Table des matières

1 Logique intuitionniste propositionnelle : approche par le calcul de séquents LJ

Il existe de nombreuses approches de la logique intuitionniste : on choisit ici celle par le calcul de séquents **LJ** introduit par Gentzen (?). Cela permet de se familiariser avec les calculs de séquents, avant de discuter

On s'intéresse à la partie propositionnelle de la logique intuitionniste : les formules sont construites à partir de la constante $\perp$, de variables propositionnelles et des connecteurs $\wedge$, $\vee$, $\rightarrow$. Une formule est *atomique* si elle est réduite à une variable propositionnelle ou $\perp$. La notation $\neg A$ signifie $A \rightarrow \perp$.

1.1 Définition du calcul LJ

Les éléments qui caractérisent généralement un calcul de séquents sont une définition de ses *séquents* et un ensemble de *règles*. Présentons ceux du calcul **LJ**.

Multiensembles et notations. On s'intéresse à des *multiensembles*, c'est-à-dire des collections où le nombre d'occurrences est pris en compte, mais non l'ordre des éléments. Cela permettra de ne pas avoir besoin de règles explicites d'échange. On utilise des lettres romaines (typiquement A, B, D, G) pour désigner les formules et des lettres grecques (Γ, Δ) pour les multiensembles de formules. La notation "A, Γ" représente le multiensemble obtenu à partir de Γ en ajoutant une occurrence de A. Lorsqu'il n'y a pas matière à confusion, on représente un multiensemble vide par un simple blanc.

Définition. *Un* **séquent** *de* **LJ** *est la donnée d'un multiensemble de formules* Γ *(les "hypothèses") et d'une formule* D *(la "conclusion") ; on écrit* $\Gamma \Rightarrow D$.

Les **règles** du calcul **LJ** sont données dans la figure 1. Pour une règle $\dfrac{prem_1 \quad ... \quad prem_p}{concl}\ (\mathcal{R})$, $\mathcal{R}$ est le nom de la règle, $prem_1$, ... , $prem_p$ sont les **prémisses**, et *concl* la **conclusion**. Les prémisses et la conclusion sont des séquents où A, B, D sont des formules quelconques et Γ un multiensemble de formules quelconques. La distinction entre règles *logiques* et *structurelles* sera expliquée plus loin.

Cette présentation est à peu près celle donnée par Dyckhoff dans [**?**]. Elle diffère de celle de Gentzen, mais elle en est suffisamment proche pour qu'on puisse quand même l'appeler le calcul **LJ**. On peut d'ailleurs facilement passer d'une définition à l'autre à l'aide des règles structurelles.

$$\frac{}{\bot, \Gamma \Rightarrow D}\ \bot L \qquad\qquad \frac{}{A, \Gamma \Rightarrow A}\ \mathrm{Id}$$

$$\frac{A, B, \Gamma \Rightarrow D}{A \wedge B, \Gamma \Rightarrow D}\ \wedge L \qquad\qquad \frac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \wedge B}\ \wedge R$$

$$\frac{A, \Gamma \Rightarrow D \quad B, \Gamma \Rightarrow D}{A \vee B, \Gamma \Rightarrow D}\ \vee L \qquad \frac{\Gamma \Rightarrow A}{\Gamma \Rightarrow A \vee B}\ \vee R_1 \quad \frac{\Gamma \Rightarrow B}{\Gamma \Rightarrow A \vee B}\ \vee R_2$$

$$\frac{\Gamma \Rightarrow A \quad B, \Gamma \Rightarrow D}{A \to B, \Gamma \Rightarrow D} \to L \qquad\qquad \frac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow A \to B} \to R$$

FIGURE 1 – Axiomes et règles logiques du calcul **LJ**

1.2 Prouvabilité

Une ***instance*** d'une règle $\mathcal{R}$ a la même forme que la règle : $\dfrac{\sigma_1 \quad ... \quad \sigma_p}{\sigma}\ (\mathcal{R})$, mais ici les σ_i et σ sont des séquents connus explicitement ; bien entendu il faut qu'il s'agisse de séquents qui correspondent à la forme donnée par la définition de la règle.

Une ***preuve*** (ou *arbre de preuve*) est un arbre dont les nœuds sont étiquetés par un séquent et une règle et ont la même arité que le nombre de prémisses de la règle, et tel que : pour tout nœud de séquent σ et de règle $\mathcal{R}$, si σ_1, ... , σ_p sont les séquents associés à chacun de ses fils respectivement, alors $\dfrac{\sigma_1 \quad \dots \quad \sigma_p}{\sigma}\,(\mathcal{R})$ est une instance de $\mathcal{R}$. Les feuilles d'un tel arbre sont les nœuds auxquels est associé un axiome.

Définition. *Un séquent σ est* **prouvable** *par le calcul* **LJ** *s'il existe un arbre de preuve tel que le séquent associé à la racine est σ. De manière équivalente, on peut définir l'ensemble des séquents prouvables comme le plus petit ensemble vérifiant : pour toute instance $\dfrac{\sigma_1 \quad \dots \quad \sigma_p}{\sigma}\,(\mathcal{R})$ d'une règle, si pour tout i, σ_i est prouvable, alors σ est prouvable (en particulier pour toute instance $\dfrac{}{\sigma}\,(\mathcal{A})$ d'un axiome $\mathcal{A}$, σ est prouvable).*

Ceci nous permet de définir la prouvabilité en logique intuitionniste.

Définition. *Une formule A est* **prouvable en logique intuitionniste** *si le séquent $\Rightarrow A$ est prouvable par le calcul* **LJ** *(on écrit $\Rightarrow A$ pour $\emptyset \Rightarrow A$).*

Interprétation d'un séquent. Les séquents, en plus d'être des structures pratiques à manipuler à l'aide de règles, ont en eux-mêmes une interprétation logique très simple grâce à la propriété suivante : un séquent $\Gamma \Rightarrow D$ est prouvable par le calcul **LJ** si, et seulement si, la formule $\left(\bigwedge_{G\in\Gamma} G\right) \to D$ est prouvable en logique intuitionniste.

1.3 Comparaison avec la logique classique (calcul LK)

La logique classique peut être définie à l'aide d'un calcul de séquents appelé **LK**, très similaire à **LJ** (en fait, c'est **LJ** qui a été dérivé de **LK** pour passer de la logique classique à la logique intuitionniste). Un séquent de **LK** comporte un autre multiensemble Δ de "conclusions" au lieu d'une unique conclusion D : on écrit $\Gamma \Rightarrow \Delta$. Un tel séquent a également une interprétation logique : il est prouvable par **LK** si, et seulement si, la formule $\left(\bigwedge_{G\in\Gamma} G\right) \to \left(\bigvee_{D\in\Delta} D\right)$ est vraie en logique classique.

Les règles sont modifiées en conséquence : par exemple $\dfrac{\Gamma \Rightarrow A, \Delta \quad \Gamma \Rightarrow B, \Delta}{\Gamma \Rightarrow A \wedge B, \Delta}\,\wedge R$ remplace $\dfrac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \wedge B}\,\wedge R$. Bien entendu, on ajoute aussi des règles structurelles agissant sur Δ. Mais surtout, on n'a plus qu'une règle pour le $\vee$ à droite : $\dfrac{\Gamma \Rightarrow A, B, \Delta}{\Gamma \Rightarrow A \vee B, \Delta}\,\vee R$. (Dans d'autres définitions, on garde deux règles distinctes, mais la règle que nous donnons ici peut être déduite de ces deux règles et de règles structurelles.)

C'est la possibilité d'avoir plusieurs formules dans la partie droite du séquent qui permet de prouver davantage de séquents dans **LK** que dans **LJ**. On comprend ainsi la différence entre le "ou" classique et le "ou" intuitionniste. En logique classique, prouver $A \vee B$, c'est prouver le séquent $\Rightarrow A, B$: les deux formules sont encore présentes. Un bon exemple est la preuve du principe du tiers exclu $A \vee \neg A$ (figure 2 ; on rappelle que $\neg A$ est une notation pour $A \to \bot$) : si on peut appliquer l'axiome Id à la formule A (ce qui nécessite deux occurrences distinctes de la formule, une de chaque côté), c'est bien parce qu'on a conservé les deux parties de la formule initiale. Tandis qu'en logique intuitionniste, pour prouver $A \vee B$ c'est-à-dire $\Rightarrow A \vee B$, les seules règles applicables sont $\vee R_1$ et $\vee R_2$: il faut donc prouver $\Rightarrow A$ ou prouver $\Rightarrow B$; une fois qu'on a choisi lequel on va prouver, on n'a plus accès à l'autre. Ainsi, on ne peut prouver $A \vee \neg A$, car ni $\Rightarrow A$ ni $\Rightarrow \neg A$ n'est prouvable.

$$\dfrac{\dfrac{\dfrac{}{A \Rightarrow A, \bot}\ \mathrm{Id}}{\Rightarrow A, (A \to \bot)}\ \to R}{\Rightarrow A \vee (A \to \bot)}\ \vee R$$

FIGURE 2 – Preuve dans **LK** de $A \vee \neg A$

2 Application du calcul de séquents à la recherche de preuve automatisée. Comparaison des différents systèmes

Un calcul de séquents est généralement défini par sa propre définition d'un séquent ainsi qu'un ensemble de règles. Il est souvent attaché à une logique : à partir d'une formule, on peut construire un séquent qui est prouvable si et seulement si la formule est "prouvable" ou "vraie" dans la logique en question (l'appellation dépend de la logique). Il existe plusieurs calculs de séquents pour la logique intuitionniste, sans parler des nombreuses autres logiques existantes.

Algorithme. Un calcul de séquents suggère naturellement un algorithme de recherche de preuve : pour essayer de prouver un séquent, on choisit une règle dont il peut être la conclusion et on essaie de prouver les prémisses correspondantes. Si elles sont toutes prouvables (notamment, s'il n'y en a pas : si le séquent est la conclusion d'un axiome), alors par définition le séquent initial est aussi prouvable. Sinon, on essaie une autre règle (sauf dans certains cas où on peut conclure grâce à la notion de règle ou prémisse inversible que nous verrons plus loin). Si on a essayé toutes les règles applicables au séquent sans succès (pour chacune, au moins une prémisse est non prouvable), on conclut que le séquent initial n'est pas prouvable.

Un tel algorithme est correct par construction et d'après la définition de la prouvabilité d'un séquent. En revanche, les causes possibles de non terminaison sont nombreuses. Assurer la terminaison est une des raisons qui rendent certaines propriétés sur les calculs de séquents très intéressantes.

Classification des règles. On distingue généralement deux sortes de règles. Les *règles logiques* remplacent une formule de la conclusion par une ou des formules plus simples. La formule remplacée, appelée *formule principale*, doit avoir une forme donnée en fonction de la règle. Les *règles structurelles* manipulent la structure du séquent en enlevant, dupliquant, déplaçant des formules dont on n'a pas besoin de connaître la forme. Elles dépendent du choix de structure du séquent : pour le calcul **LJ**, si on avait choisi de réprésenter Γ par une liste et non un multiensemble, on aurait eu besoin d'ajouter une règle d'échange $\dfrac{\Gamma_1, A, B, \Gamma_2 \Rightarrow D}{\Gamma_1, B, A, \Gamma_2 \Rightarrow D}\ (permut.\ L)$.

Coupure. La règle de coupure (figure 1, "cut") condamne l'algorithme à elle seule, puisqu'il peut y avoir une infinité de prémisses associées à une conclusion donnée. Heureusement, cette règle est souvent non nécessaire : de nombreux calculs de séquents possèdent la propriété d'élimination de la coupure, si bien qu'on peut faire comme si cette règle n'existait pas. C'est une propriété souvent difficile à démontrer, mais on ne s'y intéressera pas. Désormais, on présentera des calculs sans donner de règle de coupure.

Règles structurelles et coupure à éviter.

La règle de coupure (figure 1, “cut”) condamne l’algorithme à elle seule, puisqu’il peut y avoir une infinité de prémisses associées à une conclusion donnée. Pour commencer, on veut donc un calcul où le nombre d’instances ayant une conclusion donnée est toujours fini. Mais même ainsi, comme pour essayer de prouver un séquent, on rappelle l’algorithme sur d’autres séquents, il faut un argument soigneux de terminaison : par exemple, associer à chaque séquent une “taille” entière positive, telle que pour toute instance des règles, les “tailles” de toutes les prémisses sont strictement inférieures à celle de la conclusion. Comme on le verra, la propriété de la sous-formule est bien pratique pour ce point-là.

2.1 LJ : existence de cycles

nécessité de contraction ou réécriture de $\to L$ (preuve de $\neg\neg(A \vee \neg A)$)
LJT

3 Le calcul LSJ

L’article [1] définit un calcul de séquents **LSJ**. Une sémantique naturelle des séquents est définie à l’aide des modèles de Kripke, mais nous ne la présentons pas. En effet, ce qui nous intéresse est l’existence, pour toute formule, d’un séquent qui est prouvable dans le calcul **LSJ** si, et seulement si, la formule est prouvable en logique intuitionniste. Nous renvoyons à l’article pour les démonstrations, notamment celle de la complétude du calcul.

3.1 Les séquents

On s’intéresse à des *multiensembles*, c’est-à-dire des collections où le nombre d’occurrences est pris en compte, mais non l’ordre des éléments. Cela permettra de ne pas avoir besoin de règles explicites d’échange.

Un **séquent** est la donnée de trois multiensembles Θ, Γ et Δ de formules ; on écrit alors $\Theta \;;\; \Gamma \;\Rightarrow\; \Delta$.

Une définition d’un séquent **réfutable** est donnée dans l’article à l’aide des modèles de Kripke. Nous ne la détaillons pas ici, car ce qui nous intéresse surtout est la propriété suivante qui en découle, démontrée dans l’article. La définition de **prouvable** sera donnée plus tard car elle est liée aux règles du calcul, mais ceci illustre son intérêt.

Proposition 1. *Un séquent $\emptyset \;;\; \Gamma \;\Rightarrow\; \Delta$ est **réfutable** si, et seulement si, la formule $\bigwedge_{A\in\Gamma} A \to \bigvee_{B\in\Delta} B$ n’est pas valide en logique intuitionniste. Un séquent est **prouvable** dans **LSJ** si, et seulement si, il n’est pas réfutable.*

Corollaire 2. *Soit A une formule, elle est valide en logique intuitionniste si et seulement si le séquent $\emptyset;\emptyset \Rightarrow A$ est prouvable dans **LSJ**.*

Les multiensembles Γ et Δ, et leur signification dans la propriété 1 sont des éléments habituels en calcul des séquents. En revanche, Θ est propre à **LSJ**, et est difficile à interpréter car contrairement au cas où Θ est vide, un séquent avec Θ quelconque ne peut pas être représenté par une formule. On peut dire est que Θ contient des formules gardées en réserve, non visibles directement dans le séquent (une formule de Θ ne peut pas être *formule principale*), mais qui

$$\frac{}{\Theta\,;\bot,\Gamma\Rightarrow\Delta}\;\bot L \qquad \frac{}{\Theta\,;A,\Gamma\Rightarrow A,\Delta}\;\mathrm{Id}$$

$$\frac{\Theta\,;A,B,\Gamma\Rightarrow\Delta}{\Theta\,;A\wedge B,\Gamma\Rightarrow\Delta}\;\wedge L \qquad \frac{\Theta\,;\Gamma\Rightarrow A,\Delta \quad \Theta\,;\Gamma\Rightarrow B,\Delta}{\Theta\,;\Gamma\Rightarrow A\wedge B,\Delta}\;\wedge R$$

$$\frac{\Theta\,;A,\Gamma\Rightarrow\Delta \quad \Theta\,;B,\Gamma\Rightarrow\Delta}{\Theta\,;A\vee B,\Gamma\Rightarrow\Delta}\;\vee L \qquad \frac{\Theta\,;\Gamma\Rightarrow A,B,\Delta}{\Theta\,;\Gamma\Rightarrow A\vee B,\Delta}\;\vee R$$

$$\frac{\Theta\,;B,\Gamma\Rightarrow\Delta \quad B,\Theta\,;\Gamma\Rightarrow A,\Delta \quad B\,;\Theta,\Gamma\Rightarrow A}{\Theta\,;A\to B,\Gamma\Rightarrow\Delta}\to L$$

$$\frac{\Theta\,;A,\Gamma\Rightarrow B,\Delta \quad \emptyset\,;A,\Theta,\Gamma\Rightarrow B}{\Theta\,;\Gamma\Rightarrow A\to B,\Delta}\to R$$

FIGURE 3 – Les règles du calcul **LSJ**

peuvent être transférées dans Γ et ainsi devenir visibles. On verra que les seules règles qui agissent sur Θ sont celles qui concernent le connecteur $\to$.

Pour un séquent $\Theta\ ;\ \Gamma\ \Rightarrow\ \Delta$, on appellera les formules de Γ les **formules de gauche**, celles de Δ les **formules de droite**, et celles de Θ les **formules de réserve** du séquent (appellations non conventionnelles).

3.2 Les règles

Les règles du calcul **LSJ** sont données dans la figure 3. La notation A,Γ représente le multiensemble obtenu à partir de Γ en ajoutant une occurrence de A. Pour une règle $\dfrac{prem_1 \quad ... \quad prem_p}{concl}\ (\mathcal{R})$, $\mathcal{R}$ est le nom de la règle, $prem_1$, ... , $prem_p$ sont les (resp. première, ... , p-ième) **prémisses**, et *concl* la **conclusion**. Les **axiomes** sont les règles sans prémisse. Pour toutes les autres règles, une unique formule apparaît de manière explicite dans la conclusion : c'est la **formule principale**. Les règles dites de gauche, ou d'introduction à gauche, contenant un L dans leur nom, sont celles où la formule principale se trouve à gauche dans la conclusion, de même pour les règles de droite.

Une **instance** d'une règle $\mathcal{R}$ a la même forme que la règle : $\dfrac{\sigma_1 \quad ... \quad \sigma_p}{\sigma}\ (\mathcal{R})$, mais ici les σ_i et σ sont des séquents connus explicitement ; bien entendu il faut qu'il s'agisse de séquents qui ont bien la forme donnée par la définition de la règle. Par exemple $\dfrac{\Theta\,;A,B,\Gamma\Rightarrow\Delta}{\Theta\,;A\wedge B,\Gamma\Rightarrow\Delta}\ \wedge L$ devient une instance de la règle $\wedge L$ (qui a la même écriture que la règle) lorsqu'on connaît les formules A et B et toutes les formules de Θ, Γ, Δ.

Une **preuve** est un arbre dont les nœuds sont étiquetés par un séquent et une règle et ont la même arité que le nombre de prémisses de la règle, et tel que : pour tout nœud de séquent σ et règle $\mathcal{R}$, si σ_1, ... , σ_p sont les séquents associés à chacun de ses fils respectivement, alors $\dfrac{\sigma_1 \quad ... \quad \sigma_p}{\sigma}\ (\mathcal{R})$ est une instance de $\mathcal{R}$. Les feuilles d'un tel arbre sont les nœuds auxquels est associé un axiome.

Un séquent est **prouvable** s'il existe une preuve à la racine de laquelle il est associé.

De manière équivalente, on peut définir l'ensemble des formules prouvables comme le plus petit ensemble vérifiant : pour toute instance $\dfrac{\sigma_1 \quad ... \quad \sigma_p}{\sigma}\ (\mathcal{R})$ d'une règle de **LSJ**, si pour tout i, σ_i est prouvable, alors σ est prouvable (en particulier pour toute instance $\dfrac{}{\sigma}\,(\mathcal{A})$ d'un axiome $\mathcal{A}$, σ est prouvable).

3.3 Conditions de non-prouvabilité

Pour montrer qu'un séquent est prouvable, il suffit d'en exhiber une preuve. Comment montrer le contraire ? D'après la définition précédente, un séquent n'est pas prouvable s'il n'existe aucune instance de règle $\dfrac{\sigma_1 \quad ... \quad \sigma_p}{\sigma}\,(\mathcal{R})$ telle que tous les σ_i sont prouvables. Or les σ_i ne dépendent que de σ, $\mathcal{R}$ et du choix de la formule principale : il est donc possible de tester toutes les instances possibles. Cela fournit un premier algorithme de recherche de preuve : récursivement, pour chercher si un séquent σ est prouvable, on considère toutes les instances de règles dont σ est la conclusion et pour chacune on détermine récursivement si chaque prémisse est prouvable. Si on trouve une instance telle que toutes les prémisses sont prouvables, alors σ est prouvable (et on obtient une preuve de σ si on connaît une preuve de chacune de ces prémisses), sinon σ n'est pas prouvable. Cet algorithme est très long. En fait, c'est à peu près ce qu'on se retrouve à faire dans les cas extrêmement défavorables. Mais heureusement, on a un procédé bien plus économe en moyenne grâce à la notion de règle ou prémisse inversible.

Une prémisse $prem_i$ d'une règle $\dfrac{prem_1 \quad ... \quad prem_p}{concl}\,(\mathcal{R})$ (aussi appelée i-ème prémisse de $\mathcal{R}$) est **inversible** si on a : si $prem_i$ est non prouvable, alors $concl$ est non prouvable. Une règle est **inversible** si toutes ses prémisses sont inversibles.

On admet, une démonstration se trouvant dans l'article [1] :
- les règles $\wedge L$, $\wedge R$, $\vee L$ et $\vee R$ sont inversibles ;
- les deux premières prémisses de $\rightarrow L$ et la première prémisse de $\rightarrow R$ sont inversibles ;
- la troisième prémisse de $\rightarrow L$ et la deuxième prémisse de $\rightarrow R$ ne sont pas inversibles.

3.4 Algorithme

On en déduit le procédé suivant pour essayer d'appliquer une règle à un séquent avec un formule principale donnée : on essaie de prouver les prémisses inversibles, puis l'éventuelle prémisse non inversible (dans **LSJ** il y en a au plus une). Dès qu'on trouve qu'une prémisse inversible est non prouvable, on s'arrête : le séquent initial n'est pas prouvable non plus. Si toutes les prémisses sont prouvables, le séquent initial est également prouvable. Dans le dernier cas (seule la prémisse non inversible est non prouvable), on essaie une application de règle avec une autre formule principale.

Il ne reste plus qu'à décider dans quel ordre les formules qui peuvent l'être sont choisies comme formule principale pour essayer d'appliquer une règle. On choisit de traiter en premier les règles inversibles, car on sait alors qu'il n'y aura pas besoin d'essayer d'autre application de règle sur le même séquent. Parmi celles-ci, on privilégie celles qui n'ont qu'une prémisse ($\wedge L$ et $\vee R$) sur les autres, qui en ont deux ($\vee L$ et $\wedge R$).

3.5 ?

On voit immédiatement que l'algorithme nécessite de pouvoir déduire d'un séquent, d'une règle et d'une formule principale contenue dans le séquent et sur laquelle la règle peut agir, les séquents correspondant aux différentes prémisses. Ce n'est pas difficile : pour les axiomes il n'y a rien à faire ; pour les autres règles, la formule principale H étant de la forme

fonction estProuvable (σ)
  soit $\sigma = \Theta; \Gamma \Rightarrow \Delta$
  si ($\bot \in \Gamma$) **alors retourner** *vrai*
  si ($\Gamma \cap \Delta \neq \emptyset$) **alors retourner** *vrai*
  si (Γ et Δ ne contiennent que des formules *atomiques*) **alors retourner** *faux*
  si (il existe $A \wedge B \in \Gamma$) **alors** {
    sélectionner $H = A \wedge B$ **dans** Γ
    retourner estProuvable($prem(\wedge L, \sigma, H)$)
  }
  si (il existe $A \vee B \in \Delta$) **alors** {
    sélectionner $H = A \vee B$ **dans** Δ
    retourner estProuvable($prem(\vee R, \sigma, H)$)
  }
  ...

FIGURE 4 – Algorithme

A 'connecteur' B, il suffit d'enlever H du séquent et, selon le connecteur et le côté où se trouvait H, d'ajouter A ou B à Θ, Γ, Δ ou nulle part.

Mais ce n'est pas tout. Lorsqu'on essaie d'appliquer une règle $\dfrac{\sigma_1 \quad \sigma_2}{\sigma}$ au séquent σ, on lance une recherche de preuve sur σ_1 qu'on a obtenu comme décrit ci-dessus. Si on obtient que σ_1 est prouvable, on lance alors la recherche de preuve sur σ_2. On doit donc déterminer σ_2. On a vu qu'on sait le faire à partir de σ. Une solution consiste donc à retenir σ pendant qu'on effectue la recherche de preuve sur σ_1, mais cela peut être coûteux en mémoire. Une autre solution, que nous avons privilégiée, consiste à être capable de retrouver σ à partir de σ_1 ainsi que de la formule principale, de la règle et du numéro de la prémisse (ici 1). On a dans ce cas besoin de pouvoir retrouver la conclusion à partir de n'importe laquelle des prémisses, pas seulement par exemple de la première prémisse pour une règle qui n'en a que deux. En effet, utiliser σ_1 pour retrouver σ suppose qu'à la fin de la recherche de preuve pour σ_1, on connaît σ_1. Or, l'idée ici est de n'avoir vraiment qu'un seul séquent en mémoire à tout moment. Ainsi, à la fin de la recherche de preuve pour σ, on doit connaître σ, donc on doit aussi pouvoir déduire σ de σ_2 en connaissant la formule principale et le fait qu'on est en train de s'intéresser à la deuxième prémisse.

En résumé, on aimerait (bien que ce ne soit pas nécessaire) que toutes les règles soient **locales**, avec la définition suivante.

Définition 3. *Une règle est* ***locale*** *si pour toute instance* $\dfrac{\sigma_1 \quad \dots \quad \sigma_p}{\sigma}$ *de cette règle et pour tout i entre 1 et p, on peut déduire σ à partir de σ_i et de la formule principale et de i.*

On remarque que $\wedge L$, $\wedge R$, $\vee L$ et $\vee R$ sont locales. Les axiomes sont également locaux, la définition n'ayant pas grand intérêt pour eux. En revanche, les règles $\rightarrow L$ et $\rightarrow R$ ne sont pas locales : pour chacune, les formules représentées par Δ dans la conclusion n'apparaissent nulle part dans la dernière prémisse, il n'est donc pas possible de retrouver la conclusion en connaissant uniquement cette prémisse, la formule principale et le numéro de la prémisse, puisqu'il n'y a aucun moyen d'en déduire ce qui se trouve dans Δ.

C'est pour cette raison qu'on introduit le calcul $\mathbf{LSJ'}$, dans lequel toutes les règles sont locales.

4 Le calcul $\mathbf{LSJ'}$

Le calcul $\mathbf{LSJ'}$ est très proche du calcul **LSJ** : chaque règle de $\mathbf{LSJ'}$ correspond à une règle de **LSJ**, et des arbres de preuve dans les deux systèmes pour la même formule sont fortement liés. Mais contrairement à **LSJ**, les règles de $\mathbf{LSJ'}$ sont toutes locales. Pour cela, les séquents de $\mathbf{LSJ'}$ représentent chacun un séquent de **LSJ**, avec un peu plus d'informations : celles qui sont parfois nécessaire pour retrouver la conclusion à partir d'une prémisse. Cette représentation est exhaustive et correcte. On montre en effet qu'il existe une surjection de l'ensemble des séquents de $\mathbf{LSJ'}$ dans l'ensemble des séquents de **LSJ**, telle qu'un séquent de $\mathbf{LSJ'}$ est prouvable dans $\mathbf{LSJ'}$ si, et seulement si, son image est prouvable dans **LSJ**.

4.1 Formalisme de $\mathbf{LSJ'}$

Un séquent de $\mathbf{LSJ'}$ est la donnée de deux multiensembles Γ et Δ de couples $entier : formule$, et d'un entier naturel n, tels que tous les entiers présents dans Γ sont $\leq n+1$ et tous ceux présents dans Δ sont $\leq n$; on écrit $\Gamma \Rightarrow_n \Delta$.

Les règles du calcul $\mathbf{LSJ'}$ sont décrite dans la figure 5. Chacune correspond à une règle de **LSJ**.

n et parfois i désignent toujours des entiers naturels, avec $i \leq n$

$$\frac{}{i : \bot, \Gamma \Rightarrow_n \Delta}\ \bot L' \qquad \frac{}{i : A, \Gamma \Rightarrow_n n : A, \Delta}\ \mathrm{Id}'$$

$$\frac{i : A, i : B, \Gamma \Rightarrow_n \Delta}{i : A \wedge B, \Gamma \Rightarrow_n \Delta}\ \wedge L' \qquad \frac{\Gamma \Rightarrow_n n : A, \Delta \quad \Gamma \Rightarrow_n n : B, \Delta}{\Gamma \Rightarrow_n n : A \wedge B, \Delta}\ \wedge R'$$

$$\frac{i : A, \Gamma \Rightarrow_n \Delta \quad i : B, \Gamma \Rightarrow_n \Delta}{i : A \vee B, \Gamma \Rightarrow_n \Delta}\ \vee L' \qquad \frac{\Gamma \Rightarrow_n n : A, n : B, \Delta}{\Gamma \Rightarrow_n n : A \vee B, \Delta}\ \vee R'$$

$$\frac{i : B, \Gamma \Rightarrow_n \Delta \quad n+1 : B, \Gamma \Rightarrow_n n : A, \Delta \quad n+2 : B, \Gamma \Rightarrow_{n+1} n+1 : A, \Delta}{i : A \to B, \Gamma \Rightarrow_n \Delta} \to L'$$

$$\frac{0 : A, \Gamma \Rightarrow_n n : B, \Delta \quad 0 : A, \Gamma \Rightarrow_{n+1} n+1 : B, \Delta}{\Gamma \Rightarrow_n n : A \to B, \Delta} \to R'$$

FIGURE 5 – Les règles du calcul $\mathbf{LSJ'}$

4.2 Équivalence avec LSJ

On note $\mathfrak{S}$ l'ensemble des séquents de **LSJ**, et $\mathfrak{S}'$ l'ensemble des séquents de $\mathbf{LSJ'}$.

Soit $\sigma \in \mathfrak{S}$, on note $\vdash \sigma$ si σ est prouvable dans **LSJ** ; soit $\sigma' \in \mathfrak{S}'$, on note $\vdash' \sigma'$ si σ' est prouvable dans $\mathbf{LSJ'}$.

Soit M un multiensemble de couples $entier : formule$, l'entier d'un couple étant appelé son indice. On note M_k le multiensemble obtenu à partir de M en ne gardant que les couples d'indice k, et $M_{\leq k}$ celui obtenu en ne gardant que les couples d'indice inférieur à k. On note $\mathsf{forget}(M)$ le multiensemble de formules obtenu en oubliant l'indice et ne gardant que la formule de chaque couple de M.

On définit l'application Φ de $\mathfrak{S}'$ dans $\mathfrak{S}$, qui à $\Gamma' \Rightarrow_n \Delta'$ associe $\Theta\,;\Gamma \Rightarrow \Delta$

où : $\begin{cases} \Theta = \mathsf{forget}(\Gamma'_{n+1}) \\ \Gamma = \mathsf{forget}(\Gamma'_{\leq n}) \\ \Delta = \mathsf{forget}(\Delta'_n) \end{cases}$.

C'est une application surjective : en effet tout séquent $\Theta\,;\Gamma \Rightarrow \Delta$ de **LSJ** a au moins pour antécédent le séquent $\Gamma' \Rightarrow_0 \Delta'$, où Γ' est l'union de $0:\Gamma$ (le multiensemble de couples obtenu à partir de Γ en remplaçant chaque occurrence d'une formule A par une occurrence du couple $0:A$) avec $1:\Theta$, et où $\Delta' = 0:\Delta$.

Soit $\mathcal{R}$ une règle de **LSJ**. On note $\mathcal{R}'$ la règle de **LSJ′** qui lui correspond. On écrit $\dfrac{\sigma_1 \quad \ldots \quad \sigma_p}{\sigma}\,(\mathcal{R})$ et $\dfrac{\sigma'_1 \quad \ldots \quad \sigma'_p}{\sigma'}\,(\mathcal{R}')$ des instances de ces règles.

Lemme 4. *Soit $\sigma \in \mathfrak{S}$ et $\sigma' \in \mathfrak{S}'$ tels que $\sigma = \Phi(\sigma')$ et soit $\mathcal{R}$ une règle de* **LSJ**.

1) Si $\dfrac{\sigma_1 \quad \ldots \quad \sigma_p}{\sigma}\,(\mathcal{R})$ alors il existe $\sigma'_1, \ldots, \sigma'_p$ tels que pour tout k, $\sigma_k = \Phi(\sigma'_k)$, et $\dfrac{\sigma'_1 \quad \ldots \quad \sigma'_p}{\sigma'}\,(\mathcal{R}')$.

2) Si $\dfrac{\sigma'_1 \quad \ldots \quad \sigma'_p}{\sigma'}\,(\mathcal{R}')$, posons pour tout k, $\sigma_k = \Phi(\sigma'_k)$, alors $\dfrac{\sigma_1 \quad \ldots \quad \sigma_p}{\sigma}\,(\mathcal{R})$.
Pour un axiome $\mathcal{A}$, cela signifie simplement : $\dfrac{}{\sigma}\,(\mathcal{A})$ si et seulement si $\dfrac{}{\sigma'}\,(\mathcal{A}')$.

Preuve
On le montre pour chaque règle ; c'est une conséquence assez directe de la définition de Φ. Faisons-le par exemple pour Id, $\wedge R$ et $\to L$. À chaque fois, on se donne $\sigma = \Theta\,;\Gamma \Rightarrow \Delta$ et $\sigma' = \Gamma' \Rightarrow_n \Delta' \in \mathfrak{S}'$ tels que $\sigma = \Phi(\sigma')$.

Id : On a $\dfrac{}{\sigma}\,(\mathrm{Id})$ si et seulement s'il existe une formule A appartenant à la fois à Γ et Δ, ce qui équivaut, par définition de Φ, à : il existe A et $i \leq n$ tels que $n:A \in \Delta'$ et $i:A \in \Gamma'$, c'est-à-dire $\dfrac{}{\sigma'}\,(\mathrm{Id}')$.

$\wedge R$:

1) Si $\dfrac{\sigma_1 \quad \sigma_2}{\sigma}\,(\wedge R)$ alors il existe des formules A et B et un multiensemble $\widetilde{\Delta}$ tels que $\Delta = A \wedge B, \widetilde{\Delta}$ et $\sigma_1 = \Theta\,;\Gamma \Rightarrow A, \widetilde{\Delta}$ et $\sigma_2 = \Theta\,;\Gamma \Rightarrow B, \widetilde{\Delta}$. Posons $\widetilde{\Delta}' = \Delta' - n:A \wedge B$ le multiensemble obtenu en retirant une seule occurrence de $n:A \wedge B$ à Δ' (qui contient cet élément parce que Δ contient $A \wedge B$ et par définition de Φ), et $\sigma'_1 = \Gamma' \Rightarrow_n n:A, \widetilde{\Delta}'$ et $\sigma'_2 = \Gamma' \Rightarrow_n n:B, \widetilde{\Delta}'$. Alors on a bien $\sigma_1 = \Phi(\sigma'_1)$ et $\sigma_2 = \Phi(\sigma'_2)$ (en remarquant que $\widetilde{\Delta} = \mathsf{forget}(\widetilde{\Delta}'_n)$), et $\dfrac{\sigma'_1 \quad \sigma'_2}{\sigma'}\,(\wedge R')$ (en remarquant que $\sigma' = \Gamma' \Rightarrow_n n:A \wedge B, \widetilde{\Delta}'$).

2) Si $\dfrac{\sigma'_1 \quad \sigma'_2}{\sigma'}\,(\wedge R')$ alors il existe A, B et $\widetilde{\Delta}'$ tels que $\Delta' = n:A \wedge B, \widetilde{\Delta}'$ et $\sigma'_1 = \Gamma' \Rightarrow_n n:A, \widetilde{\Delta}'$ et $\sigma'_2 = \Gamma' \Rightarrow_n n:B, \widetilde{\Delta}'$; on pose $\widetilde{\Delta} = \Delta - A \wedge B$ le multiensemble obtenu en retirant une seule occurrence de $A \wedge B$ à Δ, et $\sigma_1 = \Phi(\sigma'_1)$ et $\sigma_2 = \Phi(\sigma'_2)$; on obtient $\sigma = \Theta\,;\Gamma \Rightarrow A \wedge B, \widetilde{\Delta}$ et $\sigma_1 = \Theta\,;\Gamma \Rightarrow A, \widetilde{\Delta}$ et $\sigma_2 = \Theta\,;\Gamma \Rightarrow B, \widetilde{\Delta}$ d'où $\dfrac{\sigma_1 \quad \sigma_2}{\sigma}\,(\wedge R)$.

$\to L$:

1) Si $\dfrac{\sigma_1 \quad \sigma_2 \quad \sigma_3}{\sigma}\,(\to L)$ alors il existe A, B et $\widetilde{\Gamma}$ tels que $\Gamma = A \to B, \widetilde{\Gamma}$ et $\sigma_1 = \Theta\,; B, \widetilde{\Gamma} \Rightarrow \Delta$ et $\sigma_2 = B, \Theta\,; \widetilde{\Gamma} \Rightarrow A, \Delta$ et $\sigma_3 = B\,; \Theta, \widetilde{\Gamma} \Rightarrow A$; et il existe $i \leq n$ tel que $i : A \to B \in \Gamma'$; on pose $\widetilde{\Gamma}' = \Gamma' - i : A \to B$ (on retire une seule occurrence de $i : A \to B$ de Γ') et $\sigma'_1 = i : B, \widetilde{\Gamma}' \Rightarrow_n \Delta'$ et $\sigma'_2 = n{+}1 : B, \widetilde{\Gamma}' \Rightarrow_n n : A, \Delta'$ et $\sigma'_3 = n{+}2 : B, \widetilde{\Gamma}' \Rightarrow_{n+1} n{+}1 : A, \Delta'$ et on vérifie que cela convient.

2) Si $\dfrac{\sigma'_1 \quad \sigma'_2 \quad \sigma'_3}{\sigma'}\,(\to L')$ alors il existe i, A, B et $\widetilde{\Gamma}'$ tels que $\Gamma' = i : A \to B, \widetilde{\Gamma}'$ et σ'_1, σ'_2 et σ'_3 ont la forme donnée ci-dessus ; on pose $\widetilde{\Gamma} = \Gamma - A \to B$ (on retire une seule occurrence de $A \to B$ de Γ), alors les images σ_1, σ_2 et σ_3 par Φ de σ'_1, σ'_2 et σ'_3 respectivement s'écrivent comme ci-dessus et donc $\dfrac{\sigma_1 \quad \sigma_2 \quad \sigma_3}{\sigma}\,(\to L)$.

∎

Théorème 5. *Soit $\sigma \in \mathfrak{S}$ et $\sigma' \in \mathfrak{S}'$ tels que $\sigma = \Phi(\sigma')$, alors $\vdash \sigma$ si et seulement si $\vdash' \sigma'$.*

Preuve
Par récurrence sur la *taille* de $\sigma \in \mathfrak{S}$, c'est-à-dire la somme des tailles des formules des trois multiensembles apparaissant dans σ.

On initialise pour tout $\sigma = \Theta\,; \Gamma \Rightarrow \Delta$ tel que toutes les formules dans Γ et dans Δ sont atomiques : soit $\sigma' = \Gamma' \Rightarrow_n \Delta' \in \mathfrak{S}'$ tel que $\sigma = \Phi(\sigma')$. Alors toutes les formules associées à un $i \leq n$ dans Γ' et toutes les formules associées à n dans Δ' sont aussi atomiques. En étudiant la forme des conclusions des règles non axiomatiques de **LSJ** comme de **LSJ′**, on remarque que si σ (resp. σ') est la conclusion d'une règle de **LSJ** (resp. **LSJ′**), alors la règle est un axiome. L'initialisation est donc un cas particulier de ce qui suit avec $p = 0$ (ce qui entraîne qu'on n'utilise en fait pas l'hypothèse de récurrence).

Soit $\sigma = \Theta\,; \Gamma \Rightarrow \Delta \in \mathfrak{S}$. Soit $\sigma' = \Gamma' \Rightarrow_n \Delta' \in \mathfrak{S}'$ tel que $\sigma = \Phi(\sigma')$.

On suppose $\vdash \sigma$. Alors il existe une règle $\mathcal{R}$ de **LSJ** et $\sigma_1, \ldots, \sigma_p \in \mathfrak{S}$ (avec éventuellement p nul) tels que $\vdash \sigma_k$ pour tout k et $\dfrac{\sigma_1 \quad \ldots \quad \sigma_p}{\sigma}\,(\mathcal{R})$. D'après le lemme, il existe $\sigma'_1, \ldots, \sigma'_p \in \mathfrak{S}'$ tels que $\sigma_k = \Phi(\sigma'_k)$ pour tout k et $\dfrac{\sigma'_1 \quad \ldots \quad \sigma'_p}{\sigma'}\,(\mathcal{R}')$. Pour tout k, on applique l'hypothèse de récurrence à σ_k qui a une *taille* strictement inférieure à celle de σ, et on obtient $\vdash' \sigma'_k$. On en déduit $\vdash' \sigma'$.

On suppose $\vdash' \sigma'$. Alors il existe une règle $\mathcal{R}'$ de **LSJ′** et $\sigma'_1, \ldots, \sigma'_p \in \mathfrak{S}'$ tels que $\vdash' \sigma'_k$ pour tout k et $\dfrac{\sigma'_1 \quad \ldots \quad \sigma'_p}{\sigma'}\,(\mathcal{R}')$. On pose $\sigma_k = \Phi(\sigma'_k)$ pour tout k. D'après le lemme on a $\dfrac{\sigma_1 \quad \ldots \quad \sigma_p}{\sigma}\,(\mathcal{R})$, en particulier on peut appliquer l'hypothèse de récurrence aux σ_k donc $\vdash \sigma_k$ pour tout k, d'où $\vdash \sigma$.

∎

5 Efficacité de LSJ

L'étude qui suit porte sur le calcul **LSJ**. En effet, **LSJ′** hérite de toutes les propriétés intéressantes de **LSJ**, en apportant une localité des règles qui facilite une implémentation économe en mémoire.

```
((a & b) & (non (c) | (a & b)))
  sf   classe   description
   1      1      Var a
   2      2      Var b
   3      3      1 & 2
   4      4      Var c
   5      0      Faux
   6      5      4 -> 5
   7      1      Var a
   8      2      Var b
   9      3      7 & 8
  10      6      6 | 9
  11      7      3 & 10
```

FIGURE 6 – Indexation de la formule $(a \wedge b) \wedge (\neg c \vee (a \wedge b))$

5.1 Propriété de la sous-formule et indexation

B est une **sous-formule** de A si $B = A$ ou si A est de la forme A_1‘connecteur’A_2 et (B est une sous-formule de A_1 ou B est une sous-formules de A_2). Un calcul de séquents vérifie la **propriété de la sous-formule** si tout séquent prouvable σ admet une preuve telle que toute formule apparaissant dans (un séquent de) la preuve est une sous-formule d'une formule de σ. En particulier, le séquent $\Rightarrow A$ a une preuve dans laquelle toute formule est une sous-formule de A.

Le calcul **LSJ**vérifie la propriété de la sous-formule : on le constate aisément en observant chaque règle.

La propriété de la sous-formule est très recherchée en calcul des séquents. D'une part, elle donne une borne sur les formules qu'il faudra manipuler au cours d'une recherche de preuve, qui sont évidemment toutes plus petites que la formule qu'on essaie de prouver. Mais surtout, elle permet de connaître à l'avance la liste exhaustive des formules qu'on pourra rencontrer. On peut donc à l'avance les numéroter : ainsi, les formules d'un séquents sont simplement représentées par un entier. Il faut quelques informations sur ces numéros : par exemple pour une formule $A \wedge B$, il faut savoir qu'il s'agit d'un "et", et pouvoir déterminer A et B. Il faut aussi pouvoir reconnaître quand l'axiome Id (une même formule apparaît des deux côtés du séquent) s'applique : pour cela on associe à chaque formule une classe, qui correspond à une classe d'équivalence de la relation d'égalité structurelle. La taille de toutes ces informations réunies est linéaire en la taille de la formule de départ (c'est-à-dire le nombre de nœuds de l'arbre qui la représente, qui est aussi le nombre de sous-formules avec multiplicité). Un exemple est donné par la figure 6.

La propriété de la sous-formule est encore plus intéressante dans le cadre de la recherche de preuve compilée : connaître à l'avance les formules qui apparaîtront permet d'écrire pour chacune des fonctions agissant sur le séquent, au lieu de les calculer au cours de la recherche de preuve (voir ? ?).

6 Implémentation

6.1 Indexation

cf "Propriété de la sous-formule et indexation" + explication sur les classes (et priorités) et les champs axiomes du séquent

6.2 Structure de données pour le séquent

Au cours de l'algorithme, on manipule un "séquent", censé représenter un séquent du calcul $\mathbf{LSJ'}$, contenant les informations suivantes :
- des informations de taille constante : l'indice n du séquent, et des booléens *id* et $fauxL$, indiquant si les axiomes de même nom sont applicables au séquent ;
- les couples *indice* : *formule* contenus dans les champs Γ et Δ du séquent.

Comme nous l'avons expliqué en introduisant le calcul $\mathbf{LSJ'}$, le but de celui-ci est de pouvoir effectuer la recherche de preuve en ne gardant à chaque instant qu'un seul séquent en mémoire.

Intéressons-nous maintenant à la complexité temporelle.

Celle-ci dépend du nombre de règles qu'on essaie d'appliquer, c'est-à-dire le nombre d'appels récursifs à la fonction *prouvable* (?) dont le pseudo-code est donné en figure ? page ?. Ce nombre dépend de la taille de la formule et des connecteurs présents dedans (et selon l'ordre dans lequel on choisit les formules principales cela peut beaucoup varier pour une même formule, mais on s'intéresse à la complexité dans le pire cas). Mais il ne dépend pas de notre choix d'implémentation (sauf pour l'ordre des "implique", mais encore une fois pas si on regarde le pire cas).

Références

[1] M. Ferrari, C. Fiorentini, and G. Fiorino, "Contraction-Free Linear Depth Sequent Calculi for Intuitionistic Propositional Logic with the Subformula Property and Minimal Depth Counter-Models," *Journal of Automated Reasoning*, vol. 51, no. 2, pp. 129–149, 2013.