lundi 13 janvier 2020

Théorie Logique

Logique propositionnelle

- Algorithme de preuve

Entrée C1,….,Cn les clauses et C le théorème à prouver

while $false \notin S$ *et* $\exists$ *? clauses résolvables non résolues* **do**
  — choisir $C_1, C_2 \in S$ tel que $\exists P \in C_1, \neg P \in C_2$
  — calculer r $:= C_1 - \{P\} \vee C_2 - \{\neg P\}$
  — calculer S $:= S \cup \{r\}$
end
if $false \in S$ **then**
  C est prouvé
else
  C n'est pas prouvé
end

- Propriétés de l'algorithme de preuve:

Décidable : L'algorithme se termine après un nombre fini d'étapes Pour toute théorie B, C.

Adéquat : Si B ⊢ C, alors B ⊨ C : Si l'algorithme peut prouver la requête q à partir du programme p, alors q est vrai quand p est vrai

Complet : Si B ⊨ C, alors B ⊢ C : Si q est vrai quand p est vrai alors l'algorithme peut trouver une preuve

- Règle de résolution :

La résolution préserve les modèles

Tout ce qui est modèle des deux premières disjonctions sera aussi modèle de la résultante.

$$\frac{\begin{array}{l} p_1 \vee q \\ p_2 \vee \neg q \end{array}}{p_1 \vee p_2}$$

lundi 13 janvier 2020

Raisonnement scientifique

Déduction:

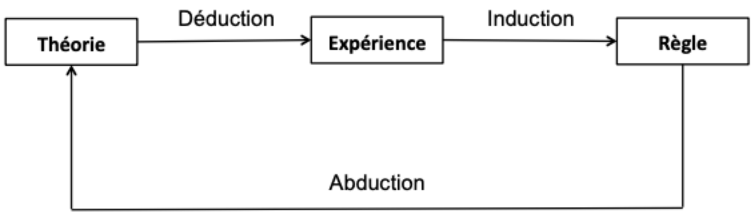
Faire des calcules et raisonnement logique par rapport à une théorie et on déduit le résultat qu'une expérience donnerait. Exemple : Déduire la trajectoire des électrons

Induction:

Trouver la règle général à partir des expériences répétées. Les résultats expérimentaux ne sont pas 100% fiable. Exemple: Tout les oiseaux volent est vrai tant qu'on a pas vu de pingouin.

Abduction:

Compare la règle général trouvée lors de l'induction avec la théorie si il y a une incohérence avec la règle, il faut corriger la règle et repartir dans la boucle. Exemple: Si un élève rentre trempé dans la classe, nous supposons qu'il a plut



Grammaire logique :

<identificateur> ::= A|B|C|….

<proposition> ::= true

| false

| <identificateur>

| (<proposition>)

| ¬<proposition>

| <proposition> V <proposition>

| <proposition> ∧ <proposition>

| <proposition> → <proposition>

| <proposition> ↔ <proposition>

Interpretation :

L'interpretation est une fonction : $val_I(E_p) \rightarrow$ {true,false}.

Modèle :

Un modèle est une interpretation telle que $VAL_I(p)$ = T rue

Preuve par contradiction:

On suppose les prémisses p,...,q vraies (c'est-à-dire on ne peut trouver de contradiction). On ajoute r aux prémisses. S'il est possible de prouver s et ¬s, cela signifie qu'il y a une erreur dans les prémisses, on suppose que l'ajout de r est fautif.

$$\frac{\begin{array}{c} p,\ldots,q,r \vdash s \\ p,\ldots,q,r \vdash \neg s \end{array}}{p,\ldots,q \vdash \neg r}$$

On justifie qu'il n'y a pas de contradiction dans p, . . . , q car on part du principe qu'il existe un modèle de p,...,q.

Logique de prédicat

Élimination de ∀

Parce que ∀x.P(x) veut dire pour tout $x_I \in D_I$: $P_I(x_I)$ est vrai, on peut

remplacer sans contrainte :

- ∀x•P(x) → P(a) a est une constante quelconque

→ P(y) y est une variable (parce que $P_I(y_I)$ est vrai)

$$\frac{\forall x{:}p}{p[x/t]}$$

Élimination de ∃

Il faut faire attention avec cette règle, parce que $\exists x.P(x)$ veut dire qu'il existe un élément $x_I \in D_I$ pour lequel $P_I(x_I)$ est vrai. On ne connait pas cet élément, mais on peut introduire un symbole qui le représente :

$\exists x \bullet P(x) \rightarrow P(a)$
a = nouvelle constante qui n'apparaît nulle part ailleurs ($val_I(a) = x_I$)

$\exists x \bullet P(x) \rightarrow P(z)$
z = nouvelle variable dans la preuve ($val_I(z) = x_I$)

$\exists x \bullet P(x)$ P(y) (pas autorisé)
y = variable qui existe déjà dans la preuve, elle a déjà une valeur

<u>Introduction de ∀</u>

- — Si p n'a pas d'occurrence libre de x alors c'est OK

- — Si p contient une occurrence libre de x : on doit s'assurer que la preuve jusqu'à cet endroit marchera pour toutes valeurs affectées à x
  → Aucune formule dans la preuve jusqu'à cet endroit ne doit mettre une contrainte sur x !
  Deux conditions :

- — x n'est pas libre dans une formule obtenue par élimination de ∃.
  Deux cas sont possibles :
  - — ∃x éliminé, donc x a été choisi pour rendre vraie la formule
  - — ∃y éliminé, donc y a été choisi selon la valeur de x (y dépend de x)

  Dans les deux cas, une contrainte sur x empêche d'introduire ∀x

- — x n'est pas libre dans une prémisse (dans ce cas, x serait connu depuis le début donc il possède déjà une valeur)

<u>Introduction de ∃</u>

Si p[t] est vrai cela veut dire qu'il existe un élément $t_I = VAL_I(t) \in D_I$ avec $(val_I(p))(t_I)$. On peut donc introduire ∃x car un élément existe qui rend vrai p[x]. Mais attention, on doit pouvoir trouver la formule originale à partir de ∃x.p[x], sinon l'introduction du quantificateur "∃" a changé le sens

<u>Règle :</u>

$$\frac{p[t]}{\exists x \bullet p[x]}$$

de la formule.

Graphe

Équilibre Structurel

Définition:

Un graphe complet annoté est équilibré si pour chaque triangle (triplet de trois noeuds), les liens sont (+ + +) ou (+ - -)

Théorème:

Si un graphe complet annoté est équilibré, alors:

- Toutes les paires de noeuds sont amis, ou
- On peut diviser les noeuds en deux groupe X et Y tel que X (resp. Y) contient uniquement des noeuds mutuels et chaque membre de X est ennemi de chaque membre Y

Preuve:

Prenons un graphe complet annoté qui est équilibré.

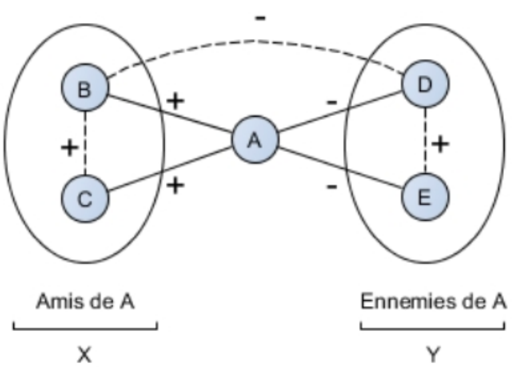
S'il n'y a pas de liens négatifs, alors il est équilibré.

Supposons donc au moins un lien négatif

Prenons un nord quelconque A

Regardons le graphe du point de vue de A:

- Le groupe X c'est A+ses amis
- Le groupe Y c'est les ennemis de A



Chaque paire de X sont des amis

Chaque paire de Y sont des amis

Chaque nord de X est ennemi avec chaque noeud de Y

<u>Équilibre Structurel faible</u>

Définition:

Un graphe complet annoté est faiblement équilibré s'il n'y a pas de triangle (+ + -)

Théorème:

Si un graphe complet annoté est faiblement équilibré, alors on peut diviser ses noeuds en n groupes, tel que les noeuds dans un même groupe sont amis et les noeuds dans des groupes différents sont ennemis

Preuve:

1. Nœud A + tous ses amis appartiennent au groupe X
   $\rightarrow$ amis mutuels, car $(+++)$.
2. A + ses amis sont ennemis avec tous les autres.
3. Enlever A + ses amis : nouveau graphe.
   $\rightarrow$ raisonnement récursif

<u>PageRank Algo</u>

1. N noeuds (chaque noeud représentant une page) : Initialisation Pr(p) $= \frac{1}{n}$
2. Choisir un nombre de pas k
3. K mises à jour :
   $Pr(p) = \sum_{p'} \frac{Pr(p')}{n(p')}$ avec p' les nœuds pointants vers p, $n(p')$ le nombre de liens sortant de p' et $Pr(p')$ le poids (ou PageRank) de p' à la $k^{ème}$ itération.

lundi 13 janvier 2020

Loi qui gouverne la popularité des pages:

distribution de la loi de puissance

mesure pour la popularité d'une page:

k-liens entrants