. On dit que ψ se déduit de ϕ_i et ϕ_j.
ou encore que ψ est une déduction logique de ϕ_i et ϕ_j

. On obtient une nouvelle forme clausale.
$\varphi = \{\varphi_1 \ldots \phi_{i-1}, \phi_{i+1} \ldots \phi_{j-1}, \phi_{j+1} \ldots \phi_n, \psi\}$

Réfutation par résolution (LP)

. Ce procédé est appelé preuve par réfutation ou réfutation par résolution.
- Prouver que $\{\phi_1, \phi_2, \ldots, \phi_n\} \vdash \psi$
- Revient à prouver que :
$\{\phi_1, \phi_2, \ldots, \phi_n, \neg\psi\} \vdash \square$

Rmq : Si $\neg\psi$ n'est pas une clause, il faut la transformer en une clause.

Principe de résolution en LPO

Si on prend une forme clausale $\{\phi_1 \ldots \phi_n\}$ dans la LPO telle que :
$\phi_i = L(x) \vee \phi_i'(x)$ et $\phi_j = \neg L(a) \vee \phi_j'(a)$
Alors le principe de résolution formulé dans la LP devient inapplicable sauf si on impose par exemple la substitution de a à x dans ϕ_i qui devient $\phi_i = L(a) \vee \phi_i'(a)$
la résolvante est : $\phi_i'(a) \vee \phi_j'(a)$

Principe de substitution

Une substitution σ est un ensemble de couples :
$\sigma = \{(t_1, v_1) \ldots (t_n, v_n)\}$
$= \{t_1/v_1, \ldots t_n/v_n\}$
où t_i sont des termes et v_i sont des variables distincts.
→ le couple t_i/v_i signifie que le terme t_i sera substitué à la variable v_i.
→ Appliquer la substitution σ à une f.b.f F revient à remplacer toute occurrence de v_i dans F par t_i.

On dit qu'on substitue t_i à v_i dans F.
Instance : La nouvelle f.b.f obtenue après l'application de la substitution σ à F sera notée $F\sigma$.
$F\sigma$ appelée instance de substitution ou instance de F.

Unification : Une substitution σ est appelée un unifieur pour un ensemble de f.b.f $\{F_1, F_2 \ldots F_n\}$ ssi l'application de σ aux F_i permet d'obtenir des expressions identiques. càd : $F_1\sigma = F_2\sigma = \ldots F_n\sigma$
. Un ensemble de f.b.f $\{F_1, \ldots F_n\}$ est dit unifiable s'il existe une substitution σ (l'unifieur) tq : $F_1\sigma = F_2\sigma \ldots = F_n\sigma$

PR en LPO

Soit une forme clausale : $\phi = \{\phi_1, \ldots \phi_n\}$
Si $\exists$ i et j tq : $\phi_i = F_1 \vee \phi_i'$ et $\phi_j = \neg F_2 \vee \phi_j'$
avec F_1 et F_2 deux littéraux tq $\{F_1, F_2\}$ unifiable et σ est un unifieur.
alors il existe une résolvante ψ de ϕ_i et ϕ_j donnée par : $\psi = (\phi_i')\sigma \vee (\phi_j')\sigma$
On note aussi :
$\{\phi_i, \phi_j\} \xrightarrow{PR} \psi = (\phi_i')\sigma \vee (\phi_j')\sigma$

Réfutation par résolution (LPO)

Prouver que $\{\phi_1, \phi_2, \ldots \phi_n\} \vdash \psi$
Revient à prouver que : $\{\phi_1 \ldots \phi_n, \neg\psi\} \vdash \square$

Définition et propriétés

- Un prédicat est une fonction propositionnelle qui peut être appliquée à un ensemble de paramètres appartenant à un domaine d'application donné pour énoncer une proposition donnée.

$$P : D_1 \times D_2 \times \dots \times D_n \longrightarrow \{Vrai, Faux\}$$
$$(x_1, x_2, \dots, x_n) \longrightarrow P(x_1, x_2, \dots, x_n)$$

$D_1 \times D_2 \times \dots \times D_n$ est le domaine d'application le prédicat P appliqué aux paramètres x_i $(P(x_1, \dots, x_n))$ devient une proposition.

Remarque : Un prédicat peut être appliqué à des variables et/ou des constantes. Il peut aussi être appliqué à la valeur d'une fct sur un ensemble de paramètres.

Exemple :

- Soit l'assertion classique suivante :

"tout homme est mortel" ∧ "Socrates est un homme" → "Socrates est mortel"

- représentée dans la LPO comme suit :

$(\forall x)$ (Homme $(x) \rightarrow$ Mortel(x)) ∧ (Homme (Socrates))) → Mortel (Socrates)

On remarque que cette écriture représente la règle d'inférence Modus-Ponens

Propriétés :

- Le nombre de paramètres d'un prédicat est appelé arité du prédicat.
- Un prédicat d'arité 0 est une proposition.
- Les connecteurs de la LPO sont les mêmes que ceux de la LP.
- Les quantificateurs universel ($\forall$) et existentiel ($\exists$) sont des éléments de la LPO.

- Un terme est une constante ou une variable ou une fonction appliquée à des termes dans la LPO
- Un Atome en LPO est un symbole de proposition (prédicat d'arité 0) ou l'application d'un prédicat à des termes
  - Homme (x)
  - Régisseur(Travaille(x))
  - P
- Un littéral est un atome ou la négation d'un atome.
- Une formule bien formée (f.b.f) est un littéral ou une expression formée à base de littéraux auxquels sont appliqués les connecteurs.

Equivalences de base

$$\neg((\forall x) P(x)) \equiv (\exists x) \neg P(x)$$
$$\neg((\exists x) P(x)) \equiv (\forall x) \neg P(x)$$

soient M et N deux f.b.f sans quantificateurs :

$$((\forall x) M) \wedge ((\forall x) N) \equiv (\forall x)(M \wedge N)$$
$$((\exists x) M) \vee ((\exists x) N) \equiv (\exists x)(M \vee N)$$
$$((\forall x) M) \vee ((\forall x) N) \rightarrow (\forall x)(M \vee N)$$
$$(\exists x)(M \wedge N) \rightarrow ((\exists x) M) \wedge ((\exists x) N)$$

soient Q_1 et Q_2 deux quantificateurs qcq :

$$((Q_1 x) M(x)) \wedge ((Q_2 y) N(y)) \equiv$$
$$(Q_1 x)(Q_2 y)(M(x) \wedge N(y))$$
$$((Q_1 x) M(x)) \vee ((Q_2 y) N(y)) \equiv$$
$$(Q_1 x)(Q_2 y)(M(x) \vee N(y))$$

Forme normale de Prénexe

Une f.b.f est une forme normale de Prénexe ssi F est sous la forme:

$F = (Q_1 x_1)(Q_2 x_2) \dots (Q_n x_n) M$

Avec Q_i sont des quantificateurs qcq ($\forall$ ou $\exists$), et M est une formule sans quantificateurs appelée Matrice

exp: $F = (\forall x)(\exists y)(\exists z)(P(x,y) \rightarrow Q(y,z))$

Forme standard de Skolem (la skolémisation)

Elle consiste à supprimer les quantificateurs existentiels ($\exists$)

1. Mettre F (f.b.f) sous forme normale de Prénexe :

   $F = (Q_1 x_1) \dots (Q_n x_n) M$

2. Mettre M sous forme normale conjonctive

   $M = M_1 \wedge M_2 \wedge \dots \wedge M_n$

3. pour tout quantificateur $Q_i = \exists$.

- S'il n'y a aucun $\forall$ à gauche de Q_i alors supprimer $(Q_i\, x_i)$ et remplacer x_i dans M par une constante non déjà existante.

- si $Q_j, Q_{j+1} \dots Q_r$ sont des quantificateurs ($\forall$) à gauche de Q_i alors supprimer $(Q_i\, x_i)$ et remplacer x_i dans M par une fonction f de $x_j, x_{j+1} \dots x_r$

  $f(x_j, x_{j+1} \dots x_r)$

Formes clausales

C'est une forme qui prépare une connaissance à être implémenté ds un système à base de connaissances avec un outil tel que Prolog.

1. Clause : est une disjonction de littéraux

. On considère que l'inconsistance $\square$ est une clause vide.

2. Forme Clausale ou ensemble de clauses.

est une conjonction de clauses.

. Pour obtenir la forme clausale d'une f.b.f F il suffit de la mettre sous forme standard de Skolem.

$F = (\forall x_1) \dots (\forall x_n) F_1 \wedge F_2 \wedge \dots \wedge F_k$

la forme clausale de F est.

$F_1 \wedge F_2 \wedge \dots \wedge F_k$ qu'on note :

$\{F_1, F_2, \dots F_k\}$

Principe de résolution

le principe de (PR) est le formalisme d'une règle d'inférence qui permet la déduction d'une nouvelle clause ψ à partir de deux autres ϕ_1 et ϕ_2

- la nouvelle clause ψ sera appelée résolvante de ϕ_1 et ϕ_2

- l'avantage du PR est qu'il est applicable à la LP et généralisable au cas de la LPO

- méthode:

. Soit une forme clausale $C = \{\phi_1, \phi_2, \dots \phi_n\}$ c.à.d: clauses ne contenant pas de variables (LP).

. Si $\exists i$ et j tq: $\phi_i = L \vee \phi_i'$ et $\phi_j = \neg L \vee \phi_j'$ avec L un littéral, alors le principe de résolution permet de déduire qu'il existe une résolvante ψ de ϕ_i et ϕ_j donnée par : $\psi = \phi_i' \vee \phi_j'$

On note que :

$\{L \vee \phi_i', \neg L \vee \phi_j'\} \xrightarrow[\text{ou}]{PR} \psi = \phi_i' \vee \phi_j'$

Module : Intelligence Artificielle

Représentation de la connaissance

&

Systèmes Experts

IA : est une discipline en informatique qui permet la reproduction de comportements humains intelligents.

Elle cherche à concevoir des programmes et des machines en mesure de traiter des problèmes pour lesquels nous n'avons pas de méthodes directes et assurées de résolution.

Système Expert est un programme qui permet l'exploitation des connaissances dans un domaine précis et rigoureusement limité.

est un système informatique où les données (la base de connaissance) sont bien séparées du programme qui les manipule (le moteur d'inférences)

IA forte : fait référence à une machine capable non seulement de produire un comportement intelligent mais d'éprouver une impression d'une réelle conscience en soi de vrai «sentiments», et une compréhension de ses propres raisonnements.

IA faibles Constitue une approche pragmatique d'ingénieur: chercher à construire des systèmes de plus en plus autonomes, des algorithmes capables de résoudre des problèmes d'une certaine classe, etc. Mais, cette fois, la machine simule l'intelligence elle semble agir comme si elle était intelligente.

I. Représentation de la connaissance

Logique des Propositions
&
Logique des Prédicats du Premier Ordre

. L'approche classique et la base fondamentale pour représenter la connaissance en IA est la logique formelle.

- La logique formelle renferme un grand nombre de classes dont les plus fréquemment utilisées pour réaliser des systèmes en IA sont la logique des propositions (LP) et la logique des prédicats du premier ordre (LPO)

1. Logique des propositions.

- Une proposition est un énoncé ou assertion permettant de définir une situation ou concept ou fait logique quelconque ayant une valeur de vérité vraie ou fausse.

Exemples. La salle est grande.

. Pour la manipulation d'une proposition on lui affecte un symbole (souvent une lettre Exp: P, Q, R ...)

Connecteurs :

- Il existe un ensemble d'opérateurs logiques qui, appliqués à un ensemble de propositions créent une nouvelle :

- $\neg$ ou $\lnot$: négation
- $\wedge$: conjonction (et)
- $\vee$: disjonction (ou)
- $\rightarrow$: implication
- $\leftrightarrow$: équivalence

L'application de ces connecteurs aux propositions permet de définir plusieurs types de propositions :

Atome : est une proposition élémentaire ne contenant pas de connecteur logique.

Littérale : est un atome ou la négation d'un atome.

Expression bien formée (e.b.f) : est un littéral ou une proposition formée à base de littéraux auxquels sont appliqués les connecteurs $\wedge, \vee, \rightarrow, \leftrightarrow$

Équivalences de bases

- $P \rightarrow Q \equiv \neg P \vee Q$
- $\neg(\neg Q) \equiv Q$
- $P \wedge Q \equiv Q \wedge P$
- $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$
- $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$
- $P \wedge (Q \vee L) \equiv (P \wedge Q) \vee (P \wedge L)$
- $P \vee (Q \wedge L) \equiv (P \vee Q) \wedge (P \vee L)$

$P \rightarrow Q \equiv \neg Q \rightarrow \neg P$

$P \vee \square \equiv P$

$P \wedge \square \equiv \square$ ← inconsistance prop t.j.r fausse

$P \vee \blacksquare \equiv \blacksquare$

$P \wedge \blacksquare \equiv P$

$P \vee \neg P \equiv \blacksquare$ ← tautologie prop t.j.r vraie.

$P \wedge \neg P \equiv \square$

Formes normales

- F.N. Conjonctive : toute proposition P sous la forme :

$$P = \bigwedge_{i=1}^{n} Q_i \text{ avec } Q_i = \bigvee_{j=1}^{p_i} H_j \ , \forall i$$

où H_j est un littéral.

- F.N. disjonctive : toute proposition P sous la forme :

$$P = \bigvee_{i=1}^{n} Q_i \text{ , avec } Q_i = \bigwedge_{j=1}^{p_i} H_j \ , \forall i \text{ où } H_j \text{ est un littéral}$$

Règles d'Inférences

une R.I est la représentation d'un procédé pour déduire de nouvelles connaissances à partir d'une ou plusieurs connaissances.

. Dans le cas de la logique des propositions une connaissance est représentée par une proposition (ou e.b.f)

- Les règles d'inférences de base sont celles de :
- Règle d'inférence de Modus Ponens

si $P \rightarrow Q$ est vraie et P est vraie alors Q est vraie, on note :

$$\{P \rightarrow Q, P\} \xrightarrow{MP} Q$$

- Règle d'inférence de Modus Tollens

si $P \rightarrow Q$ est vraie et $\neg Q$ est vraie alors $\neg P$ est vraie, on note :

$$\{P \rightarrow Q, \neg Q\} \xrightarrow{M.T} \neg P$$

2. Logique des prédicats du premier ordre (LPO)

La logique des propositions ne permet pas de formaliser des énoncés généraux : si x est père de y et x est père de z alors y et z sont frères

. c'est une condition qui ne peut être représentée ou manipulée en logique des propositions.

. En logique des prédicats de premier ordre l'assertion présentée avant peut être représentée comme suit :

$$\forall (x, y, z) \, (\text{Père}(x, y) \wedge \text{Père}(x, z)) \rightarrow \text{Frère}(y, z))$$

Ou père et frère sont des prédicats et non des propositions