Logique constructive et types dépendants

Preuves assistées par ordinateur – TP 1 – 31 janvier 2025

Année scolaire 2024–2025

Exercice 1 : Un peu de théorie constructive des ensembles

On se donne deux ensembles `A` et `B`, qu'on définit une fois pour toutes les définitions qui vont suivre.

```
Parameters A B : Set.
```

1. Définir une propriété `injective: (A -> B) -> Prop` sur les fonctions `A -> B` telle qu'on ait `injective f` lorsque `f` est injective (*c.-à.-d.*, pour tous x et y, si $fx = fy$, alors $x = y$).
2. Montrer que `f` est injective si et seulement si `f` est simplifiable à gauche (*c.-à.-d.*, pour toutes fonctions g et g', si $f \circ g = f \circ g'$, alors $g = g'$).
3. Définir une propriété `surjective: (A -> B) -> Prop` sur les fonctions `A -> B` telle qu'on ait `surjective f` lorsque `f` est surjective (*c.-à.-d.*, pour tout y, il existe x, tel que $fx = y$).
4. En supposant le tiers exclu, montrer que `f` est surjective si et seulement si `f` est simplifiable à droite (*c.-à.-d.*, pour toutes fonctions g et g', si $g \circ f = g' \circ f$, alors $g = g'$).
5. Peut-on se passer du tiers exclu ?

Exercice 2 : Lemme d'affaiblissement

On se donne un ensemble de variables.

```
Parameter var : Set.
```

1. Définir un inductif `lambda` pour représenter les termes du λ-calcul (on demande trois constructeurs, pour les variables, les abstractions et les applications).
2. Définir un inductif `type` pour représenter les types (on demande seulement deux constructeurs : les types atomiques, qui seront nommés par des variables, et l'implication).
3. On représentera un contexte par une valeur de type `list (var * type)`. Définir un inductif `derivation` pour représenter les dérivations de typage en logique naturelle : cet inductif aura trois indices, pour le contexte, le λ-terme et son type. On aura un constructeur pour chacune des trois règles suivantes : la règle axiome et les règles d'introduction et d'élimination de l'implication.
4. Montrer le lemme d'affaiblissement sur ce fragment : pour tous Γ, x et t, si $\Gamma \vdash x : t$, alors pour tout Γ', si $\Gamma \subseteq \Gamma'$, alors $\Gamma' \vdash x : t$.
5. On pourra itérativement élargir le fragment aux autres règles de la logique naturelle.

Exercice 3 : Diagonale d'une matrice carrée

On se donne le type des listes de longueur fixée.

```
Inductive vect {A}: nat -> Type :=
| nil: vect O
| cons: forall {n}, A -> vect n -> vect (S n).
```

Définir une fonction `diag: forall A n, @vect (@vect A n) n -> @vect A n` qui extrait la diagonale d'une matrice carrée.