

TP4

Système Expert

Le système expert est un mécanisme logiciel qui permet de mettre en place une forme d'intelligence basée sur l'utilisation de règles soumises par un expert. Le système expert est une application logicielle (IA), alors que l'expert est un humain capable de synthétiser (dans son domaine de compétence) une tâche, une action, une décision, par le biais de règles.

Dans ce travail, l'objectif n'est pas de créer une application iOS (ce travail sera repris pour concevoir la version iOS). Vous allez concevoir un système expert dans une application console OSX. Les classes auront une hiérarchie, et les résultats s'afficheront dans la console. L'intérêt est de manipuler le langage Swift à travers la POO, l'héritage, les protocoles, et le stockage des objets. Ce travail se base sur la traduction d'un code C# qui correspond à un système expert avec moteur d'inférence à chaînage avant.

L'intérêt de traduire du C# en Swift permet un apprentissage basé sur un existant, et sur des concepts déjà connus dans un autre langage. C'est un apprentissage qui permet de manipuler des concepts avancés en ayant une progression bien identifiée.

Ce système expert sera générique, il pourra s'appliquer à des problèmes variés de classification.

La classification qui sera utilisée dans cet exercice sera celle de figures géométriques d'ordre 3, 4, 5. L'ordre correspond au nombre de côtés de la figure. Une figure d'ordre 3 peut caractériser un triangle, l'ordre 4 correspond à un parallélogramme, alors que l'ordre 5 concerne le pentagone.

1. Récupération du code source

Le code source est déjà disponible dans le document associé à ce travail. Le code C# donne l'intégralité du projet. Il est nécessaire de le traduire en Swift ce qui constitue un bon travail d'apprentissage. Il est nécessaire de faire un travail de mise au point dans certains cas, puisque les codes Swift et C# diffèrent parfois sur certains points. La gestion de « nil » et « null » amène à réfléchir à l'écriture d'une solution fonctionnelle dans tous les cas.

Pour gérer « nil » en Swift il faut faire appel aux types optionnels (optional -> ?), qui permettent de débiller (unwrap) des variables même avec un contenu nul (nil) qui finalement est une valeur en soi.

2. Traduction du code

Le langage Swift est soumis à des évolutions régulières. La version actuelle 2.0 tend à passer en 2.1, sachant qu'une version 3.0 est à l'étude.

Les éléments de la version 1.2 ne sont plus acceptés avec Xcode 7.x. Pour cela des recherches doivent être faites dans les documentations à jour. Aucun livre ou support

actuel ne peut faire la synthèse de l'état de l'art de ce langage. C'est en langage en évolution depuis juin 2014.

3. Mise au point et vocabulaire

Le code source initial C# présente les règles de la base de connaissance avec une syntaxe que nous allons remanier et expliquer.

3.1. Syntaxe des règles

Une première modification a été faite pour simplifier la recherche de la question associée à une prémisse. Les « () » qui encadrent les questions ont été changées en « [] ». Les « () » qui encadrent la règle subsistent.

R2 : IF (triangle AND angle droit[la figure a-t-elle au moins un angle droit ?]) THEN triangle rectangle

Il faut éviter de placer des majuscules dans l'écriture des règles de la base. Les majuscules sont réservées aux mots clés : IF, THEN, AND

Le caractère « ? » marque toujours la fin d'une question associée à une prémisse.

Chaque règle se décompose de la manière suivante :

R1 : IF (ordre=3[quel est l'ordre ?]) THEN triangle

- R1 : nom de la règle
- Ordre=3 : prémisse
- Quel est l'ordre ? : question associée à la prémisse
- Triangle : conclusion

R2 : IF (triangle AND angle droit[la figure a-t-elle au moins un angle droit ?]) THEN triangle rectangle

- R2 : nom de la règle
- Triangle : première prémisse
- Angle droit : seconde prémisse
- La figure a-t-elle un angle droit ? : question associée à la prémisse
- Triangle rectangle : conclusion

Le nombre de règles (rules) dépend de la base de connaissance (base de règles) que vous souhaitez intégrer dans votre programme. C'est à l'expert (vous) d'écrire les nouvelles règles pour gérer le pentagone.

3.2. Les prémisses

Les prémisses permettent de déterminer une conclusion.

Les prémisses peuvent être de deux types :

- Entier : intFact
- Booléen : boolFact

3.2.1. Prémisses entières (intFact)

“ordre=3[quel est l'ordre ?]”

- ordre : nom (name)
- 3 : valeur (value)
- quel est l'ordre . : question (question)

“côtés égaux=2[combien la figure a-t-elle de côtés égaux ?]”

- côtés égaux : nom (name)
- 2 : valeur (value)
- combien la figure a-t-elle de côtés égaux ? : question (question)

Une prémisses entières est toujours associée à une question

3.2.2. Prémisses booléennes (boolFact)

“triangle”

triangle : nom

vrai : valeur (value)

“!triangle rectangle”

triangle rectangle : nom

faux (!) : valeur (value)

Une prémisses booléenne n'est jamais associée à une question, elle est inférée par les faits.

3.3. Les conclusions

Les conclusions sont issues de l'application des prémisses. Dans un chaînage avant, on va rechercher la prémisses qui correspond à la conclusion en cours.

Ordre=3 c'est un Triangle, la conclusion est Triangle, cette conclusion est ajoutée dans la base de faits.

Je recherche la première prémisses Triangle dans une règle, sachant que la règle précédente déjà appliquée est effacée de la base de règles.

Les conclusions constituent les faits inférés ou entrées (si elles issues d'une entrée utilisateur), enregistrés dans la base de faits.

3.4. Niveaux des prémisses et des conclusions

Les prémisses et les conclusions ont un niveau (level).

Les prémisses sont toujours de niveau 0 :

- "ordre=3[quel est l'ordre ?]" : niveau 0 (level 0)
- "côtés égaux=2[combien la figure a-t-elle de côtés égaux ?]" : niveau 0 (level 0)

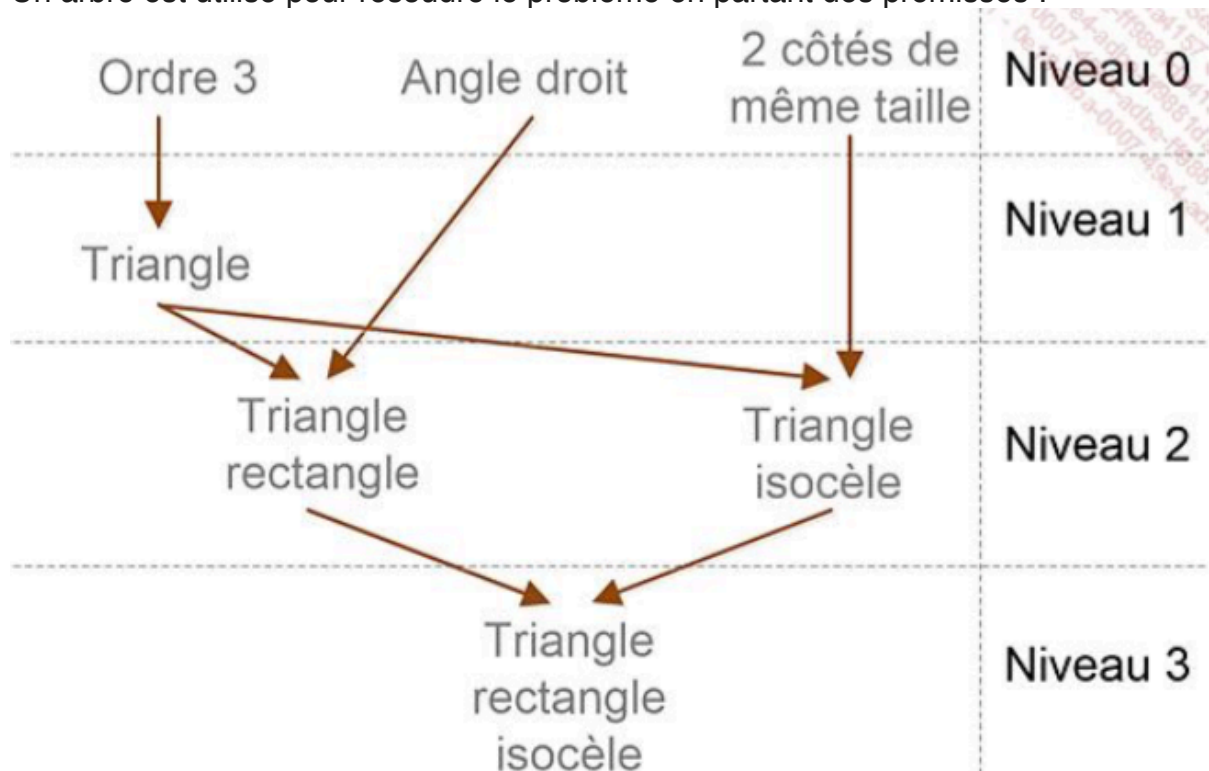
Les conclusions ont toujours un niveau > 0 :

- Triangle -> niveau 1
- Triangle rectangle -> niveau 2
- Triangle isocèle -> niveau 2
- Triangle rectangle isocèle -> niveau 3

Chaque prémisses et conclusion stockée dans les bases sont uniques.

"côtés égaux=3" n'est pas la même chose que "côtés égaux=2" en effet les valeurs (value) sont différentes (2 différent de 3)

Un arbre est utilisé pour résoudre le problème en partant des prémisses :



4. Liste des règles introduites

```
m.AddRule("R1 : IF (ordre=3[quel est l'ordre ?]) THEN triangle");
m.AddRule("R2 : IF (triangle AND angle droit[la figure a-t-elle au moins un angle droit ?]) THEN triangle rectangle");
m.AddRule("R3 : IF (triangle AND côtés égaux=2[combien la figure a-t-elle de côtés égaux ?]) THEN triangle isocèle");
m.AddRule("R4 : IF (triangle rectangle AND triangle isocèle) THEN triangle rectangle isocèle");
m.AddRule("R5 : IF (triangle AND côtés égaux=3[combien la figure a-t-elle de côtés égaux ?]) THEN triangle équilatéral");
m.AddRule("R6 : IF (ordre=4[quel est l'ordre ?]) THEN quadrilatère");
m.AddRule("R7 : IF (quadrilatère AND côtés parallèles=2[combien y a-t-il de côtés parallèles entre eux - 0, 2 ou 4?]) THEN trapèze");
m.AddRule("R8 : IF (quadrilatère AND côtés parallèles=4[combien y a-t-il de côtés parallèles entre eux - 0, 2 ou 4?]) THEN parallélogramme");
m.AddRule("R9 : IF (parallélogramme AND angle droit[la figure a-t-elle au moins un angle droit ?]) THEN rectangle");
m.AddRule("R10 : IF (parallélogramme AND cotés égaux=4[Combien la figure a-t-elle de côtés égaux ?]) THEN losange");
m.AddRule("R11 : IF (rectangle AND losange) THEN carré");
```

Les règles concernant la gestion des pentagones (ordres = 5), ne sont pas présentes. A vous de les écrire.

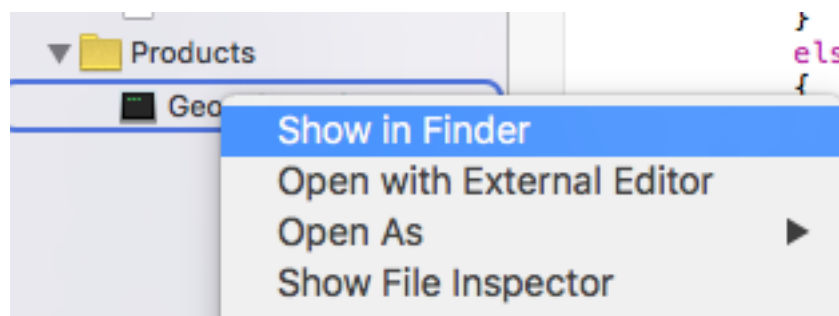
Vous pouvez aussi essayer avec un autre domaine que la géométrie.

5. Production des résultats

La version console Xcode présentant peu d'intérêt en terme d'exploitation, il faut pouvoir lancer l'exécutable produit par Xcode dans un terminal pour pouvoir mettre en couleur les réponses et améliorer la mise en forme

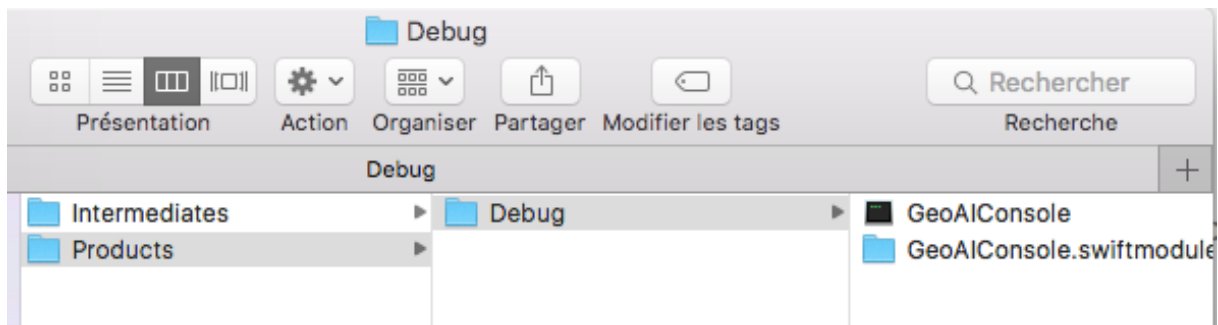
Cette partie du projet n'est pas présentée dans le code C#, vous devez cette fois pouvoir travailler en Swift pour aboutir à la forme de présentation souhaitée.

Pour accéder à l'exécutable produit par Xcode :



Se placer sur l'exécutable dans l'onglet Products (fenêtre de gauche). Clic droit sur l'exécutable, et choisir « Show in Finder ».

Le « finder » ouvre la fenêtre suivante :



Un double clic sur l'exécutable, ici « GeoAIConsole », permet de lancer le programme dans un terminal.

5.1. Arbre de cheminement dans la solution

Vous devrez présenter votre solution sous la forme d'un arbre dans un premier temps. Cet arbre permet de visualiser les prémisses issues des questions posées à l'utilisateur. Les prémisses vous engendrer des niveaux successifs de conclusions, jusqu'à la conclusion (solution) finale.

Prenons l'exemple suivant :

L'ordre de la figure est 4, la figure à un angle droit, les 4 côtés sont égaux, les 4 cotés sont égaux (cette question est posée deux fois avec toutes les règles, une fois pour l'ordre 3, et une fois pour l'ordre 4), il n'y pas d'angle concave, cela ressemble fortement à un ... carré.

Le raisonnement est le suivant :

Arbre de construction de la solution :

ordre=4 (0) angle droit (0) côtés égaux=4 (0) côtés parallèles=4 (0) cotés égaux=4 (0) !angle concave (0) angle égaux=4 (0)

```

|       |       |       |       |       |       |
v       v       v       v       v       v       v

```

triangle rectangle (1) quadrilatère (1)

```

|               |
v               v

```

parallélogramme (2)

```

|
v

```

rectangle (3) losange (3)

```

|               |
v               v

```

carré (4)

```

|
v

```

ATTENTION : les résultats sont ceux d'un programme de test qui inclut plus de règles que celles de base !

Le système conclut que la bonne réponse est un carré.

5.2. Statistiques sur les conclusions

Vous devrez présenter votre solution sous la forme de statistiques dans un second temps. Dans ce cas chaque conclusion proposée dans l'arbre (donc avec un niveau > 0, soit un level > 0) devra recevoir une probabilité d'être la bonne réponse.

Dans notre exemple précédent cela donne la réponse suivante :

La qualité de la réponse dépend de la qualité des règles et de la réponse aux questions !

Statistiques :
Cumul des niveaux : 14
Niveau maximum : 4

La forme carré de niveau 4 EST LA PLUS PROBABLE -> probabilité = 28.5714285714286 %

La forme losange de niveau 3 -> probabilité = 21.4285714285714 %
La forme parallélogramme de niveau 2 -> probabilité = 14.2857142857143 %
La forme quadrilatère de niveau 1 -> probabilité = 7.14285714285714 %
La forme rectangle de niveau 3 -> probabilité = 21.4285714285714 %
La forme triangle rectangle de niveau 1 -> probabilité = 7.14285714285714 %

ATTENTION : les résultats sont ceux d'un programme de test qui inclut plus de règles que celles de base !

Le système expert met en avant la réponse la plus probable face aux autres propositions

Bon, évidemment, ce n'est pas encore avec cela qu'une caméra va pouvoir faire des probabilités sur la nature de l'objet rencontré ! Surtout dans la conduite d'un véhicule autonome ! Il faut en conclure que la fonction autoPilot de Tesla a d'autres secrets en matière d'IA !

Le calcul se base sur le cumul des niveaux.

Le ratio (niveau/cumul) * 100 donne le pourcentage d'une solution.

Cette approche pourrait être complétée sur la probabilité qu'un fait peut être en partie vrai ou faux. Cela veut dire que l'on pourrait inclure des probabilités dans les faits inférés ou entrés dans le cas où l'utilisateur n'est pas sûr de sa réponse.

5.3. Codes couleurs à utiliser

```
let black = "\u{001B}[0;30m"  
let red = "\u{001B}[0;31m"  
let green = "\u{001B}[0;32m"  
let yellow = "\u{001B}[0;33m"  
let blue = "\u{001B}[0;34m"  
let magenta = "\u{001B}[0;35m"  
let cyan = "\u{001B}[0;36m"
```

let white = "\u{001B}[0;37m"

A vous de jouer avec !

6. Workbenchs

Les jeux d'essais proposés ici permettent de calibrer votre application, ils reposent sur les 11 règles de base qui sont proposées :

Ordre n	3	3	3	3	3	3	4	4	4
Angle droit	V	F	V	V	V	V	V	V	V
Côtés égaux	0	0	3	2	2	2	0	2	2
Côtés parallèles	0	0	0	0	0	2	0	0	0
Côtés égaux	0	0	3	2	1	2	0	0	4
Numéro solution	1	2	3	4	5	6	7	8	9
Solution humaine	Triangle rectangle	Triangle	Triangle rectangle	Triangle rectangle isocèle	Triangle rectangle isocèle	Triangle rectangle isocèle	Quadrilatère particulier	Carré rectangle	carré
Solution statistique	Triangle rectangle	triangle	Triangle rectangle + triangle isocèle	Triangle rectangle isocèle	Triangle rectangle isocèle	Triangle rectangle isocèle	Quadrilatère rectangle triangle rectangle	Triangle rectangle isocèle	Triangle rectangle isocèle
Solution dernier élément	rectangle	triangle	rectangle	rectangle	rectangle	rectangle	rectangle	rectangle	carré

Suivant la manière de retenir une solution, la solution basée sur la statistique remporte le plus souvent la vérité (60% des cas), contre le dernier élément de la solution (30%)

Solution 1

Arbre de construction de la solution :

```
ordre=3 (0) angle droit (0) côtés égaux=0 (0) côtés parallèles=0 (0) cotés égaux=0 (0)
```

v	v	v	v	v

```
triangle (1) rectangle (1)
```

v	v

triangle rectangle (2)

iv

La qualité de la réponse dépend de la qualité des règles et de la réponse aux questions !

Statistiques :

Cumul des niveaux : 4

Niveau maximum : 2

La forme rectangle de niveau 1 -> probabilité = 25.0 %

La forme triangle de niveau 1 -> probabilité = 25.0 %

La forme triangle rectangle de niveau 2 EST LA PLUS PROBABLE -> probabilité = 50.0 %

Solution alternative basée sur l'ordre de cheminement :

Le dernier élément est le plus probable dans ce cas !

triangle (1)

triangle rectangle (2)

rectangle (1)

Arbre de construction de la solution :

```
ordre=3 (0) !angle droit (0) côtés égaux=0 (0) côtés parallèles=0 (0) cotés égaux=0 (0)
```

1	1	1	1	1
v	v	v	v	v

triangle (1)

1
v

La qualité de la réponse dépend de la qualité des règles et de la réponse aux questions !

```
Statistiques :
Cumul des niveaux : 1
Niveau maximum : 1
```

La forme triangle de niveau 1 EST LA PLUS PROBABLE -> probabilité = 100.0 %

Solution alternative basée sur l'ordre de cheminement :
Le dernier élément est le plus probable dans ce cas !

triangle (1)

Solution 3

Arbre de construction de la solution :

ordre=3 (0) angle droit (0) côtés égaux=3 (0) côtés parallèles=0 (0) cotés égaux=3 (0)

v	v	v	v	v

```
triangle (1) rectangle (1)
```

1	1
2	2

triangle rectangle (2) triangle équilatéral (2)

v	v

La qualité de la réponse dépend de la qualité des règles et de la réponse aux questions !

Statistiques :
Cumul des niveaux : 6
Niveau maximum : 2

```
La forme rectangle de niveau 1 -> probabilité = 16.6666666666667 %
La forme triangle de niveau 1 -> probabilité = 16.6666666666667 %
La forme triangle équilatéral de niveau 2 EST LA PLUS PROBABLE -> probabilité = 33.3333333333333 %

La forme triangle rectangle de niveau 2 EST LA PLUS PROBABLE -> probabilité = 33.3333333333333 %
```

Solution alternative basée sur l'ordre de cheminement :
Le dernier élément est le plus probable dans ce cas !

triangle (1)

triangle rectangle (2)

triangle équilatéral (2)

rectangle (1)

Solution 4

Arbre de construction de la solution :

```
ordre=3 (0) angle droit (0) côtés égaux=2 (0) côtés parallèles=0 (0) cotés égaux=2 (0)
```

1	1	1	1	1
v	v	v	v	v

```
triangle (1) rectangle (1)
```

triangle rectangle (2) triangle isocèle (2)

triangle rectangle isocèle (3)

1
v

La qualité de la réponse dépend de la qualité des règles et de la réponse aux questions !

Statistiques :

Cumul des niveaux : 9

Niveau maximum : 3

La forme rectangle de niveau 1 -> probabilité = 11.111111111111 %

La forme rectangle de niveau 1 -> probabilité = 11.111111111111111 %
La forme triangle de niveau 1 -> probabilité = 11.111111111111111 %

La forme triangle isocèle de niveau 2 -> probabilité = 22.222222222222 %

La forme triangle rectangle de niveau 2 -> probabilité = 22.222222222222 %

La forme triangle rectangle isocèle de niveau 3 EST LA PLUS PROBABLE -> probabilité = 33.333333333333 %

Solution alternative basée sur l'ordre de cheminement :
Le dernier élément est le plus probable dans ce cas !

triangle (1)

triangle rectangle (2)

triangle isocèle (2)

triangle rectangle isocèle (3)

rectangle (1)

Arbre de construction de la solution :

```
ordre=3 (0) angle droit (0) côtés égaux=2 (0) côtés parallèles=0 (0) cotés égaux=1 (0)
```

v	v	v	v	v

```
triangle (1) rectangle (1)
```

1	1
v	v

triangle rectangle (2) triangle isocèle (2)

1	1
v	v

triangle rectangle isocèle (3)

1
v

La qualité de la réponse dépend de la qualité des règles et de la réponse aux questions !

```
Statistiques :
Cumul des niveaux : 9
Niveau maximum : 3
```

```
La forme rectangle de niveau 1 -> probabilité = 11.111111111111 %
La forme triangle de niveau 1 -> probabilité = 11.111111111111 %
La forme triangle isocèle de niveau 2 -> probabilité = 22.222222222222 %
La forme triangle rectangle de niveau 2 -> probabilité = 22.222222222222 %
La forme triangle rectangle isocèle de niveau 3 EST LA PLUS PROBABLE -> probabilité = 33.333333333333 %
```

Solution alternative basée sur l'ordre de cheminement :
Le dernier élément est le plus probable dans ce cas !

triangle (1)

triangle rectangle (2)

triangle isocèle (2)

triangle rectangle isocèle (3)

rectangle (1)

Arbre de construction de la solution :

v	v	v	v	v

1	1	1
v	v	v

1
v

Niveau maximum : 3

La forme triangle rectangle isocèle de niveau 3 EST LA PLUS PROBABLE -> probabilité = 30.0 %

Le dernier élément est le plus probable dans ce cas !

rectangle (1)

Solution 7

Arbre de construction de la solution :

ordre=4 (0) angle droit (0) côtés égaux=0 (0) côtés parallèles=0 (0) cotés égaux=0 (0)

| | | | |
v v v v v

triangle rectangle (1) quadrilatère (1) rectangle (1)

| | |
v v v

La qualité de la réponse dépend de la qualité des règles et de la réponse aux questions !

Statistiques :
Cumul des niveaux : 3
Niveau maximum : 1

La forme quadrilatère de niveau 1 EST LA PLUS PROBABLE -> probabilité = 33.333333333333 %

La forme rectangle de niveau 1 EST LA PLUS PROBABLE -> probabilité = 33.333333333333 %

La forme triangle rectangle de niveau 1 EST LA PLUS PROBABLE -> probabilité = 33.333333333333 %

Ce qui n'est pas la meilleure solution comme le suggère un autre classement de la part du programme :

Solution alternative basée sur l'ordre de cheminement :
Le dernier élément est le plus probable dans ce cas !

triangle rectangle (1)

quadrilatère (1)

rectangle (1)

Solution 8

Arbre de construction de la solution :

ordre=4 (0) angle droit (0) côtés égaux=2 (0) côtés parallèles=0 (0) cotés égaux=0 (0)



triangle rectangle (1) triangle isocèle (1) quadrilatère (1) rectangle (1)



triangle rectangle isocèle (2)



La qualité de la réponse dépend de la qualité des règles et de la réponse aux questions !

Statistiques :

Cumul des niveaux : 6

Niveau maximum : 2

La forme quadrilatère de niveau 1 -> probabilité = 16.6666666666667 %

La forme rectangle de niveau 1 -> probabilité = 16.6666666666667 %

La forme triangle isocèle de niveau 1 -> probabilité = 16.6666666666667 %

La forme triangle rectangle de niveau 1 -> probabilité = 16.6666666666667 %

La forme triangle rectangle isocèle de niveau 2 EST LA PLUS PROBABLE -> probabilité = 33.3333333333333 %

Solution alternative basée sur l'ordre de cheminement :

Le dernier élément est le plus probable dans ce cas !

triangle rectangle (1)

triangle isocèle (1)

triangle rectangle isocèle (2)

quadrilatère (1)

rectangle (1)

Solution 9

Arbre de construction de la solution :

ordre=4 (0) angle droit (0) côtés égaux=2 (0) côtés parallèles=0 (0) cotés égaux=4 (0)

```
|         |         |         |         |
v         v         v         v         v
```

triangle rectangle (1) triangle isocèle (1) quadrilatère (1) rectangle (1) losange (1)

```
|         |         |         |         |
v         v         v         v         v
```

triangle rectangle isocèle (2) carré (2)

```
|                 |
v                 v
```

La qualité de la réponse dépend de la qualité des règles et de la réponse aux questions !

Statistiques :
Cumul des niveaux : 9
Niveau maximum : 2

La forme carré de niveau 2 EST LA PLUS PROBABLE -> probabilité = 22.222222222222 %

La forme losange de niveau 1 -> probabilité = 11.111111111111 %
La forme quadrilatère de niveau 1 -> probabilité = 11.111111111111 %
La forme rectangle de niveau 1 -> probabilité = 11.111111111111 %
La forme triangle isocèle de niveau 1 -> probabilité = 11.111111111111 %
La forme triangle rectangle de niveau 1 -> probabilité = 11.111111111111 %
La forme triangle rectangle isocèle de niveau 2 EST LA PLUS PROBABLE -> probabilité = 22.222222222222 %

Solution alternative basée sur l'ordre de cheminement :
Le dernier élément est le plus probable dans ce cas !

triangle rectangle (1)

triangle isocèle (1)

triangle rectangle isocèle (2)

quadrilatère (1)

rectangle (1)

losange (1)

carré (2)



