LOGICĂ MATEMATICĂ ŞI COMPUTAŢIONALĂ

CURS 7

REZOLUŢIE

- cea mai eficace metodă de demonstraţie algoritmică a **LP** , precum şi(vezi cap. 4), a logicii predicatelor **LPr**.
- constituie metoda de demonstraţie pe care se bazează limbajul de programare logică PROLOG.
- este o metodă de demonstraţie prin respingere, la fel ca demonstraţiile Beth.

RECAPITULARE

- Un **literal** este orice atom sau negaţia acestuia.

De exemplu: $\neg A, B, \neg C$ sunt literali.

- Ştim că putem dezvolta orice propoziţie din **LP** într-o formă normală conjunctivă **FNC**, care este echivalentă cu propoziţia iniţială. O componentă a **FNC** este de fapt o disjuncţie de literali, astfel că în fiecare disjuncţie nici un literal nu apare mai mult decât o dată.

RECAPITULARE

(i) <u>legile lui De Morgan</u>:

$\neg(A \wedge B) \leftrightarrow \neg A \vee \neg B$ şi $\neg(A \vee B) \leftrightarrow \neg A \wedge \neg B$

(ii) <u>proprietăţile de asociativitate</u> ale lui $\wedge$ şi $\vee$:

$(A \wedge B) \wedge C \leftrightarrow A \wedge (B \wedge C)$ şi $(A \vee B) \vee C \leftrightarrow A \vee (B \vee C)$

(iii) <u>proprietăţile de comutativitate</u> ale lui $\wedge$ şi $\vee$:

$A \wedge B \leftrightarrow B \wedge A$ şi $A \vee B \leftrightarrow B \vee A$

(iv) <u>proprietăţile de distributivitate</u> ale lui $\wedge$ faţă de $\vee$ şi ale lui $\vee$ faţă de $\wedge$:

$A \wedge (B \vee C) \leftrightarrow (A \wedge B) \vee (A \wedge C)$ şi

$A \vee (B \wedge C) \leftrightarrow (A \vee B) \wedge (A \vee C)$

(v) <u>propoziţiile:</u>

$A \vee A \leftrightarrow A$ $A \wedge A \leftrightarrow A$

$A \wedge (B \vee \neg B) \leftrightarrow A$ $A \vee (B \wedge \neg B) \leftrightarrow A$

şi $\neg\neg A \leftrightarrow A$

RECAPITULARE

<u>Exemplul 1</u>: Dezvoltaţi propoziţia S într-o **FNC** unde:

S: $\neg((A \vee B) \wedge (\neg A \vee \neg B) \wedge C$

<u>*Pasul* 1</u>: Mutăm negaţiile spre interiorul parantezelor folosind legile lui De Morgan:

a: $S \leftrightarrow [\neg(A \vee B) \vee \neg(\neg A \vee \neg B)] \wedge C$

b: $S \leftrightarrow [(\neg A \wedge \neg B) \vee (\neg\neg A \wedge \neg\neg B)] \wedge C$

<u>*Pasul* 2</u>: Folosim proprietăţile de asociativitate şi comutativitate pentru a aduce la un loc literalii aceluiaşi atom. Apoi putem simplifica dublele negaţii, termenii dubli de forma $A \vee A$ şi $A \wedge A$ precum şi termenii inutili de forma $B \wedge \neg B$ sau $B \vee \neg B$ folosind teorema substituţiei echivalenţelor:

$$S \leftrightarrow [(\neg A \wedge \neg B) \vee (A \wedge B)] \wedge C$$

<u>*Pasul* 3</u>**:** Conform proprietăţilor de distributivitate avem:

$$S \leftrightarrow [((\neg A \wedge \neg B) \vee A) \wedge ((\neg A \wedge \neg B) \vee B)] \wedge C$$

Continuăm prin repetarea paşilor al doilea şi al treilea. până ce stabilim **FNC** finală:

<u>*Pasul* 1’</u>: $S \leftrightarrow ((\neg A \wedge \neg B) \vee A) \wedge ((\neg A \wedge \neg B) \vee B) \wedge C$

<u>*Pasul* 2'</u>: $\leftrightarrow (\neg A \vee A) \wedge (\neg B \vee A) \wedge (\neg A \vee B) \wedge (\neg B \vee B) \wedge C$

<u>*Pasul* 3'</u>: $\leftrightarrow (\neg B \vee A) \wedge (\neg A \vee B) \wedge C$

care este **FNC** a lui S pe care o cautăm

REZOLUŢIE

- Disjuncţia unui număr finit de *literali* poate fi reprezentată conform *teoriei mulţimilor* ca o mulţime ale cărei elemente sunt literalii în cauză. Această mulţime se numeşte ***clauză***. Astfel, o clauză este echivalentă cu o propoziţie disjunctivă din **LP**.
- Din motive tehnice vom introduce şi noţiunea de ***clauză vidă***, o clauză care nu conţine literali şi este întotdeauna *neverificabilă.* Clauza vidă se notează prin $\square$.
- Conjuncţia unui număr finit de *clauze* poate fi reprezentată *conform teoriei mulţimilor* ca o mulţime ale cărei elemente sunt aceste clauze. Aceasta mulţime se numeşte ***mulţime de clauze***.

REZOLUŢIE

<u>Exemplul 2:</u> Mulţimea de clauze:

$$\{ \underbrace{\{A, B\}}_{1}, \underbrace{\{\neg B, \neg C\}}_{2}, \underbrace{\{D\}}_{3} \}$$

reprezintă propoziţia:

$$(\underbrace{(A \vee B)}_{1} \wedge \underbrace{(\neg B \vee \neg C)}_{2} \wedge \underbrace{D}_{3})$$

REZOLUŢIE

Rezoluţia este o regulă deductivă prin care, într-o clauză, putem deduce o propoziţie din alte două propoziţii.

- Să considerăm ca fiind date următoarele clauze:

$C_1 = \{ A_1,\ldots, A_i, \neg B_1,\ldots, \neg B_j\}$

$C_2 = \{ D_1,\ldots, D_k, \neg F_1,\ldots, \neg F_l\}$

unde $A_1,\ldots, A_i, \neg B_1,\ldots, \neg B_j, D_1,\ldots, D_k, \neg F_1,\ldots, \neg F_l$ sunt atomi.

REZOLUŢIE

- Să presupunem că A_1 coincide cu F_1.

  Putem atunci rescrie cele două clauze după cum urmează:

$$C_1 = \{A_1\} \cup C'_1 \text{ unde } C'_1 = \{A_2, \ldots, A_i, \neg B_1, \ldots, \neg B_j\}$$

$$C_2 = \{\neg A_1\} \cup C'_2 \text{ unde } C'_2 = \{D_1, \ldots, D_k, \neg F_2, \ldots, \neg F_l\}$$

- Atunci, regula rezoluţiei pe care vrem să o stabilim va trebui să ne permită să producem drept deducţie clauza următoare:

$$C = C'_1 \cup C'_2$$

REZOLUŢIE

Rezoluţia, o descriere formalizată:

- Fie C_1 şi C_2 două clauze şi fie L un literal astfel încât $L \in C_1$ şi $(\neg L) \in C_2$.

Putem atunci deduce ***rezolventul D* al lui C_1 şi C_2:**

$$D = (C_1 - \{L\}) \cup (C_2 - \{\neg L\})$$

REZOLUŢIE

Exemplul 3: Să considerăm următoarele clauze:

$$\{\neg A, B\}$$

$$\{A, C\}$$

Aplicând rezoluţia, putem deduce:

$$\frac{\{\neg A, B\} \quad \{A, C\}}{\{B, C\}}$$

REZOLUŢIE

- Fie $S = \{C_1, C_2, \ldots, C_n\}$ *o* multime de clauze. Atunci mulţimea:

  $R(S) = S \cup \{D \mid D$ este rezolventul clauzelor $C_i, C_j \in S, i \neq j, 1 \leq i,j \leq n\}$

  este *rezolventul* lui S.

- Fireşte, putem continua cu aplicarea metodei, luând succesiv mulţimile următoare:

$$R^0 = S,\ R^1(S) = R(S),\ R^2(S) = R(R(S)), \ldots, R^n(S) = R(R^{n-1}(S)),$$

şi, în final:

$$R^*(S) = \bigcup_{n=1}^{\infty} R^n(S) = \{C_i \mid C_i \in R^j(S) \quad \text{şi} \quad j \in N\}$$

unde C_i sunt clauzele conţinute în al j-lea rezolvent al lui S.

De notat că $R^*(S)$ este o mulţime finită dacă şi numai dacă S este finită.

REZOLUŢIE

Exemplul 4: Fie S o mulţime de clauze:

$$S = \left\{ \underbrace{\{A, \neg B, \neg C\}}_{1}, \underbrace{\{B, D\}}_{2}, \underbrace{\{\neg A, \neg D\}}_{3} \right\}$$

Aplicând regula rezoluţiei perechilor de clauze ale lui S, avem:

$$\frac{1\ \{A, \neg B, \neg C\} \quad 2\ \{B, D\}}{4\ \{A, D, \neg C\}} \qquad \frac{2\ \{B, D\} \quad 3\ \{\neg A, \neg D\}}{5\ \{B, \neg A\}} \qquad \frac{3\ \{\neg A, \neg D\} \quad 1\ \{A, \neg B, \neg C\}}{6\ \{\neg B, \neg C, \neg D\}}$$

şi, în final:

$$R(S) = \left\{ \underbrace{\{A, \neg B, \neg C\}}_{1}, \underbrace{\{B, D\}}_{2}, \underbrace{\{\neg A, \neg D\}}_{3}, \underbrace{\{A, D, \neg C\}}_{4}, \underbrace{\{B, \neg A\}}_{5}, \underbrace{\{\neg B, \neg C, \neg D\}}_{6} \right\}$$

REZOLUŢIE

- Fie S o mulţime de clauze. O ***demonstraţie prin rezoluţie*** din S este o secvenţă finită de clauze $C_1, C_2, \ldots, C_n$, astfel încât pentru fiecare C_i, $i = 1, \ldots, n$, avem:

$$C_i \in (S) \quad \text{sau} \quad C_i \in R(\{C_j, C_k\}) \quad 1 \leq j, k \leq i \leq n.$$

- O clauză C este ***demonstrabilă prin rezoluţie*** *dintr-o mulţime de clauze* S, formalizat $S \vdash_R C$, dacă există o demonstraţie prin rezoluţie din S a cărei ultimă clauză este C. Evident, $C \in R^*(S)$

REZOLUŢIE

Ex1: Aflați toți rezolvenții mulțimii de clauze:

$S = \{\{A, B\}, \{\neg A, \neg B\}\}.$

Ex2: Se dă următoarea propoziţie:

S: $((A \leftrightarrow (B \rightarrow C)) \wedge (A \leftrightarrow B) \wedge (A \leftrightarrow \neg C)$

Demonstraţi că S nu este verificabilă.

BIBLIOGRAFIE

BIBLIOGRAFIE RECOMANDATĂ LA UNITATEA DE INVATARE NR.2

1. G. Metakides, A. Nerode – *Principii de logică şi programare logică*, Editura Tehnică, Bucureşti, 1998
2. G. Georgescu – *Elemente de logică matematică*, Editura Academiei Tehnice Militare, Bucureşti, 1978
3. D. Busneag, D. Piciu, *Probleme de logica si teoria multimilor*, Craiova, 2003.
4. G. Georgescu, A. Iorgulescu, *Logica matematica*, Ed. ASE, Bucuresti, 2010
5. Gr. C Moisil, *Elemente de logica matematica si de teoria multimilor*, Ed. Stiintifica, Bucuresti, 1968
6. J.D. Monk, *Mathematical Logic*, Springer Verlag, 1976
7. V. E. Cazanescu, *Curs de bazele informaticii*, Tipografia Universitatii din Bucuresti, 1976
8. S. Rudeanu, *Curs de bazele informaticii*, Tipografia Universitatii din Bucuresti, 1982
9. M. Huth, M. Ryan*, Logic in Computer Science*: Modelling and Reasoning about Systems, Cambridge Univ. Press, 2009
10. A.R. Bradley, Z. Manna, *The Calculus of Computation Decision Procedures with Applications to Verification*, Springer,2007
11. M. Ben-Ari, *Mathematical Logic For Computer Science*, Springer, 2003

MULŢUMESC!

SEMINAR 7

Ex1: Exprimați următoarele propoziții ca mulțimi de clauze:

a) $\neg(A \wedge B \wedge \neg C)$

b) $A \leftrightarrow (\neg B \wedge \neg C)$

SEMINAR 7

Ex2: Care dintre următoarele mulțimi de clauze este verificabilă și de ce? Pentru fiecare mulțime verificabilă, stabiliți o valorizare de adevăr care o satisface.

a) $\{\{A, B\}, \{\neg A, \neg B\}, \{\neg A, B\}\}$

b) $\{\{\neg A\}, \{A, \neg B\}, \{B\}\}$

c) $\{\{A\}, \square\}$

d) $\{\square\}$

SEMINAR 7

Ex3: Stabiliți rezolventul R(S) al mulțimii S în următoarele cazuri:

a) $S=\{\{A, \neg B\}, \{A, B\}, \{\neg A\}\}$

b) $S=\{\{A\}, \{B\}, \{A, B\}\}$

SEMINAR 7

Ex4: Aplicând rezoluția, demonstrați că următoarele mulțimi nu sunt verificabile:

a) $S=\{\neg A \vee B \vee D, \neg B \vee D \vee A, \neg D \vee C, \neg D \vee A, A \vee B, B \vee \neg C, \neg A \vee \neg B\}$

b) $S=\{\neg A \vee B, \neg B \vee C, \neg C \vee A, A \vee C, \neg A \vee \neg C\}$

c) $S=\{A \vee B \vee C, A \vee B \vee \neg C, A \vee \neg B, \neg A \vee \neg C, \neg A \vee C\}$

SEMINAR 7

Ex5: Aplicând rezoluția, demonstrți că:

$\{A \wedge B \to C, A \to B\} \vdash_R A \to C$

SEMINAR 7

Ex6: Fiind dată propoziția Q următoare:

$(A \wedge B \to C) \wedge A \wedge (\neg B \to C)$

a) Stabiliți pentru Q forma potrivit teoriei mulțimilor.

b) Demonstrați prin rezoluție că $Q \vdash_R C$

c) Date fiind propozițiile Q și $A \to \neg C$, ce putem conchide asupra valorii de adevăr a lui B?

LUMEA WUMPUS

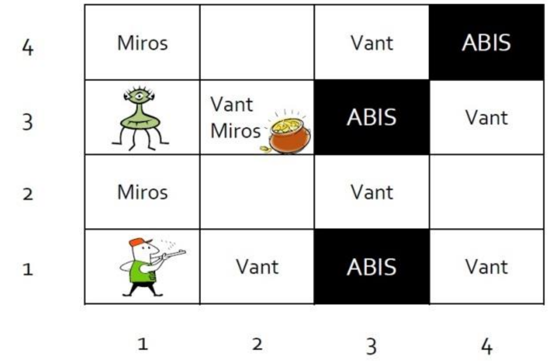
4	Miros		Vant	ABIS
3		Vant Miros 	ABIS	Vant
2	Miros		Vant	
1		Vant	ABIS	Vant
	1	2	3	4

LUMEA WUMPUS
MEDIUL ÎNCONJURĂTOR

4	Miros		Vant	ABIS
3		Vant Miros	ABIS	Vant
2	Miros		Vant	
1		Vant	ABIS	Vant
	1	2	3	4

- Tabela cu patratele (pestera) este inconjurata de ziduri.
- Agentul (exploratorul) porneste intotdeauna din coltul din stanga jos (1, 1).
- Patratele adiacente Wumpus-ului si patratul lui au miros (neplacut).
- Patratele adiacente abisurilor contin adieri de vant.
- Daca exploratorul se afla in patrat cu aur, acesta straluceste.
- Impuscatura il ucide pe Wumpus daca omul este indreptat catre el.
- Impuscatura se face in directia orientarii omului, iar glontul merge pana ucide wumpus-ul daca e pe directie sau cand ajunge la capatul liniei/coloanei.

LUMEA WUMPUS
MEDIUL ÎNCONJURĂTOR

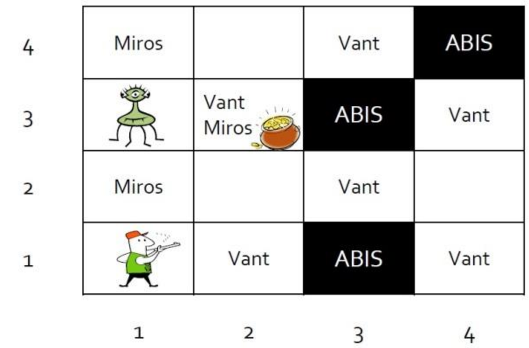


- Aurul se poate lua doar daca se afla in patratul cu aur.
- Cand agentul intra intr-un zid, simte o lovitura.
- Cand wumpus-ul este ucis, acesta scoate un strigat care se poate auzi in intreaga pestera.
- Perceptiile agentului de la mediul inconjurator vin in forma a 5 simboluri:
  - Daca simte miros, vant, stralucire, nu se loveste si nu aude tipat, lista va arata astfel:

[*Miros, Vant, Stralucire, Nimic, Nimic*]

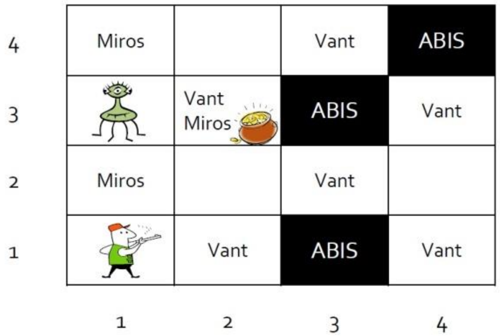
- Agentul poate merge inainte, se poate roti la stanga si la dreapta cu 90°.

LUMEA WUMPUS
MEDIUL ÎNCONJURĂTOR



- Agentul poate *lua* aurul daca se afla in patratul cu aur.
- Poate *trage* un (singur) foc in line dreapta.
- Poate *iesi* din pestera, dar numai pe la (1,1).
- Agentul moare daca intra intr-un patrat cu abis sau intr-unul cu ur wumpus in viata.
- Scopul agentului: sa gaseasca aurul si sa iese cu el din pestera.
- Castiguri si penalizari:
  - 1 000 puncte daca iese cu aurul din pestera
  - -1 punct pentru fiecare actiune facuta
  - -10 000 puncte daca moare.

LUMEA WUMPUS
MEDIUL ÎNCONJURĂTOR



- Agentul este mereu initializat la patratul (1,1) cu tata spre deapta.
- Locatiile pentru wumpus si pentru aur sunt alese aleator, fara pozitia (1,1).
- Fiecare locatie cu exceptia (1,1) are 20% sanse sa contina un abis.
  - Deci un abis poate fi in casuta cu aur sau in cea cu wumpus-ul.
- Desigur, pot fi situatii cand agentul nu poate ajunge la aur.
  - Viata este uneori nedreapta.
- Evident, agentul nu stie de la inceput ce se afla in fiecare patrat.
  - El doar poate simti prezenta vantului, a mirosului sau poate vedea stralucirea dintr-un patrat.

LUMEA WUMPUS

4	Miros		Vant	ABIS
3		Vant Miros 	ABIS	Vant
2	Miros		Vant	
1		Vant	ABIS	Vant
	1	2	3	4

- Agentul este plasat la (1,1).
  - Nu simte nici vant, nici miros.
  - Deduce ca (1,2) si (2,1) nu contin pericole.

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
1,1 A	2,1	3,1	4,1

A – Agent
Ab – Abis
Au – Aur
M – Miros
OK – Patrat OK
V – Vant
Viz - Vizitata
W – Wumpus

LUMEA WUMPUS

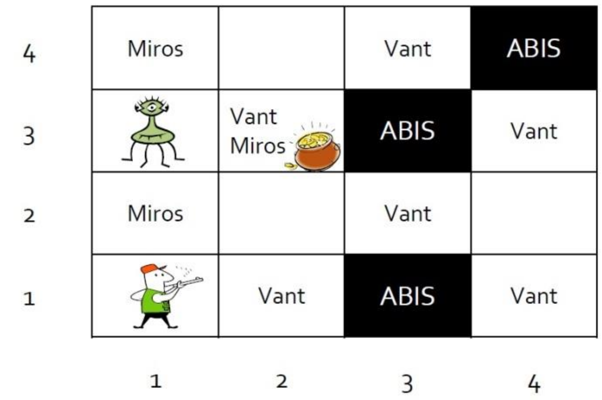
4	Miros		Vant	ABIS
3		Vant Miros 	ABIS	Vant
2	Miros		Vant	
1		Vant	ABIS	Vant
	1	2	3	4

- Agentul este plasat la (1,1).
  - Nu simte nici vant, nici miros.
  - Deduce ca (1,2) si (2,1) nu contin pericole.
  - Le marcheaza pe acestea cu OK.
  - De asemenea, (1,1) este OK.
- Un agent precaut se muta intr-un patrat numai daca este OK.

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2	3,2	4,2
1,1 A OK	2,1 OK	3,1	4,1

A – Agent
Ab – Abis
Au – Aur
M – Miros
OK – Patrat OK
V – Vant
Viz - Vizitata
W – Wumpus

LUMEA WUMPUS



- Presupunem ca agentul se muta la (2,1).
- Aici detecteaza vant.

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2	3,2	4,2
1,1 Viz OK	2,1 A V OK	3,1	4,1

A – Agent
Ab – Abis
Au – Aur
M – Miros
OK – Patrat OK
V – Vant
Viz - Vizitata
W – Wumpus

LUMEA WUMPUS

4	Miros		Vant	ABIS
3		Vant Miros 	ABIS	Vant
2	Miros		Vant	
1		Vant	ABIS	Vant
	1	2	3	4

- Presupunem ca agentul se muta la (2,1).
- Aici detecteaza vant.
  - In (1,1) nu poate fi abis pentru ca de acolo vine.
  - Deci este un abis la (2,2) sau la (3,1). (indicam cu Ab?)

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2 Ab?	3,2	4,2
1,1 Viz OK	2,1 A V OK	3,1 Ab?	4,1

A – Agent
Ab – Abis
Au – Aur
M – Miros
OK – Patrat OK
V – Vant
Viz - Vizitata
W – Wumpus

LUMEA WUMPUS

4	Miros		Vant	ABIS
3		Vant Miros 	ABIS	Vant
2	Miros		Vant	
1		Vant	ABIS	Vant
	1	2	3	4

- Mai este un singur patrat "OK" in care A nu a fost, (1,2).
- A se intoarce prin (1,1) si merge in (1,2).
- Detecteaza miros in (1,2), deci este un wumpus pe aproape.
  - Acesta nu e in (1,1) si nu poate fi nici in (2,2) pentru ca ar fi detectat miros cand se afla in (2,1).
- Din acest rationament reiese ca W este in (1,3).

1,4	2,4	3,4	4,4	A – Agent Ab – Abis Au – Aur M – Miros OK – Patrat OK V – Vant Viz - Vizitata W – Wumpus
1,3 W!	2,3	3,3	4,3	
1,2 A M OK	2,2 OK	3,2	4,2	
1,1 Viz OK	2,1 V Viz OK	3,1 Ab?	4,1	

LUMEA WUMPUS

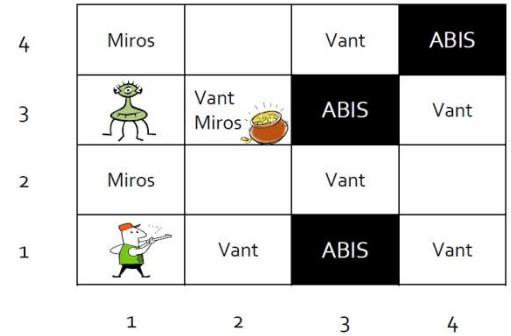
4	Miros		Vant	ABIS
3		Vant Miros 	ABIS	Vant
2	Miros		Vant	
1		Vant	ABIS	Vant
	1	2	3	4

- Lipsa vantului in (1,2) indica faptul ca nu este abis in (2,2) deci trebuie sa fie unul la (3,1).
- Aceasta inferenta se bazeaza pe cunostinte castigate in timpi diferiti si pe lipsa unei perceptii pentru a trage o concluzie.

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A M OK	2,2 OK	3,2	4,2
1,1 Viz OK	2,1 V Viz OK	3,1 Ab!	4,1

A – Agent
Ab – Abis
Au – Aur
M – Miros
OK – Patrat OK
V – Vant
Viz - Vizitata
W – Wumpus

LUMEA WUMPUS
și logica propozițională

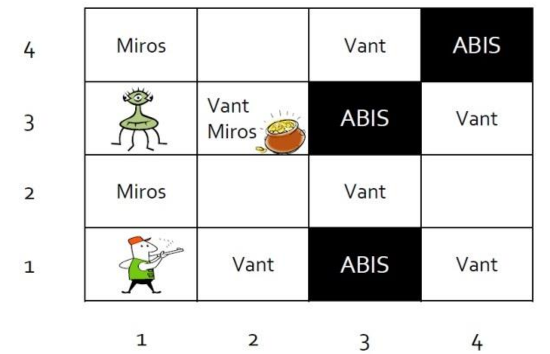


- Agentul se afla in situatia din figura alaturata, dar nu stie inca unde este W.
- La fiecare pas, perceptiile agentului se transforma in propozitii.
- Notatii:
  - M12 – in celula (1,2) exista miros
  - V21- in celula (2, 1) exista vant

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A M OK	2,2 OK	3,2	4,2
1,1 Viz OK	2,1 V Viz OK	3,1 Ab!	4,1

A – Agent
Ab – Abis
Au – Aur
M – Miros
OK – Patrat OK
V – Vant
Viz - Vizitata
W – Wumpus

LUMEA WUMPUS
și logica propozițională

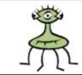


- Cunostintele adunate pana acum sunt urmatoarele:
  - $\neg M_{11}$
  - $\neg M_{21}$
  - M_{12}
  - $\neg V_{11}$
  - V_{21}
  - $\neg V_{12}$
- In plus, agentul stie ca daca nu simte miros intr-o celula, atunci acea celula si nicio alta celula adiacenta nu contin un wumpus.

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A M OK	2,2 OK	3,2	4,2
1,1 Viz OK	2,1 V Viz OK	3,1 Ab!	4,1

A – Agent
Ab – Abis
Au – Aur
M – Miros
OK – Patrat OK
V – Vant
Viz - Vizitata
W – Wumpus

LUMEA WUMPUS
și logica propozițională

4	Miros		Vant	ABIS
3		Vant Miros 	ABIS	Vant
2	Miros		Vant	
1		Vant	ABIS	Vant
	1	2	3	4

- Agentul trebuie sa stie acest lucru pentru fiecare celula din pestera.
- Ne reducem doar la ce a descoperit agentul pana la momentul curent:
- $\neg M_{11} \rightarrow \neg W_{11} \wedge \neg W_{12} \wedge \neg W_{21}$
- $\neg M_{21} \rightarrow \neg W_{11} \wedge \neg W_{21} \wedge \neg W_{22} \wedge \neg W_{31}$
- $M_{12} \rightarrow W_{11} \vee W_{12} \vee W_{22} \vee W_{13}$
- Poate deduce agentul W13! folosind aceste cunostinte si logica computationala?

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A M OK	2,2 OK	3,2	4,2
1,1 Viz OK	2,1 V Viz OK	3,1 Ab!	4,1

A – Agent
Ab – Abis
Au – Aur
M – Miros
OK – Patrat OK
V – Vant
Viz - Vizitata
W – Wumpus

LUMEA WUMPUS
și logica propozițională

4	Miros		Vant	ABIS
3		Vant Miros 	ABIS	Vant
2	Miros		Vant	
1		Vant	ABIS	Vant
	1	2	3	4

- Cunostintele dobandite:
  - $\neg M_{11}$
  - $\neg M_{21}$
  - M_{12}
  - $\neg M_{11} \rightarrow \neg W_{11} \wedge \neg W_{12} \wedge \neg W_{21}$
  - $\neg M_{21} \rightarrow \neg W_{11} \wedge \neg W_{21} \wedge \neg W_{22} \wedge \neg W_{31}$
  - $M_{12} \rightarrow W_{11} \vee W_{12} \vee W_{22} \vee W_{13}$

LUMEA WUMPUS
și logica propozițională

4	Miros		Vant	ABIS
3		Vant Miros 	ABIS	Vant
2	Miros		Vant	
1		Vant	ABIS	Vant
	1	2	3	4

- Avand starea din figura alaturata, sa se demonstreze ca abisul se afla la (3,1).

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A M OK	2,2 OK	3,2	4,2
1,1 Viz OK	2,1 V Viz OK	3,1 Ab!	4,1

A – Agent
Ab – Abis
Au – Aur
M – Miros
OK – Patrat OK
V – Vant
Viz - Vizitata
W – Wumpus

LUMEA WUMPUS
și logica propozițională

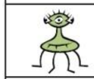
- Logica propozitionala poate fi folosita cu succes pentru inferente care sa ne descopere unde este wumpusul sau un abis.
- Pentru a folosi insa informatia, avem nevoie de reguli care sa ii spuna agentului cum sa se deplaseze.
- Daca wumpusul se afla chiar in fata, cel mai bine ar fi ca agentul sa nu se deplaseze chiar inainte…
  - Acest lucru se poate reprezenta prin o serie de reguli, una pentru fiecare locatie si orientare a agentului.

LUMEA WUMPUS
și logica propozițională

4	Miros		Vant	ABIS
3		Vant Miros 	ABIS	Vant
2	Miros		Vant	
1		Vant	ABIS	Vant
	1	2	3	4

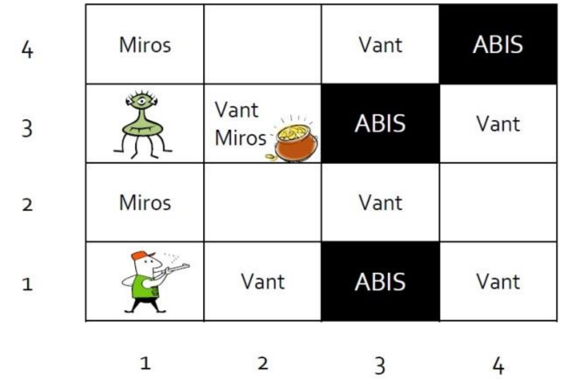
- Daca agentul se gaseste la (1, 1) cu fata spre est (dreapta se schimba in functe de orientarea agentului), o regula ar fi:
  - $A11 \land Est_A \land W21 \rightarrow \neg$ Inainte
- Pentru o lume de 4 x 4 cu (16 patrate si 4 orientari posibile), numai regula care spune sa nu mearga inainte daca este un wumpus in fata ar duce la crearea de 16 x 4 = 64 de reguli.
- Luand in calcul multitudinea de reguli care ar trebui adaugate, numai intr-o lume de 4x4 am ajunge la mii de reguli necesare pentru a realiza un agent competent.

LUMEA WUMPUS
și logica propozițională

4	Miros		Vant	ABIS
3		Vant Miros 	ABIS	Vant
2	Miros		Vant	
1		Vant	ABIS	Vant
	1	2	3	4

- Daca marim dimensiunea lumii, lucrurile se complica exponential.
- Plus ca lumea (pestera) se schimba odata cu trecerea timpului.
  - Nu vrem sa uitam de la un moment la altul ce a fost intr-un patrat, deci vom folosi notatii diferite pentru timpi diferiti:
  - $A^1 11 \wedge Est^1_A \wedge W^1 21 \rightarrow \neg Inainte^1$
  - $A^2 11 \wedge Est^2_A \wedge W^2 21 \rightarrow \neg Inainte^2 \ldots$
  - Asadar, regulile ar trebui rescrise pentru fiecare moment in timp.

LUMEA WUMPUS
și logica propozițională



- Daca agentul ar rula pentru 100 de pasi, am avea 6400 de reguli numai pentru a ii spune agentului sa nu mearga inainte daca wumpusul este acolo.
- Asadar, problema este ca logica propozitiilor utilizeaza pentru reprezentare o singura componenta: propozitia.
- In logica de ordinul I se pot reprezenta obiecte si relatii intre obiecte in plus fata de propozitii.
    - Cele 6400 de reguli vor fi reduse numai la 1.