

La déduction naturelle automatique

Éric GILLET

Abstract. Automatic demonstration privileges efficacy and doesn't care about the structural information of the proofs. First, I try to show the interest of the natural deduction for the study of both the reasoning and the signification, and then I propose an algorithm for the automatic natural deduction.

Résumé. La démonstration automatique privilégie l'efficacité et se préoccupe peu de l'information structurelle des preuves. Après avoir tenté de montrer l'intérêt et l'importance de la déduction naturelle tant pour l'étude du raisonnement que de la signification, je propose un algorithme de déduction naturelle automatique.

Keywords: Natural deduction, automatic deduction, algorithm, signification. **Mots-clés :** Déduction naturelle, déduction automatique, algorithme, signification.

1. La démonstration automatique

La démonstration automatique, née il y a plus trente ans avec la méthode de résolution de Robinson (Robinson : 1965), vise à établir algorithmiquement l'existence d'une relation de déductibilité entre les prémisses et la conclusion d'un raisonnement. L'enjeu est évidemment de taille puisqu'on peut faire exécuter cet algorithme par un ordinateur et disposer ainsi d'un outil qui permette de construire automatiquement la preuve qu'un raisonnement est correct.

En fait, la situation est moins idéale qu'il n'y paraît, puisqu'il y a des raisons théoriques et pratiques qui viennent limiter la démonstration automatique. D'abord, d'un point de vue théorique, il faut noter que tout

✉ Université de Liège; Séminaire de Logique; 32, place du XX Août; B-4000 Liège (Belgique).
E-mail : egillet@ulg.ac.be

raisonnement est un rapport entre des propositions formulées dans un langage et que ce langage possède ou non la propriété de *décidabilité*, à savoir qu'il existe pour ce langage, une procédure qui permet de décider en un nombre *fini* d'étapes si un raisonnement est valide ou non. Le langage du *calcul des propositions* est décidable, c'est-à-dire que pour tout raisonnement que l'on peut formuler au moyen de propositions atomiques non analysées (p, q, r, \dots) reliées par des connecteurs vérifonctionnels (non : \neg , et : \wedge , ou : \vee , si... alors : \supset , si et seulement si : \equiv , ...), il est possible en droit de dire si oui ou non il est valide ; par contre le langage du premier ordre (ou calcul des prédicats) est, quant à lui, indécidable : dès qu'un raisonnement dépend de propositions analysées en termes d'objets et de relations, on ne peut plus garantir l'existence d'une procédure qui se terminera¹. Ensuite, d'un point de vue pratique, l'existence d'une procédure qui se termine en un nombre fini d'étapes ne signifie pas que cette procédure soit réalisable dans un temps acceptable, aussi l'effort principal accompli dans le cadre de la recherche en démonstration automatique a-t-il consisté à développer des méthodes de preuve les plus efficaces possible, puis à prouver l'équivalence des différentes méthodes.

Ces méthodes efficaces appartiennent aussi bien à l'approche sémantique qu'à l'approche syntaxique. La première approche consiste à étudier un raisonnement comme une relation de conséquence logique entre les prémisses et la conclusion : quelle que soit l'interprétation des propositions atomiques présentes dans le raisonnement, si les prémisses sont vraies, la conclusion l'est également. Prouver la validité d'un raisonnement revient donc à passer en revue toutes les interprétations et à vérifier que celles qui rendent les prémisses vraies vérifient également la conclusion. Une telle façon de procéder est évidemment assez coûteuse puisqu'il y a 2^n interprétations différentes des n propositions atomiques d'un raisonnement : le nombre de modèles à vérifier croît exponentiellement avec le nombre de propositions atomiques. Il est donc plus intéressant d'utiliser une méthode *indirecte*, à savoir, une méthode qui cherche à construire un seul modèle, contre-exemple du raisonnement, qui rende les prémisses vraies et la conclusion fausse. C'est alors l'impossibilité d'établir un tel modèle qui permet de conclure que le raisonnement est valide.

¹ Le calcul des prédicats est néanmoins *semi-décidable*, il existe une procédure qui permet d'établir en un nombre fini d'étapes la validité de raisonnements valides ; par ailleurs, cette procédure peut ne pas se terminer pour certains raisonnements non valides.

C'est avec ces méthodes indirectes que l'approche sémantique offre les meilleurs résultats, qu'il s'agisse d'une des déclinaisons de la méthode des connexions ou des tableaux sémantiques.

La seconde approche, dite syntaxique ou axiomatique, repose sur l'utilisation de règles déductives qui permettent d'établir une preuve du raisonnement en construisant la suite des étapes passant progressivement des prémisses et des éventuels axiomes à la conclusion. Le plus souvent cette approche est assez difficile à mettre en œuvre puisque, à chaque étape de l'établissement d'une preuve, il faut choisir les formules et la règle que l'on va utiliser sans avoir la garantie que l'étape réalisée rapproche effectivement de la conclusion. Par contre, une fois la preuve construite, elle constitue un argument très persuasif en faveur de la correction du raisonnement puisqu'elle montre comment on peut passer des prémisses à la conclusion en faisant exclusivement usage de règles admises.

Historiquement, une approche syntaxique, la méthode de la résolution, a joué un rôle déterminant dans le développement de la démonstration automatique. Il s'agit également d'une méthode indirecte puisqu'il s'agit de montrer que si on adjoint la négation de la conclusion aux prémisses, on obtient un ensemble de propositions dont on peut dériver une contradiction. La difficulté du choix des règles est contournée en n'utilisant qu'une seule règle et en réécrivant toutes les formules au moyen des seules négation \neg et disjonction \vee . Bien entendu, une preuve réalisée par cette méthode efficace véhicule moins l'évidence qu'une preuve faisant droit à la spécificité de chaque connecteur.

2. La déduction naturelle

La déduction naturelle a été développée par Gentzen en 1934 (Gentzen : 1934, 1935), c'est une approche syntaxique dont le principal intérêt réside dans le fait qu'il n'y a pas d'axiome et que la puissance déductive est entièrement issue d'un jeu de règles associées à l'introduction et à l'élimination des connecteurs dans le cours d'une preuve. Ainsi, la contribution de chaque mot logique est isolée et les preuves obtenues en rendent parfaitement compte. Cette méthode est dite naturelle par la volonté affichée par Gentzen de n'introduire que des règles déductives qui ont une contrepartie évidente dans le raisonnement en langue naturelle.

Il existe plusieurs façons de représenter une preuve en déduction naturelle, et je privilégie ici une représentation graphique qui illustre assez bien les liens entre les diverses parties d’une déduction.

Une déduction naturelle est un enchaînement de règles qui prennent en entrée les prémisses du raisonnement et qui fournissent en conclusion une nouvelle formule qui dépend des prémisses, et qui à son tour peut être utilisée en entrée d’une règle. Les règles se subdivisent en règles d’introduction et d’élimination comme on peut le voir dans le tableau suivant.

Règles d’introduction	Règles d’élimination
$\frac{A \quad B}{A \wedge B} I\wedge$	$\frac{A \wedge B}{A} E\wedge \quad \frac{A \wedge B}{B} E\wedge$
$\frac{\begin{array}{ l} \rightarrow A \\ \vdots \\ B \end{array}}{A \supset B} I\supset$	$\frac{A \quad A \supset B}{B} E\supset$
$\frac{A}{A \vee B} I\vee \quad \frac{B}{A \vee B} I\vee$	$\frac{A \vee B \quad \left[\begin{array}{ l} \rightarrow A \\ \vdots \\ C \end{array} \right] \quad \left[\begin{array}{ l} \rightarrow B \\ \vdots \\ C \end{array} \right]}{C} E\vee$
$\frac{\begin{array}{ l} \rightarrow A \\ \vdots \\ C \wedge \neg C \end{array}}{\neg A} I\neg$	$\frac{\neg\neg A}{A} E\neg$
$\frac{A \supset B \quad B \supset A}{A \equiv B} I\equiv$	$\frac{A \equiv B}{A \supset B} E\equiv \quad \frac{A \equiv B}{B \supset A} E\equiv$

À ces règles s’ajoute la règle de réflexivité qui permet de répéter une formule soit présente parmi les prémisses soit déjà démontrée :

$$\overline{A} R$$

Les trois règles $I\supset$, $E\vee$ et $I\neg$ supposent une déduction auxiliaire représentée par un encadrement partant d’une prémisses auxiliaire désignée par une flèche et se terminant à une conclusion dépendant de la prémisses auxiliaire. La règle de réflexivité ne peut répéter en dehors d’une déduction

auxiliaire une formule y apparaissant, puisqu'une telle formule dépend de l'assertion de la prémisses auxiliaire et n'est donc pas assurée en dehors de la déduction auxiliaire.

La règle de réflexivité semble assez anodine, elle permet seulement de réutiliser dans un raisonnement un résultat antérieur, et pourtant, le fait même d'en autoriser l'utilisation est caractéristique d'un type de raisonnement où prémisses et résultats intermédiaires sont lus de façon absolue, indépendamment du contexte que constitue l'ordre d'enchaînement dans la déduction. C'est le cas de la logique dite classique, mais cela n'est plus vrai si l'on envisage un raisonnement non monotone où une information nouvelle vient modifier la lecture de tout ce qui précède. Pour ce qui nous concerne ici, nous nous limiterons à la logique classique.

Le caractère *naturel* des règles s'aperçoit aisément. Toutes les règles relient une opération effectuée dans le métalangage et un connecteur du langage objet, et c'est précisément ce lien qui définit le sens du connecteur. Cette liaison est particulièrement apparente quand la règle repose sur une déduction auxiliaire, mais cela est déjà vrai dans les cas plus simples. Ainsi dans la règle $I\wedge$, c'est parce que l'on dispose d'une déduction des propositions A et B que l'on peut conclure à une déduction de la conjonction $A \wedge B$.

- I \wedge Pour pouvoir déduire la proposition $A \wedge B$ il faut préalablement avoir déduit (ou introduit comme prémisses) les deux propositions A et B .
- E \wedge Il est toujours possible de simplifier une conjonction pour n'en retenir qu'un des éléments.
- I \supset Pour déduire une proposition conditionnelle $A \supset B$ à partir d'un certain nombre de prémisses, il suffit d'ajouter une prémisses auxiliaire A et de montrer que l'on peut en dériver la conclusion auxiliaire B . On a donc montré que si on avait A on pouvait déduire B , et de cette démarche on conclut à la proposition $A \supset B$.
- E \supset Il s'agit ici de la règle de détachement bien connue : si vous avez une déduction d'une proposition conditionnelle et que par ailleurs vous savez que la condition est remplie, vous pouvez conclure que la conséquence l'est également.
- IV Cette règle d'addition est une règle qui affaiblit une proposition connue : si vous connaissez A , *a fortiori* pouvez-vous conclure que la disjonction $A \vee B$ est vraie.
- E \vee De deux choses l'une : soit vous pouvez déduire C de A , soit vous pouvez le déduire de B ; et donc puisque vous avez la disjonction $A \vee B$, vous pouvez déduire C .

- I¬ Il s'agit du modèle de la démonstration par l'absurde : si une contradiction suit d'une prémisses que vous avez posée, vous pouvez déduire que cette prémisses ne tient pas et conclure à sa négation.
- E¬ Cette règle caractérise la logique classique : à partir de la double négation d'une formule, il est possible de déduire la formule initiale.
- I≡ Le biconditionnel se construit à partir des deux conditionnels converses l'un de l'autre.
- E≡ À partir d'un biconditionnel, il est toujours possible de déduire un des deux conditionnels qui le constituent.

Une preuve réalisée en déduction naturelle a l'avantage de pouvoir être convertie en un argument en langue naturelle. Prenons l'exemple du raisonnement partant de trois prémisses que voici :

- Soit l'arôme de café embaume l'immeuble, soit la porte n'est pas ouverte.
- Quand l'arôme de café embaume l'immeuble, le facteur n'est pas pressé.
- Quand le facteur est pressé, il laisse la porte ouverte.

pour aboutir à la conclusion suivante :

- Le facteur n'est pas pressé.

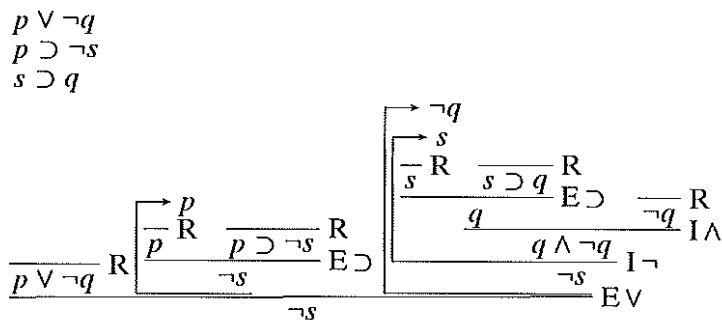
Transcrit en langage formel, cela donne

- $p \vee \neg q$
- $p \supset \neg s$
- $s \supset q$

Donc

- $\neg s$

Une preuve en déduction naturelle de ce raisonnement peut se construire et se représenter comme suit.



En langage naturel on peut alors le retranscrire comme suit :

Soit l'arôme de café embaume l'immeuble, soit la porte n'est pas ouverte.
Quand l'arôme de café embaume l'immeuble, le facteur n'est pas pressé.
Quand le facteur est pressé, il laisse la porte ouverte.

De deux choses l'une :

- soit, premièrement, l'arôme de café embaume l'immeuble :
dans ce cas, comme on sait que quand l'arôme de café embaume l'immeuble, le facteur n'est pas pressé, on peut conclure que le facteur n'est pas pressé ;
 - soit, deuxièmement, la porte n'est pas ouverte :
supposons que le facteur est pressé :
comme on sait que quand il est pressé, il laisse la porte ouverte, on devrait en conclure que la porte est ouverte, or, dans ce deuxième cas, la porte n'est pas ouverte, ce qui établit que notre supposition est absurde et donc, on peut conclure de ce deuxième cas que le facteur n'est pas pressé ;
- comme on sait que l'un au moins de ces cas est réalisé, on peut conclure que le facteur n'est pas pressé.

L'avantage de la méthode de la déduction naturelle est de fournir la preuve d'un raisonnement dans un format qui est compatible avec le raisonnement en langue naturelle. À elle seule, cette propriété justifie déjà l'intérêt de disposer d'une méthode de preuve automatique, mais comme nous allons le voir, d'autres éléments permettent d'enrichir les critères d'évaluation de la méthode.

3. Déduction, évidence et signification

Le but premier d'une méthode de preuve est de pouvoir vérifier qu'un ensemble de prémisses entraîne bien une conclusion, et de ce point de vue, le choix de la méthode est secondaire, pourvu qu'elle donne le résultat escompté.

Autre chose est de comparer les méthodes de preuve du point de vue de l'évidence ou du degré de persuasion accordé aux différentes preuves. La déduction naturelle vise à rencontrer cet aspect tout en restant une méthode formelle² : chaque étape peut être justifiée dans

² Une preuve formelle est complètement explicite, ce qui s'oppose à la conception intuitive d'une preuve convaincante car le développement intégral de certaines étapes évidentes nuit à la lisibilité et viole les principes conversationnels de la pertinence.

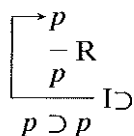
la langue naturelle, l'enchaînement des étapes reproduit les transitions du raisonnement en langue naturelle, et enfin, l'économie générale des preuves (successions d'analyses et de synthèses) explicite une démarche naturelle. La déduction naturelle ajoute donc, à la propriété commune à toute règle d'inférence de véhiculer la vérité des prémisses à la conclusion, une propriété analogue qui transmet l'évidence des prémisses à la conclusion.

Mais il existe une troisième manière de comparer les méthodes de preuve qui s'appuie sur une conception élargie de la signification. Si on réduit la signification d'un énoncé à ses conditions de vérité ou à l'ensemble de ses conséquences, rien ne distingue un théorème ou une vérité de logique des autres théorèmes. Et pourtant, il apparaît que chaque théorème jouit d'une signification épistémique particulière, rendue visible tant par sa valeur de connaissance (Gillet : 1997) que par son rôle dans l'économie générale de la théorie. Une façon de rendre compte de cet aspect de la signification est de considérer que la preuve d'un théorème exhibe une certaine structure formelle unique, différente de celle de tous les autres théorèmes, et que cette structure n'est en rien arbitraire, mais au contraire, reflète la relation déductive qu'entretient le théorème avec l'ensemble du système.

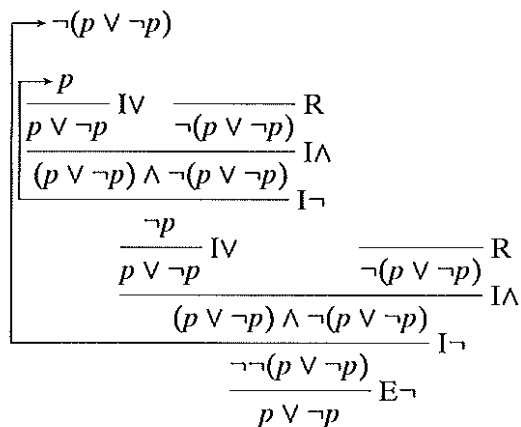
De ce point de vue, la déduction naturelle est particulièrement intéressante puisque toutes les preuves sont construites à partir d'un ensemble riche de règles qui sont à la fois indépendantes les unes des autres et qui, de plus, définissent isolément le rôle de chaque connecteur logique.

Pour illustrer l'apport de la preuve dans la signification d'un théorème, comparons les deux théorèmes suivants. Le premier théorème, $p \supset p$, principe de l'idempotence, est un théorème présent dans tous les systèmes : il affirme simplement que si une certaine proposition est réalisée, alors elle l'est. Le second théorème, $p \vee \neg p$, appelé principe du tiers exclus, est lui récusé par certains, et en particulier par les intuitionnistes qui n'acceptent pas qu'on puisse dire *a priori* que toute proposition est soit vraie, soit fausse. Dans un système déductif qui ne différencie pas les différents connecteurs et qui introduit la proposition $p \vee q$ comme une abréviation de $\neg p \supset q$, il est clair que les preuves des deux principes seront identiques, à ceci près que la seconde preuve fera appel à la définition abrégative de la disjonction (qui, bien entendu, n'est pas acceptée par les intuitionnistes). Par contre les preuves des mêmes principes en déduction naturelle sont fortement contrastées, comme on peut le voir ci-dessous.

Idempotence



Tiers exclu



La simple observation de ces deux preuves permet de comprendre pourquoi le premier principe est plus universellement admis que le second. Le second, quant à lui, dépend de l'imbrication de deux preuves par l'absurde (admises par les intuitionnistes) suivie d'une règle d'élimination de la double négation (pour les intuitionnistes, la négation de la fausseté ne suffit pas à établir la vérité).

Si aux critères d'efficacité mis en avant par la démonstration automatique, on ajoute les critères d'évidence et de signification, on justifie aisément l'intérêt qu'il y a à développer des méthodes de déduction naturelle automatique dont le but n'est pas seulement de dire si oui ou non il y a un lien déductif entre les prémisses et la conclusion mais également à expliciter ce lien³.

³ Il existe un algorithme de déduction automatique dû à Bolotov, Bocharov et Gorchakov (1996) qui utilise une version de la déduction naturelle sensiblement différente de celle présentée ici : les règles d'élimination de la disjonction et d'introduction de la négation sont traitées en restant au niveau de la phrase, sans faire appel à des déductions auxiliaires, ce qui rend le système moins pertinent pour ce qui nous occupe ici.

4. La multiplicité des déductions

Une première difficulté de la déduction naturelle est qu'il existe une infinité de déductions équivalentes. Il s'agit donc de privilégier une direction de recherche qui donne la meilleure déduction possible. Il faut distinguer parmi la multiplicité des preuves possibles trois types différents : d'abord il existe des preuves qui font des *détours* inutiles. Ainsi la déduction ci-dessous est une nouvelle preuve de l'idempotence.

$$\begin{array}{c}
 \begin{array}{l}
 \rightarrow p \\
 \begin{array}{l}
 \rightarrow p \\
 \frac{}{p} R \\
 \frac{}{p} I \supset
 \end{array} \\
 \frac{}{p \supset p} I \supset \\
 \frac{}{p} R \quad \frac{}{p \supset p} R \\
 \frac{}{p} R \quad \frac{}{p \supset p} E \supset \\
 \frac{}{p} I \supset \\
 \frac{}{p \supset p} I \supset
 \end{array}
 \end{array}$$

Il est tout à fait possible d'éliminer ce type de déduction : il suffit de remarquer que les détours correspondent toujours à l'utilisation d'une même proposition à la fois comme conclusion d'une règle d'introduction puis comme prémisses d'une règle d'élimination. On qualifie une preuve qui ne contient pas de tels détours de *normale*; un algorithme évite très simplement la non-normalité en gardant une hiérarchie des objectifs à construire.

Une deuxième raison conduisant à la multiplication des preuves est qu'il est possible de construire une même étape intermédiaire à plusieurs endroits différents comme l'atteste cet exemple.

$ \begin{array}{c} r \\ (r \vee s) \supset q \\ \begin{array}{l} \rightarrow p \\ \frac{}{r} R \\ \frac{}{r \vee s} IV \quad \frac{}{(r \vee s) \supset q} R \\ \frac{}{q} E \supset \\ \frac{}{p \supset q} I \supset \end{array} \end{array} $	$ \begin{array}{c} r \\ (r \vee s) \supset q \\ \frac{}{r} R \\ \frac{}{r \vee s} IV \quad \frac{}{(r \vee s) \supset q} R \\ \frac{}{q} E \supset \\ \begin{array}{l} \rightarrow p \\ \frac{}{q} R \\ \frac{}{p \supset q} I \supset \end{array} \end{array} $
--	---

La version de gauche construit la proposition *q* à l'intérieur de la déduction auxiliaire, alors que celle de droite le fait d'emblée. Il est clairement préférable de privilégier la seconde version, et donc de déduire ce qui peut l'être au plus haut niveau de la hiérarchie des emboîtements de déductions, sans quoi, on sera amené à devoir reconstruire plusieurs

fois une déduction construite indûment à un niveau inférieur d'où elle ne peut être exportée.

La troisième variété de déductions multiples a trait à la sur-information contenue dans les prémisses qui peuvent dès lors être exploitées diversement pour construire la conclusion.

$$\begin{array}{c}
 p \\
 p \supset q \\
 p \supset \neg p \\
 \hline
 p \text{ R} \quad \frac{p \supset q}{p \supset \neg p} \text{ R} \\
 \hline
 \neg p \text{ E} \supset
 \end{array}$$

$$\begin{array}{c}
 p \\
 p \supset q \\
 p \supset \neg p \\
 \hline
 p \text{ R} \quad \frac{p \supset q}{p \supset \neg p} \text{ R} \\
 \hline
 q \text{ E} \supset
 \end{array}$$

$$\begin{array}{c}
 \neg q \\
 \hline
 p \text{ R} \quad \frac{\neg p}{\neg q} \text{ R} \\
 \hline
 p \wedge \neg p \text{ I} \wedge \\
 \hline
 \neg \neg q \text{ I} \neg \\
 \hline
 q \text{ E} \neg
 \end{array}$$

Ces déductions sont équivalentes et rien ne permet de choisir une version plutôt qu'une autre. Ici, ce seront les particularités de l'algorithme qui privilégieront telle ou telle déduction selon que l'espace de recherche aura été exploré en parcourant telle voie d'abord. L'important n'est pas tant d'explorer telle voie avant telle autre, mais de garantir que l'algorithme est général et qu'il explorera le cas échéant toutes les voies.

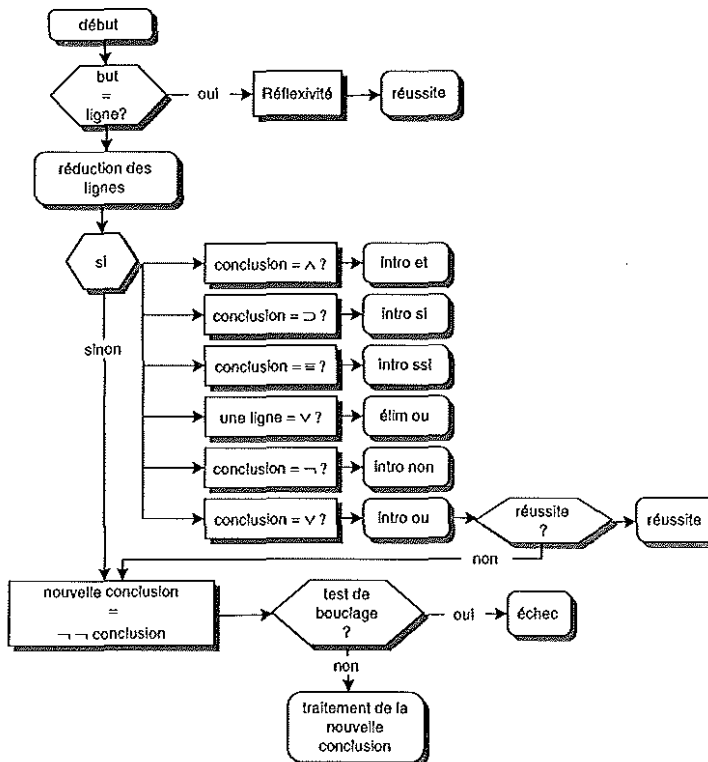
5. L'algorithme de la déduction naturelle

Puisqu'une déduction naturelle est un enchâssement hiérarchique de déductions, l'algorithme proposé utilisera une approche récursive qui s'appliquera de façon identique à chaque niveau. Le problème se ramène à construire le lien déductif entre les lignes actives, issues des prémisses, et la conclusion recherchée, en le décomposant en déductions intermédiaires qui, éventuellement, se décomposeront à leur tour. Ce qui nous fera sortir de cette récursion sera le cas simple où la conclusion recherchée fait déjà partie des prémisses considérées et où la règle de réflexivité s'applique.

Cette procédure s'appuie sur une pile de buts intermédiaires dont le premier est la conclusion de la déduction et sur une liste de lignes actives initialisée par les prémisses de la déduction.

La seule difficulté consiste à choisir sur quelle proposition, ligne active ou but intermédiaire, travailler pour atteindre le résultat escompté. C'est ce choix qui constitue le cœur de l'algorithme présenté ci-dessous.

5.1. Fonctionnement général



- La première étape consiste à vérifier si le but recherché n'est pas déjà présent parmi les lignes actives de la déduction. Si c'est le cas, l'algorithme introduit parmi les lignes actives le but recherché, étiqueté par la règle de réflexivité, et la phase se termine sur une réussite.
- Sinon la deuxième étape consiste à opérer la réduction des lignes actives : c'est une phase analytique qui essaye d'utiliser les règles d'élimination. Cette phase, décrite plus bas, peut se terminer sur une réussite (le but recherché est un constituant d'une ligne éliminée) et dans ce cas l'algorithme se termine.
- La troisième étape introduit une hiérarchie dans l'ordre de traitement des propositions non encore utilisées : il faut d'abord traiter la conclusion si c'est une conjonction, une conditionnelle ou une biconditionnelle ; ensuite, si une ligne active est une disjonction, il faut privilégier

son élimination; enfin, on traite la conclusion si c'est une négation ou une disjonction. Chacune de ces possibilités donne lieu à un traitement subordonné qui peut se terminer par un échec ou une réussite et ce résultat devient le résultat du processus courant.

- La dernière étape consiste à introduire la double négation du but comme nouveau but intermédiaire. Un premier test est effectué pour vérifier qu'il n'y a pas de bouclage : on vérifie que ce nouveau but n'est pas déjà dans la pile des buts; si c'est le cas et si, depuis l'introduction de ce but, il n'y a pas de nouvelle ligne active différente de celles présentes alors, on est dans un processus circulaire et on arrête sur un constat d'échec qui se propage au processus courant.

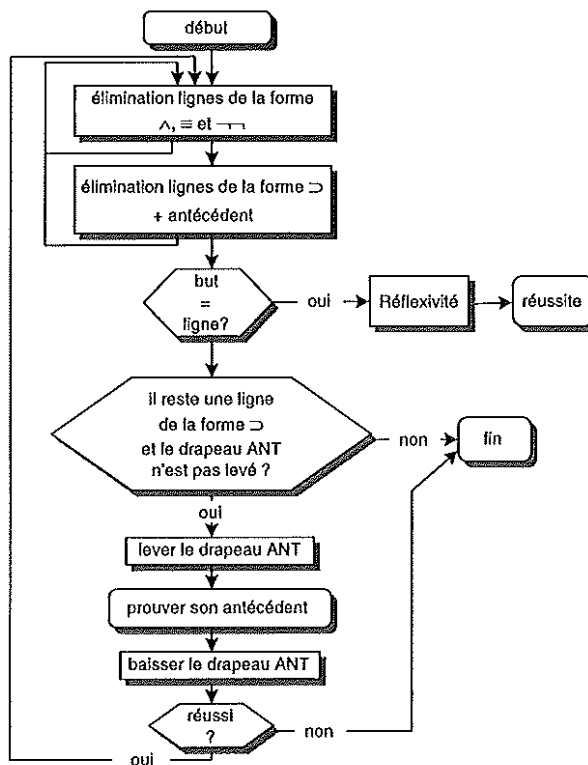
S'il n'y a pas bouclage, on lance un nouveau processus qui hérite des mêmes lignes actives et vise à construire la double négation. Si le résultat est un échec, il se propage au but courant, sinon on introduit le but courant, étiqueté par la règle d'élimination de la négation, parmi les lignes actives, et on termine le processus sur une réussite.

5.2. Réduction des lignes

La réduction des lignes actives consiste à analyser les lignes actives et à en extraire le maximum d'information. Elle est constituée de deux phases assez différentes : la première concerne les lignes actives qui sont des conjonctions, des biconditionnelles, des doubles négations, ou des conditionnelles dont l'antécédent fait partie des lignes actives; la seconde concerne l'élimination des autres conditionnelles.

Les différentes étapes s'articulent selon le diagramme présenté ci-dessous.

- Tant qu'il y a une conjonction ou une biconditionnelle parmi les lignes actives, elle est inactivée et on ajoute les deux lignes issues de l'application des deux règles d'élimination du connecteur considéré; tant qu'il y a une double négation, elle est inactivée et on ajoute la proposition issue de l'élimination de la double négation; tant qu'il y a une conditionnelle et une proposition identique à son antécédent, la conditionnelle est inactivée et on ajoute la conclusion de la règle d'élimination du si.
- Si le but recherché est présent parmi les lignes actives de la déduction, on introduit parmi les lignes actives le but recherché, étiqueté par la règle de réflexivité, et la réduction se termine sur une réussite.
- L'élimination des conditionnelles dont l'antécédent n'est pas une ligne active ne peut se faire que par la construction préalable de

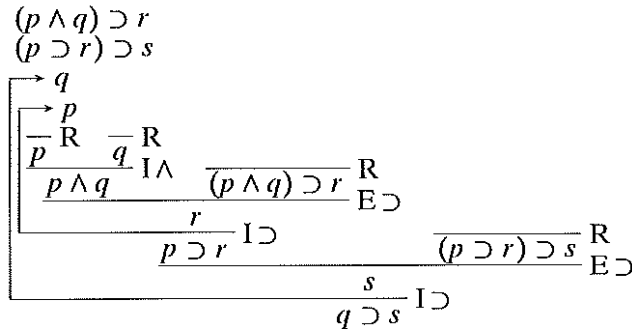


l'antécédent. *A priori* rien ne nous garantit que cette construction nous rapprochera du but final, elle risque de boucler sur elle-même ; de plus, s'il y a plusieurs conditionnelles dans le même cas, on risque d'être amené à essayer de construire un antécédent à l'intérieur de la construction d'un autre antécédent sans que cet emboîtement soit logiquement justifié.

Pour éviter ces deux difficultés, d'une part, on effectue un test de bouclage sur le nouveau but poursuivi et on accepte que la construction puisse échouer sans que cela ait de répercussion sur le processus courant, et d'autre part, on utilise un drapeau «ANT» qui indique qu'on est dans une phase de construction d'antécédent et permet d'éviter de se lancer dans une autre procédure de même type avant que la présence d'information nouvelle ne nous y invite.

Il ne faudrait pas conclure de ce qui précède que des constructions d'antécédent ne peuvent pas être emboîtées. L'exemple suivant illustre le fait que la construction de l'antécédent $p \wedge q$ n'est réalisable que si la construction de l'antécédent $p \supset r$ est déjà en cours, sans quoi la

proposition p qui y est introduite comme prémisses auxiliaires, ne serait pas disponible.



- Interdire l’enchâssement de constructions d’antécédent est excessif, il faut l’autoriser si une nouvelle information apparaît, ce qui est le cas dès qu’une prémisses auxiliaires est introduite.
 La solution réside dans l’utilisation éclairée du drapeau «ANT» : quand un processus parent lance un processus enfant, il lui transmet l’état du drapeau; quand un processus introduit une prémisses auxiliaires il baisse le drapeau.
- Après avoir levé le drapeau «ANT», on lance la procédure de construction de l’antécédent proprement dite : on introduit l’antécédent sur la pile des buts intermédiaires, on vérifie qu’il n’y a pas de boucle (comme dans le cas de l’introduction de la double négation), puis, on lance le processus de construction proprement dit.
- Ensuite, on baisse le drapeau «ANT», et si la construction a échoué on quitte la réduction, sinon on la poursuit en tenant compte de l’antécédent construit que l’on place dans la liste des lignes actives.

5.3. Introduction d’une conjonction

Il s’agit à partir des mêmes lignes actives de lancer deux processus, l’un prenant comme but intermédiaire le premier constituant de la conjonction et l’autre le second. Si un de ces processus échoue, le processus courant échoue également. Si tous les deux réussissent, on réactive les lignes initialement actives, on ajoute, à l’état inactif, les lignes introduites par les deux processus ainsi que la ligne constituée par la conjonction recherchée étiquetée par la règle d’introduction de la conjonction, et le processus courant se termine sur une réussite.

5.4. Introduction d'une conditionnelle

On ajoute à la liste des lignes actives l'antécédent de la conditionnelle à titre de prémisses auxiliaires, puis on lance un processus visant à construire le conséquent de la conditionnelle. Si le processus échoue, le processus courant échoue également. Sinon on réactive les lignes initialement actives, on désactive les lignes introduites à partir de la prémisses auxiliaires et on ajoute la ligne constituée par le but visé étiquetée par la règle d'introduction de la conditionnelle; le processus courant se termine sur une réussite.

5.5. Introduction d'une biconditionnelle

Il s'agit d'une procédure similaire à l'introduction de la conjonction à ceci près que les deux buts intermédiaires sont ici des conditionnelles et que la règle d'introduction étiquetant la dernière ligne est la règle d'introduction de la biconditionnelle.

5.6. Élimination d'une disjonction

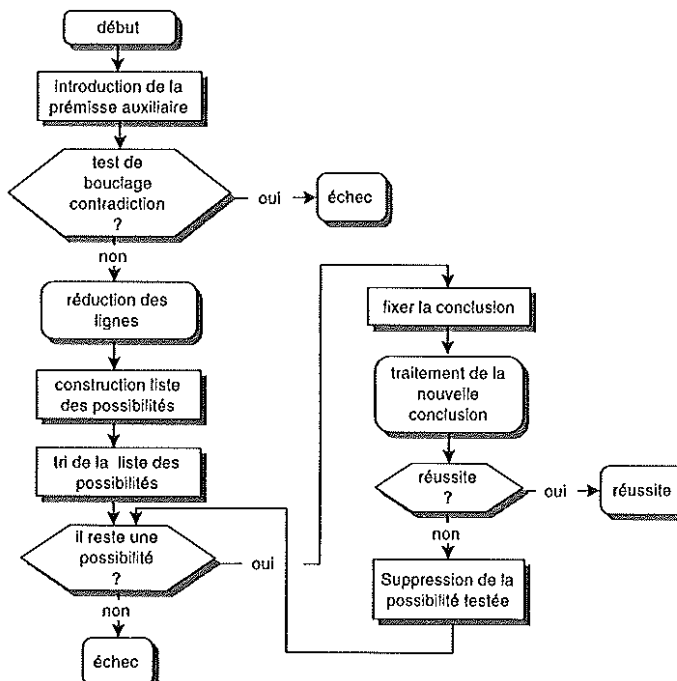
Bien qu'il ne s'agisse pas ici d'une règle analytique respectant la propriété de la sous-formule, à savoir que sa conclusion n'est pas contenue dans ses prémisses et donc qu'elle pourrait être quelconque et ouvrir un espace de recherche potentiellement infini, cela ne pose pas de problème dans le cadre de notre algorithme puisque l'élimination d'une ligne contenant une disjonction se fait toujours dans la perspective de la construction d'un but donné, et c'est bien entendu ce but qui sert de conclusion à la règle d'élimination.

Comme le laisse deviner la formulation de la règle d'élimination de la disjonction, la procédure est une combinaison des procédures d'introduction de la conjonction et de la conditionnelle. On va opérer deux processus poursuivant le même but que le processus courant et prenant comme prémisses auxiliaires, l'un le premier constituant de la disjonction et l'autre le second. Si un de ces processus échoue, le processus courant échoue également. Si tous les deux réussissent, on réactive les lignes initialement actives, on ajoute, à l'état inactif, les lignes introduites par les deux processus à partir de leur prémisses auxiliaires ainsi que la ligne constituée par le but recherché étiquetée par la règle d'élimination de la disjonction, et le processus courant se termine sur une réussite.

5.7. Introduction d'une négation

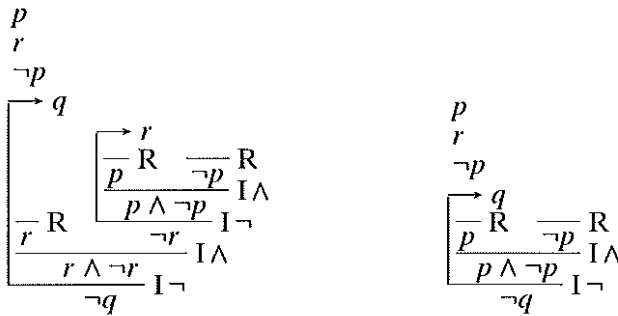
L'introduction de la négation est la partie difficile de la déduction naturelle. En effet, ici, on a réellement à construire une conclusion intermédiaire dont on ne sait rien sinon que c'est une contradiction. Mais il y a une infinité de contradictions et il est impératif de limiter l'espace de recherche sans être arbitraire sans quoi il n'y aura jamais, le cas échéant, de constat d'échec et l'algorithme ne se terminera pas, alors qu'on sait que le calcul des propositions est décidable. À la différence du cas précédent, la difficulté liée à l'absence de la propriété de la sous-formule ne peut pas être esquivée en sélectionnant une seule conclusion intermédiaire. Aussi est-on amené à définir un nombre fini de possibilités et à les essayer successivement jusqu'à un constat de réussite ou d'échec. Ces essais successifs introduisent la notion de retour en arrière et la nécessité de mémoriser la liste des contradictions possibles.

Afin d'exploiter au maximum les informations disponibles au moment où l'on fixe la liste des contradictions possibles, on repousse ce choix, que l'on ne fait qu'après avoir introduit la prémisse auxiliaire et l'avoir utilisée dans une phase de réduction des lignes.



Le diagramme ci-dessus explicite la procédure.

- Après avoir introduit parmi les lignes actives une nouvelle prémisses auxiliaire constituée par le constituant nié du but courant, on fait un premier test : il s'agit de vérifier que l'on n'est pas en train de boucler dans un emboîtement indéfini de recherche de contradictions, ou moins grave, que l'on n'est pas en train de faire le détour consistant à construire une contradiction non-pertinente dépendant d'une contradiction qui aurait dû être sélectionnée prioritairement. L'exemple de gauche illustre le détour issu du choix inadéquat d'une contradiction corrigé dans la version de droite.



Ce test de bouclage est similaire à celui rencontré plus haut : on vérifie qu'il n'y a pas déjà une contradiction dans la pile des buts. Si c'est le cas et si depuis l'introduction de ce but, il n'y a pas de nouvelle ligne active différente de celles présentes alors, on est dans un processus circulaire et on arrête sur un constat d'échec.

- On procède ensuite à une réduction des lignes, en suivant la procédure décrite plus haut.
- La construction de la liste des contradictions à essayer revient à faire la liste des propositions qui apparaissent soit positivement, soit négativement dans les contradictions : cette liste de propositions est calquée sur la liste des lignes actives. Un tel choix garantit qu'au moins toutes les contradictions pertinentes seront essayées, puisque toutes les phrases qui jouent un rôle dans l'état courant de la déduction sont sélectionnées. Et d'autre part, ce choix nous garantit qu'une des deux parties de la construction de la contradiction (qui est une conjonction) sera obtenue par réflexivité.

Dans l'exemple précédent, cette liste de propositions est la suivante : $p, r, \neg p, q$. Il est impossible de réduire cette liste au moyen de considérations purement internes à la déduction naturelle, mais par

contre il est possible de la trier de manière à essayer les possibilités les plus prometteuses d'abord.

- Le tri de la liste des possibilités consiste simplement privilégier une proposition dont la négation est déjà dans la liste, et ensuite à privilégier les propositions négatives. Le premier critère est évident, puisqu'il conduit à construire des contradictions dont les deux constituants sont déjà présents. Le second critère est moins efficace, il repose sur le fait que la liste des possibilités ne contient que des lignes actives, et donc des lignes déjà réduites : elle ne peut donc contenir que des propositions conditionnelles, des propositions négatives ou des propositions atomiques. Le seule raison pour privilégier les propositions négatives, qui sont les seules propositions moléculaires irréductibles, est qu'en procédant ainsi, dans une étape ultérieure, on sera amené à construire leur constituant non nié, et donc à exploiter la structure qu'il recèle.
- S'il ne reste pas de possibilité inexplorée, le processus se termine sur un échec.
- Sinon, on choisit la première possibilité inexplorée, on construit, à partir d'elle, la contradiction qui sert de but intermédiaire et on lance le processus de construction du nouveau but.
- Si le processus réussit, on réactive les lignes initialement actives, on désactive les lignes introduites à partir de la prémisse auxiliaire et on ajoute la ligne constituée par la contradiction étiquetée par la règle d'introduction de la négation ; le processus courant se termine sur une réussite.
- Si le processus échoue, on remet les lignes actives dans l'état qui précédait le processus de construction de la contradiction, on la supprime de la liste des possibilités, et on retourne à l'étape qui teste la liste.

5.8. Introduction d'une disjonction

L'introduction d'une disjonction est une opération qui peut se réaliser de trois manières différentes : la première consiste à construire le premier constituant, puis à introduire le second par la première règle d'introduction de la disjonction, la seconde fait de même avec le second constituant, et la troisième est une méthode indirecte. Cette dernière, illustrée par la preuve du tiers-exclus, s'impose quand on ne peut pas savoir *a priori* lequel des deux constituants est vrai ; il s'agit alors de les traiter solidairement dans une preuve par l'absurde.

Notre algorithme implémente cette dernière voie en testant, après l'exploration des deux premières voies, qui constitue la procédure d'introduction de la disjonction proprement dite, la réussite ou l'échec du processus. En cas d'échec, celui-ci ne se propage pas au processus courant, mais la troisième voie est testée dans la dernière phase de l'algorithme.

L'introduction d'une disjonction proprement dite consiste à introduire comme but intermédiaire le premier constituant de la disjonction et à lancer le processus de construction. Si le processus échoue, on revient à l'état antérieur des lignes actives, et on relance le processus avec le deuxième constituant. Si celui-ci échoue également, on revient à l'état antérieur des lignes actives et on continue le processus pour le but courant.

Si un des deux processus réussit, on réactive les lignes initialement actives, on désactive les lignes introduites par le processus ainsi que la ligne constituée par la disjonction recherchée étiquetée par la règle d'introduction de la disjonction, et le processus courant se termine sur une réussite.

Ceci conclut la présentation de l'algorithme de déduction naturelle.

6. Conclusions

Disposer d'un algorithme automatique de déduction naturelle est, bien évidemment, une chose intéressante. Le principal intérêt d'un tel outil est la construction aisée de n'importe quelle déduction, ce qui est sa raison d'être et sa destination première. Mais au-delà de cet aspect, il convient de remarquer que si la déduction naturelle est particulièrement importante par son apport à la théorie de la signification, où elle permet une meilleure compréhension des différents connecteurs logiques, un apport du même ordre est réalisé par l'existence d'un algorithme de déduction naturelle. En effet, à une élucidation des connecteurs par les règles de déduction naturelle, vient s'ajouter une plus grande mise en lumière de l'interaction dynamique des différentes règles, que l'algorithme spécifie en fixant à la fois les priorités hiérarchiques dans l'ordre d'utilisation des règles et les étapes nécessitant des retours en arrière. L'algorithme nous donne une connaissance plus profonde des déductions, qui repose non seulement sur les preuves écrites telles qu'elles apparaissent statiquement *a posteriori* mais également sur la dynamique de recherche qui a présidé à leur élaboration.

Une utilisation particulièrement intéressante de cet algorithme est l'enseignement des méthodes de preuve où la connaissance des règles est insuffisante et où l'expertise basée sur l'expérience et sur le *flair* se transmet difficilement. L'algorithme peut servir tant implicitement pour structurer l'enseignement de l'interaction des règles qu'explicitement comme méthode à suivre pour découvrir infailliblement les déductions. Personnellement, il me semble préférable de ne pas proposer immédiatement l'algorithme comme méthode à suivre, car l'intérêt de la déduction naturelle réside plus dans la compréhension des mécanismes de la signification et du raisonnement que dans la réalisation d'une performance technique.

Enfin, de la conception d'un algorithme à son implémentation informatique, il n'y a qu'un pas. Je tiens à la disposition de toute personne intéressée une version de l'algorithme que j'ai réalisée pour l'environnement *Openstep*.

Bibliographie

- BOLOTOV (A.), BOCHAROV (V.) and GORCHAKOV (A.) : 1996, « A proof search algorithm for natural deduction classical propositional calculus », *Logical Investigation*, (Russian Academy of Sciences, Moscow).
- GENTZEN (G.) : 1934, 1935, « Untersuchungen über das logische Schliessen », *Mathematische Zeitschrift*, 39, p. 176–210; 40, p. 405–431, (Traduction française et commentaires par F. Feys et J. Ladrière, *Recherches sur la déduction logique*, Paris, P.U.F, 1955).
- GILLET (É.) : 1997, « Belief: a doxastic relation to deduction », *The Role of Pragmatics in Contemporary Philosophy*, in P. Weingartner, G. Schurz, and G. Dorn (editors), Proceedings of the 20th International Wittgenstein Symposium, The Austrian Ludwig Wittgenstein Society (Kirchberg am Wechsel), vol. 1, p. 286–293.
- ROBINSON (J. A.) : 1965, « A machine-oriented logic based on the resolution principle », *Jour. Assoc. for Comput. Mach.*, p. 23–41.