanță externă , atunci agentul care alege să acționeze astfel încât să își maximizeze utilitatea va fi rațional conform măsurii de performanță externă. Asemenea agenților cu un scop exact , există avantaje în ceea ce privește flexibilitatea și învățarea.

Există situații în care scopurile sunt inadecvate , dar un agent bazat pe funcționalitate poate să ia decizii raționale: scopurile aflate în conflict (viteza si siguranța) , sau în cazul în care există mai multe scopuri cerute , dar niciunul nu poate fi atins cu certitudine.



# Întrebări recapitulative

1. Cum se definește logica?
2. Dați exemple de utilizare a logicii în informatica economică și de afaceri!
3. Cum se definește logica formală și prin ce se deosebește de cea clasică?
4. Care este deosebirea esențială între tratarea logică și cea gramaticală a propozițiilor?
5. Ce este logica bivalentă și ce extensii ale ei cunoașteți?
6. Ce este teoria logică și care sunt abordările sale?
7. Definiți noțiunile de tautologie și de teoremă.
8. Ce este abordarea semantică și care sunt elementele sale de bază?
9. Ce simboluri propoziționale cunoașteți?
10. Ce este un wff în PL?
11. Care este semnificația negației logice?
12. Care este semnificația conjuncției a două propoziții?
13. Care este semnificația disjuncției a două propoziții și prin ce se deosebește disjuncția de „sau exclusiv”?
14. Ce este o propoziție compusă și cum se determină valoarea sa de adevăr?
15. Ce logici polivalente cunoașteți?
16. Ce este raționamentul și ce este consecința logică a unei propoziții?
17. Care sunt legile algebrei Booleene?
18. Ce alte logici bivalente cunoașteți?
19. Enunțați principiul dualității și legile lui DeMorgan!
20. Definiți forma normală conjunctivă și cea disjunctivă!
21. Câte funcții logice binare se pot construi în logica bivalentă?
22. Cum se definesc implicația și echivalența?
23. În ce constă principiul fixării numărului de valori logice?
24. Ce logici trivalente cunoașteți?
25. Ce este logica fuzzy?
26. În ce constă principiul consistenței?
27. În ce constă principiul excluderii celei de  $n+1$ -a valori?
28. Enunțați propozițiile fundamentale ale consecinței logice!
29. Enunțați cele două reguli inferențiale ale logicii propozițiilor!
30. Ce forme normale cunoașteți și cum se poate aduce un wff la forma normală?

31. Definiți noțiunea de agent rational și dați un exemplu.
32. Definiți și explicați noțiunea de *măsură de performanță* a unui agent.
33. Definiți proprietățile mediului în care acționează agenții și explicați-le pe un exemplu.
34. Noțiunea de multiagent și clasificarea mediilor acestuia
35. Prezentați noțiunea de agent inteligent și structura acestuia.
36. Definiți și exemplificați noțiunea de *Agent reflex simplu*.
37. Definiți și exemplificați noțiunea de *Agent reflex cu stare internă*
38. Definiți și exemplificați noțiunea de *Agent cu scop exact*
39. Definiți și exemplificați noțiunea de *Agent bazat pe funcționalitate*
40. Definiți cele patru tipuri de probleme rezolvabile de către agenți