

LOGIQUE
L3 Informatique

Salim Lardjane
Université Bretagne Sud

3. La logique propositionnelle (LP0)

3.3. Tableaux sémantiques

Définition

Nous avons caractérisé un *raisonnement correct* (du point de vue sémantique) comme un raisonnement dans lequel il n'existe pas d'interprétation permettant d'évaluer simultanément les prémisses à vrai et la conclusion à faux.

Par conséquent, si nous essayons toutes les possibilités de construction de modèles de l'ensemble de prémisses et de la négation de la conclusion, nous faisons « d'une pierre deux coups ».

Définition

En effet, si nous ne réussissons pas, alors nous n'avons pas réussi à réfuter le raisonnement, parce que nous n'avons pas réussi à évaluer à vrai les prémisses et à faux la conclusion (puisque nous avons essayé d'évaluer à vrai la négation de la conclusion).

Donc, le raisonnement est correct.

En revanche, si nous réussissons, nous avons construit un contre-exemple qui prouve que le raisonnement n'est pas correct.

Définition

On a vu par ailleurs que, pour essayer d'évaluer une formule à vrai, on n'est pas obligé de construire toute la table de vérité (cf. TD 2).

La méthode des *tableaux sémantiques*, dont il existe plusieurs variantes, formalise ces idées.

Elle est fondée sur la technique suivante : pour chaque connectif (constructeur de formule) pouvant apparaître dans une fbf \mathcal{A} , on donne toutes les façons de construire des modèles de \mathcal{A} .

L'ensemble de connectifs que nous utilisons est $\{\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow\}$.

La méthode utilise les règles suivantes (\mathcal{A} et \mathcal{B} dénotent des fbf).

Règles de type α

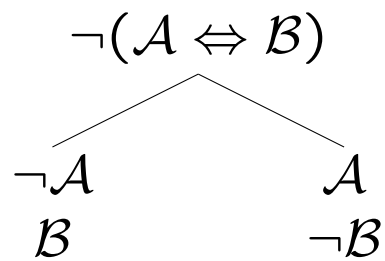
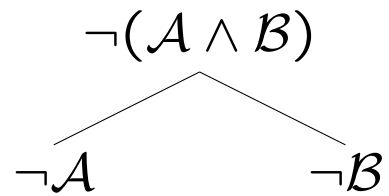
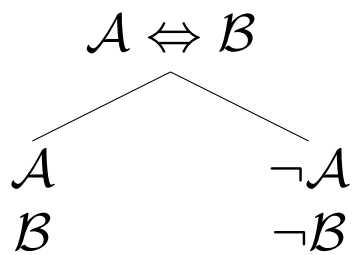
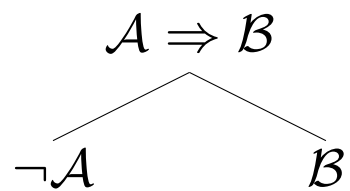
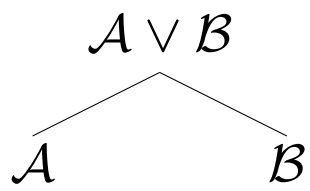
$$\begin{array}{c} \neg\neg A \\ | \\ A \end{array}$$

$$\begin{array}{c} A \wedge B \\ | \\ A \\ B \end{array}$$

$$\begin{array}{c} \neg(A \vee B) \\ | \\ \neg A \\ \neg B \end{array}$$

$$\begin{array}{c} \neg(A \Rightarrow B) \\ | \\ A \\ \neg B \end{array}$$

Règles de type β



Idée de la méthode

Etant donné un ensemble S de fbf, on essaie de construire, en appliquant les règles ci-dessus, tous les modèles de S en combinant les modèles de chacune des formules.

Il est clair que les règles remplacent des fbf par des fbf contenant (strictement) moins de symboles que la fbf originelle. On finit donc par arriver nécessairement à des littéraux (cette propriété sert à prouver la terminaison de la méthode).

La combinaison des modèles se traduit ainsi par un arbre (non unique et dépendant de l'ordre dans lequel on analyse les modèles de S).

Idée de la méthode

Il se peut que certaines combinaisons ne soient pas possibles (parce que l'on essaie d'évaluer une formule atomique simultanément à vrai pour construire le modèle d'une fbv et à faux pour construire un modèle d'une autre).

Dès qu'une branche contient des littéraux A et $\neg A$ on la ferme (pour fermer une branche on utilisera le symbole \times). On dira qu'elle est *fermée*. Si une branche n'est pas fermée, elle est dite *ouverte*.

Un tableau est dit *fermé* si et seulement si toutes ses branches sont fermées. Il est dit *ouvert* autrement .

Avant de donner l'algorithme correspondant à la méthode nous appliquons les règles sur quelques exemples simples.

Exemple 1

Considérons le raisonnement suivant :

$$\begin{array}{l} A \Rightarrow \neg B \\ \neg C \Rightarrow A \\ \hline B \Rightarrow C \end{array}$$

En accord avec ce que nous avons dit précédemment, nous considérons l'ensemble (c'est-à-dire il n'y a pas d'ordre imposé pour analyser les formules) de formules :

- (1) $A \Rightarrow \neg B$
- (2) $\neg C \Rightarrow A$
- (3) $\neg(B \Rightarrow C)$

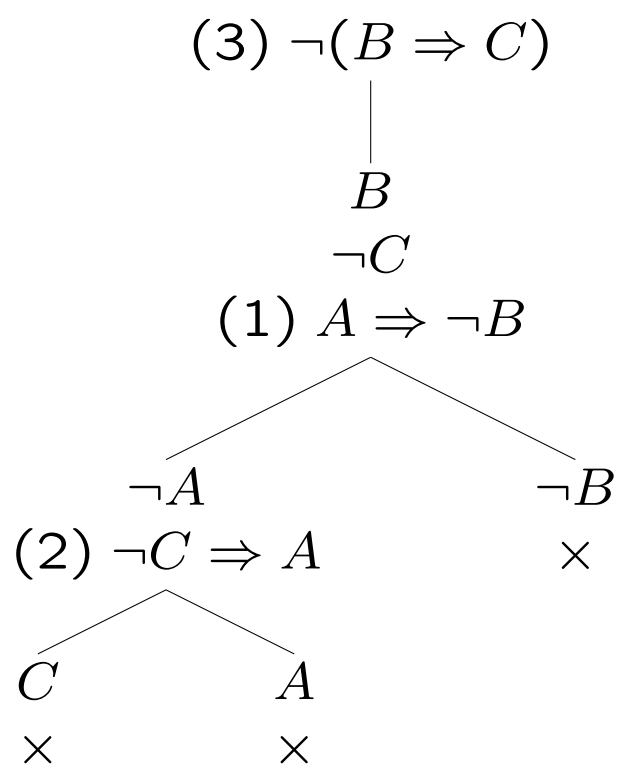
Exemple 1

Nous essaierons de construire un (des) modèle(s) pour l'ensemble de formules (1), (2), (3).

L'ordre d'analyse choisi est (3), (1), (2).

La procédure est totalement mécanique : pour chaque fbf on identifie la règle à appliquer (il y en a une pour chaque connectif).

Exemple 1



Exemple 1

Conclusion : toutes les branches de l'arbre sont fermées.

Ceci veut dire qu'en examinant toutes les possibilités de construire les modèles potentiels de (1), (2), (3), nous avons échoué.

L'ensemble de fbf (1), (2), (3) est donc contradictoire (ou insatisfaisable ou incohérent).

Le raisonnement initial est donc correct .

Exemple 2

Considérons le raisonnement suivant :

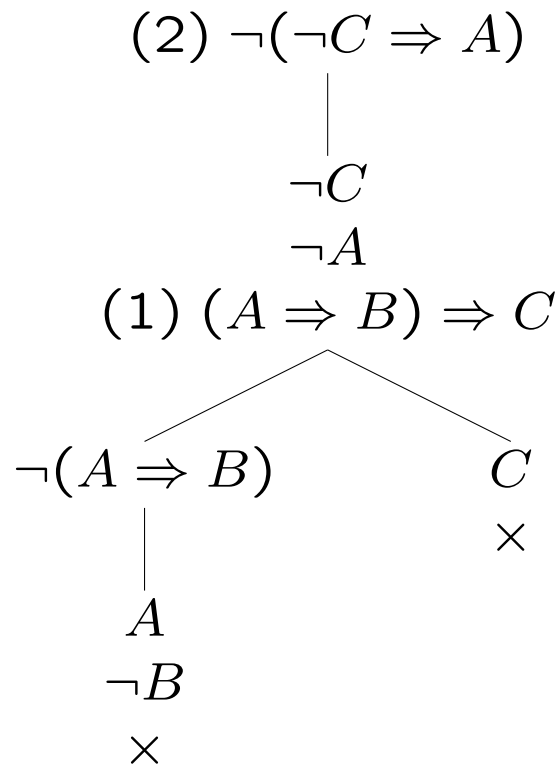
$$\frac{(A \Rightarrow B) \Rightarrow C}{\neg C \Rightarrow A}$$

On analyse l'ensemble de formules :

- (1) $(A \Rightarrow B) \Rightarrow C$
- (2) $\neg(\neg C \Rightarrow A)$

On choisit, pour appliquer les règles, l'ordre (2), (1).

Exemple 2



Conclusion : toutes les branches de l'arbre sont fermées.

L'ensemble de fbf (1), (2) est donc contradictoire (ou insatisfaisable ou incohérent).

Le raisonnement initial est donc correct .

Exemple 3

Nous avons analysé deux exemples de raisonnement correct. Que se passe-t-il pour un raisonnement incorrect ? La méthode est-elle également utile ?

Considérons le raisonnement suivant :

$$\frac{P \Rightarrow (\neg Q \vee R) \\ Q \Rightarrow (P \wedge \neg R)}{R \Rightarrow Q}$$

Exemple 3

On considère donc l'ensemble de formules :

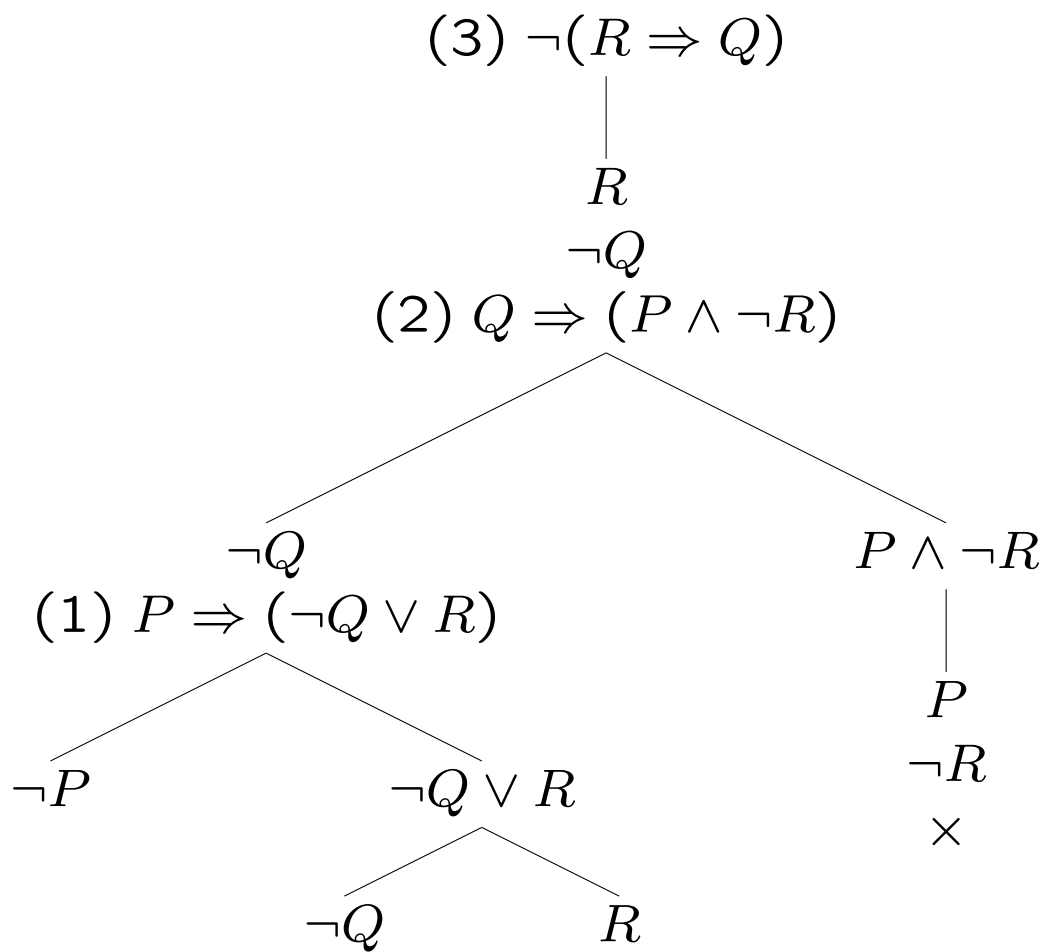
$$(1) \quad P \Rightarrow (\neg Q \vee R)$$

$$(2) \quad Q \Rightarrow (P \wedge \neg R)$$

$$(3) \quad \neg(R \Rightarrow Q)$$

qu'on décide d'analyser dans l'ordre (3), (2), (1), ce qui donne :

Exemple 3



Exemple 3

Etant donné que nous essayons d'énumérer tous les modèles potentiels de l'ensemble de fbf considérées, on aurait dû, en principe, greffer le sous-arbre de racine $\neg Q$ à la place de \times , mais ce n'est pas nécessaire parce que toutes les branches issues de ce nœud contiendront la contradiction $(R, \neg R)$.

Mais, à présent, *que signifient les branches ouvertes ?*

Exemple 3

En lisant les fbf atomiques et négations de fbf atomiques, on a les modèles de prémisses et de la négation de la conclusion, donc des *modèles de prémisses qui sont des contre-modèles de la conclusion*, c'est-à-dire des *contre-exemples* montrant que *le raisonnement est incorrect*.

Les contre-exemples que nous pouvons lire dans les branches ouvertes sont :

$$\{R, \neg Q, \neg P\}$$

et

$$\{\neg Q, R\}$$

Tableaux sémantiques et modèles

Ainsi, il apparaît que la méthode des tableaux sémantiques pour la LP0 est une méthode pour *énumérer les modèles* d'un ensemble de formules.

Elle permet donc de décider de la satisfaisabilité ou de l'insatisfaisabilité d'un ensemble de fbf.

Algorithme

entrée : un ensemble fini de formules de la LPO $\mathcal{F} = \{f_1, \dots, f_n\}$.

sortie : ensemble des modèles de \mathcal{F} ou " \mathcal{F} contradictoire"

début

racine de l'arbre $\leftarrow \mathcal{F}$

tantque $\mathcal{F} \neq \emptyset$ et (il y a des branches ouvertes)

faire

si une branche \mathcal{B} contient deux littéraux complémentaires,
fermer \mathcal{B} (mettre une \times)

$\#$ le modèle correspondant n'est pas viable

choisir $f_i \in \mathcal{F}$ et appliquer la règle correspondante,

donnant (disons) f_i^j ($1 \leq j \leq 2$)

greffer f_i^j ($1 \leq j \leq 2$) dans toutes les branches ouvertes
passant par le noeud étiqueté f_i

si f_i^j ($1 \leq j \leq 2$) n'est pas un littéral

alors $\mathcal{F} \leftarrow (\mathcal{F} \setminus \{f_i\}) \cup \{f_i^j\}$

sinon $\mathcal{F} \leftarrow (\mathcal{F} \setminus \{f_i\})$

finfaire

si arbre fermé

alors retourner " \mathcal{F} contradictoire"

sinon retourner les ensembles de littéraux des branches
ouvertes

fin

Utilisation

– S'il s'agit de tester la *correction d'un raisonnement* $H_1, \dots, H_n \models? C$, faire

$$\mathcal{F} \leftarrow \{H_1, \dots, H_n, \neg C\}$$

Si l'algorithme donne " \mathcal{F} contradictoire", alors le raisonnement est correct, sinon l'algorithme fournit tous les contre-exemples (branches ouvertes) montrant que le raisonnement est incorrect.

– S'il s'agit de tester la *validité d'une formule* \mathcal{G} , faire

$$\mathcal{F} \leftarrow \{\neg \mathcal{G}\}$$

Si l'algorithme donne " \mathcal{F} contradictoire", alors la fbf est *valide*, sinon sinon l'algorithme fournit tous les contre-exemples (branches ouvertes) montrant que la formule n'est pas valide.

Utilisation

- S'il s'agit de construire *tous les modèles* d'une formule \mathcal{G} , faire

$$\mathcal{F} \leftarrow \{\mathcal{G}\}$$

- S'il s'agit de tester si un ensemble de fbf \mathcal{S} est contradictoire (insatisfaisable), faire

$$\mathcal{F} \leftarrow \mathcal{S}$$

Si l'arbre est fermé, \mathcal{S} est insatisfaisable, sinon on obtient tous les modèles de \mathcal{S} .

- S'il s'agit de tester si un ensemble de fbf $\mathcal{S} = \{f_1, \dots, f_n\}$ est *valide*, faire

$$\mathcal{F} \leftarrow \{\neg f_1 \vee \dots \vee \neg f_n\}$$

Si l'arbre est fermé, \mathcal{S} est *valide*, sinon on obtient tous les contre-modèles de \mathcal{S} .

Non déterminisme

Notons que l'algorithme donné est non déterministe (instruction **choisir**), c'est-à-dire qu'il peut être implémenté avec différentes *stratégies*.