Outils d'Analyse d'une Base de Règles

Hadrien Negros, Swan Rocher

14 février 2012

Table des matières

Chapitre 1

1.1 Définitions

1.1.1 Prédicat

- Un prédicat noté $p\backslash n$ est un symbole relationnel d'arité n.
- On suppose que tout nom de prédicat est unique.
- On note p_i la i^{eme} position de p.
- On note P_K l'ensemble des prédicats de K.

1.1.2 Atome

Un atome $a = p(a_1, a_2, ..., a_n)$ associe un terme à chaque position d'un prédicat $p\backslash n$. On note :

- a_i le terme en position i dans a. Un terme peut être une *constante* ou une *variable*. Une variable peut être universelle (notée $\forall - var$) ou existentielle (notée $\exists - var$).
- $dom(a) = \{a_i : \forall i \in [1, n]\}$, l'ensemble des termes de a
- $var(a)$ l'ensemble des variables de a
- $cst(a)$ l'ensemble des constantes de a

Ces notations s'étendent aux *conjonctions d'atomes*.

1.1.3 Règle

Une règle $R = (H, C)$ est définie par :

- H : conjonction d'atomes représentant l'hypothèse de R,
- C : conjonction d'atomes représentant la conclusion de R.

On note :

- $dom(R) = dom(H) \cup dom(C)$
- $var(R) = var(H) \cup var(C)$
- $cst(R) = cst(H) \cup cst(C)$
- $fr(R)$ l'ensemble des variables frontières de R, avec a_i est une variable frontière ssi a_i est une variable telle que $a_i \in var(H)\ intersection\ var(C)$.
- $cutp(R) = fr(R) \cup cst(R)$ l'ensemble des points de coupure de R

1.1.4 Représentation des Conjonctions d'Atomes

Une conjonction d'atomes $K = k_1 k_2 ... k_n$ avec $k_i = q(t_1, t_2, ..., tp)$ peut être représentée par le graphe non orienté $G_K = (V_K, E_K)$ avec V_K son ensemble de sommets et E_K son ensemble d'arêtes construits de la manière suivante :
$V_K = A_K \cup T_K$ avec $A_K = \{i : k_i \in K\}$ et $T_K = \{t_j \in dom(K)\}$
$E_K = \{(i, t_j) : \forall k_i \in K, \forall t_j \in dom(k_i)\}$

On remarque que quelques propriétés sont directement induites par cette représentation :

- G_K admet une bipartition de ses sommets, en effet tous les arcs ont une extrémité dans A_K etre l'autre dans T_K, or par construction $A_K intersection T_K = \emptyset$.

1.1.5 Implémentation du Graphe

Par la suite on supposera que le graphe G_K est implémenté par deux ensembles de sommets (les atomes et les termes), ainsi que par une liste de voisinage pour chaque sommet. De plus, chaque sommet peut avoir connaissance de son type de contenu parmis : $atome$, $terme$, $constante$, $\forall - var$, $\exists - var$ en temps constant.

On note n (resp. t) le nombre d'atomes (resp. de termes) dans K.

Opérations (et temps d'exécution associé) :

- Parcourir les sommets *atomes* : n.
- Parcourir tous les sommets : $n + t$.
- Accéder au i^{eme} voisin d'un sommet : *constant*.

1.1.6 Substitution

Une substitution de taille n d'un ensemble de symboles X dans un ensemble de symboles Y est une fonction de X vers Y représentée par l'ensemble de couples suivants (avec $n \leq |X|$) :

- $s = \{(x_i, y_i) \ : \ \forall i \in [1, n] \ x_i \in X, \ y_i \in Y, \forall j \neq i \ x_i \neq x_j\}$
- $s(x_i) = y_i \ \forall i \in [1, n]$
- $s(x_i) = x_i \ \forall i \in [n + 1, |X|] \ x_i \in X$

1.1.7 Unificateur logique

Un unificateur logique entre deux atomes a_1 et a_2 est une substitution μ telle que :

- $\mu : var(a_1) \cup var(a_2) \rightarrow dom(a_1) \cup dom(a_2)$
- $\mu(a_1) = \mu(a_2)$

Cette définition s'étend aux conjonctions d'atomes.

1.1.8 Unificateur de conclusion atomique

Un unificateur de conclusion atomique est un unificateur logique $\mu = \{(x_i, t_i) : \forall i \in [1, n]\}$ entre l'hypothèse d'une règle $R_1 = (H_1, C_1)$ et la conclusion atomique d'une règle $R_2 = (H_2, C_2)$ et est défini de la manière suivante :

- $\mu : fr(R_2) \cup var(H_1) \rightarrow dom(C_2) \cup cst(H_1)$
- $\forall (x_i, t_i) \in \mu \ si \ x_i \in fr(R_2) \ alors \ t_i \in cutp(R_2) \cup cst(H_1)$

1.1.9 Bonne unification atomique

Un bon unificateur de conclusion atomique est un unificateur de conclusion atomique $\mu = \{x_i, t_i) : \forall i \in [1, n]\}$ entre un sous ensemble Q de l'hypothèse d'une règle $R_1 =$

(H_1, C_1) et la conclusion d'une règle atomique $R_2 = (H_2, C_2)$ tel que :
$\forall (x_i, t_i) \in \mu : si\ x_i \in H_1 - Q\ alors\ t_i\ n'est\ pas\ une\ \exists - var$

Un tel ensemble Q est appelé un *bon ensemble d'unification atomique.*

Un *bon ensemble d'unification atomique minimal* Q de H_1 par C_2 enraciné en a est défini comme suit :

- Q est un bon ensemble d'unification atomique de H_1 par C_2
- $a \in Q$
- $|Q| = min(|Q_i| : Q_i\ est\ un\ bon\ ensemble\ d'unification\ atomique\ de\ H_1\ par\ C_2\ et\ a \in Q_i)$

1.2 Unification de Règles

1.2.1 Algorithme d'Unification

Algorithm 1 Unification

Require: H_1 : conjonction d'atomes, $R = (H_2, C_2)$: règle à conclusion atomique
Ensure: succès si C_2 peut s'unifier avec H_1, i.e. si $\exists H\ contenu\ H_1, \mu\ une\ substitution : \mu(H_1) = \mu(C_2)$, échec sinon

```
1  ▷ Précoloration
2  for all sommet atome a ∈ H1 do
3      if UnificationLocale(a, R) ≠ échec then
4          couleurLogique[a] ← noir
5      else
6          couleurLogique[a] ← blanc
7      end if
8  end for
9  ▷ Initialisation du tableau contenant les positions des variables existentielles de C2
10 E ← {i : ci est une ∃ − var de C2}
11 ▷ Extension des ensembles
12 for all sommet atome a ∈ H1 : couleurLogique[a] = noir do
13     if Q ← Extension(H1, a, couleurLogique, E) ≠ échec then
14         if UnificationLocale(Q, R) ≠ échec then
15             return succès
16         end if
17     end if
18     couleurLogique[a] ← blanc
19 end for
20 return échec
```

1.2.2 Extension des Conjonctions d'Atomes

Algorithm 2 Extension

Require: H_1 : conjonction d'atomes, $a \in H_1$: sommet atome racine, *couleurLogique* : tableau de taille égal au nombre d'atomes dans H_1 tel que $couleurLogique[a] = noir$ ssi $UnificationLocale(a, R) =$ succès, E : ensemble des positions des variables existentielles

Ensure: Q : bon ensemble d'unification minimal des atomes de H_1 construit à partir de a s'il existe, échec sinon.

```
1  ▷ Initialisation du parcours
2  for all sommet atome a ∈ H1 do
3      if couleurLogique[a] = noir then
4          for all sommet terme t ∈ voisins(a) do
5              couleur[t] = blanc
6          end for
7          couleur[a] = blanc
8      end if
9  end for
10 couleur[a] ← noir
11 Q ← {a} ▷ conjonction d'atomes à traiter
12 attente ← {a} ▷ file d'attente du parcours
13 while attente ≠ ∅ do
14     u ← haut(attente)
15     if u est un atome then
16         for all i ∈ E do
17             v ← voisin(u, i)
18             if v est une constante then
19                 return échec
20             else if couleur[v] = blanc then
21                 ▷ v est une ∀ − var non marquée par le parcours
22                 couleur[v] ← noir
23                 attente ← attente ∪ {v}
24             end if
25         end for
26     else
27         ▷ u est un terme
28         for all v ∈ voisins(u) do
29             if couleur[v] = blanc then
30                 if couleurLogique[v] = blanc then
31                     return échec
32                 else
33                     couleur[v] ← noir
34                     attente ← attente ∪ {v}
35                     Q ← Q ∪ {v}
36                 end if
37             end if
38         end for
39     end if
40 end while ▷ Fin du parcours
41 return Q
```

Remarque :

La phase d'initialisation du parcours pourrait simplement parcourir tous les sommets de H_1 et mettre leur couleur à blanc. En pratique, cette solution serait sans doute plus efficace, mais dépendrait donc du nombre de sommets total dans H_1. Ce qui en théorie

amènerait la complexité de cette boucle en $\bigcirc(nombre\ d'atomes\ \times arite\ max\ de\ H_1)$. Or ici, la complexité ne dépend pas de cette arité max, mais uniquement de l'arité du prédicat de la conclusion C_2.

1.2.3 Algorithme d'Unification Locale

Algorithm 3 UnificationLocale

Require: H_1 : conjonction d'atomes, $R = (H_2, C_2)$: règle à conclusion atomique
Ensure: succès si C_2 peut s'unifier avec H_1

▷ Vérification des prédicats
for all *atome* $a \in H_1$ **do**
  if prédicat(a) $\neq$ prédicat(C_2) **then**
    return échec
  end if
end for
$u \leftarrow \emptyset$ ▷ substitution
for all *terme* $t_i \in C_2$ **do**
  ▷ def : a_i = terme de a en position i
  $E \leftarrow \{a_i : \forall\ atome\ a \in H_1\}$
  if t_i *est une constante* **then**
    if $\exists\ v \in E : v$ *est une constante et* $v \neq t_i$, *ou* v *est une* $\exists - variable$ **then**
      return échec
    else
      $u \leftarrow \{(v, t_i) : v \in E\ et\ v \neq t_i\}$
    end if
  else if t_i *est une* $\exists - variable$ **then**
    if $\exists\ v \in E : v$ *est une* $\exists - variable$ *et* $v \neq t_i$, *ou* v *est une constante* **then**
      return échec
    else
      $u \leftarrow \{(v, t_i) : v \in E\ et\ v \neq t_i\}$
    end if
  else
    if $\exists\ v_1, v_2 \in E : v_1 \neq v_2\ et\ v_1, v_2$ *ne sont pas des* $\forall - variables$ **then**
      return échec
    else if $\exists\ c \in E : c$ *est une constante* **then**
      $u \leftarrow \{(v, c) : v \in E \cup \{t_i\}\ et\ v \neq c\}$
    else
      $u \leftarrow \{(v, t_i) : v \in E\ et\ v \neq t_i\}$
    end if
  end if
  $H_1 \leftarrow u(H_1)$
  $C_2 \leftarrow u(C_2)$
end for
return succès

1.2.4 Correction

Propriétés :

– Soit $R = (H, C)$ une règle à conclusion atomique,
Soit $K = (k_1, k_2, ..., k_n)$ une conjonction d'atomes,
si $\exists a \in K$: a ne peut pas s'unifier avec C alors K ne peut pas s'unifier avec C

1.2.5 Complexites

On note :
– n = nombre d'atomes dans H_1
– p = arité de C_2
– t = nombre de termes "colorables" dans H_1
– m = nombre d'arêtes "suivables" dans H_1
On remarque que dans le pire des cas on a :
– $t = n \times p$
– $m = n \times p$

Temps

– UnificationLocale (pire des cas) = $\bigcirc(np)$
– Extension (pire des cas) = $\bigcirc(np)$
– Unification (pire des cas) = $\bigcirc(n^2p)$

Extension
On effectue simplement un parcours en largeur à partir d'un sommet donné qui peut éventuellement s'arrêter plus tôt qu'un parcours classique. La complexité en temps dans le pire des cas est donc au plus la même, c'est à dire linéaire au nombre de sommets plus le nombre d'arcs. Le graphe représentant la conjonction d'atomes H_1 possèdent $n \times ariteMax(H_1)$ arêtes et $n \times (ariteMax(H_1) + 1)$ sommets. On sait donc que $C^{temps}_{Extension} = \bigcirc(n \times ariteMax(H_1))$.

Deux cas :
i) Découverte d'un sommet atome :
parcours classique => on récupère tous les voisins blancs
parcours extension => on récupére tous les voisins blancs de couleur logique noire (en effet arrêt immédiat si un voisin de couleur logique noire est découvert.

ii) Découverte d'un sommet variable :

pas de différence

On a donc que l'algo ne suit que les sommets atomes pouvant être unifies localement. C'est à dire (entre autres) que leur prédicat est égal à prédicat(C_2).

Le parcours suit au pire $2(n \times p)+1$ arêtes. Supposons que l'algorithme ait déjà parcouru $2(n \times p)$ arêtes ...

Espace

Chapitre 2

Bibliographie