1 Notions

En plus des notions données dans le document précédent, nous avons :

1. Une formule fermée de notre langage *Pred* est une *vérité logique* ssi (si et seulement si) elle est vraie dans toutes ses interprétations.
2. Une formule fermée de notre langage *Pred* est une *contradiction logique* ssi elle est fausse dans toutes ses interprétations.
3. Deux formules fermées de *Pred* sont logiquement équivalentes ssi, dans chacune de leurs interprétations, elles sont soit toutes les deux vraies, soit toutes les deux fausses.
4. Uun ensemble de formules fermées est *consistant* ssi il a ce que l'on appelle un *modèle* (i.e. une interprétation qui rende les formules en question simultanément vraies).
5. Une lettre de constante est *ancienne* ssi – alors qu'on déploie l'arbre – elle apparaît déjà le long d'une branche.
6. Une lettre de constante est *nouvelle* ssi – alors qu'on déploie l'arbre – elle n'apparaît pas le long d'une branche.

2 Règles spécifiques à *Pred* de construction pour les arbres

En plus des règles données dans le document précédent, nous avons :

1. Règle ($\forall$) : Si une formule de type $\forall u(...u...u...)$ apparaît le long d'une branche, nous pouvons ajouter l'instance de substitution $(...c...c...)$ au pied de cette branche, « c » étant n'importe quelle constante qui apparaît *déjà* le long de cette branche (i.e. avec « c » ancienne). Si aucune lettre n'apparaît déjà, nous pouvons en prendre une nouvelle.

$$\begin{array}{c} | \\ \forall u(...u...u...) \\ | \\ (...c...c...) \end{array}$$

2. Règle ($\exists$) : Si une formule du type $\exists u(...u...)$ apparaît le long d'une branche, nous devons choisir une lettre de constante nouvelle (i.e. qui n'apparaît pas déjà le long de cette branche) – disons s – puis écrire l'instance de substitution $(...s...)$ de notre formule quantifiée en bas de chaque branche ouverte dans laquelle cette dernière apparaît. Nous ajoutons enfin un ✓ à côté de la formule pour signaler qu'elle a été traitée.

$$\begin{array}{c} | \\ \checkmark\ \exists u(...u...) \\ | \\ (...s...) \end{array}$$

3. Règle ($\neg\forall$) : Si vous rencontrez dans une branche une formule de type $\neg\forall u(....u...)$, vous pouvez traiter cette formule (et mettre un ✓ une fois l'opération effectuée) en écrivant en dessous de cette formule (sur la même branche) la formule : $\exists u\neg(...u...)$.

$$\begin{array}{c} | \\ \checkmark\neg\ \forall u(...u...u...) \\ | \\ \exists u\neg(...u...u...) \end{array}$$

4. Règle ($\neg\exists$) : Si vous rencontrez dans une branche une formule de type $\neg\exists u(....u...)$, vous pouvez traiter cette formule (et mettre un ✓ une fois l'opération effectuée) en écrivant en dessous de cette formule (sur la même branche) la formule : $\forall u\neg(...u...)$.

$$\begin{array}{c} | \\ \checkmark\neg\ \exists u(...u...u...) \\ | \\ \forall u\neg(...u...u...) \end{array}$$

3 Méthode

1. Si nous voulons démontrer qu'une formule (la formule *à évaluer*) est *une vérité logique* :
   (a) Nous prenons la négation de cette formule et nous la plaçons en haut de l'arbre.
   (b) Nous construisons l'arbre.
   (c) L'arbre peut se terminer de deux façons :
      i. Toutes les formules sont traitées.
      ii. Toutes les branches ferment avant que toutes les formules ne soient traitées.

(d) Dans le cas ii. et dans le cas où, une fois toutes les formules traitées, toutes les branches ferment, nous pouvons conclure que la formule en haut de l'arbre est une contradiction logique, ce qui veut dire, puisque la formule en haut de l'arbre est sa négation, que la formule *à évaluer* est une vérité logique.

(e) Quand une branche reste ouverte et que toutes les formules ont été traitées, nous pouvons conclure que la formule à évaluer *n'est pas* une vérité logique : il y a une interprétation qui rend vraie sa négation, ce qui veut dire que cette même interprétation la rend fausse. Nous pouvons spécifier cette interprétation (ce contre-exemple) en remontant le long de la branche ouverte : le nombre de constante nous donnera les éléments de notre domaine, et les propositions élémentaires ou négations de propositions élémentaires nous indiqueront l'interprétation des lettres de prédicat à fournir pour que la formule à évaluer soit fausse.

2. Si nous voulons démontrer qu'une formule est une contradiction logique, nous plaçons directement la formule en haut de l'arbre. Si toutes les branches ferment, c'est que la formule est une contradiction logique (il est impossible de la rendre vraie). Si au moins une branche reste ouverte, c'est que la formule n'est pas une contradiction logique (il est possible de la rendre vraie).
3. Si nous voulons démontrer que deux formules (ϕ et ψ) sont logiquement équivalentes nous chercherons à montrer que $(\phi \leftrightarrow \psi)$ est une vérité logique.
4. Si nous voulons démontrer qu'un ensemble de formules est consistant, nous « empilons » ces formules en haut d'un arbre et nous déployons l'arbre. Si toutes les branches ferment, c'est qu'il est impossible que toutes ces formules soient vraies en même temps. Cet ensemble de formules n'est donc pas consistant. S'il y a au moins une branche ouverte, nous pouvons alors donner un modèle qui rend ces formules simultanément vraies. L'ensemble de formules est donc consistant.

4 Conseils

En plus des conseils donnés dans le document précédent, il est bon de suivre les conseils suivants :

1. Traitez d'abord (si possible) les formules en utilisant les règles spécifiques au langage du calcul des propositions.
2. Traitez (si possible) les formules quantifiées existentiellement (les quantifications existentielles) *avant* les formules quantifiées universellement.

5 Exemples

5.1 Vérité logique

Soit la formule à évaluer suivante :

$$\exists x(Px \to Qx) \to \neg\forall x(Px \land \neg Qx)$$

Voici l'arbre pour cette formule :

$$\begin{array}{c}
\checkmark \neg\{\exists x(Px \to Qx) \to \neg\forall x(Px \land \neg Qx)\} \\
| \\
\checkmark \exists x(Px \to Qx) \\
| \\
\checkmark \neg\neg\forall x(Px \land \neg Qx) \\
| \\
\forall x(Px \land \neg Qx) \\
| \\
\checkmark (Pa \to Qa) \\
\swarrow \qquad \searrow \\
\begin{array}{cc}
\neg Pa & Qa \\
| & | \\
\checkmark (Pa \land \neg Qa) & \checkmark (Pa \land \neg Qa) \\
| & | \\
Pa & Pa \\
\otimes & | \\
 & \neg Qa \\
 & \otimes
\end{array}
\end{array}$$

Toutes les branches de l'arbre ferment. La formule en haut de l'arbre est contradictoire ; la formule à évaluer est une vérité logique.

5.2 Contre-exemple

Soit la formule à évaluer :

$$\forall x(Kx \lor Ax) \land \neg Kb] \to Ac$$

L'arbre commence ainsi :

$$\begin{array}{c}
\neg\{[\forall x(Kx \lor Ax) \land \neg Kb] \ \to Ac\} \\
| \\
\forall x(Kx \lor Ax) \\
| \\
\neg Kb \\
| \\
\neg Ac \\
| \\
Kb \lor Ab \\
\begin{array}{cc}
Kb & \begin{array}{c} Ab \\ | \\ Kc \lor Ac \\ \begin{array}{cc} Kc & Ac \\ & \otimes \end{array} \end{array} \\
\otimes &
\end{array}
\end{array}$$

Nous avons donc notre contre-exemple. Dans un domaine à deux éléments et avec l'interprétation suivante des lettres de prédicat, la formule est fausse : D : {b, c} K : {c} A : {b}

$$[(Kb \lor Ab) \land (Kc \lor Ac) \land \neg Kb] \to Ac$$

Les prémisses sont vraies, la conclusion est fausse.

5.3 Équivalence logique

Supposons que nous voulions prouver l'équivalence logique suivante :

$$(\forall xPx \land \forall xQx) \leftrightarrow \forall x(Px \land Qx)$$

Nous commencerons alors notre arbre par la négation de cette formule :

$$\begin{array}{c}
\checkmark \neg[(\forall xPx \land \forall xQx) \leftrightarrow \forall x(Px \land Qx)] \\
\begin{array}{cc}
\begin{array}{c}
\checkmark \forall xPx \land \forall xQx \\
| \\
\checkmark \neg\forall x(Px \land Qx) \\
| \\
\forall xPx \\
| \\
\forall xQx \\
| \\
\checkmark \exists x\neg(Px \land Qx) \\
| \\
\checkmark \neg(Pa \land Qa) \\
\begin{array}{cc} \neg Pa & \neg Qa \\ | & | \\ Pa & Qa \\ \otimes & \otimes \end{array}
\end{array}
&
\begin{array}{c}
\checkmark \neg(\forall xPx \land \forall xQx) \\
| \\
\forall x(Px \land Qx) \\
\begin{array}{cc}
\checkmark \neg\forall xPx & \checkmark \neg\forall xQx \\
| & | \\
\checkmark \exists x\neg Px & \checkmark \exists x\neg Qx \\
| & | \\
\neg Pb & \neg Qc \\
| & | \\
\checkmark Pb \land Qb & \checkmark Pc \land Qc \\
| & | \\
Pb & Qc \\
\otimes & \otimes
\end{array}
\end{array}
\end{array}
\end{array}$$

Toutes les branches ferment : la formule en haut de l'arbre est donc contradictoire, ce qui veut dire que la formule à évaluer est une vérité logique.

5.4 Ensembles consistants

5.4.1

Nous cherchons à savoir si les formules $\exists xRxa$, $\forall x(Rxa \to \neg Fx)$ et $\neg\exists x\neg Fx$ peuvent être vraies en même temps.

Nous faisons alors l'arbre comme suit :

$$\begin{array}{c}
\checkmark\, \exists x Rxa \\
| \\
\forall x[Rxa \rightarrow \neg Fx] \\
| \\
\checkmark\, \neg\exists x \neg Fx \\
| \\
Rba \\
| \\
\forall x Fx \\
| \\
Fb \\
| \\
Fa \\
| \\
\checkmark\, Raa \rightarrow \neg Fa
\end{array}$$

$$\begin{array}{cc}
\neg Raa & \neg Fa \\
| & \otimes \\
\checkmark\, Rba \rightarrow \neg Fb & \\
\begin{array}{cc} \neg Rba & \neg Fb \\ \otimes & \otimes \end{array} &
\end{array}$$

Conclusion, un tel ensemble de phrases est *inconsistant* : il est impossible qu'elles soient simultanément vraies.

5.4.2

En contraste, si l'on prend cet ensemble de formules [$\exists x(Fx \wedge Gx)$, $\forall x Hx$, $\forall x(Hx \rightarrow Gx)$ et $\exists x Hx$], nous pouvons voir qu'il est consistant et nous pouvons fournir un modèle, une interprétation qui les rende simultanément vraies.

Nous construisons également l'arbre pour la trouver :

$$\begin{array}{c}
\checkmark\, \exists x(Fx \wedge Gx) \\
| \\
\forall x Hx \\
| \\
\forall x(Hx \rightarrow Gx) \\
| \\
\checkmark\, \exists x Hx \\
| \\
Ha \\
| \\
\checkmark\, Fb \wedge Gb \\
| \\
Fb \\
| \\
Gb \\
| \\
Hb \\
| \\
\checkmark\, Ha \rightarrow Ga
\end{array}$$

$$\begin{array}{cc}
\neg Ha & Ga \\
\otimes & | \\
 & \checkmark\, Hb \rightarrow Gb \\
 & \begin{array}{cc} \neg Hb & Gb \\ \otimes & \end{array}
\end{array}$$

Ici, l'arbre a une branche ouverte : c'est cette branche ouverte qui va nous fournir l'interprétation que l'on cherche. Nous avons cherché à rendre ces formules vraies et c'était possible.

Ainsi, dans un domaine à deux éléments $D : \{a, b\}$ et avec l'interprétation des lettres de prédicats suivante : $F : \{b\}$, $G : \{a, b\}$ et $H : \{a, b\}$, les quatre formules sont vraies en même temps. Il est vrai qu'il y a (au moins) une chose qui est à la fois F et G (ici, c'est b), il est vrai que toutes les choses sont H, il est vrai que toutes les choses sont telles que si elles sont H, alors elles sont G, et il est vrai qu'il y a au moins une chose qui est H (par exemple a).