

Projet Techniques de Programmation ITII P20 2011

Sujet : réalisation d'un système expert intégrant un moteur d'inférences utilisant des règles d'ordre 0.

**A rendre le 4 septembre 2010
A Stéphane VERA & Christian PAUL
Informations supplémentaires (paul.christian.emse@gmail.com)**

Tout les documents rapports, documentation et CD devront être remis dans une seule pochette

Spécifications

Le programme doit permettre de réaliser les actions suivantes :

Au niveau du moteur d'inférences :

- lancer le chaînage avant,
- lancer le chaînage arrière,

Au niveau de la base de règles :

- Créer,
- Modifier,
- Afficher,
- Sauvegarder dans un fichier au format spécifié, (voir à la fin du document)
- Charger depuis un fichier au format spécifié, (voir à la fin du document)

Au niveau de la base de faits :

- Créer,
- Modifier,
- Afficher,
- Sauvegarder dans un fichier au format spécifié, (voir à la fin du document)
- Charger depuis un fichier au format spécifié, (voir à la fin du document)

Au niveau d'une règle :

- Créer,
- Modifier,
- Supprimer,

Contraintes techniques

Parties du projet à écrire impérativement en C ANSI :

- La gestion des fichiers,
- L'analyse des fichiers,
- Le moteur d'inférences.

Partie du projet laissée à l'appréciation des concepteurs :

- Les interfaces d'écrans pour l'application générique,
- La formalisation d'un domaine d'expertise avec les règles de chaînage et la base de faits,
- Les interfaces applicatifs, si il y a une application spécifique utilisant un domaine d'expertise particulier

Techniques de programmation à utiliser impérativement :

- Modularité,
- Listes chaînées,
- Récursivité,

Environnement(s) cible(s) :

- Linux,
- Windows autorisé,

Documents à fournir

Ces documents sont à fournir à la première évaluation (voir plus loin)

1) Le rapport sur le projet contenant :

- L'analyse des besoins (retranscrivant le travail préalable au codage),
- Les spécifications de l'application (décrivant les choix techniques arrêtés, l'architecture du programme),
- La procédure de test mise en œuvre pour valider l'adéquation du programme au "cahier des charges"
- L'organisation du binôme sur les tâches de conception et de réalisation,
- Le calendrier prévisionnel et le calendrier réalisé,

2) Les documentations d'utilisation de l'application :

- Guide d'installation (et de désinstallation !),
- Guide d'utilisation.

3) Les sources

4) Le ou les exécutables

5) Le document de présentation du projet (voir plus loin).

Ces documents seront à fournir sur CD-ROM pour (1), (2), (3) et (4). Noter sur le CD-ROM les noms et prénoms du binôme, le numéro de promotion, ainsi que le nom du projet et sa version.

Une version papier est demandée pour (1), (2) et (3). Les rapports devront être correctement paginés et agrafés.

Le tout devra être rendu dans un dossier carton unique portant aussi les noms et prénoms du binôme ainsi que le numéro de promotion.

Améliorations intervenant dans la note (suggestions)

Un dictionnaire des littéraux utilisés dans la base de faits et dans la base de règles permet de vérifier la cohérence générale. Il peut être mis en œuvre à chaque chargement de nouvelles bases de règles et de faits.

Le travail sur une logique améliorée (coefficients de vraisemblance, ordre 0+, logique à 3 valeurs) permet de mettre en forme des problèmes plus complexes. Il s'agit de la spécifier et la mettre en œuvre.

D'autres améliorations peuvent intervenir dans l'évaluation du projet.

Evaluation du projet

Attention un projet n'est pas rattrapable, il ne peut pas être rendu en retard. Il faut donc prendre ses précautions pour rendre ce qui est demandé à la date de remise des projets.

Le travail de l'étudiant sur le projet est évalué en deux temps :

- Première évaluation : une présentation en binôme de 15 minutes permettant au correcteur d'évaluer globalement le projet, à cette occasion le CD-ROM du projet et toute la documentation est rendue.
- Deuxième évaluation : Un entretien individuel sur la base de la première évaluation. Il permet au correcteur de prendre la mesure de l'implication personnelle de chaque étudiant dans le projet. L'entretien individuel peut porter sur n'importe quel aspect du projet, l'architecture générale, les spécifications, le codage, la documentation ou la formalisation du domaine d'expertise.

La note de projet est donc attribuée individuellement et non au binôme.

Notation (N)

$10 \leq N < 12$: Le projet est mené à terme dans le respect du cahier des charges et des spécifications, la documentation est correcte. La présentation en binôme et la présentation individuelle montre que le sujet est maîtrisé.

$12 \leq N < 15$: **Le cahier des charges est respecté.** La réalisation est soignée et la présentation est bonne. Des améliorations, suggérées ou non, ont été produites. La présentation en binôme et la présentation individuelle montrent une bonne maîtrise du sujet.

$15 \leq N < 20$: **Le cahier des charges est respecté.** La réalisation est particulièrement soignée, la présentation excellente. Des améliorations, suggérées ou non, ont été produites. La présentation en binôme et la présentation individuelle montre une très bonne maîtrise du sujet.

$0 \leq N < 10$: **Le cahier des charges n'est pas respecté.** En particulier la présentation en binôme montre que les objectifs ne sont pas atteints ou les tests du correcteur montrent des erreurs de conception ou la présentation individuelle montre que l'élève ne s'est pas investi dans le projet.

Dates :

- Le 3 septembre 2011 : Première évaluation et fourniture des « livrables » du projet
- Le 1^{er} octobre 2011 pour l'entretien individuel

Format des fichiers

Grammaire associée aux bases de règles et aux bases de faits

Les symboles non terminaux sont notés entre < >. Les symboles terminaux sont notés entre simple cote '.

Les **commentaires** sont notés comme en C /* début de commentaire, */ fin de commentaire.

Les **symboles** entre crochets [] peuvent apparaître 0 ou 1 fois.

Les symboles entre crochets suivis d'une étoile []* peuvent apparaître 0 ou n fois.

Les caractères séparateurs des fichiers sont : l'espace, la tabulation et le retour chariot.

Base de règles ordre 0

```
<base de règles> ::= [<règle>]* /* différente en ordre 0+ */
<règle> ::= <hypothèse> [<hypothèse>]* -> <conclusion> ';'
/* nombre d'hypothèses maximal <= 6 */
<hypothèse> ::= <littéral>
<conclusion> ::= <littéral>
<identificateur> ::= <lettre> [<caractère admis>]*
/* longueur totale de la chaîne <= 32 */
<caractère admis> ::= <lettre> | '_' | <chiffre>
/* '_' est le caractère souligné */
<lettre> ::= 'a' | ... | 'z' | 'A' | .. | 'Z'
<chiffre> ::= '0' | ... | '9'
```

Ordre 0

```
<littéral> ::= <identificateur>
```

Ordre 0+

```
/* modifier l'expression du non terminal <base de règles> */
<base de règles> ::= [<règle>]* [<déclaration>]*
```

/ rajouter les éléments suivants à la grammaire d'ordre 0 */*

```
<signe> ::= '+' | '-'
<comparateur> ::= '=', '>', '<', '>=', '<=', '<>'
<littéral> ::= <identificateur> <comparateur> <valeur>
<valeur> ::= <entier relatif> | <identificateur>
<entier relatif> ::= [<signe>] <chiffre> [<chiffre>]*
<déclaration> ::= <identificateur> 'dans' '{' <identificateur> ['\,'
<identificateur>]* '\}' ';' /* nombre d'identificateurs <= 32 */
<déclaration> ::=
    <identificateur> dans { <entier relatif> [, <entier relatif>]* } ;
/* nombre d'entiers relatifs <= 32 */
<déclaration> ::= <identificateur> 'dans' '[' <borne inf> ',' <borne
sup> ']' ';'
<borne inf> ::= <entier relatif>
<borne sup> ::= <entier relatif>
```

Base de faits

```
<base de faits> ::= <fait> [ <fait> ]*
```

Faits ordre 0

```
<fait> ::= <littéral> ';' ;
```

Faits ordre 0+

```
<fait> ::= <identificateur> <comparateur> <valeur> ;
```

Annexes sur les notions relatives aux systèmes experts et au format des fichiers

Positionnement du problème : Système Expert et Intelligence Artificielle

Les systèmes experts sont issus des recherches informatiques en I.A (Intelligence Artificielle). Ce domaine a pour objet d'étudier et de reproduire les processus cognitifs élaborés que l'on met en œuvre dans la compréhension et la résolution de problèmes complexes. La communication en langage naturel, l'activité de diagnostic au sens large, les capacités de prévisions font en particulier l'objet de ces recherches.

Qu'est-ce qu'un système expert (SE) ?

C'est un système informatique qui reproduit certains raisonnements humains pour résoudre des problèmes sur un domaine précis. La définition actuelle est plus restrictive puisqu'elle prend maintenant en compte la structure même du système informatique.

Fondements théoriques des systèmes experts

Le type de raisonnement couramment reproduit par les systèmes experts est le raisonnement logique basé sur une règle d'inférence fonctionnant de la sorte : "Si A est vrai" et "Si A implique B", alors on peut déduire "B est vrai".

On remarque que la proposition logique "A implique B" ne prouve pas que B est vrai. Pour cela, il faut que la prémisse de l'implication soit vraie.

Systèmes formels basé sur la logique : exemple de règles d'inférences

Modus ponens (chaînage avant)

Si A et $(A \rightarrow B)$ Alors on **déduit** B (noté A, $(A \rightarrow B) \vdash B$)

Modus tollens (chaînage arrière)

Si $\neg B$ et $(A \rightarrow B)$ Alors on **déduit** $\neg A$

Enchaînement

Si $A \rightarrow B$ et $B \rightarrow C$ alors on **déduit** $A \rightarrow C$

Propriétés d'un système formel :

- Si la règle d'inférence ne produit que des phases vraies : Elle est cohérente
- Si la règle d'inférence produit toutes les phases vraies : Elle est complète
- Une théorie complète est décidable s'il existe un algorithme qui permet de déterminer si un énoncé quelconque est (logiquement) vrai ou non.

Structure d'un système expert

Un SE est composé d'un *moteur d'inférences* basé sur une règle d'inférence et capable de déduire (modus ponens) ou de vérifier (modus tollens) des connaissances à partir d'une *base de connaissances* et d'une *base de faits*.

Base de connaissances

Elle contient les connaissances du domaine précis ou domaine d'expertise. C'est en général un expert du domaine (sur du diagnostic médical, ce sera un médecin spécialisé) qui la mis en forme aidé éventuellement par un cogniticien spécialisé dans la formalisation logique des problèmes. On dit que les connaissances sont représentées sous formes de règles, C'est pourquoi la base de connaissances est souvent appelée base de règles.

Base de faits

Elle contient les connaissances relatives aux problèmes à résoudre.

Moteur d'inférences

A partir des faits et de la base de connaissances, il produit le ou les résultats qui pourront être vus comme des réponses à une question complexe (diagnostic). Le chaînage avant permet de produire toutes les conclusions possibles à partir d'un ensemble de faits constatés. Le chaînage arrière part de la conclusion supposée et la valide à partir des faits précédemment constatés.

Compréhension du chaînage avant (diagnostic médical) : Etant donné les symptômes suivants ... (saisis dans la base de faits) déduire toutes les pathologies possibles. C'est l'utilisation du *modus ponens*.

Compréhension du chaînage arrière (diagnostic médical) : Est-ce que la pathologie ... est envisageable avec les symptômes suivants (saisis dans la base de faits) ?

Notion d'ordre

Dans les bases de connaissances, les règles sont de la forme "si A et B et C alors D" où A, B, C et D sont des atomes de connaissances. Ces atomes ne peuvent prendre que deux valeurs {Vrai, Faux}. On dit qu'un système expert est d'ordre 0, 0+ ou 1 suivant la nature des atomes de connaissance qu'ils manipulent.

Ordre 0 : Un atome est un nom symbolique : fièvre, fièvre-élevée, pas-de-fièvre. Remarque à cet ordre le calcul des propositions est complet et décidable. Ce qui signifie qu'il existe un algorithme capable de produire et de reconnaître toutes les connaissances vraies à partir des connaissances initiales.

Ordre 0+ : Un fait est un quadruplet <objet, attribut, comparateur, valeur>. L'attribut de l'objet prend ses valeurs dans un domaine de définition. Le moteur d'inférences, que ce soit en chaînage avant ou en chaînage arrière, va affecter des valeurs ou des plages de valeurs à chaque inférences. Il s'agira de "propager" ces valeurs ou cet ensemble de valeurs au cours de la résolution. Il y a échec à la résolution lorsque l'ensemble de valeur devient vide. Dans la pratique, objet et attribut peuvent être remplacés par une variable : fièvre = 40, fièvre > 38, fièvre = 37.

Ordre 1 : Un atome de connaissance est une relation, fièvre(X) ou X peut être le patient et fièvre(X) est vrai suivant le patient.

Le pouvoir d'expression des bases de connaissances et des bases de faits augmente avec l'ordre. Il est plus simple de décrire une situation en ordre 0+ ou en ordre 1 qu'en ordre 0. Le moteur d'inférences qui y est associé est plus ou moins complexe car il doit gérer des variables et propager au cours de la résolution des valeurs. A chaque inférence il faut une méthode pour recalculer l'ensemble de valeurs. A l'ordre 1 c'est l'algorithme d'unification (non décrit ici) qui prend en charge ce calcul.

Détails de mise en œuvre (dans le cadre du projet IA)

Base de règles et base de faits

Les règles sont formées d'une partie si, les prémisses ou hypothèses, et d'une partie alors, la conclusion. Exemple :

Si Hypothèse1 et Hypothèse2 alors Conclusion
Est une règle.

La base de règles et les base de faits suivantes illustrent la démarche de fonctionnement d'un système expert.

Base-de-règles = { si soleil et été alors plage ; si hiver et neige alors ski }

Base-de-faits1 = { soleil ; neige }

Base-de-faits2 = { soleil ; été }

Base-de-faits3 = { hiver ; neige }

A partir de la base de faits n°1 on ne peut rien déduire. A partir de la base de faits n°2 on peut déduire que plage est vrai. A partir de la base de faits n°3 on peut déduire que ski est vrai.

Les atomes de connaissances constituant les hypothèses et les conclusions des règles ainsi que les faits : soleil, été, plage, hiver, neige, ski ; sont aussi appelés des littéraux. On considère qu'il ne peuvent prendre que deux valeurs : vrai ou faux. Tout littéral apparaissant dans la base de fait prend la valeur vrai. Tout littéral obtenu à partir de la base de faits par enchaînement de règles devient vrai. Dans la pratique, c'est à dire en dehors de la logique pure qui les considérerait comme faux, ils peuvent être inconnus ou avoir une valeur indéterminée. Dans l'exemple précédent les littéraux plage et ski avec la base de faits n°1 ont une valeur indéterminée.

Il existe deux manières de vérifier si un littéral est vrai (déductible) de la base de faits et de la base de règles :

- Par déduction c'est à dire chaînage arrière,
- Par induction c'est à dire chaînage avant.

Explications sur le chaînage avant

L'algorithme de chaînage avant part des faits de la base de faits et produit tous les nouveaux faits, littéraux vrais, à partir des règles de la base de règles. Suivant que l'on veut produire tous les faits vérifiables ou que l'on veut simplement vérifier une seule assertion, l'algorithme s'arrêtera de manière différente.

Dans tous les cas la production de nouveaux faits ou de faits déduits suit le principe exposé ci-dessous :

Pour une règle donnée si toutes les hypothèses sont dans la base de faits alors la conclusion est vérifiée (c'est à dire que le littéral apparaissant dans la partie conclusion est

vrai). La conclusion peut être rajoutée à la base de faits. Si l'une au moins des hypothèses n'appartient pas à la base de faits, la conclusion ne peut pas être déduite de la règle et de la base de faits.

Evidemment au fur et à mesure que l'on produit de nouveaux faits, d'autres règles sont utilisables. Dans tous les cas l'algorithme s'arrête quand il n'est plus possible de produire de nouveaux faits.

Explications sur le chaînage arrière (ordre 0)

L'algorithme de chaînage arrière tente de vérifier qu'une question peut-être déduite d'un ensemble de faits à partir des règles de la base de règle. Pour ce faire, à chaque étape de la vérification il faut :

- 1) Vérifier si la question est un fait de la base de fait
ou
- 2) pour chaque règle dont la conclusion correspond à la question courante, vérifier que toutes les hypothèses de la règle peuvent être déduites de l'ensemble de faits à partir des règles de la base de règles.

L'algorithme ci-dessous détaille la démarche. But est la question courante, Base-de-règles est la base de règles, Base-de-faits est la base de faits.

Chaînage arrière (But, Base-de-règles, Base-de-faits)

Début

Résultat ← faux

Si But n'est pas dans Base-de-faits **alors**

{ Règle ← première(Base-de-règles)

Tant que Base-de-règles non vide **et** Résultat = faux **faire**

{ **Si** Conclusion (Règle) = But **alors**

{ */*Vérifier que toutes les Hypothèses peuvent être vérifiées
à partir de la Base-de-faits */*

Hypothèse ← première-hypothèse(Règle)

Continue ← vrai

Tant que Existe(Hypothèse) **et** Continue **faire**

{

Continue = Chaînage arrière (Hypothèse,
Base-de-règles, Base-de-faits)

Hypothèse ← hypothèse-suivante(Règle)

} fin tant que

/ Si toutes les Hypothèses sont vérifiées la conclusion est vérifiée
Sinon la conclusion n'est pas vérifiée */*

Résultat = Continue

} fin si

Règle ← suivante(Base-de-règles)

}fin tant que

} Sinon */* But est dans Base-de-faits */*

Résultat ← vrai

Retourne (Résultat)

Fin

Format des fichiers Base de règles et Base de faits

Afin de bien montrer le découpage entre moteur d'inférences et bases de connaissances (base de règles et base de faits) le format des fichiers pour les bases de faits et pour les bases de règles est donné pour l'ordre 0, à titre indicatif pour l'ordre 0+.

Il est impératif que les applications développées respectent cette contrainte et permettent de charger / sauvegarder des fichiers ayant ce format.

Les deux exemples ci-dessous pourront être utilisés pour les essais du système expert avant de vous lancer dans la formalisation d'un problème. La grammaire in-extenso est livrée en dessous. La notation B.N.F est utilisée pour la représentation du langage.

Exemple 1

Règles :

```
B D E -> F ;
G D -> A ;
F C -> A ;
B -> X ;
E -> D ;
A X -> H ;
C -> D ;
X C -> A ;
X B -> D ;
```

Faits :

```
B ;
C ;
```

A titre d'entraînement : trouver tous les faits déduits des faits initiaux. Cet exemple est à utiliser dans cet ordre pour vérifier que le chaînage arrière marche même en cas d'erreur dans une branche de la résolution.

Exemple 2

```
animal_à_poil -> animal_est_mammifère ;
animal_donne_lait -> animal_est_mammifère ;
animal_à_plumes -> animal_est_oiseau ;
animal_vole animal_pond_des_œufs -> animal_est_oiseau ;
animal_mange_de_la_viande -> animal_est_carnivore ;
animal_à_dents_pointues animal_à_griffes animal_a_des_yeux_vers_avant ->
animal_est_carnivore ;
animal_est_mammifère animal_a_des_sabots -> animal_est_ongulé ;
animal_est_mammifère animal_rumine -> animal_est_ongulé ;
animal_est_mammifère animal_est_carnivore animal_a_couleur_brune
animal_a_tâches_sombres -> animal_est_guépard ;
animal_est_mammifère animal_est_carnivore animal_a_couleur_brune
animal_a_raies_noires -> animal_est_tigre ;
animal_est_ongulé animal_a_un_long_coup animal_a_des_longues_pattes
animal_a_des_tâches_sombres -> animal_est_girafe ;
animal_est_ongulé animal_a_des_raies_noires -> animal_est_zèbre ;
animal_est_oiseau animal_ne_vole_pas animal_a_un_long_coup
animal_a_des_longues_pattes animal_est_noir_et_blanc -> animal_est_autruche
;
animal_est_oiseau animal_ne_vole_pas animal_nage animal_est_noir_et_blanc -
> animal_est_pingouin ;
animal_est_oiseau animal_vole_bien -> animal_est_albatros ;
```

A titre d'entraînement : trouver les faits initiaux qui rendent vrai les faits suivants

1. animal_est_albatros;
2. animal_est_pingouin;
3. animal_est_autruche;
4. animal_est_zèbre;
5. animal_est_girafe;
6. animal_est_tigre;
7. animal_est_guépard;