

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Jijel
Faculté des Sciences exactes et de l'informatique
Département d'informatique



– Module – **Systèmes Experts**

Master 1 : SIAD

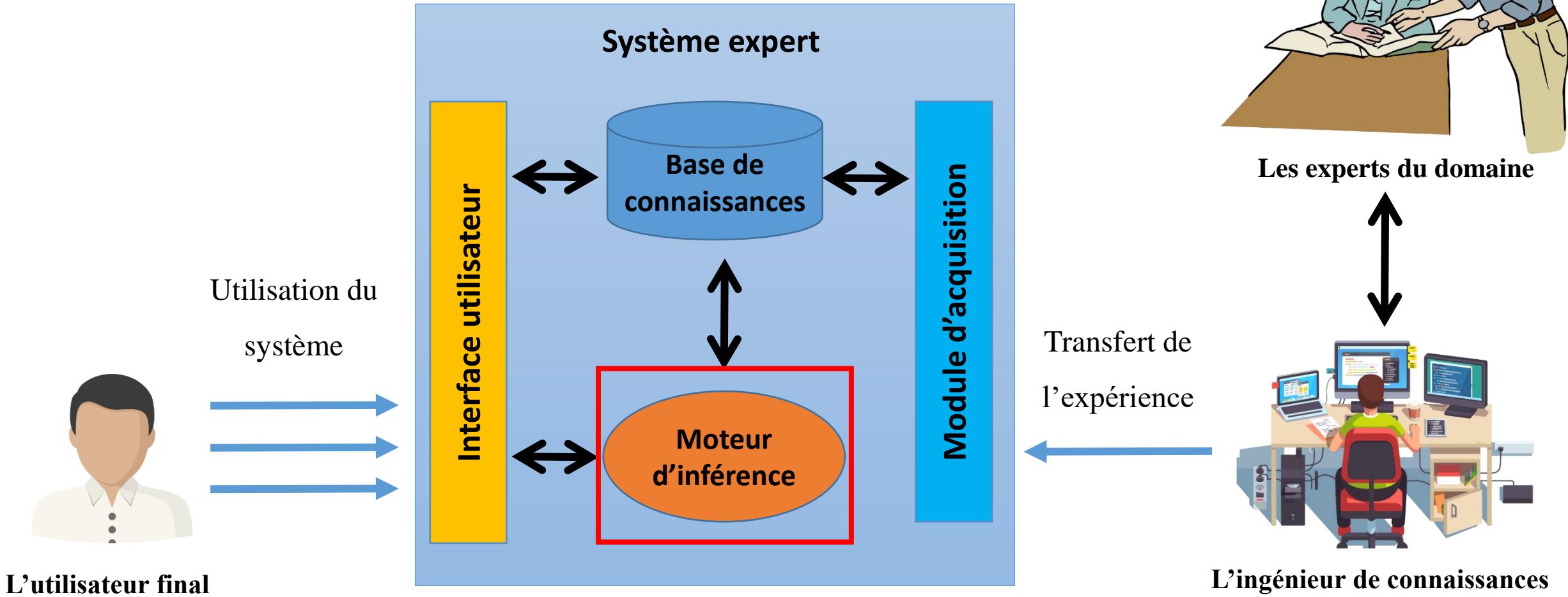
Enseignant du module : Dr. Hemza FICEL

Contact: hemza.ficel@univ-jijel.dz

Chapitre 3 – Moteur d’Inférence

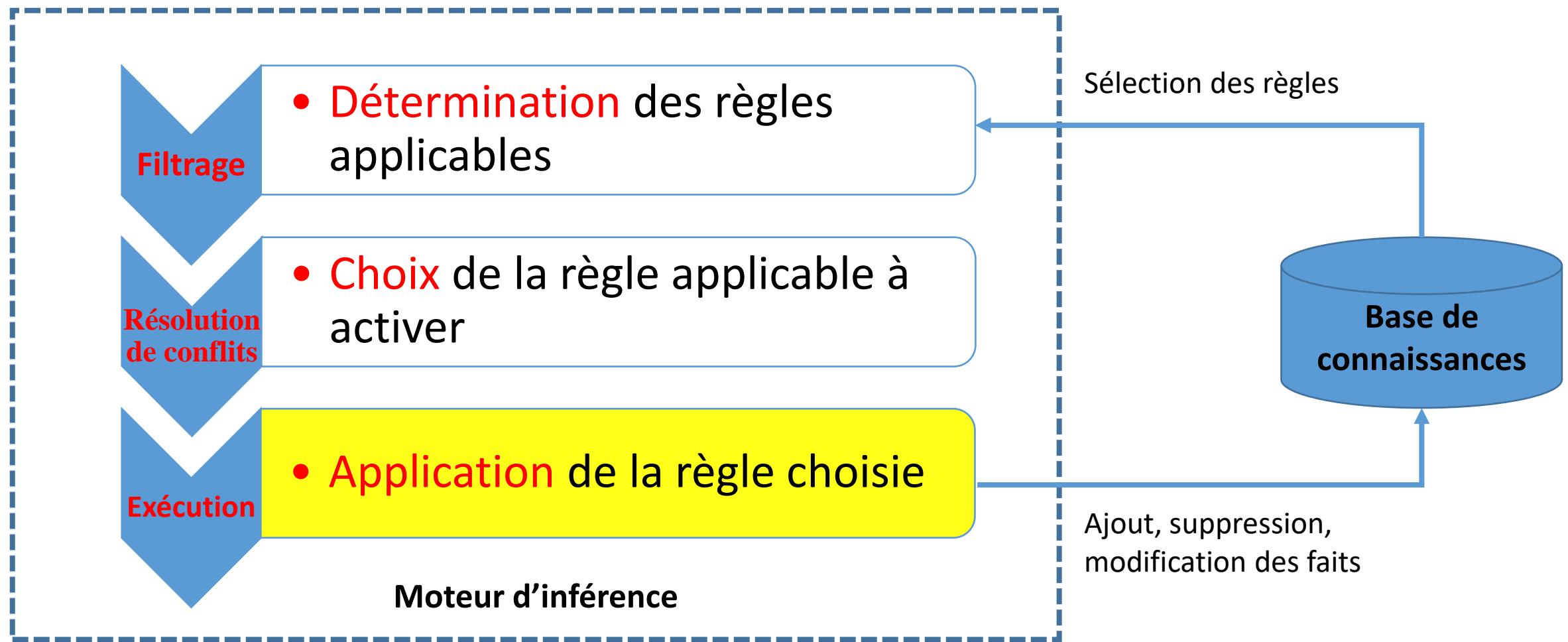
Moteur d'Inférence

Architecture d'un système Expert



Moteur d'Inférence

Cycle de base d'un moteur d'inférence



Moteur d'Inférence

Modes de raisonnement

Le système expert « pâtisserie ».

Base de règles	Base de faits
R1 : Si farine et beurre et œufs et sel alors pâte	Œufs, Pommes , Poires
R2 : Si pommes et sucre alors pommes sucrées	Abricots, Farine
R3 : Si pommes sucrées et pâte alors tarte aux pommes	Beurre, Sucre
R4 : Si abricots et pâte alors tarte aux abricots	Sel
R5 : Si poires et pâte alors tarte aux poires	
R6 : Si cerises et pâte alors tarte aux cerises	

Moteur d'Inférence

Modes de raisonnement

Démarche 2: Est-ce que je peux faire **une tarte aux abricots**

?



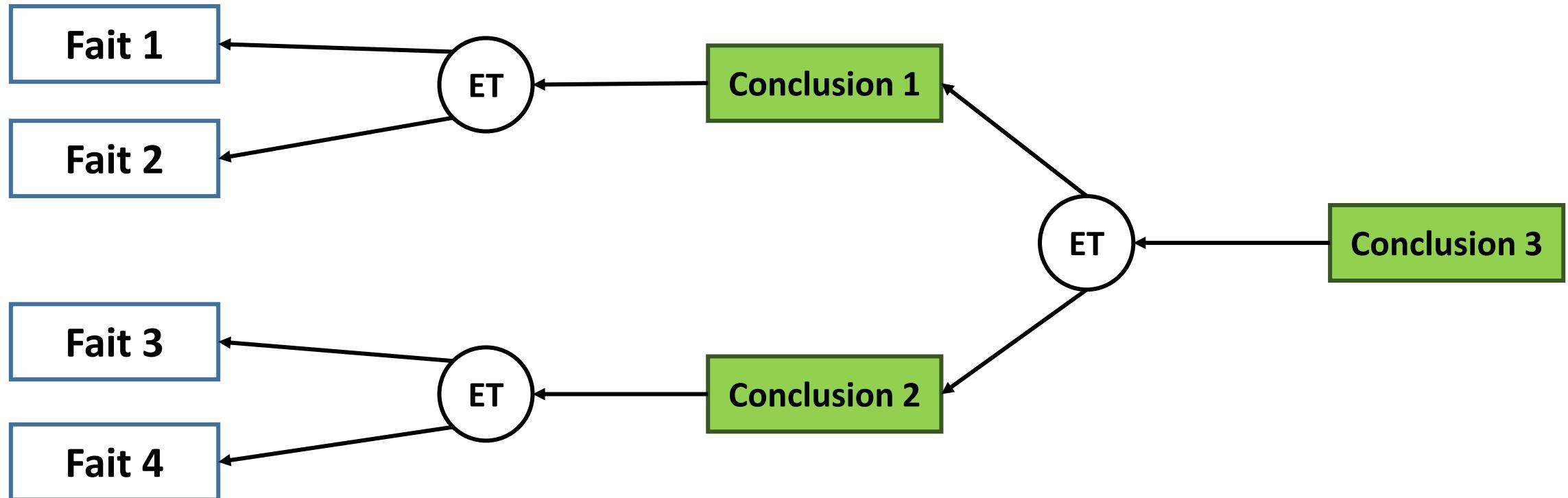
Chaînage-arrière



Un mécanisme d'inférence qui commence à partir **du but (solution/effet/action)** et tente de **remonter aux causes du but (faits)** pour le démontrer .

Modes de raisonnement

Chaînage-avant



Moteur d'Inférence

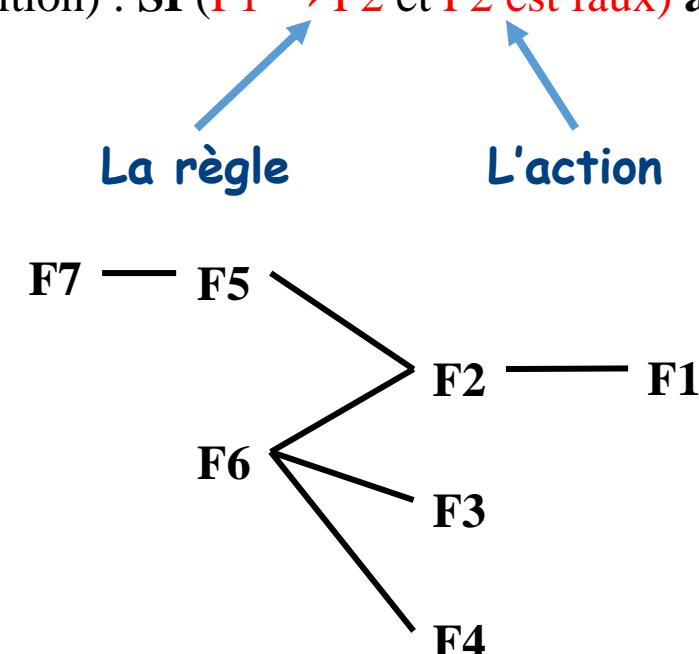
Modes de raisonnement

Chaînage-arrière

Un mécanisme d'inférence **guidé par les buts** : il commence à partir du but (solution/effet/action) et tente de remonter aux causes du but (faits) pour le démontrer .

La traduction du **modus tollens** (raisonnement indirect/contraposition) : **SI** ($F1 \rightarrow F2$ et $F2$ est faux) **alors** $F1$ est faux.

Le chaînage-avant traduit un raisonnement **inductif** : de $F1$ est déduit $F2$, de $F2$ sont déduits $F6$ et $F5$, et enfin de $F5$ est déduit $F7$.



Moteur d'Inférence

Modes de raisonnement

Chaînage-arrière

Les règles applicables sont celles dont la partie droite (conclusion) correspond au but/sous-but à démontrer.

Les conditions (prémisses) non prouvées de R deviennent des sous-buts à démontrer

La règle est prouvée si ses prémisses sont vérifiées.

Si parmi les règles sélectionnées, une règle est prouvée, alors le but initial est vérifié.

Algorithme Chaînage-arrière

Entrée : BF = base de faits; BR = base de règles; FD = fait à démontrer

Début

Déterminer les règles applicables (Filtrage)

TQ (FD n'est pas résolu ET il existe une règle applicable) **Faire**

 Choisir une règle **R** applicable dans BR (Résolution des conflits)

 Ajouter les sous-buts (partie gauche de **R**) à démontrer

Si (un sous-but SB n'est pas résolu) **Alors**

$FD \leftarrow SB$

Sinon

 FD est résolu

Finsi

FTQ

Fin

Moteur d'Inférence

Modes de raisonnement

Chaînage-arrière

Quand s'arrêter ?

- ⊕ Avec **succès** lorsque **l'ensemble des sous-buts est vide** (tous les sous-buts ont été vérifiés) **et le problème est résolu**.
- ⊕ Avec **échec** lorsque **un des sous-buts n'est pas vérifiable avec la règle courante et on** ne peut plus sélectionner de règles **pour le vérifier**.

Moteur d'Inférence

Modes de raisonnement

Chaînage-arrière

Le système expert « pâtisserie ».

Base de règles	Base de faits
R1 : Si farine et beurre et œufs et sel alors pâte	Œufs, Pommes , Poires
R2 : Si pommes et sucre alors pommes sucrées	Abricots, Farine
R3 : Si pommes sucrées et pâte alors tarte aux pommes	Beurre, Sucre
R4 : Si abricots et pâte alors tarte aux abricots	Sel
R5 : Si poires et pâte alors tarte aux poires	
R6 : Si cerises et pâte alors tarte aux cerises	

Est-ce que je peux faire **une tarte aux poires**?

Moteur d'Inférence

Modes de raisonnement

Chaînage-arrière

Base de règles	Base de faits
R1 : Si farine et beurre et œufs et sel alors pâte	Eufs, Pommes , Poires
R2 : Si pommes et sucre alors pommes sucrées	Abricots, Farine
R3 : Si pommes sucrées et pâte alors tarte aux pommes	Beurre, Sucre
R4 : Si abricots et pâte alors tarte aux abricots	Sel
R5 : Si poires et pâte alors tarte aux poires	
R6 : Si cerises et pâte alors tarte aux cerises	

Résolution de conflits : la règle de précédence (ordre d'écriture des règles)

R1, est-elle applicable sur le but « tarte aux poires »? / la partie droite de R1 correspond-elle au but/sous-but à démontrer?

Non, alors on passe à la règle suivante

Moteur d'Inférence

Modes de raisonnement

Chaînage-arrière

Base de règles	Base de faits
R1 : Si farine et beurre et œufs et sel alors pâte	
R2 : Si pommes et sucre alors pommes sucrées	Œufs, Pommes , Poires
R3 : Si pommes sucrées et pâte alors tarte aux pommes	Abricots, Farine
R4 : Si abricots et pâte alors tarte aux abricots	Beurre, Sucre
R5 : Si poires et pâte alors tarte aux poires	Sel
R6 : Si cerises et pâte alors tarte aux cerises	

R2, est-elle applicable sur le but « tarte aux poires »? / la partie droite de R2 correspond-elle au but/sous-but à démontrer?

Non, alors on passe à la règle suivante

Moteur d'Inférence

Modes de raisonnement

Chaînage-arrière

Base de règles	Base de faits
R1 : Si farine et beurre et œufs et sel alors pâte	
R2 : Si pommes et sucre alors pommes sucrées	Œufs, Pommes , Poires
R3 : Si pommes sucrées et pâte alors tarte aux pommes	Abricots, Farine
R4 : Si abricots et pâte alors tarte aux abricots	Beurre, Sucre
R5 : Si poires et pâte alors tarte aux poires	Sel
R6 : Si cerises et pâte alors tarte aux cerises	

R3, est-elle applicable sur le but « tarte aux poires »? / la partie droite de R3 correspond-elle au but/sous-but à démontrer?

Non, alors on passe à la règle suivante

Moteur d'Inférence

Modes de raisonnement

Chaînage-arrière

Base de règles	Base de faits
R1 : Si farine et beurre et œufs et sel alors pâte	
R2 : Si pommes et sucre alors pommes sucrées	Œufs, Pommes , Poires
R3 : Si pommes sucrées et pâte alors tarte aux pommes	abricots, Farine
R4 : Si abricots et pâte alors tarte aux abricots	Beurre, Sucre
R5 : Si poires et pâte alors tarte aux poires	Sel
R6 : Si cerises et pâte alors tarte aux cerises	

R4, est-elle applicable sur le but « tarte aux poires »? / la partie droite de R4 correspond-elle au but/sous-but à démontrer?

Non, alors on passe à la règle suivante

Moteur d'Inférence

Modes de raisonnement

Chaînage-arrière

Base de règles	Base de faits
R1 : Si farine et beurre et œufs et sel alors pâte	
R2 : Si pommes et sucre alors pommes sucrées	Œufs, Pommes , Poires
R3 : Si pommes sucrées et pâte alors tarte aux pommes	abricots, Farine
R4 : Si abricots et pâte alors tarte aux abricots	Beurre, Sucre
R5 : Si poires et pâte alors tarte aux poires	Sel
R6 : Si cerises et pâte alors tarte aux cerises	

R5, est-elle applicable sur le sous-but « tarte aux poires »? / la partie droite de R5 correspond-elle au but à démontrer?

Oui (sa conclusion est égale à notre but), les prémisses de R5 sont-elles présentes dans la BF ?

Le fait « **poires** » est présent dans la BF, donc il est vérifié.

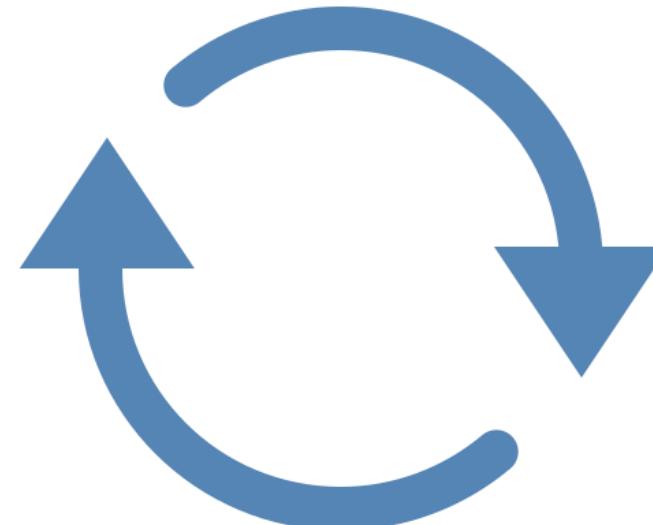
La deuxième prémissse « **pâte** » est non disponible dans la BF : elle doit être vérifiée !

Moteur d'Inférence

Modes de raisonnement

Chaînage-arrière

La deuxième prémissse « pâte » est non disponible dans la BF :
elle doit être vérifiée !



Moteur d'Inférence

Modes de raisonnement

Chaînage-arrière

Base de règles	Base de faits
R1 : Si farine et beurre et œufs et sel alors pâte	
R2 : Si pommes et sucre alors pommes sucrées	
R3 : Si pommes sucrées et pâte alors tarte aux pommes	
R4 : Si abricots et pâte alors tarte aux abricots	
R5 : Si poires et pâte alors tarte aux poires	<u>Œufs</u> , Pommes , Poires Abricots, <u>Farine</u> <u>Beurre</u> , Sucre <u>Sel</u>
R6 : Si cerises et pâte alors tarte aux cerises	

R1, est-elle applicable sur le sous-but « pâte »? / la partie droite de R1 correspond-elle au sous-but à démontrer?

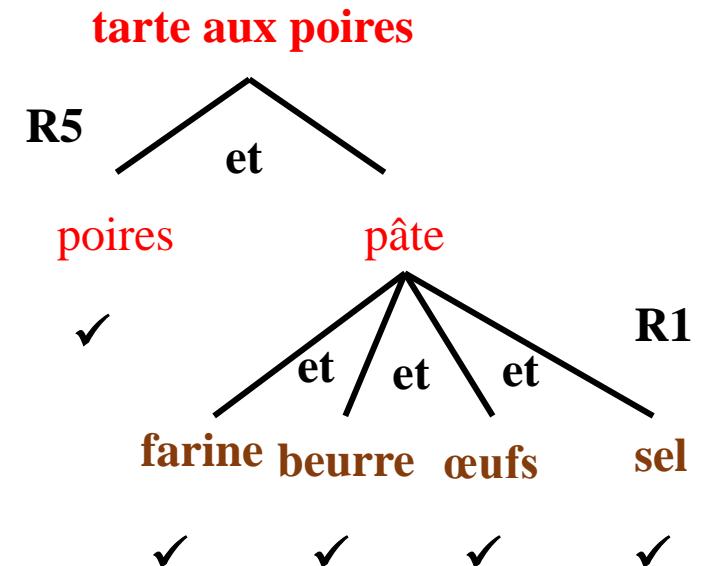
Oui (sa conclusion est égale à notre sous-but), les prémisses de R1 sont-elles présentes dans la BF ?

Tous les prémisses de R1 sont présentes dans la BF, donc elles sont vérifiées.

Modes de raisonnement

Chaînage-arrière

Base de règles	Base de faits
R1 : Si farine et beurre et œufs et sel alors pâte	
R2 : Si pommes et sucre alors pommes sucrées	
R3 : Si pommes sucrées et pâte alors tarte aux pommes	Œufs , Pommes , Poires Abricots, Farine
R4 : Si abricots et pâte alors tarte aux abricots	Beurre , Sucre Sel
R5 : Si poires et pâte alors tarte aux poires	
R6 : Si cerises et pâte alors tarte aux cerises	



Modes de raisonnement

Chaînage-arrière

✚ **Avantage** : un processus interactif basé généralement sur un arbre de recherche plus petit qu'en chaînage avant (car elle n'utilise que les règles nécessaires)

✚ **Inconvénients** :

- Il est plus complexe à mettre en œuvre (bouclage).
- Contrairement au chainage avant, il exige la disponibilité de toutes les données dès le départ .

Modes de raisonnement

Chaînage-mixte

- Le chaînage-mixte combine les deux chaînages avant et arrière pour optimiser le raisonnement. On commence avec un chaînage avant, une fois bloqué on utilise la chaînage arrière

Algorithme Chaînage-mixte

Entrée : BF = base de faits; BR = base de règles; FD = fait à démontrer

TQ (FD n'est pas déduit) Faire

Saturer la base de faits par le chaînage-avant

Déduire les faits à déduire

Choisir un fait à déduire par chaînage-arrière

Ajouter les faits déduits à la base de faits

FTQ

Moteur d'Inférence

Modes de raisonnement

Quelle stratégie choisir ?

- Le **chaînage avant** est recommandé lorsque
 - **on a plusieurs** solutions **envisageables** ;
 - **les faits sont limités** et que la plupart d'entre eux sont utilisés dans le processus de raisonnement ;
 - dans le cas d'un **SE interactif** qui opère sur **des informations incomplètes** ou le dialogue avec l'utilisateur doit s'installer (la réception de nouvelles données peut déclencher de nouvelles inférences).
- Le **chaînage arrière** est préférable si
 - **les faits initiaux** sont beaucoup **plus nombreux** que les **buts à vérifier/atteindre** ;
 - **on a si peu de buts** (solutions) **envisageables**.

Moteur d'Inférence

Exercice

Base de règles	Base de faits
R1 : Si belle ville et bon restaurants alors ville méritant une visite.	
R2 : Si ville historique alors ville méritant une visite.	
R3 : Si ville maritime et ville touristique alors ville méritant une visite.	bord de mer
R4 : Si ville maritime et belle ville alors ville méritant une visite.	
R5 : Si végétation abondante et ville maritime alors belle ville	espaces verdoyants
R6 : Si tradition culinaire et restaurants 3 étoiles alors bons restaurants.	
R7 : Si musées et ville ancienne alors ville historique.	avenues larges
R8 : Si parc animalier et grottes touristiques alors ville touristique.	grottes touristiques
R9 : Si monuments célèbres et moyens de transport alors ville touristique.	
R10 : Si espaces verdoyants et avenues larges alors végétation abondante.	corniche
R11 : Si corniche et bord de mer alors ville maritime.	

Prouvez le fait « Jijel est une ville méritant une visite » par chaînage arrière.
Résolution de conflits selon la règle de précédence

Moteur d'Inférence

Exercice

Pour prouver le fait « **Jijel est une ville méritant une visite** » par chaînage arrière, il faut partir du fait « **ville méritant une visite** » et remonter la chaîne des règles pour trouver les faits qui permettent de prouver ce but.

Moteur d'Inférence

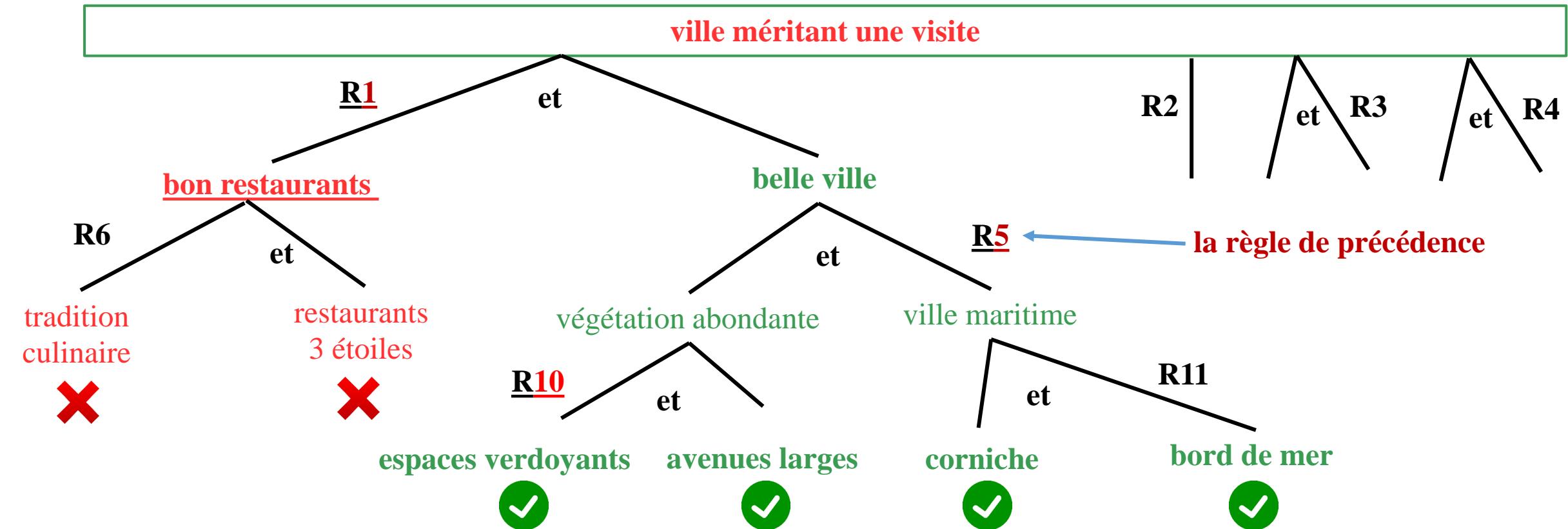
Exercice

Base de règles	Base de faits
R1 : Si belle ville et bon restaurants alors ville méritant une visite.	
R2 : Si ville historique alors ville méritant une visite.	
R3 : Si ville maritime et ville touristique alors ville méritant une visite.	bord de mer
R4 : Si ville maritime et belle ville alors ville méritant une visite.	
R5 : Si végétation abondante et ville maritime alors belle ville	espaces verdoyants
R6 : Si tradition culinaire et restaurants 3 étoiles alors bons restaurants.	avenues larges
R7 : Si musées et ville ancienne alors ville historique.	grottes touristiques
R8 : Si parc animalier et grottes touristiques alors ville touristique.	
R9 : Si monuments célèbres et moyens de transport alors ville touristique.	
R10 : Si espaces verdoyants et avenues larges alors végétation abondante.	
R11 : Si corniche et bord de mer alors ville maritime.	corniche

Prouvez le fait « Jijel est une ville méritant une visite » par chaînage arrière.
Résolution de conflits selon la règle de précédence

Moteur d'Inférence

Exercice



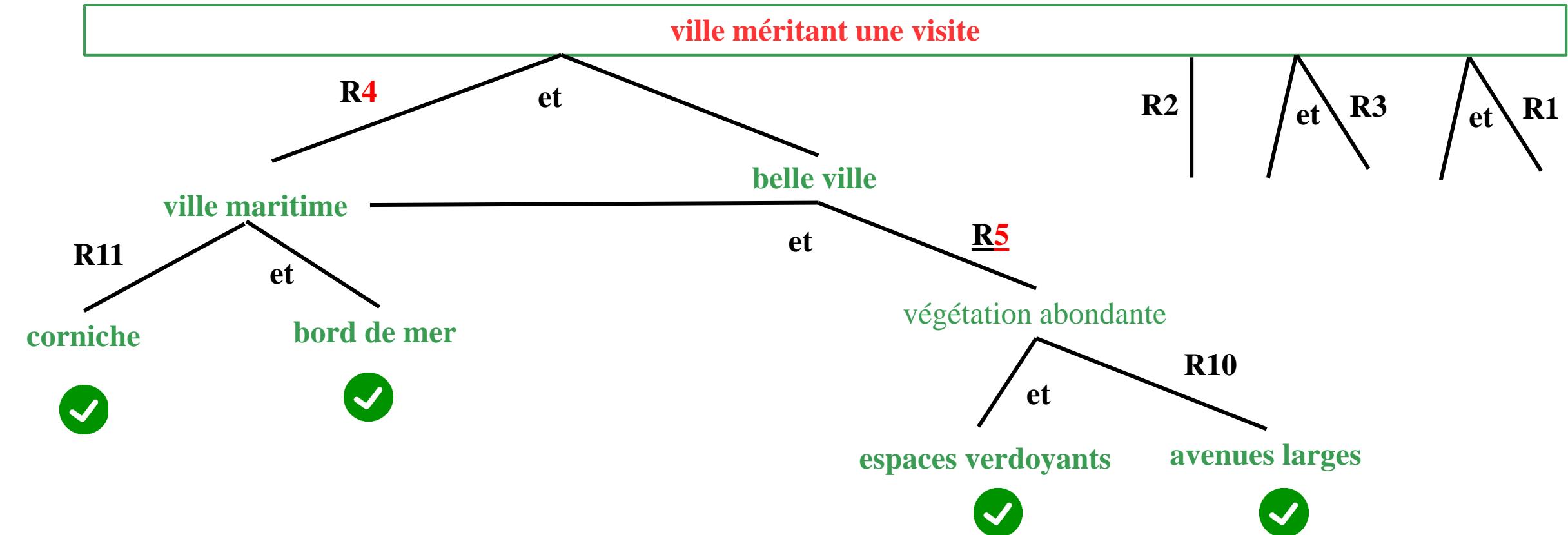
Il y a échec, car on n'a pas pu prouver le fait « bon restaurants ».

Moteur d'Inférence

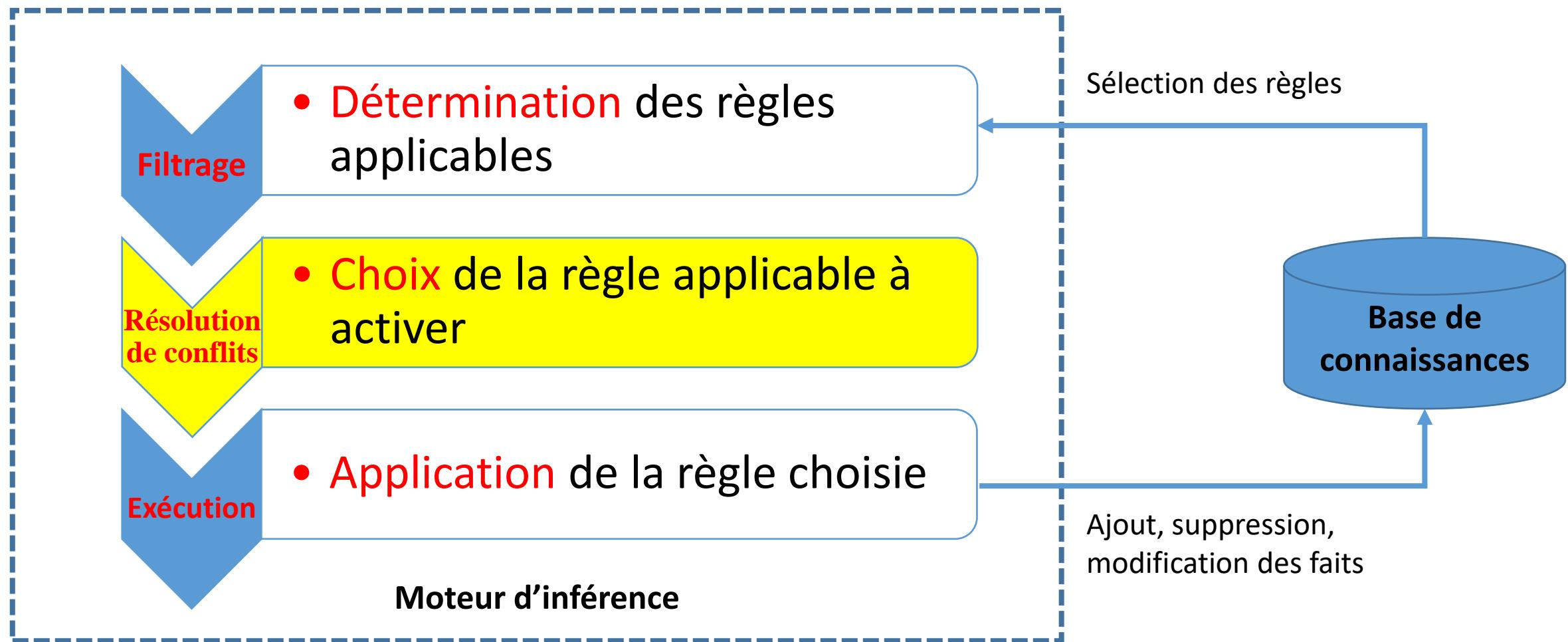
Exercice

Le problème réside dans la **stratégie de résolution de conflits**, car R4 permet de prouver notre but

Exercice



Modes de raisonnement



Moteur d'Inférence

Modes de raisonnement

- ✚ Les SE effectuant un chaînage avant ou arrière appliquent une stratégie de résolution de conflit bien définie. **La plus part choisissent les règles applicables selon l'ordre défini par l'utilisateur.**

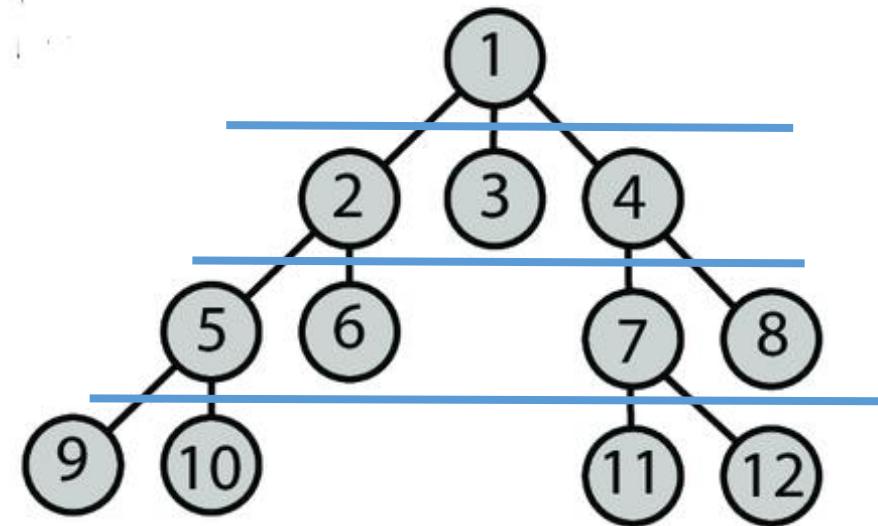


Application des règles **en largeur ou en profondeur.**

Moteur d'Inférence

Modes de raisonnement

- **Stratégie en largeur d'abord** : le système sélectionne les règles qui permettent de déduire les faits appartenant à un **même niveau de l'arbre** avant de passer à un autre niveau inférieur. Ainsi, il explore toutes les possibilités présentes avant d'entrer dans les détails.
 - + Garantie de trouver une solution (si elle existe).
 - + Si une solution existe, c'est le chemin le plus court en terme de nœuds.
 - - Couteux en temps de traitement et en mémoire.
 - - Parcours possible de tout le graphe.



Moteur d'Inférence

Modes de raisonnement

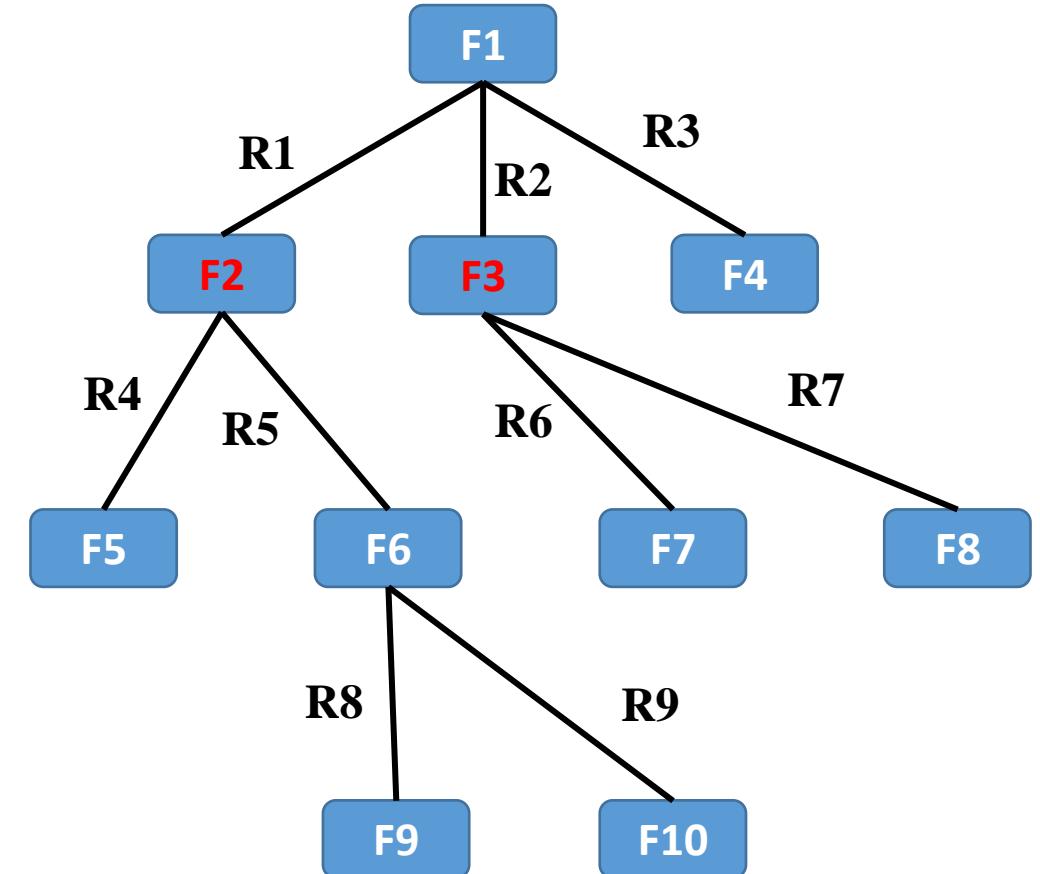
Exemple : Stratégie en largeur d'abord

$BF = \{F2, F3\}$

Mode de raisonnement = chainage avant

Ordre d'exécution des règles ?

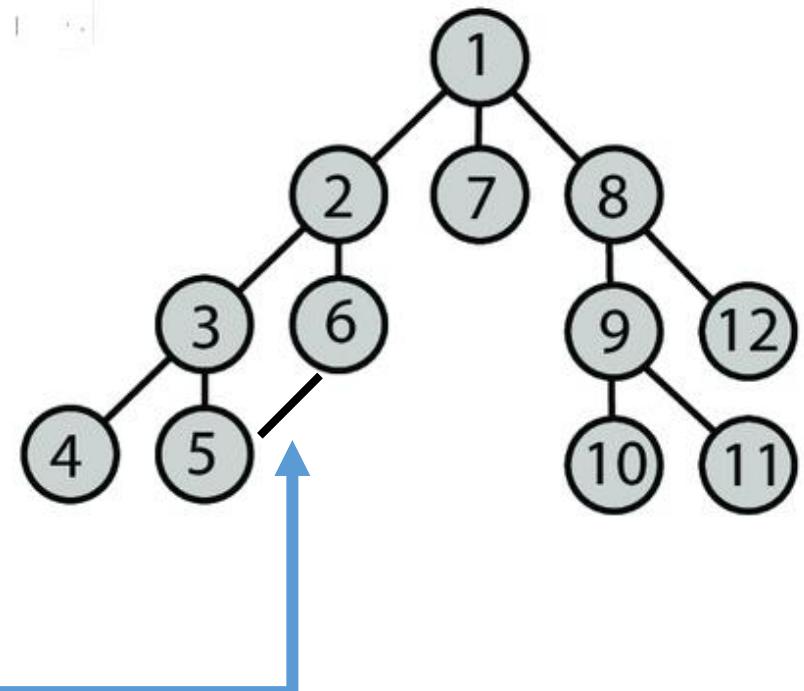
SE applique en ordre les règles : {R4,R5,R6,R7,R8,R9}



Moteur d'Inférence

Modes de raisonnement

- **Stratégie en profondeur d'abord** : le système sélectionne les règles qui permettent de déduire tous les faits appartenant **à une branche de l'arbre (vertical)** avant de passer à une autre branche. Ainsi, il explore au maximum une possibilité afin d'obtenir le plus de détails possible.
- + plus sélective.
- + plus rapide (diminution du temps de traitement).
- - parcours possible de tout le graphe.
- - branches infinies.



Moteur d'Inférence

Modes de raisonnement

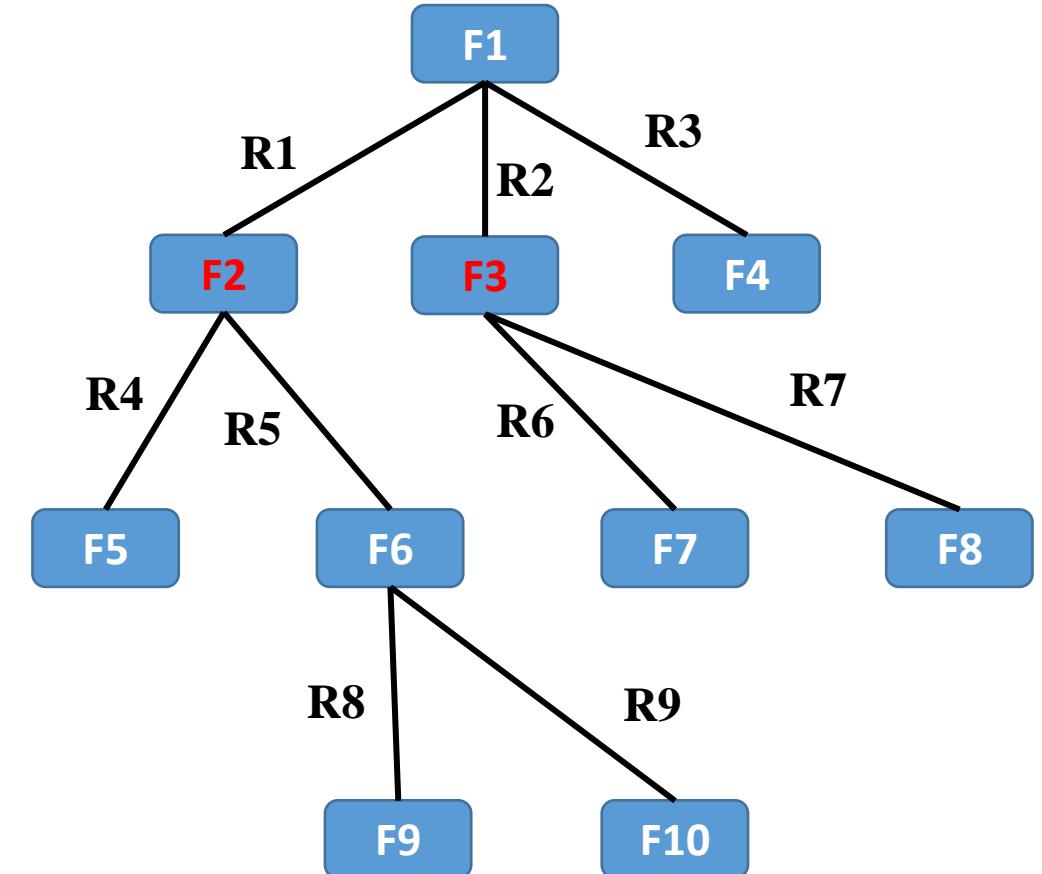
Exemple : Stratégie en profondeur d'abord

$BF = \{F2, F3\}$

Mode de raisonnement = chainage avant

Ordre d'exécution des règles ?

SE applique en ordre les règles : $\{R4, R5, R8, R9, R6, R7\}$



Moteur d'Inférence

Modes de raisonnement

- ✚ Les SE effectuant un chaînage avant ou arrière appliquent une stratégie de résolution de conflit bien définie. **La plus part choisissent les règles applicables selon l'ordre défini par l'utilisateur.**



Stratégie **irrévocable** VS Stratégie **par tentatives**

Moteur d'Inférence

Modes de raisonnement

Stratégie **irrévocable** VS Stratégie **par tentatives**

- ✚ Si le médecin décide qu'un patient doit subir une intervention chirurgicale, il recommandera la procédure et la considérera comme la seule option possible. Cette stratégie est dite « **irrévocable** » car elle ne prend pas en compte les rétroactions ultérieures (p. ex., les effets secondaires indésirables).
- ✚ Le médecin peut initialement recommander une stratégie de traitement en se basant sur les caractéristiques de la maladie. Par la suite, il surveille la réponse du patient au traitement et peut ajuster la stratégie en conséquence. (P. ex., si le patient ne répond pas bien au traitement initial, le médecin peut modifier la dose du traitement en cours). Cette stratégie est dite « **par tentatives** » car elle permet de s'adapter en fonction de l'évolution de la situation en cours.

Moteur d'Inférence

Modes de raisonnement

Stratégie irrévocabile VS Stratégie par tentatives

⊕ Une stratégie irrévocabile : les choix pris (règles appliquées) ne sont jamais remis en cause. (généralement en profondeur d'abord **puisque'en largeur**, le retour arrière est inutile, car **on explore tout le graphe**).

⊕ Une stratégie par tentatives : on peut remettre en cause l'exécution d'une règle si ce choix mène à une situation de blocage. Le moteur d'inférence effectue alors **un retour arrière (backtracking)** pour essayer une règle écartée précédemment (généralement les moteurs procédant en chaînage arrière utilisent cette stratégie).

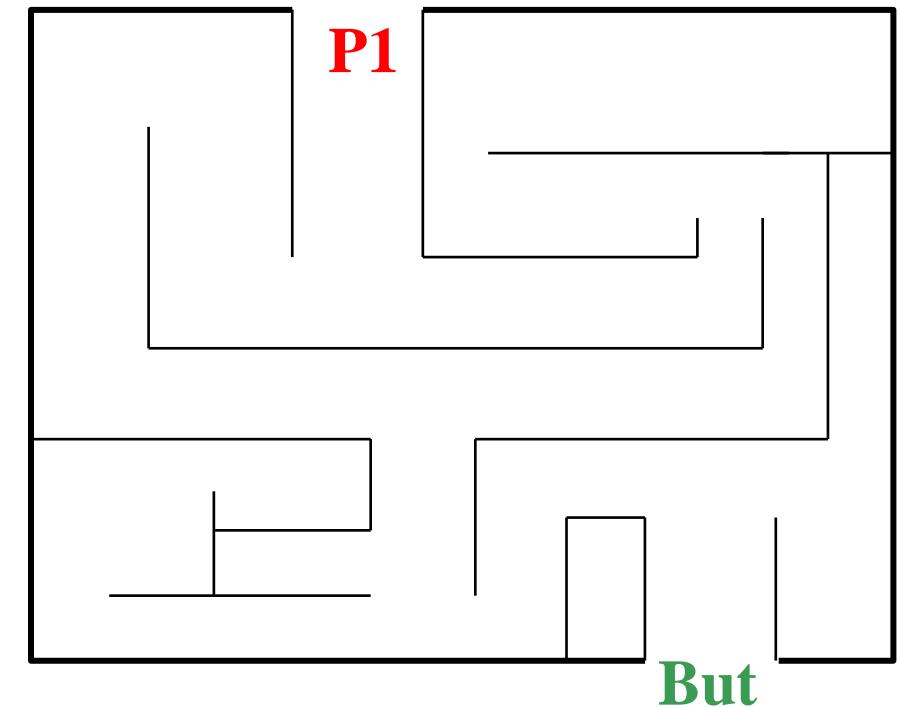
Moteur d'Inférence

Modes de raisonnement

Exemple :

Stratégie irréversible VS Stratégie par tentatives

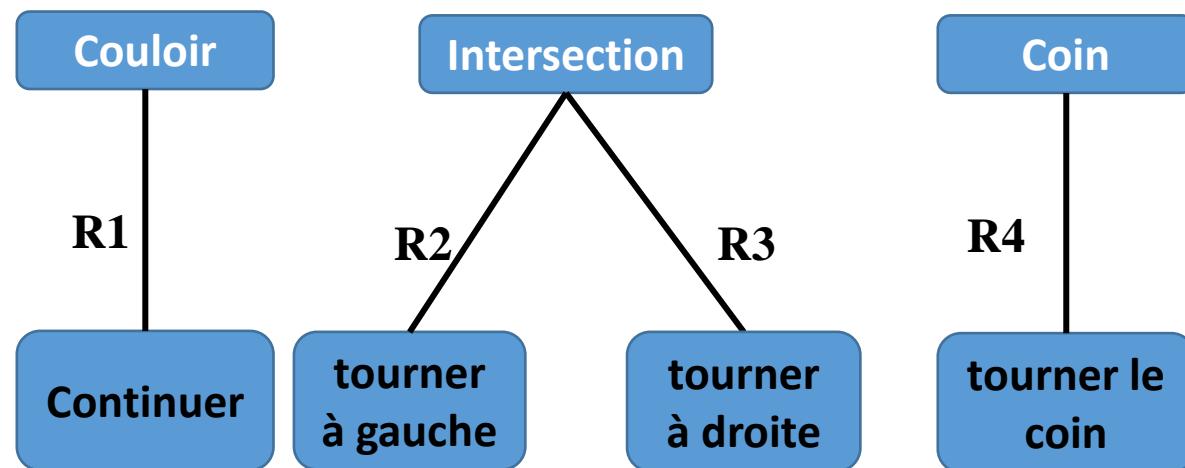
Base de règles	Base de faits
R1 : Si dans un couloir Alors continuer dans la même direction	
R2 : Si à une intersection Alors tourner à droite	
R3 : Si à une intersection Alors tourner à gauche	
R4 : Si à un coin Alors tourner le coin	



Moteur d'Inférence

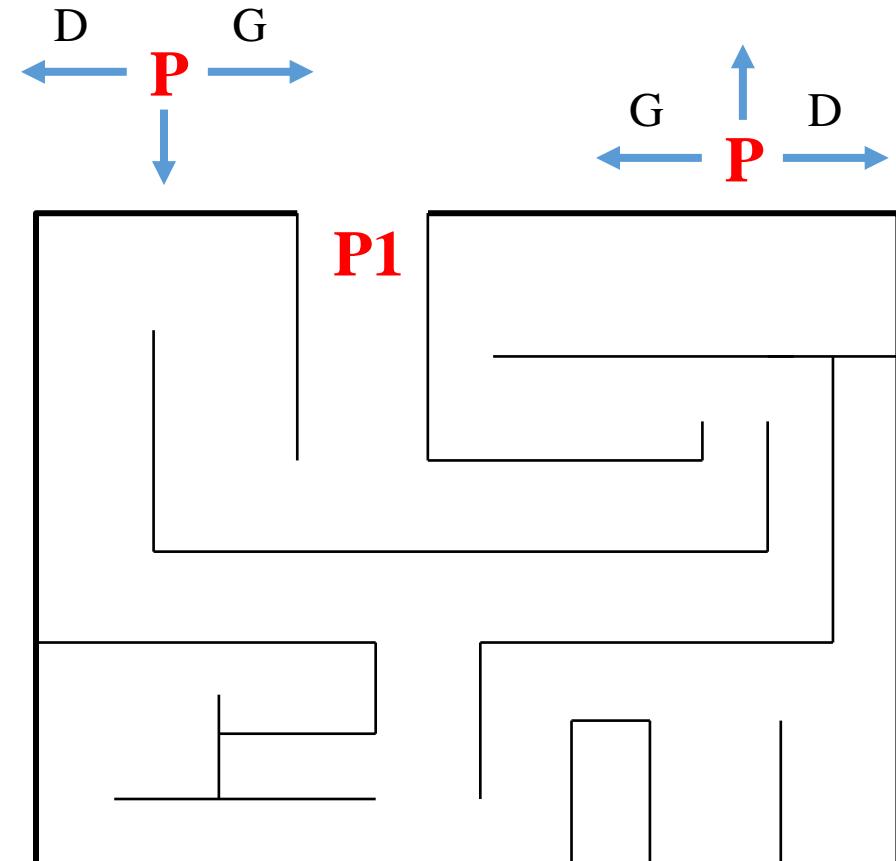
Modes de raisonnement

Exemple : chainage avant + choix irrévocable



Base de faits = {P1(Couloir)}

Règles applicables = {R1}



Moteur d'Inférence

Modes de raisonnement

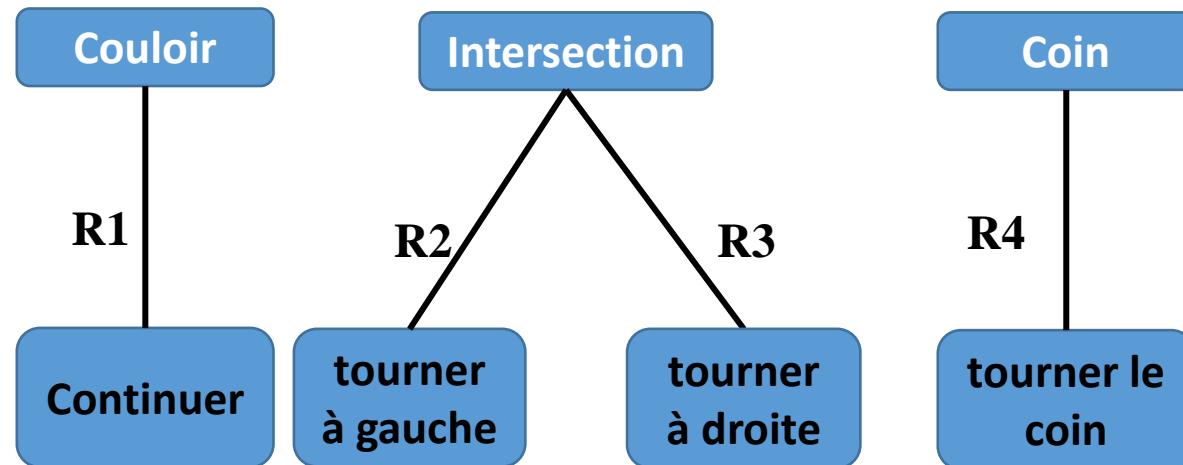
Exemple : chainage avant + **choix irréversible**

Choix irréversible : Toujours tourner à gauche aux intersections

Moteur d'Inférence

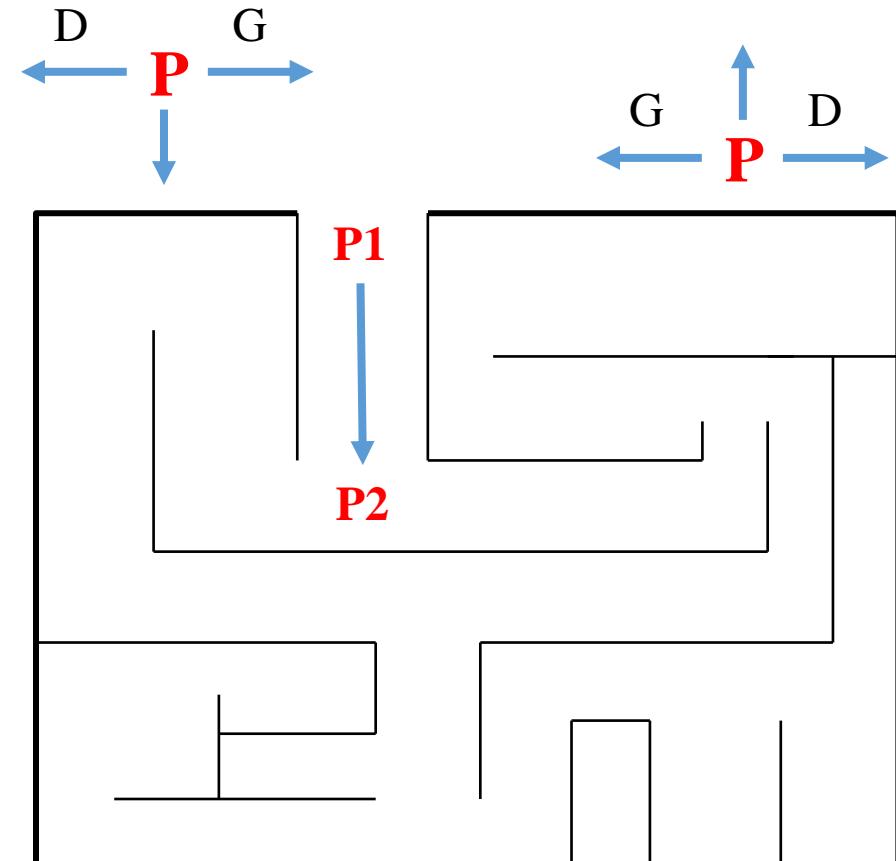
Modes de raisonnement

Exemple : chainage avant + choix irrévocable



Base de faits = {P1(Couloir), P2(Intersection)}

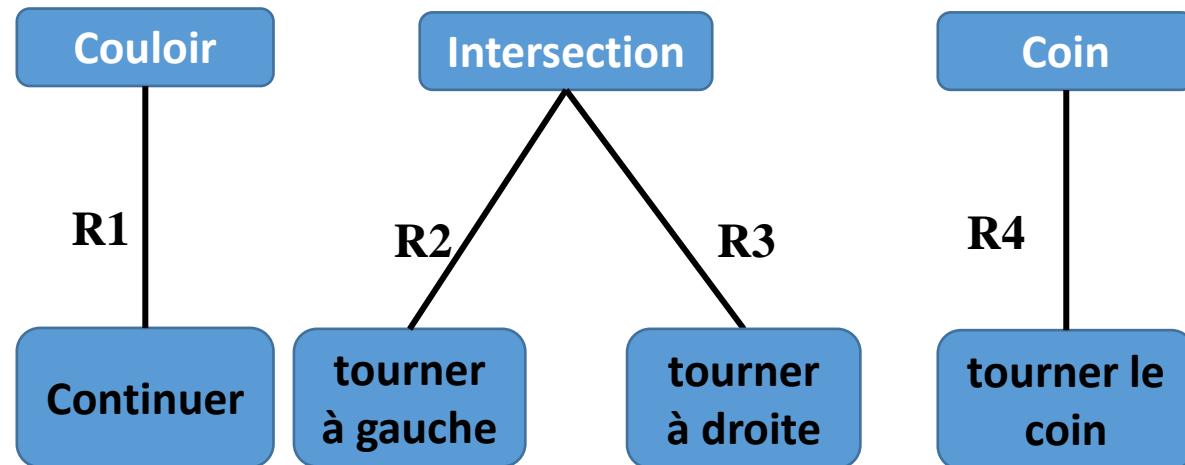
Règles applicables = {R2,R3}



Moteur d'Inférence

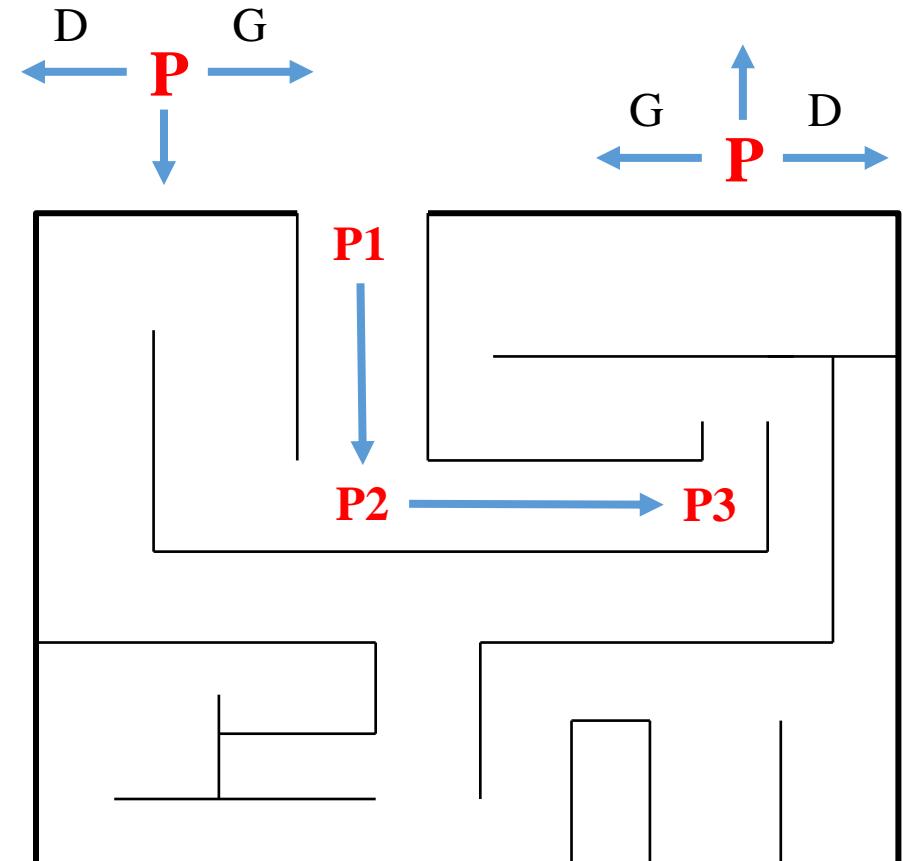
Modes de raisonnement

Exemple : chainage avant + choix irrévocable



Base de faits = {P1(Couloir), **P2(Intersection)**, P3(Coin)}

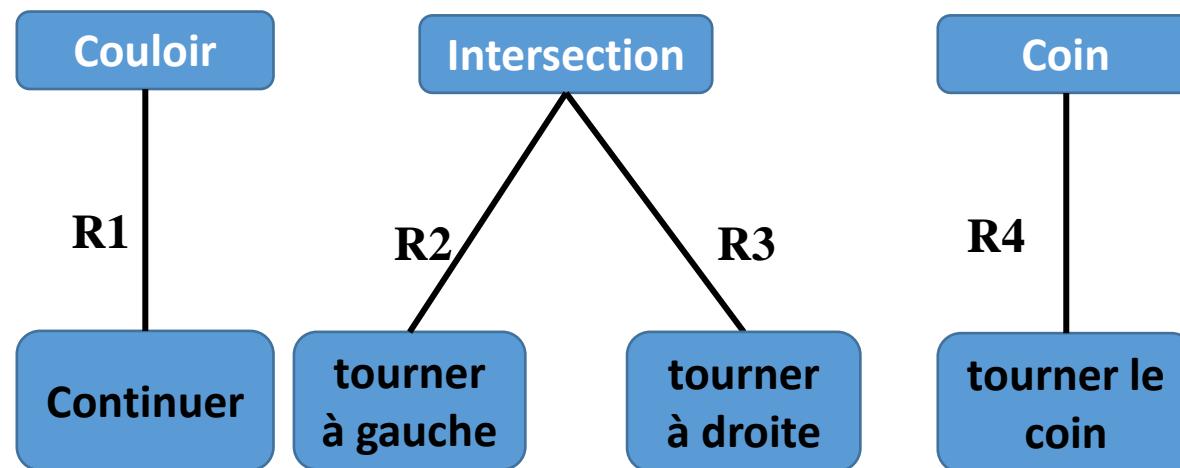
Règles applicables = {R4}



Moteur d'Inférence

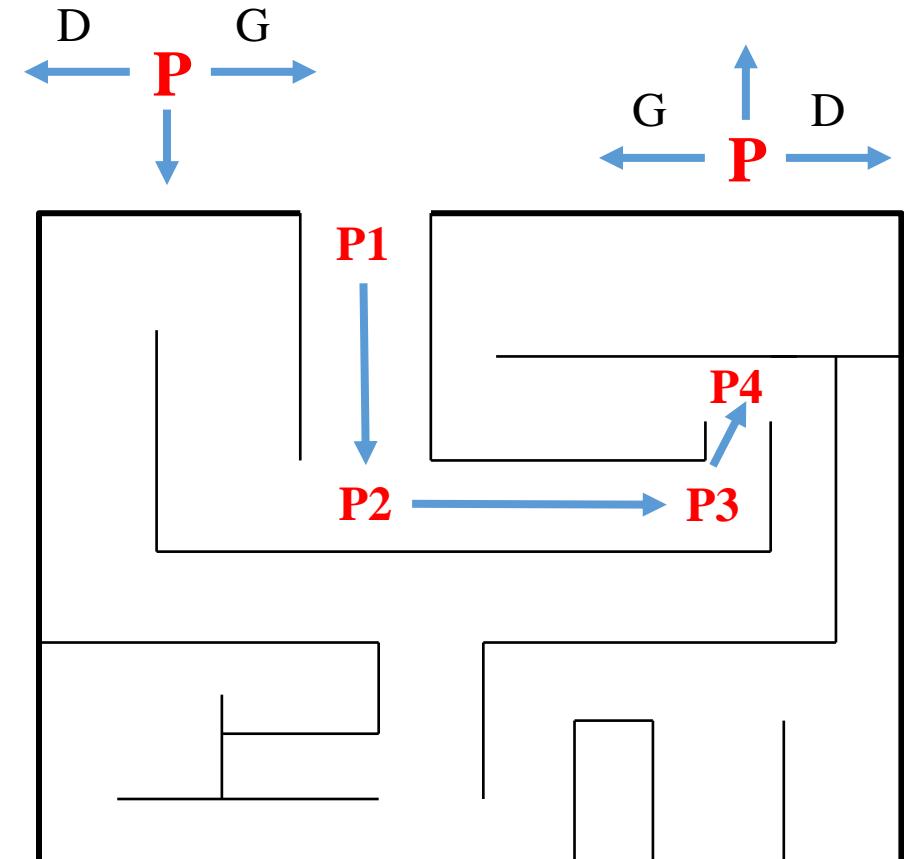
Modes de raisonnement

Exemple : chainage avant + choix irrévocable



Base de faits = {P1(Couloir), **P2(Couloir)**, P3(Coin),
P4(Intersection)}

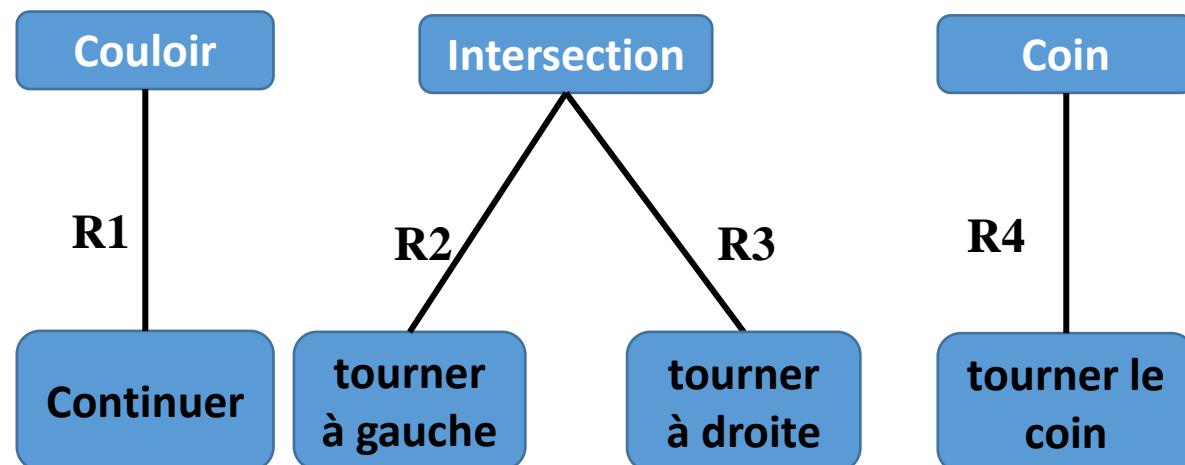
Règles applicables = {R2,R3}



Moteur d'Inférence

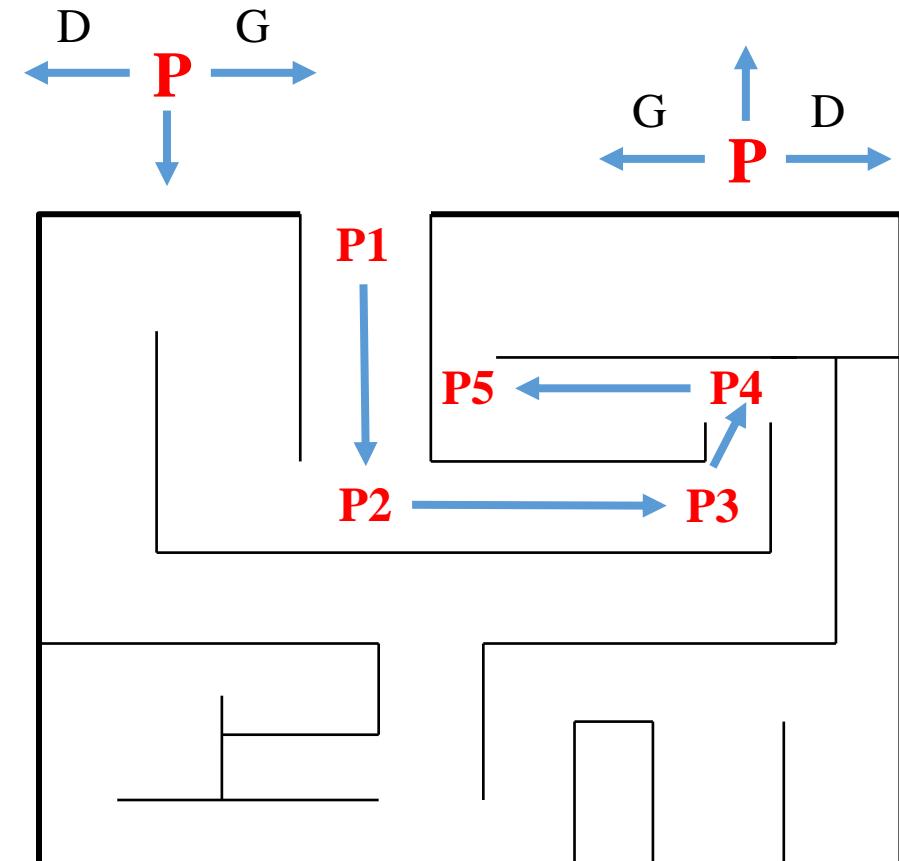
Modes de raisonnement

Exemple : chainage avant + choix irrévocable



Base de faits = {P1(Couloir), **P2(Intersection)**, P3(Coin),
P4(Intersection), P5(Coin)}

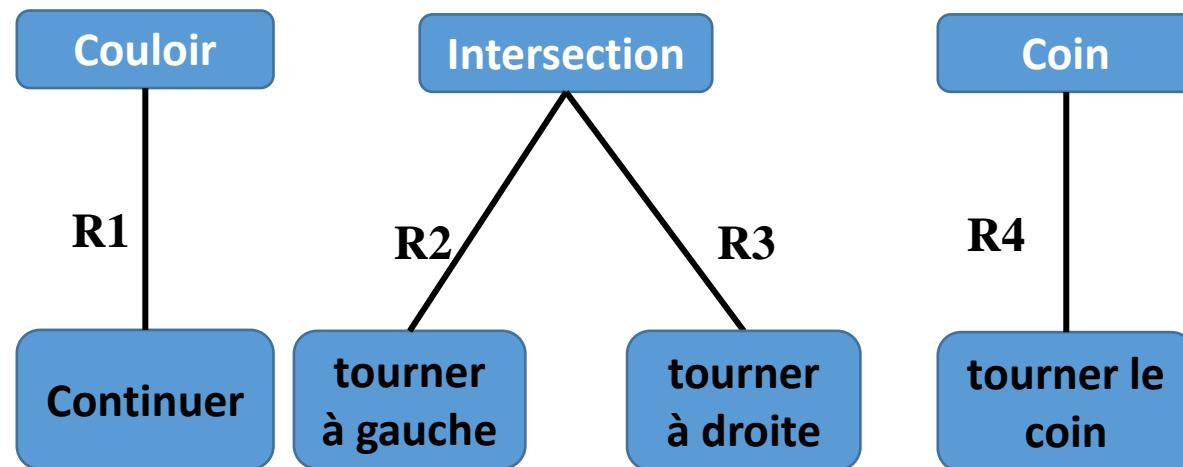
Règles applicables = {R4}



Moteur d'Inférence

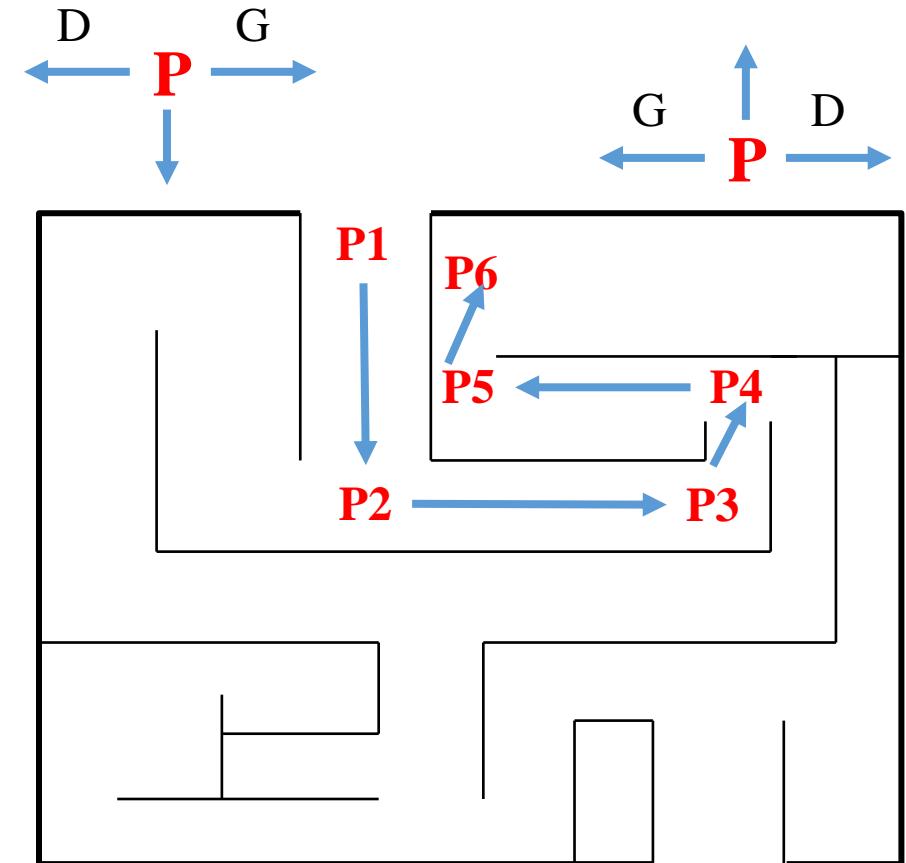
Modes de raisonnement

Exemple : chainage avant + choix irrévocable



Base de faits = {P1(Couloir), **P2(Intersection)**, P3(Coin),
P4(Intersection), P5(Coin), P6(Couloir)}

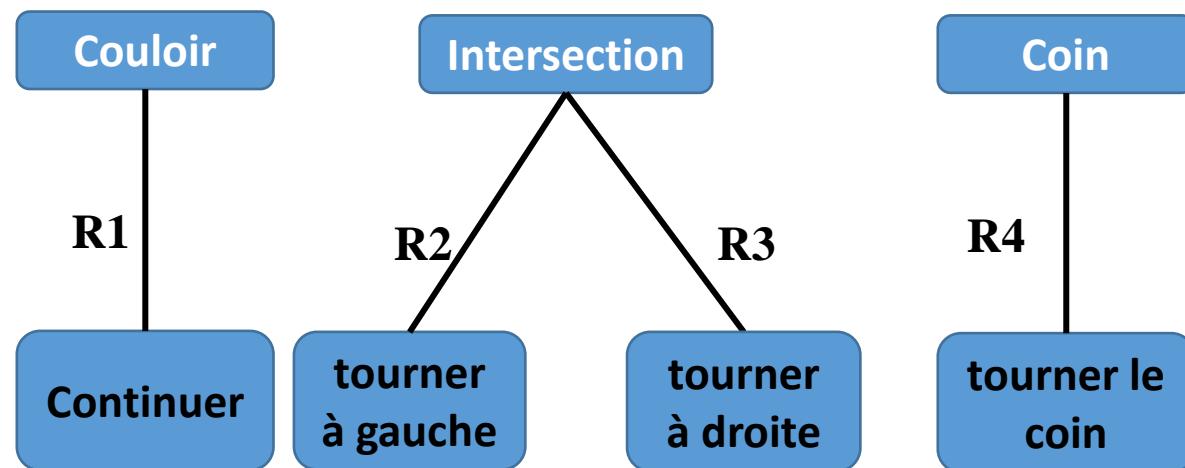
Règles applicables = {R1}



Moteur d'Inférence

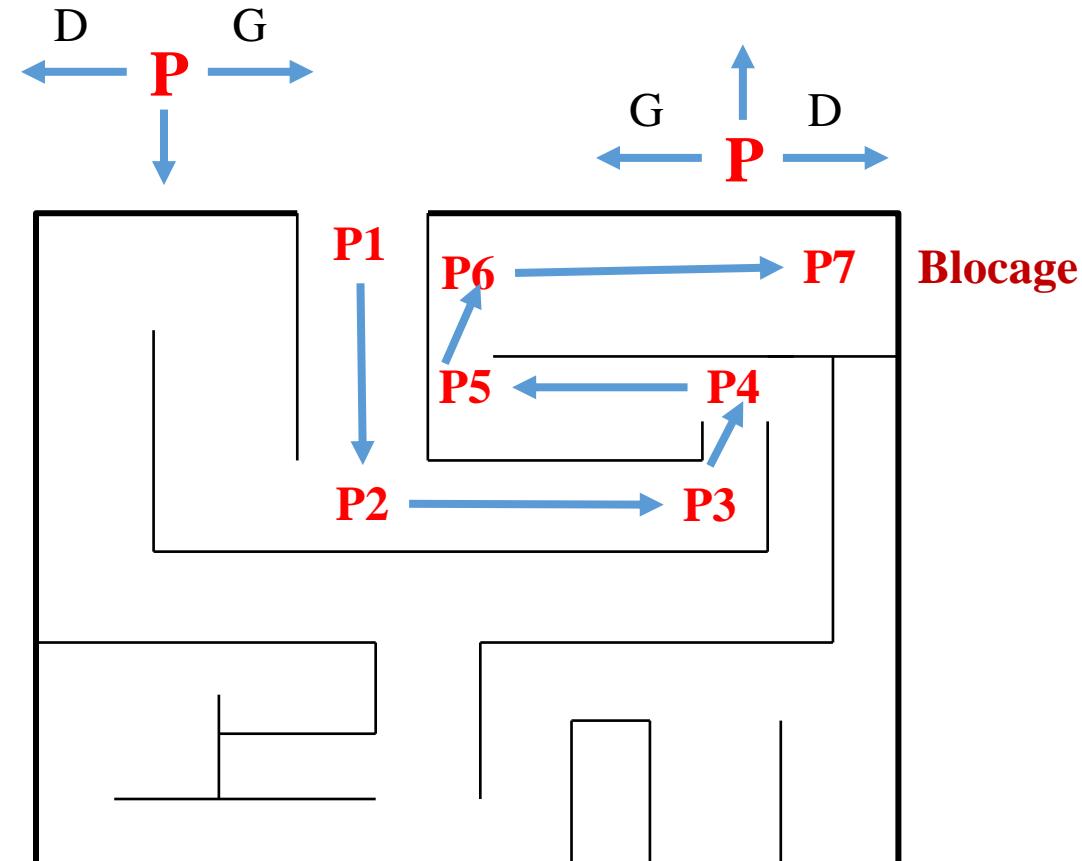
Modes de raisonnement

Exemple : chainage avant + choix irrévocable



Base de faits = {P1(Couloir), **P2(Intersection)**, P3(Coin),
P4(Intersection), P5(Coin), P6(Couloir)}

Règles applicables = **Blocage**



Moteur d'Inférence

Modes de raisonnement

Exemple : chainage avant

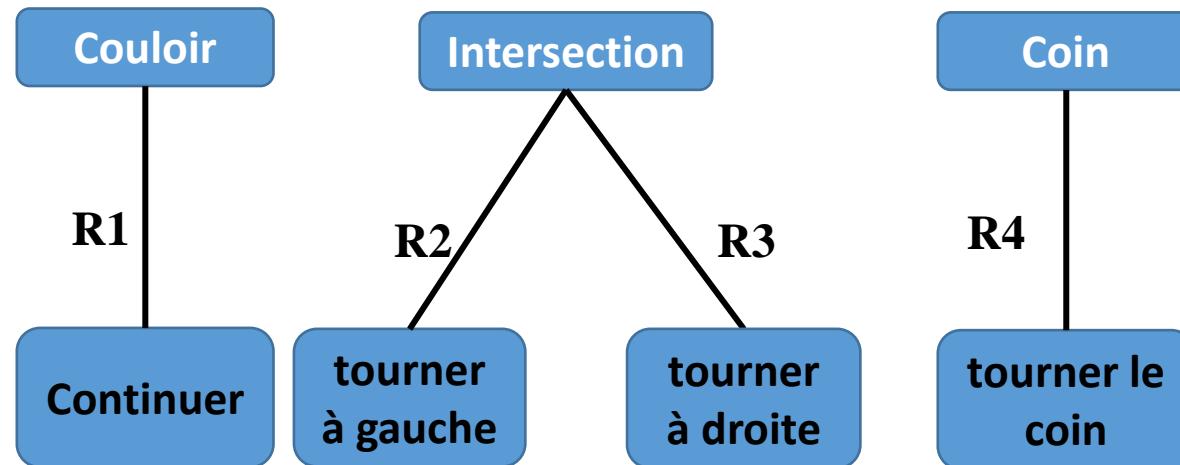
Choix irréversible : Toujours tourner à gauche aux intersections

Choix par tentatives: Tourner à gauche aux intersections sauf si on a une situation de blocage

Moteur d'Inférence

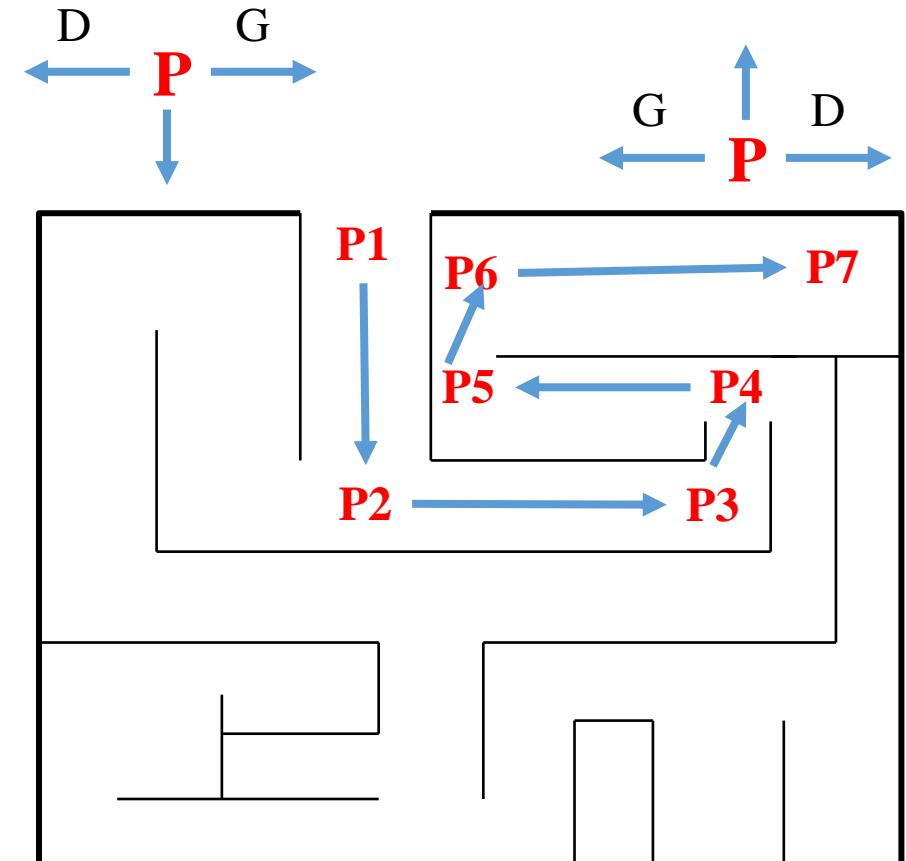
Modes de raisonnement

Exemple : chainage avant + Choix par tentatives



Base de faits = {P1(Couloir), **P2(Intersection)**, P3(Coin),
P4(Intersection), P5(Coin), P6(Couloir)}

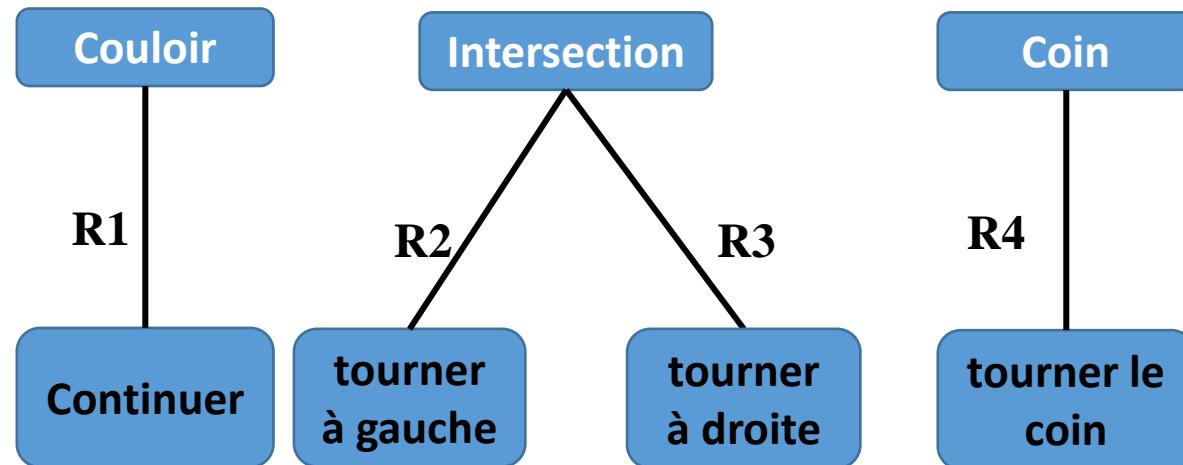
Règles applicable = **Blocage**



Moteur d'Inférence

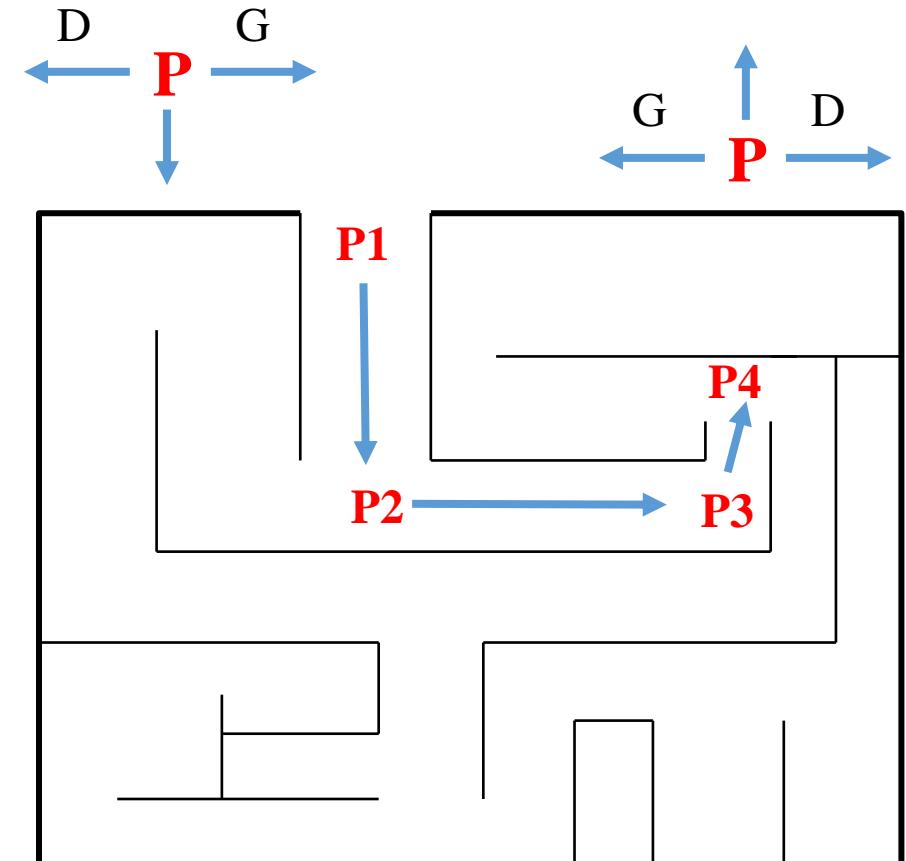
Modes de raisonnement

Exemple : chainage avant + Choix par tentatives



Base de faits = {P1(Couloir), **P2(Intersection)**, P3(Coin),
P4(Intersection)}

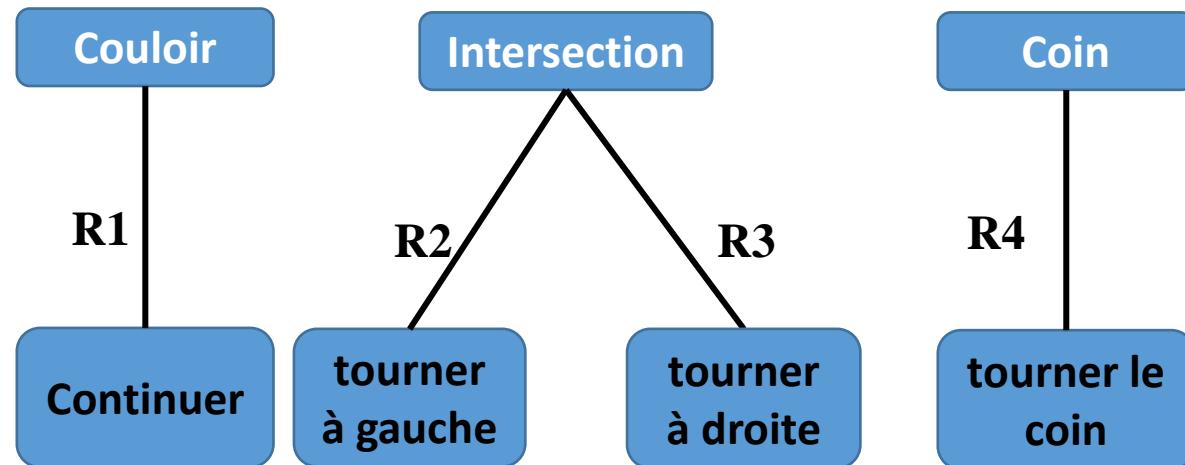
Règles applicable = {R2,R3}



Moteur d'Inférence

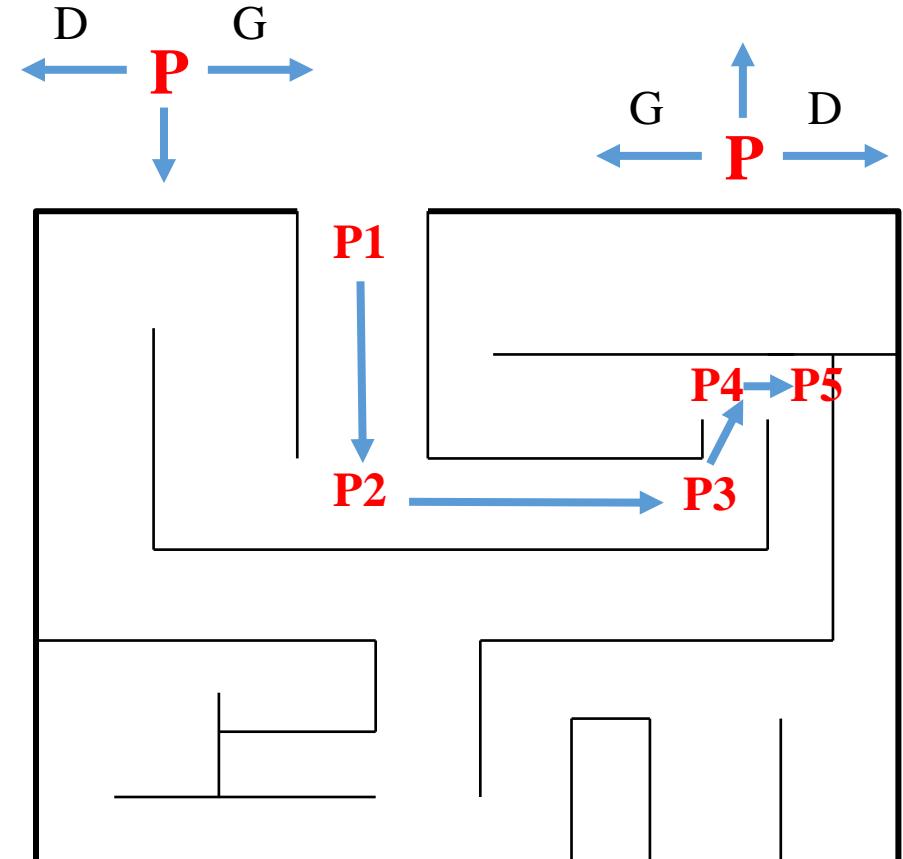
Modes de raisonnement

Exemple : chainage avant + Choix par tentatives



Base de faits = {P1(Couloir), **P2(Intersection)**, P3(Coin),
P4(Intersection)}

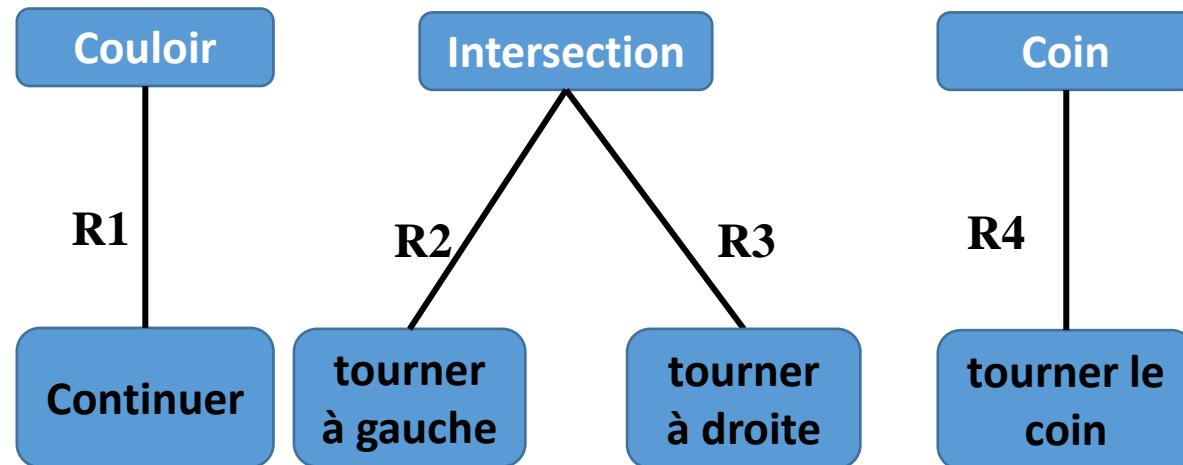
Règles applicable = {R2,R3}



Moteur d'Inférence

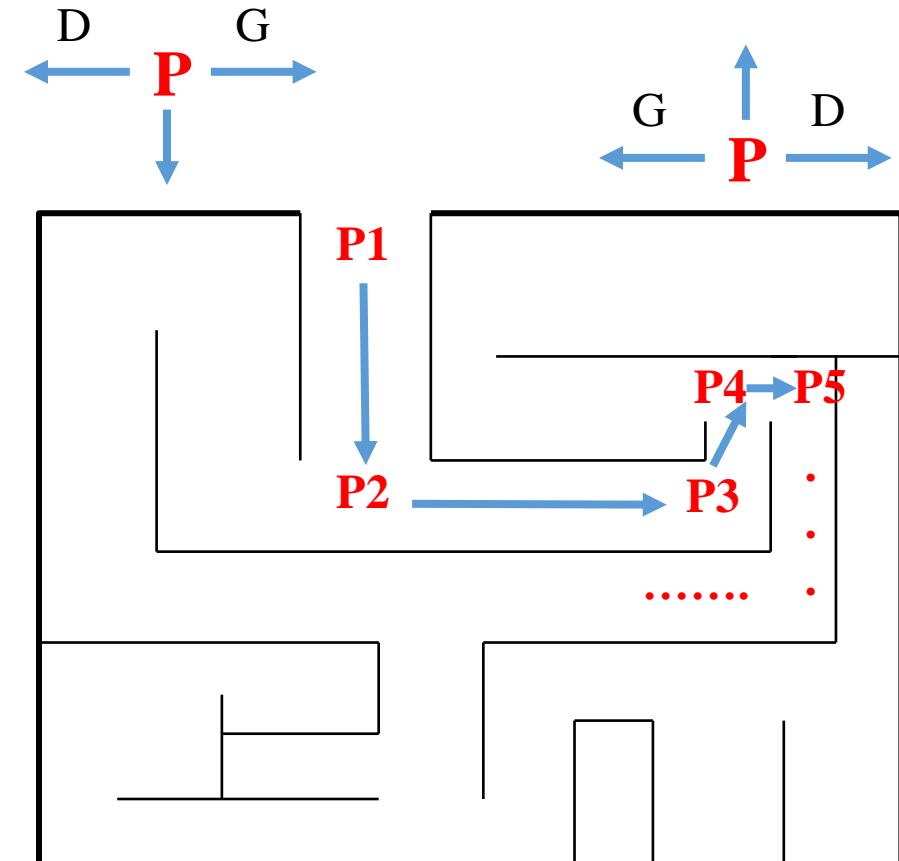
Modes de raisonnement

Exemple : chainage avant + Choix par tentatives



Base de faits = {P1(Couloir), **P2(Intersection)**, P3(Coin),
P4(Intersection),.....}

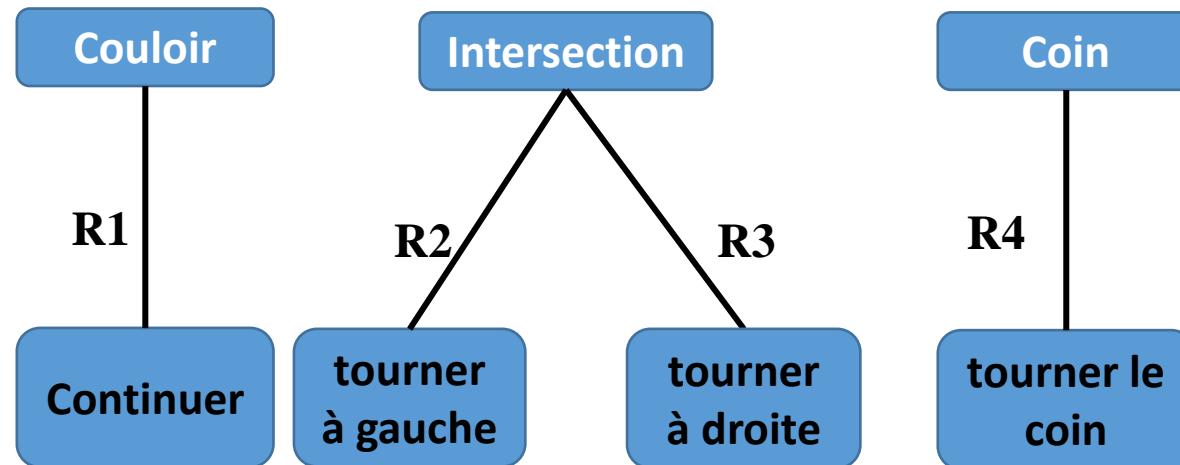
Règles applicable = {...}



Moteur d'Inférence

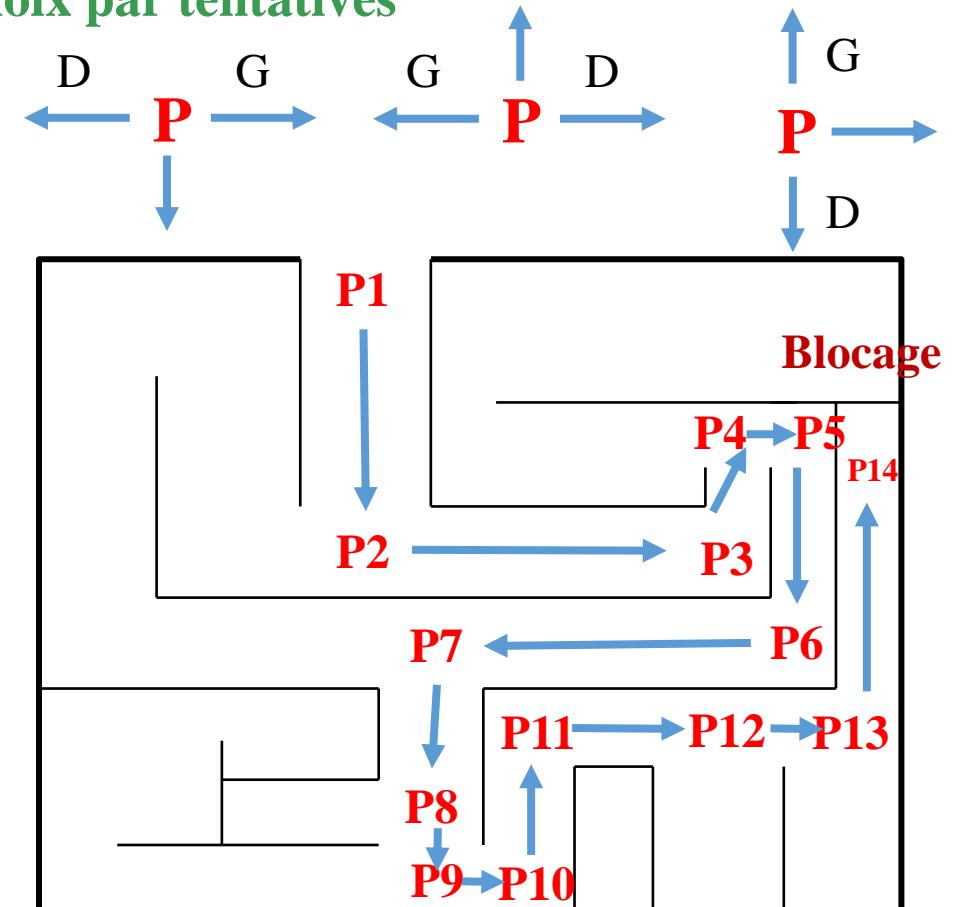
Modes de raisonnement

Exemple : chainage avant + Choix par tentatives



Base de faits = {P1(Couloir), **P2(Intersection)**, P3(Coin),
P4(Intersection),.....}

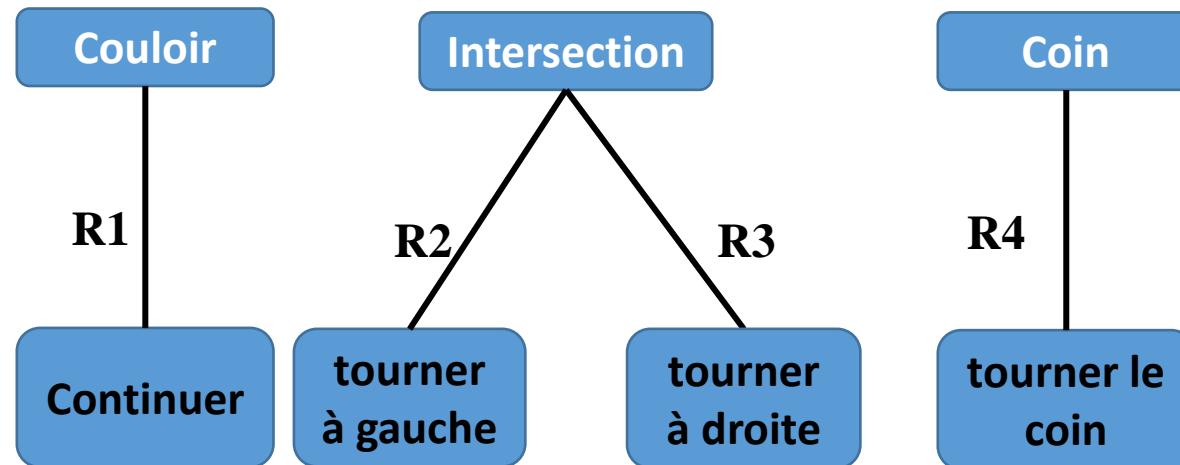
Règles applicable = {...}



Moteur d'Inférence

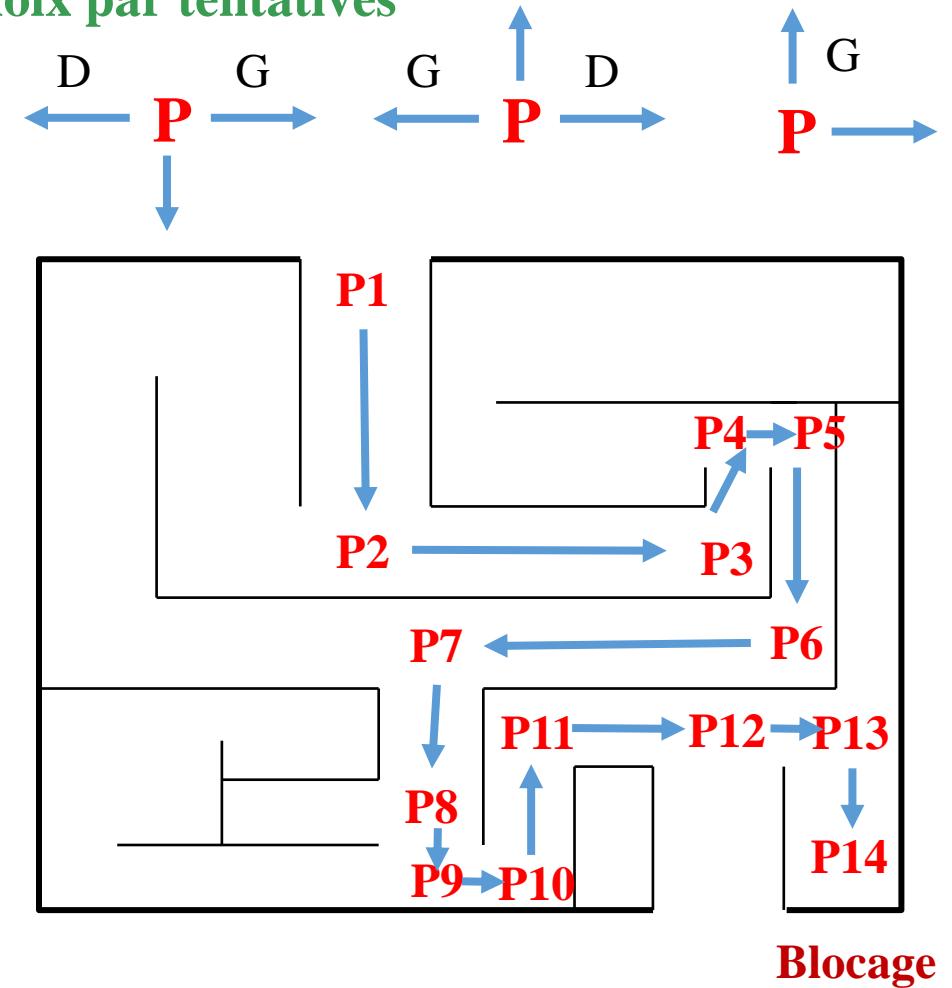
Modes de raisonnement

Exemple : chainage avant + Choix par tentatives



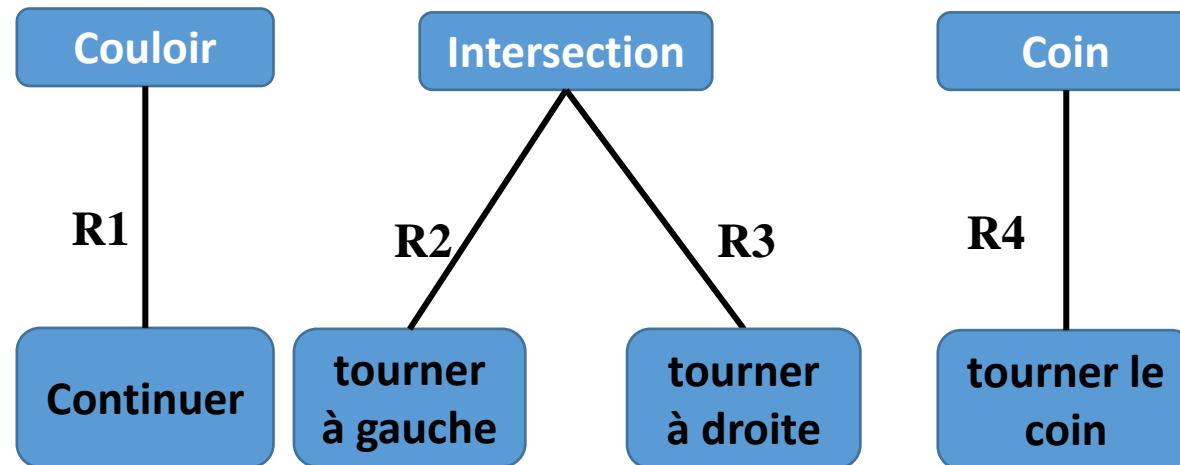
Base de faits = {P1(Couloir), **P2(Intersection)**, P3(Coin),
P4(Intersection),.....}

Règles applicable = {...}



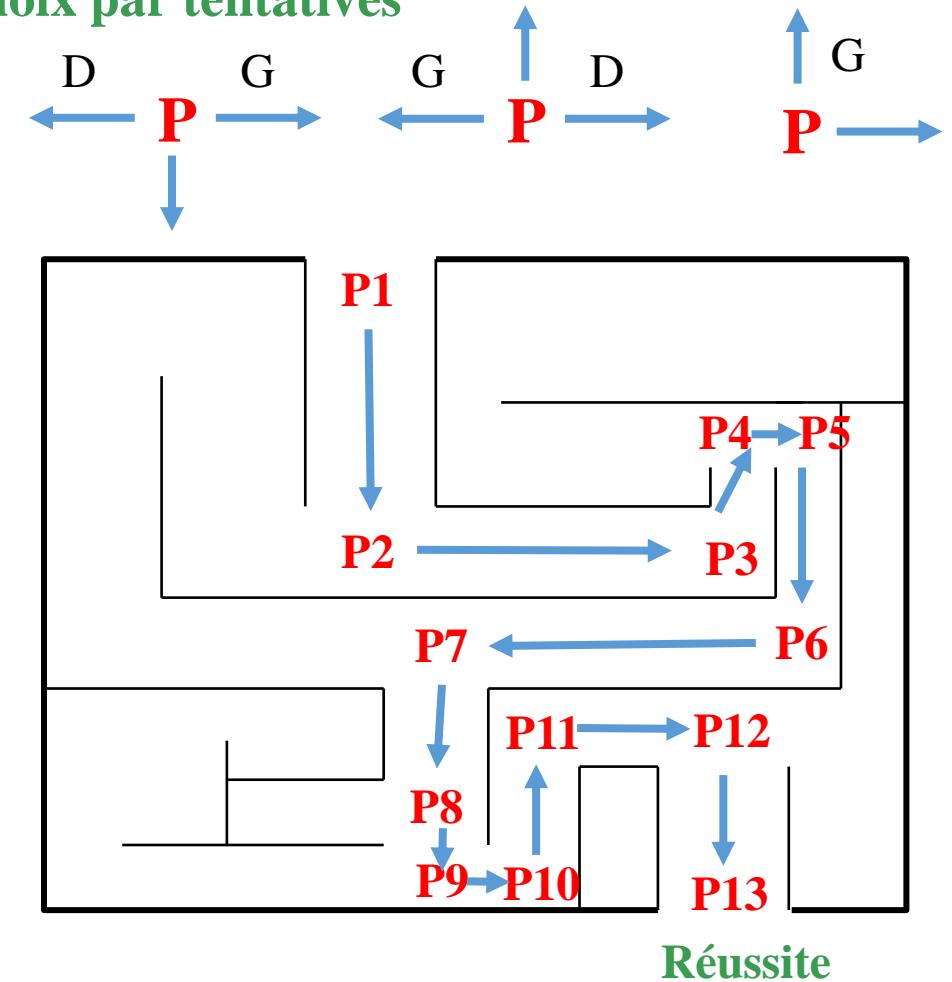
Modes de raisonnement

Exemple : chainage avant + Choix par tentatives



Base de faits = {P1(Couloir), **P2(Intersection)**, P3(Coin),
P4(Intersection),.....}

Règles applicable = {...}



Moteur d'Inférence

Exercice

Base de règles	Base de faits
R1 : Si E et B alors C	
R2 : Si B et D alors A	
R3 : Si J et H alors B	
R4 : Si D et E alors B	
R5 : Si B et D alors F	
R6 : Si E et F alors D	E, F

On cherche à démontrer C

Résoudre le problème suivant par chaînage arrière avec « backtracking »

Résolution de conflits : la règle de précédence (ordre d'écriture des règles)

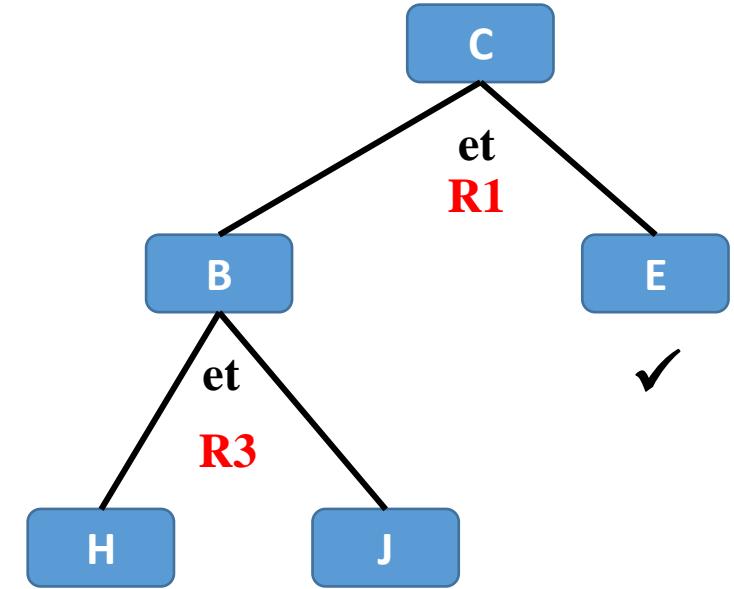
Moteur d'Inférence

Exercice

Base de règles	Base de faits
R1 : Si E et B alors C	
R2 : Si B et D alors A	
R3 : Si J et H alors B	
R4 : Si D et E alors B	
R5 : Si B et D alors F	
R6 : Si E et F alors D	

Situation de Blocage !

Appliquant un retour arrière (backtracking)
pour essayer la règle écartée **R4**



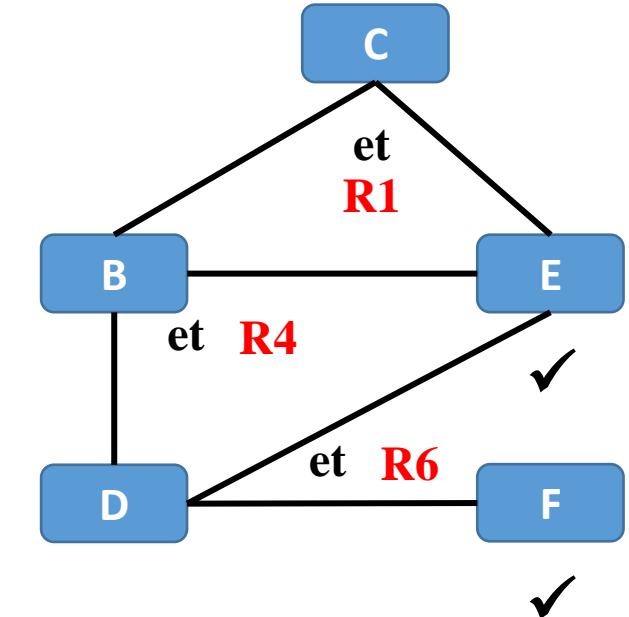
Cycle 1 : règles applicables {R1}

Cycle 2 : règles applicables {R3, R4}

Exercice

Base de règles	Base de faits
R1 : Si E et B alors C	
R2 : Si B et D alors A	
R3 : Si J et H alors B	
R4 : Si D et E alors B	
R5 : Si B et D alors F	
R6 : Si E et F alors D	

Base de faits finale : {E, F}
Il y a réussite



Situation de Blocage !

Appliquant un retour arrière (backtracking)
pour essayer la règle écartée R4

Cycle 1 : règles applicables {R1}

Cycle 2 : règles applicables {R3, R4}

Cycle 3 : règles applicables {R6}