

Prolog:Integrame

- Responsabili: Andrei Olaru, Teodora Argintaru, Andrei Popovici
- Data publicării: 08.05.2021
- Data ultimei modificări: 12.05.2021 ([changelog](#))
- Tema (o arhivă .zip cu fișierul integrame.pl) se va încărca pe vmchecker [<https://vmchecker.cs.pub.ro/ui/#PP>]
- Forum tema 3 [<https://curs.upb.ro/mod/forum/view.php?id=292431>]

Organizare

Tema constă într-o aplicație care rezolvă integrame.

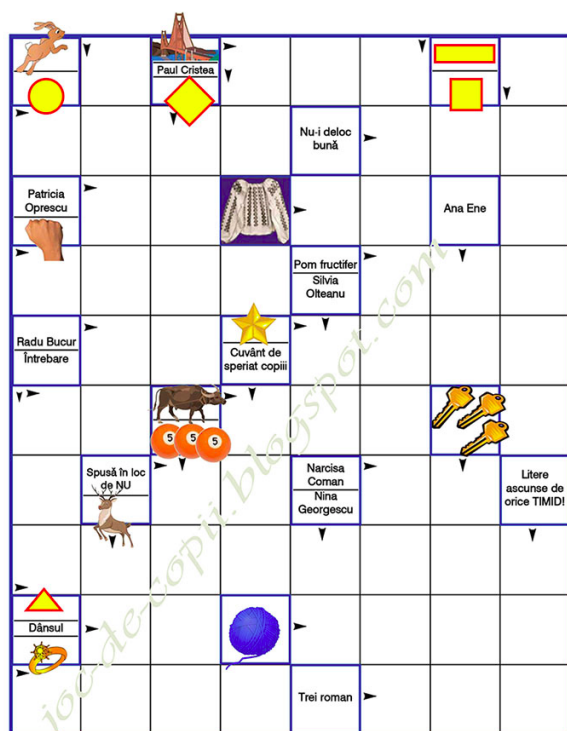
Tema este împărțită în **3 etape**, cu deadline-uri pentru toți studenții pe **19 mai**, **26 mai**, respectiv **2 iunie**. Ziua de 2 iunie este **deadline hard**.

Tema se va prezenta, în mod obligatoriu, în zilele de 3 și 4 iunie, după o programare care va fi anunțată.

Rezolvările tuturor etapelor pot fi trimise până în ziua deadline-ului hard, dar orice exercițiu trimis după deadline-ul soft se punctează cu **jumătate** din punctaj. Nota finală pentru o etapă se calculează conform formulei: $n = (n1 + n2) / 2$ ($n1$ = nota obținută înainte de deadline soft; $n2$ = nota obținută după deadline soft, înainte de deadline hard).

Descriere generală

O integramă este un joc de acest gen:



În problema noastră concretă, o integramă are doar întrebări în jos sau în dreapta, și arată astfel:

1	0↓, 1→	2↓	3↓	
4→	-	-	-	
5→	-	-	-	
1				
ID	DIRECTIE	INTREBARE		
0	j	Negativ		
1	d	Afirmativ		

2	j	Al doilea număr
3	j	Primii 3 din artă
4	d	Din care plouă
5	d	Când nu îmi amintesc

Iar o integramă rezolvată arată astfel:

4→	0↓,1→	2↓	3↓	
5→	N	D	A	
	U	O	R	
		I	T	
ID	DIRECTIE	INTREBARE		
0	j	Negativ		
1	d	Afirmativ		
2	j	Al doilea număr		
3	j	Primii 3 din artă		
4	d	Din care plouă		
5	d	Când nu îmi amintesc		

O integramă este definită ca o mulțime de celule, fiecare având un rând și o coloană, și care pot fi:

- celule negre, care nu pot fi completate
- celule cu întrebări, în care pot exista una sau două întrebări, fiecare întrebare fiind caracterizată prin
 - un text al întrebării
 - o direcție pentru răspuns (jos sau dreapta)
 - un identificator numeric
- celulele care nu sunt prezente în definiția integralei sunt libere.
- prin completare, se mai pot adăuga la definiția integralei celule completate, care conțin o literă mare (ca șir de caractere de 1 caracter).

Pe lângă definiția celulelor, integrama conține și un vocabular, o listă de cuvinte pe care le putem folosi pentru a completa integrama.

În temă, predicatul `integrama(0, X)` va lega variabila `X` la o structură, descriind integrama de mai sus, de forma:

```
X = integ(5,5,[
  ((0,0),x),          <- înălțimea și lățimea integralei, în număr de celule
  ((0,4),x),          <- celulă neagră, la coordonata 0,0
  ((1,4),x),
  ((2,4),x),
  ((3,4),x),
  ((4,4),x),
  ((0,1),x),
  ((1,1),[('Negativ',j,0),('Afirmativ',d,1)]), <- celulă cu două întrebări, una în jos și una la dreapta
  ((0,2),[('Al doilea număr',j,2)]),          <- celulă cu o singură întrebare, în jos
  ((0,3),[('Primii 3 din artă',j,3)]),
  ((1,0),x),
  ((2,0),[('Din care plouă',d,4)]),           <- celulă cu o singură întrebare, la dreapta
  ((3,0),[('Când nu îmi amintesc',d,5)]),
  ((4,0),x),
  ((4,1),x),
  ((4,2),x),
  ((4,3),x)],
  ['DA','NU','NOR','ART','UIT','DOI']         <- setul de cuvinte disponibile pentru a completa integrama
)
```

Puteți foarte bine "citi" elementele integralei prin pattern-match, inclusiv direct în predicatul `integrama`, ca `integrama(0, integ(H, W, Celule, Vocabular))`.

Ne interesează să rezolvăm integrama orbește, fără să ne uităm la legătura dintre textul întrebării și textul răspunsului. Vom încerca să potrivim cuvinte în integramă, considerând că:

1. lungimea cuvântului trebuie să fie aceeași cu dimensiunea spațiului disponibil pentru răspuns.
2. cuvintele care se intersectează trebuie să aibă aceeași literă în locurile unde se intersectează.

Acest lucru este foarte simplu în Prolog, pentru că prolog face backtracking pentru a căuta soluții care să respecte

anumite condiții.

Surse

Important: Citiți comentariile din surse (mai ales din `integrame.pl`). Acestea conțin toate detaliile despre predicatele cerute și despre structurile folosite.

Lucrați numai în fișierul `integrame.pl`. Acesta conține documentația tuturor predicatelor cerute.

Fișierul `input.pl` conține integralele folosite la testare. **Atenție:** integralele sunt definite aici într-un format oarecum diferit de cel folosit în predicate. Folosiți predicatul `integramă/2` pentru a obține integralele.

Fișierul `utils.pl` conține predicatele `integramă/2` care obține o integramă și `print_integramă/1` care afișează o integramă.

Fișierul `tester.pl` conține infrastructura generală de testare, iar `checker.pl` conține testele propriu-zise pentru temă. Formatul acestora este destul de obscur, dar puteți întreba pe forum ce ar trebui să facă un anumit test, dacă nu înțelegeți din output.

Fișierul `solutie.pl` conține soluțiile la integralele din `input.pl`. **Nu** folosiți predicatul `solutie` în rezolvare.

Notă: va fi necesar să converțiți între un literal/șir de caractere și lista caracterelor componente ale literalului. Pentru aceasta, folosiți predicatul `atom_chars/2`, care permite conversia în ambele sensuri (documentație [https://www.swi-prolog.org/pldoc/doc_for?object=atom_chars/2]).

De exemplu:

```
?- atom_chars('hel lo', L).
L = [h, e, l, ' ', l, o].

?- atom_chars(S, ['h', 'e', 'l', 'l', 'o', ' ', 'w', 'o', 'r', 'l', 'd']).
S = 'hello world'.
```

Testare

Pentru lucru, puteți încărca în Prolog direct fișierul `integrame.pl`.

Pentru testare, folosiți interogarea `vmtest`. în consola Prolog.

Notă: Prolog permite testarea faptului că anumite interogări sunt adevărate, dar și că anumite interogări sunt false, sau se obțin pentru ele zero soluții. Testerul face inițial o verificare și dacă niciunul dintre testele pozitive nu reușește, va afișa direct “Nerezolvat” pentru întregul set respectiv de teste. Astfel: **dacă ați implementat un anumit predicat, dar testerul vă afișează “Nerezolvat”,** decompați linia `%test_mode(detailed)` de la începutul fișierului `integrame.pl` (sau adăugați `test_mode(detailed)` în sursa voastră). Pe `vmchecker`, modul `detailed` **nu** este activat.

Etapa 1

Pentru a vă obișnui cu elementele temei, în această etapă veți implementa predicatele:

- `intrebări/2`, care obține informații despre întrebările prezente într-o integramă
- `id_intrebare/3`, care exprimă relația dintre textul și identificatorul unei întrebări (foarte util pentru conversia între cele două).
- `completare/3`, care adaugă într-o integramă răspunsuri la întrebări.
- Bonus pentru cazul în care predicatele au o singură soluție.

Etapa 2

În această etapă veți implementa două predicate importante pentru rezolvarea integralei:

- `lungime_spatiu/3`, care determină lungimea spațiului disponibil pentru răspunsul la o întrebare
- `intersectie/5`, care determină caracteristicile intersecției între răspunsurile la două întrebări, prin calculul poziției (indexului) la care fiecare răspuns se intersectează cu celălalt.
- Bonus pentru controlul numărului soluțiilor, și pentru obținerea, prin apelul cu variabile nelegate, ca soluții separate, a tuturor spațiilor, respectiv a tuturor intersecțiilor din integramă.

Etapa 3

În această etapă veți implementa rezolvarea propriu-zisă a integralei. O soluție a integralei este una în care pentru fiecare întrebare este asociat un cuvânt (un cuvânt din vocabular poate fi asociat cel mult unei întrebări) de lungimea potrivită, și care la intersecțiile cu oricare alt spațiu are aceeași literă cu cuvântul asociat cu întrebarea corespunzătoare acelui spațiu.

Trebuie implementate predicatele:

- `solutii_posibile/2`, care determină toate cuvintele (ca liste de litere) care pot fi soluție pentru fiecare întrebare.
- `rezolvare/2`, care determină în al doilea argument o soluție pentru integramă.
 - o soluție este reprezentată ca o listă de perechi întrebare-răspuns, unde atât întrebarea (identificată prin textul ei) cât și răspunsul sunt literali/atomii, ca în fișierul `solutie.pl`.
 - ca bonus poate fi considerată oricare dintre următoarele două proprietăți (cealaltă va fi parte din punctajul nominal):
 - tema funcționează într-un timp acceptabil (5 secunde) și pentru cazuri în care vocabularul este mai extins;
 - predicatul `rezolvare` produce un număr corect de soluții (2 soluții pentru integrama 0 și câte o singură soluție pentru celelalte integrale)
 - (adică, dacă implementați doar una dintre cele două proprietăți, aveți punctaj nominal; dacă implementați pe ambele, aveți și bonus)

Punctaje

- pentru etapa 1 puteți acumula maxim 3.6 puncte
- pentru etapa 2 puteți acumula maxim 3.6 puncte
- pentru etapa 3 puteți acumula maxim 4.8 puncte
- în total puteți acumula maxim 12 puncte.

Observații

- Lucrați numai în fișierul `integrame.pl`
- Nu utilizați predicatul `solutie`
- Nu lăsați variabile singleton în `rezolvare`.

Resurse

- schelet de cod
- schelet de cod pentru etapa 2
- schelet de cod pentru etapa 3

Notă: practic, diferența între fișiere este doar fișierul `pointsN.pl` utilizat, pentru a activa testele pentru diverse etape.

Changelog

- 8.05 - Publicare etapele 1 și 2.
- 12.05 (22:30)
 - publicare vmchecker etapa 1
 - modificare structură testare, fără efect asupra implementării temei (fișierul `pointsN.pl` corespunzător fiecărei etape).
 - includere detalii despre modul de testare
 - includere detalii despre modul de testare "*detaliat*"
 - clarificare legată de reprezentarea elementelor din integramă ca literali, nu ca stringuri, și utilizarea predicatului `atom_chars` (în loc de `string_chars`)
 - adăugare în enunț a detalierii structurii unei integrame
- 18.05 (20:00)
 - publicare vmchecker pentru etapele 2 și 3, publicare etapa 3
 - modificare test `completare|d` pentru a verifica corect și cazurile în care `completare` întoarce un număr mare de soluții.
 - corectare mențiuni despre *stringuri* în fișierul `integrame.pl`
 - adăugare documentare corectă (în termeni de "tipuri" ale argumentelor) pentru bonus pentru predicatele `lungime` și `intersectie`.
- 23.05
 - modificare structură punctaje astfel încât la fiecare etapă se primesc doar punctele de pe etapa respectivă (vedeți [Punctaje](#))
 - fix bug în care era posibil ca modul de test detaliat să fie activat pe vmchecker

pp/21/teme/prolog-integrame.txt · Last modified: 2021/05/23 17:25 by andrei.olaru