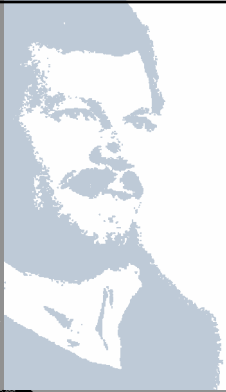



L
I
P
6

C
N
R
S



Reformulation et changement de représentation

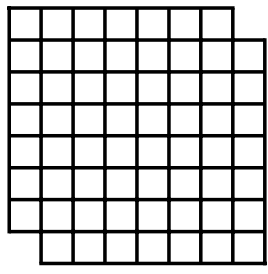
Jean-Gabriel Ganascia

 Jean-Gabriel Ganascia - Université Pierre et Marie CURIE 1


L
I
P
6


C
N
R
S

L'échiquier tronqué



Est-il possible de recouvrir tout l'échiquier avec des dominos ?

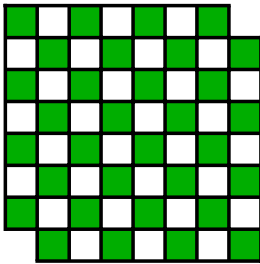


 Jean-Gabriel Ganascia - Université Pierre et Marie CURIE 2


L
I
P
6

C
N
R
S


L'échiquier tronqué: solution classique



Colorer l'échiquier



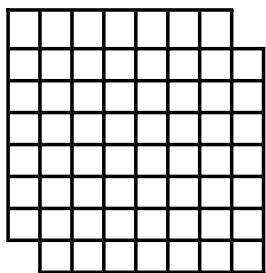
Le nombre de carrés verts étant différent du nombre de carrés blancs, il n'y a pas de solution...

 Jean-Gabriel Ganascia - Université Pierre et Marie CURIE 3


L
I
P
6

C
N
R
S

Existe-t-il des solutions sans coloration ?



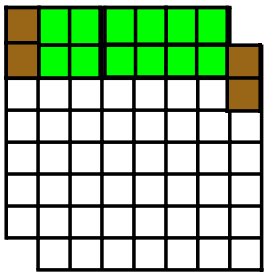
John McCarthy
Shmuel Winograd
Marvin Minsky
Dimitri Stefanyuk
...

 Jean-Gabriel Ganascia - Université Pierre et Marie CURIE 4

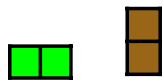
L
I
P
6


C
N
R
S

Solution sans coloration: Shmuel Winograd



Le nombre de domino se projetant **verticalement** d'une ligne à une autre est toujours impair...




 Jean-Gabriel Ganascia - Université Pierre et Marie CURIE 5

L
I
P
6

C
N
R
S

Solution sans coloration: Shmuel Winograd 2

- i. Le nombre de domino se projetant **verticalement** d'une ligne à la suivante est toujours impair.
- ii. Il y a sept lignes à prendre en considération
- iii. Le nombre total de dominos **verticaux** est donc impair (somme d'un nombre impair de nombres impairs)
- iv. Par symétrie, le nombre total de dominos **horizontaux** est impair
- v. Le nombre total de dominos est donc pair ce qui est faux puisqu'il y a 62 cases sur l'échiquier tronqué...

 Jean-Gabriel Ganascia - Université Pierre et Marie CURIE 6

Solution sans coloration: Marvin Minsky

Il y a 3 projections de d7 sur d8

Jean-Gabriel Ganascia - Université Pierre et Marie CURIE 7

Solution sans coloration: Minsky Résumé

- Il y a 2 projections de d2 sur d3
- Il y a 1 projection de d3 sur d4
- Il y a 3 projections de d4 sur d5
- Il y a 2 projections de d5 sur d6
- Il y a 4 projections de d6 sur d7
- Il y a 3 projections de d7 sur d8,
- Par symétrie, il y a 3 projection de d-7 sur d8, ce qui laisse deux cases non couvertes

Jean-Gabriel Ganascia - Université Pierre et Marie CURIE 8

Programmation avec des ASP – Answer Set Programming

Un programme Π est un ensemble d'expression ρ

$$\rho : L_0 \text{ or } L_1 \text{ or } \dots L_k \leftarrow L_{k+1}, L_{k+2}, \dots, L_m, \text{not } L_{m+1}, \dots, \text{not } L_n$$

Où

- les L_i sont des littéraux (atomes ou négation d'atomes)
- Le « not » est une négation par échec

Signification intuitive: pour toute interprétation de Herbrand qui rend vraie $\{L_{k+1}, L_{k+2}, \dots, L_m\}$ sans satisfaire $\{L_{m+1}, \dots, L_n\}$ on peut dériver $\{L_0, L_1, \dots, L_k\}$

Jean-Gabriel Ganascia - Université Pierre et Marie CURIE 9

ASP et reformulation

Un programme Π est un ensemble d'expression ρ

$$\rho : L_0 \text{ or } L_1 \text{ or } \dots L_k \leftarrow L_{k+1}, L_{k+2}, \dots, L_m, \text{not } L_{m+1}, \dots, \text{not } L_n$$

Exécution: génération de tous les modèles minimaux qui satisfont les d'expressions ρ d'un programme Π

Deux phases:

1. Reformulation
 - LPARSE: calcule toutes les instanciations de programmes – $\text{ground}(\Pi, \mathcal{L})$
2. Calcul des modèles minimaux
 - S MODELS: construit les ensembles réponses à partir des sorties de LPARSE

Jean-Gabriel Ganascia - Université Pierre et Marie CURIE 10

Programmation avec les « Answer Set Programming »

```

index(1..8).

#domain index(I,J: II: JJ).

case(I, J) :- not tronquee(I, J).
tronquee(1, 8).
tronquee(8, 1).

% Possibilité de recouvrement d'une case par un domino
domino(I, J, I, I+1)|domino(I, J, I+1, J)|domino(I-1, J, I, J)|domino(I, J-1, I, J) :-
  case(I, J).

% Contraintes: il ne doit y avoir de domino que sur des cases
:- domino(I, J, I+1, J), not case(I, J).
:- domino(I, J, I+1, J), not case(I+1, J).
:- domino(I, J, I, I+1), not case(I, J).
:- domino(I, J, I, I+1), not case(I, J+1).
:- domino(I-1, J, I, J), not case(I-1, J).
:- domino(I, J-1, I, J), not case(I, J-1).

% absence de recouvrement
:- domino(I, J, I+1, J), domino(I, J, I, J+1).
:- domino(I, J+1, I+1, J+1), domino(I, J, I, J+1).
:- domino(I, J, I+1, J), domino(I+1, J, I+2, J).
:- domino(I, J, I, I+1), domino(I, J+1, I, J+2).
  
```

Règle

Contraintes

Jean-Gabriel Ganascia - Université Pierre et Marie CURIE 11

Jean-Gabriel Ganascia - Université Pierre et Marie CURIE 12

Programmation avec les « Answer Set Programming »

```

index(1..8).

#domain index(I;J; II: JJ).

case(I, J) :- not tronquee(I, J).
#tronquee(1, 8).
#tronquee(8, 1).

% Possibilité de recouvrement d'une case par un domino
domino(I, J, I, J+1)|domino(I, J, I+1, J)|domino(I-1, J, I, J)|domino(I, J-1, I, J) :-
case(I, J).

% Contraintes: il ne doit y avoir de domino que sur des cases
:- domino(I, J, I+1, J), not case(I, J).
:- domino(I, J, I, J+1), not case(I, J).
:- domino(I, J, I, J+1), not case(I, J+1).
:- domino(I-1, J, I, J), not case(I-1, J).
:- domino(I, J-1, I, J), not case(I, J-1).

% absence de recouvrement
:- domino(I, J, I+1, J), domino(I, J, I, J+1).
:- domino(I, J+1, I+1, J+1), domino(I, J, I, J+1).
:- domino(I, J, I+1, J), domino(I+1, J, I+2, J).
:- domino(I, J, I, J+1), domino(I, J+1, I, J+2).
    
```

The screenshot shows the 'Answer Set Program Window' with the code from slide 13. Below the code, there are buttons for 'Load Program', 'Save Program', 'Syntax Check', 'Ignore Options', 'Simplify Options', and 'Execute'. The 'Support program Log Window' is open, displaying the following statistics:

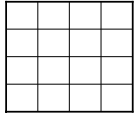
```

AnswerSet Program Log Window
#domains: 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100
#atoms: 4881
#rules: 2058
#clauses: 4438
#literals: 112
    
```

Un problème traité par JLL

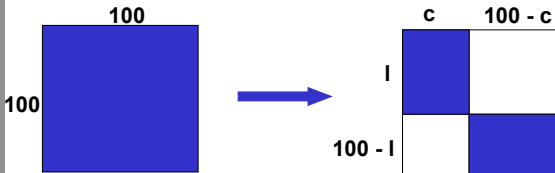
Énoncé: "Un damier de côté 100 est rempli de 10 000 pions noirs. Un coup légal, dans le jeu, consiste à changer, sur une même rangée (ligne ou colonne) tous les pions noirs en pions blancs et tous les pions blancs en pions noirs. Est-il possible d'obtenir 1990 pions blancs sur ce damier ? "

Comment « reformuler le problème » Procéder sur un damier abstrait ou réduit (4x4)



- Remarques :
- 2 coups de suite sur une même rangée s'annulent
 - L'ordre des coups est indifférent
 - deux lignes ou deux colonnes (évident)
 - une ligne L et une colonne C (essayer L puis C et C puis L)
 - Indifférence du nom de la rangée: seul le nombre de pions blancs est important
 - Il suffit de s'intéresser à la parité du nombre de retournements de chaque rangées:
 - ordre indifférent
 - annulation de deux coups
 2 types de rangées, lignes ou colonnes
 - En vertu de 3, on peut réarranger les rangées
 - Seules comptent les l lignes et les c colonnes retournées un nombre impair de fois.

Reformulation du problème



Nouvel énoncé:
trouver l et c tels que
 $l \in [0, +100]$,
 $c \in [0, +100]$
 $(100 - l).c + l.(100 - c) = 1990$ (nbre blancs)

Programmation en ASP

Énoncé: trouver l et c tels que
 $l \in [0, +100]$,
 $c \in [0, +100]$
 $(100 - l).c + l.(100 - c) = 1990$ (nbre blancs)

```

#domain entiers(I;J;K;L).
#domain produits(P).
entiers(0..100).
produits(0..10000).
blancs((100-I)*J+(100-J)*I) :- lignes(I), colonnes(J).
:- blancs(P), neq(P, 1990).
1{lignes(X):entiers(X)}1.
1{colonnes(X):entiers(X)}1.
    
```


Attribute	Type	Domain	Liste des attributs avec leurs caractéristiques
year	integer	ℕ	
location	string	NA	
temperature	ordered set	severe-cold < cold < average < hot < very-hot	
humidity	ordered set	low < high < very-high	
food-quantity	ordered set	starvation<severe-restrictions<restrictions<OK	
food-variety	ordered set	low < average < high	
hygiene	ordered set	very-bad < bad < average < good < very-good	
type-of-location	unordered set	land, sea	
fresh_fruits/vegetables	Boolean	yes, no	
disease-severity	integer	(0,1,2,...,5)	

Représentation

Un exemple

disease-severity = 0
 fresh_fruits/vegetables = yes
 food-variety = low
 food-quantity >= ok
 type-of-location = land
 location = California
 year < 1604
 year > 1602
 hygiene = average

Utilisation de connaissances

food-quantity > restrictions
 food-quantity > severe-restrictions
 food-quantity > starvation

hygiene > bad
 hygiene > very-bad
 hygiene < good
 hygiene < very-good

disease-severity < 1
 disease-severity < 2
 disease-severity < 3
 disease-severity < 4
 disease-severity < 5
 food-variety = low
 food-variety < average
 food-variety < high

12 Règles engendrées

Set I: Rules 3,4,8 use in their premises the variety of the diet.

R3: IF diet-variety ≥ high THEN disease-severity ≤ 0. [5]
 R4: IF diet-variety ≤ average THEN disease-severity ≥ 2. [4]
 R8: IF diet-variety ≥ average THEN disease-severity ≤ 2. [11]

Set II: Rules 7, 10 use in their premises the presence (or absence) of fresh fruits and vegetables in the diet.

R7: IF fresh_fruits/vegetables = no THEN disease-severity ≥ 2. [5]
 R10: IF fresh_fruits/vegetables = yes THEN disease-severity ≤ 2. [13]

Set III: Rule 2 uses in its premises the quantity of food available.

R2: IF food-quantity ≥ ok THEN disease-severity ≤ 0. [4]

Set IV: Rules 5,6,9,12 use in their premises the level of hygiene.

R5: IF hygiene ≤ bad THEN disease-severity ≥ 3. [3]
 R6: IF hygiene ≤ average THEN disease-severity ≥ 2. [4]
 R9: IF hygiene ≥ average THEN disease-severity ≤ 2. [7]
 R12: IF hygiene ≥ good THEN disease-severity ≤ 1. [6]

Set V: Rules 1, 11 use in their premises the temperature.

R1: IF location = land, temperature ≥ hot THEN disease-severity ≤ 0. [4]
 R11: IF temperature ≤ severe-cold THEN disease-severity ≥ 1. [5]

Comparaison des explications

DIET (food variety):
It was J.F. Bachström (1734) who first expressed the opinion that, 'Abstinence of vegetables is the only, the true, the first cause of scurvy.'

Set I: Rules 3,4,8 use in their premises the variety of the diet.

R3: IF diet-variety ≥ high THEN disease-severity ≤ 0. [5]
 R4: IF diet-variety ≤ average THEN disease-severity ≥ 2. [4]
 R8: IF diet-variety ≥ average THEN disease-severity ≤ 2. [11]

Set II: Rules 7,10 use in their premises the presence (or absence) of fresh fruits and vegetables in the diet.

R7: IF fresh_fruits/vegetables = no THEN disease-severity ≥ 2. [5]
 R10: IF fresh_fruits/vegetables = yes THEN disease-severity ≤ 2. [13]

DIET (food quantity):
We are lead to conclude that a decrease in quantity of food, or to speak clearly, starvation, can occasionally serve the cause of scurvy, but it cannot produce it by itself.

Set IV: Rule 2 uses in its premises the quantity of food available.

R2: IF food-quantity ≥ ok THEN disease-severity ≤ 0. [4]

Comparaison des explications

HYGIENE:
If Cook's crews were entirely spared from scurvy, in a relatively large extent considering the times, it is thought that these great results were precisely the happy consequence of the care given to the cleanliness and drying of the ships.

Set II: Rules 5,6,9,12 use in their premises the level of hygiene.

R5: IF hygiene ≤ bad THEN disease-severity ≥ 3. [3]
 R6: IF hygiene ≤ average THEN disease-severity ≥ 2. [4]
 R9: IF hygiene ≥ average THEN disease-severity ≤ 2. [7]
 R12: IF hygiene ≥ good THEN disease-severity ≤ 1. [6]

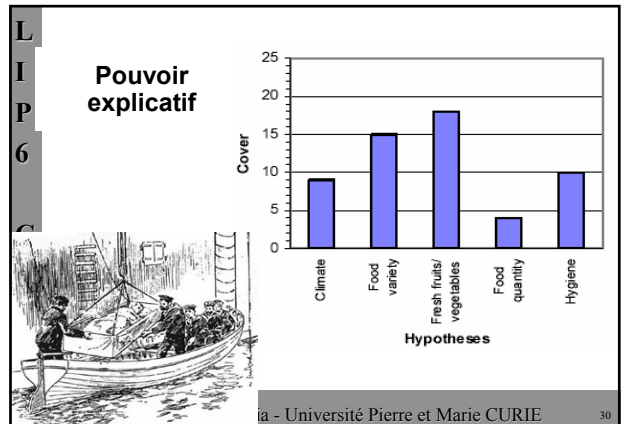
CLIMATE:
Spring and winter are obviously the seasons of predominance for scurvy.

Set III: Rules 1, 11 use in their premises the temperature.

R1: IF location = land, temperature ≥ hot THEN disease-severity ≤ 0. [4]
 R11: IF temperature ≤ severe-cold THEN disease-severity ≥ 1. [5]

Introduction de la notion de pouvoir explicatif

- Pour chaque attribut, on cherche les règles dans lesquelles il apparaît.
- Le pouvoir explicatif est lié au nombre de faits expliqués.

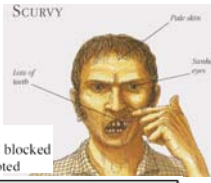


Le problème de l'humidité...

"The influence of a cold and humid atmosphere has been said to be the key factor for the apparition of scurvy. "Air humidity is the main predisposing cause of this disease", according to Lind."

Ajout de la théorie de la "respiration bloquée"

- Gallen 131-201, Sanctorius 17th century...
- Corps: Intérieur and extérieur
- Intérieur: fluides (corrompus, sains), tissus
- Extérieur: air, nourriture, excrétiens (urine, transpiration...)



Attribute	Type	Domain
perspiration	Ordered set	normal < hard < blocked
fluids	Ordered set	healthy < corrupted

IF (humidity = high)	THEN (perspiration ≥ hard)
IF (hygiene ≥ good) (humidity ≤ high)	THEN (perspiration ≤ hard)
IF (humidity ≥ very-high)	THEN (perspiration ≥ blocked)
IF (perspiration ≤ hard)	THEN (fluids ≤ healthy)
IF (fresh_fruits/vegetables = yes)	THEN (fluids ≤ healthy)
IF (fresh_fruits/vegetables > yes)	(perspiration ≥ blocked)
IF (perspiration ≥ blocked)	THEN (fluids ≥ corrupted)
IF (hygiene ≤ average)(location = sea)	THEN (humidity ≥ very-high)
IF (hygiene ≥ good)	THEN (humidity ≤ high)

31

Règles traduisant la théorie de la transpiration bloquée

SI (humidité = élevée) ALORS (transpiration ≥ difficile)
 SI (humidité ≥ très-élevée) ALORS (transpiration ≥ bloquée)
 SI (hygiène ≥ bonne)
 (humidité ≤ élevée) ALORS (transpiration ≤ difficile)
 SI (transpiration ≤ difficile) ALORS (fluides ≤ sain)
 SI (fruits_végétaux = oui) ALORS (fluides ≤ sain)
 SI (fruits_végétaux ≠ oui)
 (transpiration > bloquée) ALORS (fluides > corrompus)
 SI (hygiène ≤ moyen) (lieu = mer) ALORS (humidité > très-élevée)
 SI (hygiène > bonne) ALORS (humidité ≤ élevée)

Jean-Gabriel Ganascia - Université Pierre et Marie CURIE 32

Nouvelles règles obtenues par apprentissage

FONCTIONS INTERNES => SYMPTOMES

SI fluides = corrompus ALORS gravité_maladie >= 2 [9]
 SI fluides = sains ALORS gravité_maladie <= 2 [14]

ENVIRONNEMENT => SYMPTOMES

SI humidité <= élevée
 hygiène >= moyen ALORS gravité_maladie <= 1 [6]

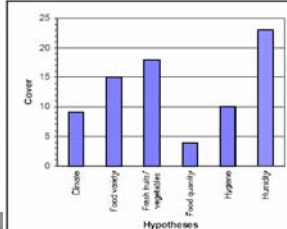
Jean-Gabriel Ganascia - Université Pierre et Marie CURIE 33

Nouvelles règles

Induction avec la théorie de la "respiration bloquée"

IF humidity ≥ high	THEN disease-severity ≥ 2.	[4]
IF fresh_fruits/vegetables = unknown,	THEN disease-severity ≤ 1.	[6]
IF humidity ≤ high, hygiène ≥ average	THEN disease-severity ≤ 1.	[6]
IF perspiration ≤ hard	THEN disease-severity ≤ 1.	[6]
IF fluids ≥ corrupted	THEN disease-severity ≥ 2.	[9]
IF fluids ≤ healthy	THEN disease-severity ≤ 2.	[14]

Pouvoir explicatif



Jean-Gabriel Ganascia 34