Dans ce TP, on étudie un système déductif, appelé calcul des séquents, proposé par Gentzen pour améliorer celui de la déduction naturelle. On y manipule une notion généralisée de séquents.

Définition

On appelle **séquent** un couple $\Gamma \vdash \Delta$ où Γ et Δ sont des ensembles de formules propositionnelles.

L'interprétation d'un séquent $\Gamma \vdash \Delta$ prouvable est que si on suppose vraies **toutes** les formules de Γ, alors **au moins une** formule de Δ est vraie. La notion de règle d'inférence est la même qu'en déduction naturelle.

Définition

Soient Γ et Δ deux ensembles de formules propositionnelles. On dit que Γ **implique sémantiquement** Δ, noté $\Gamma \vDash \Delta$, si et seulement si $\bigcap\limits_{A\in\Gamma} \mathcal{M}(A) \subseteq \bigcup\limits_{B\in\Delta} \mathcal{M}(B)$. On dit qu'un séquent $\Gamma \vdash \Delta$ est **valide** si $\Gamma \vDash \Delta$.

On rappelle qu'une itération vide d'un opérateur est égale à l'élément neutre de cet opérateur.

Contrairement à la déduction naturelle, il n'y a pas des règles d'introduction et d'élimination des connecteurs, mais plutôt des règles gauches et des règles droites, selon le membre où se trouve le connecteur logique. Les règles du calcul des séquents sont les suivantes :

Règles gauche et droite

Opérateur	Règle gauche	Règle droite
Implication $\to$	$\dfrac{\Gamma \vdash \Delta, A \quad \Gamma, B \vdash \Delta}{\Gamma, A \to B \vdash \Delta} \to\vdash$	$\dfrac{\Gamma, A \vdash \Delta, B}{\Gamma \vdash \Delta, A \to B} \vdash\to$
Conjonction $\wedge$	$\dfrac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} \wedge\vdash$	$\dfrac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B} \vdash\wedge$
Disjonction $\vee$	$\dfrac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta} \vee\vdash$	$\dfrac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B} \vdash\vee$
Négation $\neg$	$\dfrac{\Gamma \vdash \Delta, A}{\Gamma, \neg A \vdash \Delta} \neg\vdash$	$\dfrac{\Gamma, A \vdash \Delta}{\Gamma \vdash \Delta, \neg A} \vdash\neg$
Atomiques $\bot$ et $\top$	$\dfrac{}{\Gamma, \bot \vdash \Delta} \bot\vdash$	$\dfrac{}{\Gamma \vdash \Delta, \top} \vdash\top$
Affaiblissement	$\dfrac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta}$ aff $\vdash$	$\dfrac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A} \vdash$ aff

À ces règles, on rajoute la règle axiomatique $\dfrac{}{\Gamma, A \vdash \Delta, A}$ ax.

Exercice 1

Montrer que les séquents suivants sont prouvables en calcul des séquents :

1. $\vdash A \vee \neg A$;
2. $A \wedge \neg A \vdash$;
3. $\neg A \vee \neg B \vdash \neg(A \wedge B)$;
4. $\neg(A \wedge B) \vdash \neg A \vee \neg B$;
5. $\vdash ((A \to B) \to A) \to A$.

Exercice 2

On représente une formule propositionnelle en OCaml par un objet du type suivant :

```
type formule =
     | Top
     | Bot
     | Var of int
     | Neg of formule
     | Impl of formule * formule
     | Et of formule * formule
     | Ou of formule * formule
```

où l'entier associé à une variable `Var i` correspond à l'indice i de la variable x_i (compris entre 0 et $n-1$ pour un ensemble à n variables). On représente une valuation μ comme un tableau `mu` tel que `mu.(i)` est égal à $\mu(x_i) \in \{0, 1\}$.

```
type valuation = int array
```

1. Écrire une fonction `interpretation : valuation -> formule -> int` qui prend en argument une valuation μ et une formule A (supposées sur le même ensemble de variables) et renvoie $\mu(A)$.
2. Écrire une fonction `nombre_variables : formule -> int` qui prend en argument une formule A et renvoie l'entier n correspondant au cardinal de l'ensemble de variables.
3. En déduire une fonction `tautologie : formule -> bool` qui prend en argument une formule A et renvoie un booléen qui vaut `true` si et seulement si A est une tautologie.

On représente un séquent $\Gamma \vdash \Delta$ par le type suivant :

```
type sequent =
     {gamma : formule list;
      gamma_var : bool array;
      delta : formule list;
      delta_var : bool array}
```

tel que si `seq` est un objet de type `sequent`, alors :

- `seq.gamma` est une liste de formules ;
- `seq.gamma_var` est un tableau de booléens de taille n ;
- $\Gamma = \{A \mid A$ apparaît dans `seq.gamma`$\} \cup \{x_i \mid$ `seq.gamma_var.(i)` vaut `true`$\}$;
- de même pour Δ.

Par exemple le séquent $\{x_1 \to \neg x_0, x_2, \neg x_2\} \vdash \{x_1, x_2 \wedge \neg x_1\}$ peut être représenté par :

```
let seq = {gamma = [Impl(Var 1, Neg (Var 0)); Neg (Var 2)];
           gamma_var = [|false; false; true|];
           delta = [Var 1; Et (Var 2, Neg (Var 1))];
           delta_var = [|false; true; false|]}
```

Notons que les formules atomiques limitées à une variable peuvent apparaître soit dans la liste de formules, soit comme un `true` dans le tableau de booléen (soit les deux). L'intérêt du tableau de booléens viendra par la suite.

4. Écrire une fonction `creer_sequent : formule list -> formule list -> sequent` qui prend en argument deux listes de formules Γ et Δ et crée le séquent correspondant, avec des tableaux de booléens ne contenant que des `false` (mais de la bonne taille qu'on commencera par chercher !).
5. Écrire une fonction `valide : sequent -> bool` qui prend en argument un séquent $\Gamma \vdash \Delta$ et renvoie un booléen qui vaut `true` si et seulement si $\Gamma \vDash \Delta$.

Indication : on pourra montrer que $\Gamma \vDash \Delta$ *si et seulement si* $\bigwedge_{A\in\Gamma} A \to \bigvee_{B\in\Delta} B$ *est une tautologie.*

6. Vérifier que les séquents du premier exercice sont valides en remplaçant A et B par des variables.

Exercice 3

On cherche à effectuer de la recherche de preuve de séquents valides en ne se basant que sur la syntaxe des formules et pas sur la sémantique. L'idée de l'algorithme est la suivante :

- si `gamma` et `delta` sont vides, alors le séquent n'est prouvable que s'il existe une variable qui apparaît des deux côtés du séquent ;
- sinon si `gamma` est non vide, on regarde la première formule de `gamma` :
  - si c'est $\bot$, on utilise la règle $\bot \vdash$ pour affirmer que le séquent est prouvable ;
  - si c'est $\top$, on utilise la règle aff $\vdash$ pour le supprimer, et on recommence sur ce qu'il reste ;
  - si c'est une variable x_i, on met `true` dans le tableau `gamma_var` à l'indice i, et on recommence en supprimant x_i de la liste ;
  - si c'est une formule avec un connecteur logique, on applique la règle correspondante et on vérifie que la ou les prémisses de la règle sont prouvables ;
- sinon si `gamma` est vide, on regarde la première formule de `delta` et on traite de manière similaire (attention, les formules atomiques ne sont pas traitées de la même manière que pour `gamma`).

1. Écrire une fonction `impl_gauche : sequent -> sequent * sequent` qui prend en argument un séquent $\Gamma \vdash \Delta$ dont la première formule de Γ est de la forme $A \to B$ et renvoie un couple de séquents correspondant aux prémisses de la règle $\to \vdash$ correspondante. On renverra un message d'erreur si Γ n'est pas du bon format.
2. Écrire une fonction `impl_droite : sequent -> sequent` qui prend en argument un séquent $\Gamma \vdash \Delta$ dont la première formule de Δ est de la forme $A \to B$ et renvoie un séquent correspondant à la prémisse de la règle $\vdash \to$ correspondante.
3. Écrire de même des fonctions `et_gauche`, `et_droite`, `ou_gauche`, `ou_droite`, `neg_gauche`, `neg_droite`, `aff_gauche` et `aff_droite`, de signature `sequent -> sequent * sequent` ou `sequent -> sequent` selon la règle.
4. En déduire une fonction `prouvable : sequent -> bool` qui prend en argument un séquent $\Gamma \vdash \Delta$ et renvoie un booléen qui vaut `true` si et seulement si il est prouvable. **Attention** : pour éviter les liaisons de données, on fera des copies des tableaux `gamma_var` et `delta_var` avec `Array.copy` et on créera un nouveau séquent chaque fois qu'une modification sera nécessaire.
5. Tester la fonction `prouvable` avec les mêmes formules que dans l'exercice précédent.

Exercice 4

Montrer par induction que le calcul des séquents est correct, c'est-à-dire que si $\Gamma \vdash \Delta$, alors $\Gamma \vDash \Delta$.

Comme dans la preuve du cours, on considèrera chaque règle et on montrera les bonnes inclusions d'ensembles de modèles.

Exercice 5

On cherche à montrer que le calcul des séquents est complet, c'est-à-dire que si $\Gamma \vDash \Delta$, alors $\Gamma \vdash \Delta$. Pour cela on introduit la notion suivante :

Définition

Une règle d'inférence non axiomatique est dite **inversible** si lorsque le séquent de la conclusion de la règle est valide, alors toutes les prémisses de la règle sont valides.

1. Montrer que les règles $\vee_i$, $\wedge_e$, $\vee_e$, $\to_e$ de la déduction naturelle ne sont pas inversibles.

2. Montrer que toutes les règles du calcul des séquents, sauf les axiomes et les affaiblissements, sont inversibles.
3. Soit $\Gamma \vdash \Delta$ un séquent sur un ensemble de variables V. On suppose qu'aucune règle du calcul des séquents ne peut être appliquée pour obtenir $\Gamma \vdash \Delta$ en conclusion.
   (a) Montrer que $\Gamma \subseteq V \cup \{\top\}$, $\Delta \subseteq V \cup \{\bot\}$ et $\Gamma \cap \Delta = \emptyset$.
   (b) En déduire que $\Gamma \nvdash \Delta$.
4. Montrer que le calcul des séquents sans affaiblissement est complet.