

INF9340 Devoir 6 : Procédure de décision

Ce devoir est à remettre avant 23h59 le dimanche 26 avril 2026. Assurez-vous d'inclure votre nom et code permanent. Un gabarit LaTeX est fourni. Tous les problèmes sont assujettis à la politique du tableau noir (voir le site web du cours pour les détails de cette politique). N'oubliez pas que si vous utilisez la politique du tableau noir, vous *devez* indiquer avec qui vous avez collaboré.

Les solutions écrites sont à remettre sur papier au secrétariat ou par courriel en format PDF. Le code et les tests sont à remettre par courriel dans un fichier tar compressé (tar.gz). **Vous devez citer toutes les ressources que vous utilisez autres que les notes que vous avez prises en classes.**

Conception (15 points)

Ce devoir vous demande d'implémenter une des procédures de décision pour la logique intuitionniste propositionnelle vues en classe. Rédigez un document de moins d'une page et demi qui explique brièvement votre procédure de décision. Assurez-vous d'inclure toutes les règles d'inférence utilisées par votre système. Si vous utilisez une procédure pour vérifier s'il y a des boucles dans la fouille, donnez son pseudocode et une brève explication de sa correction. Ma solution combine le calcul d'inversion et G4ip.

Implémentation (30 points)

Implémentez un programme interactif qui reçoit sur l'entrée standard une séquence de lignes, où chaque ligne est une proposition. Après avoir lu chaque ligne, sortez la ligne OUI sur la sortie standard si la proposition est démontrable en logique intuitionniste propositionnelle et la ligne NON sinon. Le programme termine à la fin de l'entrée standard. Par exemple, voici une interaction :

```
$ ./solution
A | B => B | A
OUI
∅
NON
A | ~A
NON
~~(A | ~A)
OUI
A & B => A
OUI
A => B <=> A => B
OUI
```

Vous pouvez utiliser un des langages suivants : OCaml, Standard ML, Haskell, C, C++, Java, Python 3, Rust, Go. Si vous souhaitez utiliser un autre langage, veuillez m'en parler. Si vous utilisez OCaml, vous pouvez utiliser mon gabarit `decide.ml` : il suffit d'implémenter la fonction `decide`. Si vous utilisez mon gabarit, vous pouvez activer son mode verbeux avec l'option `-v`, par exemple :

```
$ ./solution -v
A => ~~A
OUI: A => ~~A
~~A => A
NON: ~~A => A
```

La correspondance entre les opérateurs ASCII et la notation logique utilisée dans le cours est la suivante, en ordre de précedence croissant :

ASCII	Logique
$P \Leftrightarrow Q$	$(P \supset Q) \wedge (Q \supset P)$
$P \Rightarrow Q$	$P \supset Q$
$P \& Q$	$P \wedge Q$
$P \mid Q$	$P \vee Q$
$\sim P$	$P \supset Q$
(P)	(P)
1	\top
\emptyset	\perp
$A, A1, A2, \dots$	$\alpha, \alpha_1, \alpha_2, \dots$

Les variables propositionnelles satisfont l'expression régulière suivante : $[A-Za-z][A-Za-z0-9]^*$. Des espaces peuvent être présents, sauf à l'intérieur de variables propositionnelles.

Des fichiers tests true et false sont fournis avec le gabarit pour vous aider à tester votre implémentation. Le fichier true contient des propositions démontrables ; le fichier false des propositions non démontrables. Leurs tests proviennent en partie de la « *The ILTP Library : Benchmarking Theorem Provers for Intuitionistic Logic* » à Jens Otten et Thomas Rath. Veuillez remettre au moins 20 tests supplémentaires, dont au moins dix propositions démontrables et dix propositions non démontrables.

À remettre : votre archive solution.tar.gz doit contenir les fichiers suivants :

- Un fichier README qui explique comment compiler votre solution si vous n'utilisez pas le gabarit decide.ml
- Le code source de votre programme
- Les fichiers true et false qui contiennent au moins 10 tests chacun
- Votre document PDF de conception si vous ne le remettez pas sur papier.

Votre implémentation sera évaluée selon les critères suivants, en ordre décroissant de pondération : correction, élégance, efficacité. Vos tests seront évalués selon la couverture de code, l'originalité et le rapport entre la taille de chaque proposition et la complexité de sa démonstration (le temps requis pour la démontrer). Votre implémentation sera testée sur l'ensemble des tests valides remis.

Boni (5 points)

Si votre implémentation est plus rapide en moyenne que la mienne sur l'ensemble des tests, vous recevrez cinq points boni. Pour l'instant, sur les tests inclus,

```
$ ocamlc -o solution solution.ml && cat tests/* | time ./solution > /dev/null
./solution > /dev/null 3,65s user 0,00s system 99% cpu 3,650 total
```

Je me réserve le droit d'optimiser ma solution avant la remise.