Introduction à la logique des propositions V

Robert Michels

mail@robert-michels.de

Université de Neuchâtel – semestre d'automne 2019 – 25 novembre 2019

L'idée fondamentale de la méthode des arbres

- Comme la déduction naturelle, la méthode des arbres est une méthode syntaxique
- En général, elle nous permet de prouver qu'un ensemble de formules est consistant ou inconsistant
- Elle nous permet en effet de vérifier p. ex. pour quelques formules de L_0 A, B, C, que $\vdash_{L_0} \neg(A \wedge (B \wedge C))$ ou que, au contraire, $\nvdash_{L_0} \neg(A \wedge (B \wedge C))$
- Comme pour les concepts sémantiques correspondants, il existe une connexion étroite entre les concepts syntaxiques de consistance, de déduction et de prouvabilité
- Cette connexion se base sur le théorème suivant :

Le théorème de déduction

Théorème de déduction $\Delta, A \vdash_{L_0} B$ si et seulement si $\Delta \vdash_{L_0} A \rightarrow B$

- Ce théorème peut vous sembler familier ; sa version sémantique a été présentée à la p. 26 de l'exemplier de la troisième séance du cours.

Le théorème de déduction et la méthode des arbres

- Le théorème de déduction implique que $A \vdash_{L_0} B$ si et seulement si $\vdash_{L_0} A \rightarrow B$
- On peut prouver que $A \rightarrow B \vdash_{L_0} \neg(A \wedge \neg B)$ (et que $\neg(A \wedge \neg B) \vdash_{L_0} A \rightarrow B$ – on peut dire que les formules de ces deux formes logiques sont syntaxiquement équivalentes)
- Ensemble, ces deux faits impliquent que pour prouver que $A \vdash_{L_0} B$, il est suffisant de prouver que $A \wedge \neg B$ est inconsistante, ou en symboles : $A \vdash_{L_0} B$ si et seulement si $\vdash_{L_0} \neg(A \wedge \neg B)$
- Ce résultat nous permet de construire des arbres qui montrent qu'une formule C peut être prouvée à partir d'un ensemble de formules $\{A, B, \ldots\}$ – on prouve que la conjonction de la conjonction de toutes les formules dans l'ensemble et de la négation de C est inconsistante : si $\vdash_{L_0} \neg(A \wedge (B \wedge \ldots) \wedge \neg C)$ (un fait qui peut être prouvé par la méthode des arbres), alors $A, B, \ldots \vdash_{L_0} C$

Le théorème de déduction et la méthode des arbres

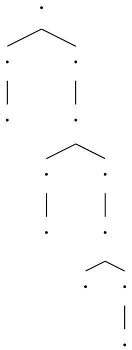
- De la même manière, le résultat nous permet d'utiliser la méthode des arbres pour vérifier la validité d'un argument déductif :
- P. ex. un argument déductif de la forme
    1. A
    2. B
    3. $\therefore C$

  est valide si et seulement si $\vdash_{L_0} \neg(A \wedge (B \wedge \neg C))$ (si la conjonction de la conjonction de ses prémisses avec la négation de sa conclusion est inconsistante)

Les arbres

- En termes mathématiques, les arbres que nous allons construire pour prouver des faits logiques sont des arbres binaires étiquetés enracinés
- Un arbre consiste en des nœuds et les arêtes (des linges) qui les connectent
- La méthode des arbres se base sur des diagrammes des arbres qui ont une structure comme la suivante (nœuds marqués par "·") :d

Les arbres



- Donc, nos arbres “poussent” de haut en bas : on appelle le nœud le plus haute la racine ; les branches de l’arbre poussent de la racine vers le bas

Construction d'un arbre

- Nous commençons toujours la construction d'un arbre avec la racine
- On écrit dans ce premier nœud, les unes au-dessus des autres, toutes les formules qui sont supposées comme hypothèses ou qui correspondent aux prémisses de l'argument (s'il y en a) et la négation soit de la formule qui doit être déduite, soit de la formule qui correspond à la conclusion de l'argument
- Attention : n'oubliez pas de mettre cette dernière formule en parenthèses avant de former sa négation si c'est une formule complexe ; comme vous le savez, "$\neg(p \vee q)$" n'est p. ex. pas équivalent à "$\neg p \vee q$"

Construction d'un arbre

- Un exemple typique : On construit un arbre pour prouver le fait que $A, B, C \vdash_{L_0} D$; dans ce cas on inscrit dans la racine A, B, C et $\neg D$
- On peut aussi utiliser les arbres pour montrer qu'une formule A est prouvable (ou pas) – pour le faire, on écrit simplement sa négation $\neg A$ (et aucune autre formule) dans la racine
- Pareillement, on peut construire un arbre pour prouver qu'une formule A est inconsistante (ou pas) – pour le faire, on met la formule A elle-même (et aucune autre formule) dans la racine de l'arbre

Construction d'un arbre

- Comme en déduction naturelle, le reste de l'arbre est construit par l'application des *règles syntaxiques* qui sont liées aux connecteurs logiques de L_0
- À l'exception de la négation, il existe deux règles pour chaque connecteur logique : une règle qui nous permet d'éliminer une formule qui a le connecteur comme connecteur logique principal et une règle qui nous permet d'éliminer la négation d'une telle formule
- À la différence de la déduction naturelle, toutes ces règles sont des règles d'élimination – les formules qui résultent de l'application d'une règle à une formule sont toujours logiquement moins complexes que cette formule

Construction d'un arbre – les règles

Règle pour la négation

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

Règles pour la conjonction

$$\begin{array}{c} A \wedge B \\ | \\ A \\ B \end{array} \qquad \begin{array}{c} \neg(A \wedge B) \\ \diagup \quad \diagdown \\ \neg A \qquad \neg B \end{array}$$

Règles pour la disjonction

$$\begin{array}{c} A \vee B \\ \diagup \quad \diagdown \\ A \qquad B \end{array} \qquad \begin{array}{c} \neg(A \vee B) \\ | \\ \neg A \\ \neg B \end{array}$$

Construction d'un arbre – les règles

Règles pour le conditionnel matériel

$A \to B$

$\neg A \quad B$

$\neg(A \to B)$

A

$\neg B$

Règles pour le biconditionnel matériel

$A \leftrightarrow B$

$A \quad \neg A$

$B \quad \neg B$

$\neg(A \leftrightarrow B)$

$A \quad \neg A$

$\neg B \quad B$

Construction d'un arbre

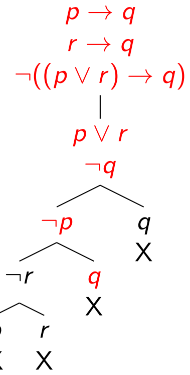
- Pour prouver qu'un ensemble de formules est inconsistant, nous devons montrer que toutes *les branches* d'un arbre *se terminent*
- Qu'est-ce qu'une branche d'un arbre et quand est-ce qu'elle se termine ?
- Une *branche* est un trajet qui commence à la racine et continue jusqu'en bas (jamais en haut !) à travers des arêtes/lignes de nœud à nœud

Construction d'un arbre

Les branches

- Voilà un arbre dont les formules d'une des branches sont marquées en rouge :

À prouver : $p \to q, r \to q \vdash_{L_0} (p \vee r) \to q$



Construction d'un arbre

Les branches

- Voilà une branche différente du même arbre :

À prouver : $p \rightarrow q, r \rightarrow q \vdash_{L_0} (p \vee r) \rightarrow q$

$$p \rightarrow q$$
$$r \rightarrow q$$
$$\neg((p \vee r) \rightarrow q)$$
$$p \vee r$$
$$\neg q$$

- $\neg p$
  - $\neg r$
    - p X
    - r X
  - q X
- q X

Construction d'un arbre

Application des règles

- Pour créer de nouveaux nœuds sur une branche, on applique une règle à une formule complexe sur la même branche
- On peut toujours utiliser toutes les formules qui sont au-dessus du dernier nœud sur la branche, jusqu'à la racine
- Sur la même branche, chaque formule est utilisée seulement *une fois*, mais la même formule peut aussi être utilisée sur une autre branche
- Il peut être utile de marquer si on a déjà utilisé une formule sur une branche, p. ex. par une flèche de la formule utilisée à la formule qui est le résultat de l'application d'une règle sur cette formule, ou en cochant chaque formule qui a été utilisée sur toutes les branches

Construction d'un arbre

- On a toujours le choix de l'ordre dans lequel utiliser les formules sur une branche
- Une fois une formule choisie, sa forme logique détermine la règle qu'on lui applique pour continuer la construction de l'arbre
- Comme toujours, faites attention aux parenthèses, particulièrement si vous appliquez les règles pour la négation

Construction d'un arbre

Quand est-ce qu'une branche se termine ?

- Une branche *se termine* si elle contient, dans le même ou dans différents nœuds, une formule et sa négation (on peut dire : elle contient une contradiction)
- Sur les deux différentes branches marquées en rouge sur l'arbre ci-dessus, ce sont "$\neg q$" et "q" (première branche) et "$\neg p$" et "p" (deuxième branche)
- Pour montrer qu'une branche s'est terminée, on met la lettre "X" sous son dernier nœud, si on a vérifié qu'elle contient une contradiction

Construction d'un arbre

Complèter la preuve – déduction et validité d'un argument

- Si toutes les branches d'un arbre se terminent, que toutes les formules sur ces branches aient été utilisées (comme dans l'exemple au-dessus) ou non, cela montre que la conjonction des formules dans la racine est inconsistante
- Si ces formules correspondent aux prémisses d'un argument et à la négation de sa conclusion, cela montre que l'argument est valide
- D'une façon plus générale, si ces formules sont les hypothèses d'une preuve et la négation de la formule qui doit être déduite de ces hypothèses, cela montre que la formule peut être déduite de ces hypothèses
- Si, par contre, il y a (au moins) une branche sur l'arbre qui *ne se termine pas* après que toutes les formules qu'elle contient ont été utilisées, cela montre que la conjonction des formules dans la racine est consistante – la formule ne peut pas être déduit des hypothèses/l'argument n'est pas valide

Construction d'un arbre

Compléter la preuve – prouvabilité

- Pareillement, une formule A est *prouvable* si toutes les branches d'un arbre contenant seulement $\neg A$ dans la racine se terminent (autrement elle *n'est pas prouvable*)
- Une formule A est *inconsistante* si toutes les branches d'un arbre contenant seulement A dans la racine se terminent (autrement elle est *consistante*)

Exemples des preuves

À prouver : $p \rightarrow q, p \vdash_{L_0} q$

$$p \rightarrow q$$
$$p$$
$$\neg q$$

$\neg p$	q
X	X

Exemples des preuves

À prouver : $p \to q, \neg q \vee r, \neg(r \wedge \neg s) \vdash_{L_0} p \to s$

$$
\begin{array}{c}
p \to q \\
\neg q \vee r \\
\neg(r \wedge \neg s) \\
\neg(p \to s) \\
| \\
p \\
\neg s \\
\begin{array}{cc}
\begin{array}{c} \neg p \\ \text{X} \end{array} &
\begin{array}{c} q \\
\begin{array}{cc}
\begin{array}{c} \neg q \\ \text{X} \end{array} &
\begin{array}{c} r \\
\begin{array}{cc}
\begin{array}{c} \neg r \\ \text{X} \end{array} &
\begin{array}{c} \neg\neg s \\ | \\ s \\ \text{X} \end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
$$

Exemples des preuves

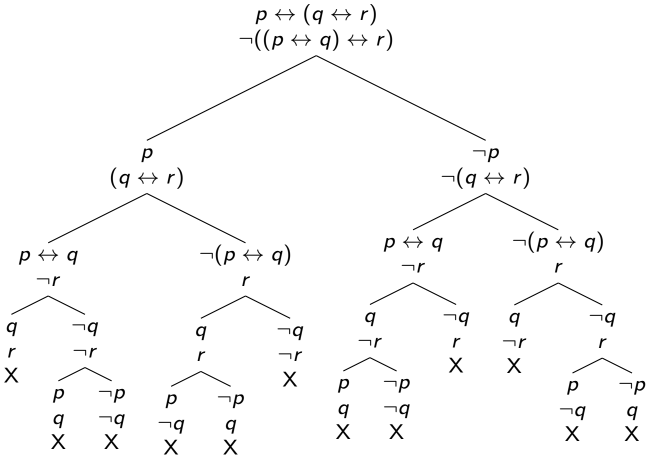
À prouver : $p \rightarrow q, q \nvdash_{L_0} p$

$$p \rightarrow q$$
$$q$$
$$\neg p$$
$$\neg p \qquad q$$

Commentaire : La seule règle qu'on peut appliquer à une formule dans cet arbre est la règle pour la négation du conditionnel matériel, mais le résultat est un arbre avec deux branches qui ne se terminent pas – notez qu'on peut en principe appliquer encore et toujours la même règle à la même formule, mais cela ne servirait à rien, car le résultat serait un arbre infini qui ne se terminerait jamais

Exemples des preuves

À prouver : $p \leftrightarrow (q \leftrightarrow r) \vdash_{L_0} (p \leftrightarrow q) \leftrightarrow r$



Quelques astuces pour la construction des arbres

- Ne combinez pas plusieurs règles dans une seule étape – on construit l'arbre pas à pas, par l'application d'une règle à une formule
- Au début, il peut être plus facile de suivre l'ordre des formules dans la branche pour construire l'arbre ; avec un peu d'exercice, vous allez parfois voir directement quelle formule il faut utiliser pour terminer une branche
- Si vous avez le choix entre plusieurs formules, pour simplifier l'arbre, commencez avec les formules qui donnent seulement un nouveau nœud
- Essayez à compléter une branche à la fois
- Pour s'assurer qu'une branche ne se termine pas, il faut avoir utilisé toutes les formules complexes sur la branche

Comment découvrir un contrexemple sur un arbre

- Bien que la méthode des arbres soit une méthode syntaxique, on peut extraire des informations sémantiques d'un arbre qui a une branche qui ne se termine pas
- Une telle branche nous donne *un contrexemple* au fait qui doit être prouvé
- P. ex. si on construit un arbre pour prouver $A, B \vdash_{L_0} C$, cette branche indique une distribution des valeurs de vérité parmi les formules atomiques pertinentes pour laquelle A et B sont vraies, mais C est fausse

Comment découvrir un contrexemple sur un arbre

Comment identifier un tel contrexemple ?

- Si nous trouvons une formule atomique sur la branche, nous lui assignons la valeur de vérité **V**
- Et si nous trouvons la négation d'une formule sur la branche, nous lui assignons le valeur de vérité **F**
- Si ni une formule atomique, ni sa négation ne sont sur la branche, sa valeur de vérité n'est pas pertinente pour le contrexemple et peut être assignée arbitrairement

La méthode des arbres et *reductio ad absurdum*

- Par exemple : dans l'arbre suivant, sur la branche marquée en rouge, nous trouvons"p" et "$\neg q$" et nous savons que "$((p \rightarrow q) \rightarrow q) \rightarrow q$" est fausse si "$p$" est vrai et "$q$" est faux

À prouver : $\nvdash_{L_0} ((p \rightarrow q) \rightarrow q) \rightarrow q$

$$\neg(((p \rightarrow q) \rightarrow q) \rightarrow q)$$

$$(p \rightarrow q) \rightarrow q$$

$$\neg q$$

$$\begin{array}{cc} \neg(p \rightarrow q) & q \\ & \text{X} \\ p & \\ \neg q & \end{array}$$

La méthode des arbres et *reductio ad absurdum*

- En effet, la méthode des arbres est proche de la forme d'argumentation *reductio ad absurdum* : on montre que si on suppose toutes les formules dans la racine, on peut déduire une contradiction

Exemples

À prouver : $q \vdash_{L_0} p \rightarrow q$

$$
\begin{array}{c}
q \\
\neg(p \rightarrow q) \\
| \\
p \\
\neg q \\
\text{X}
\end{array}
$$

Exemples

À prouver : $\neg p \vdash_{L_0} p \rightarrow q$

$$\begin{array}{c} \neg p \\ \neg(p \rightarrow q) \\ | \\ p \\ \neg q \\ \text{X} \end{array}$$

Exemples

À prouver : $p \wedge q, r \vdash_{L_0} s \rightarrow s$

$$\begin{array}{c} p \wedge q \\ r \\ \neg(s \rightarrow s) \\ | \\ s \\ \neg s \\ \text{X} \end{array}$$

Notez que dans ce cas, les hypothèses sont insignifiantes ; "$s \rightarrow s$" est prouvable à partir de n'importe quelle hypothèse ; en fait, cette formule, comme toutes les tautologies, est prouvable à partir de zéro hypothèses hypothèses

Exemples

À prouver : $q, r \vdash_{L_0} p \to q$

$$\begin{array}{c} q \\ r \\ \neg(p \to q) \\ | \\ p \\ \neg q \\ \text{X} \end{array}$$

Notez que cet arbre nous montre une propriété générale de $\vdash_{L_0}$: si une formule A peut être déduite d'un ensemble des hypothèses Δ, on peut ajouter n'importe quelle autre hypothèse B à Δ et il sera encore le cas que A peut être déduite de ce nouvel ensemble de formules. Cette propriété est appelée *monotonicité*

Monotonicité

Définition : Monotonicité de $\vdash_{L_0}$ Pour tout ensemble de formules Δ, si $\Delta \vdash_{L_0} A$, pour toute formule B, $\Delta, B \vdash_{L_0} A$.

Notez qu'il suit de cette définition qu'on peut ajouter un nombre arbitraire d'hypothèses à une preuve correcte sans mettre à mal la relation de dérivabilité entre elles et la formule qui est prouvée

Exemples

À prouver : $\vdash_{L_0} ((p \rightarrow q) \wedge (p \rightarrow r)) \rightarrow (p \rightarrow (q \wedge r))$

$$\neg((p \rightarrow q) \wedge (p \rightarrow r)) \rightarrow (p \rightarrow (q \wedge r))$$

$$(p \rightarrow q) \wedge (p \rightarrow r)$$
$$\neg(p \rightarrow (q \wedge r))$$

$$p \rightarrow q$$
$$p \rightarrow r$$

$$p$$
$$\neg(q \wedge r)$$

$\neg p$ X | q

$\neg p$ X | r

$\neg q$ X | $\neg r$ X

Exemples

À prouver : $p \vdash_{L_0} \neg(q \vee p) \rightarrow \neg q$

$$
\begin{array}{c}
p \\
\neg(\neg(q \vee p) \rightarrow \neg q) \\
| \\
\neg(q \vee p) \\
\neg\neg q \\
| \\
q \\
| \\
\neg q \\
\neg p \\
\text{X}
\end{array}
$$

Exemples

À prouver : $p \leftrightarrow q \vdash_{L_0} (p \wedge q) \vee (\neg p \wedge \neg q)$

$$p \leftrightarrow q$$
$$\neg((p \wedge q) \vee (\neg p \wedge \neg q))$$
$$\neg(p \wedge q)$$
$$\neg(\neg p \wedge \neg q)$$

- $\neg p$
  - p, q, X
  - $\neg p$, $\neg q$
    - $\neg\neg p$, p, X
    - $\neg\neg q$, q, X
- $\neg q$
  - p, q, X
  - $\neg p$, $\neg q$
    - $\neg\neg p$, p, X
    - $\neg\neg q$, q, X