

Examen Final

IFT-2035

16 Décembre 2020

Directives

- Répondre dans le *fichier de réponses* associé.
- Chaque question vaut 5 points, pour un total de 25 points.
- Les questions ne sont pas placées par ordre de difficulté.

0 Identification et code d'honneur

1. Écrire votre nom et matricule dans le fichier de réponses.
2. Promettre de faire cet examen sans faire recours à de l'aide extérieure.

1 Cat

Soit le langage Cat du TP2, étendu avec de nouvelles opérations: `concat` qui prend deux objets de type `string` sur la pile et les remplace par leur concaténation, `tostring` qui converti un entier en sa représentation décimale, et `length` qui calcule la longueur d'une liste.

Les règles de typage respectives sont:

$$\overline{\vdash \text{concat} : \text{string}^{\wedge} \text{string}^{\wedge} ST \Rightarrow \text{string}^{\wedge} ST}$$

$$\overline{\vdash \text{tostring} : \text{int}^{\wedge} ST \Rightarrow \text{string}^{\wedge} ST} \quad \overline{\vdash \text{length} : \text{list}(T)^{\wedge} ST \Rightarrow \text{int}^{\wedge} ST}$$

Donner le type des fonctions suivantes.

Par exemple, pour 0 la réponse est: `[add] : forall(t, int^int^t → int^t)`

0. `[add]`
1. `[concat, dup, concat]`
2. `[]`
3. `[cons, length, tostring]`
4. `[length, cons]`
5. `[car, concat, cons]`

Donner une fonction qui a le type donné.

Par exemple, pour 6 la réponse est: `forall(t, int^int^t → int^t) ⇒ [add]`

6. `forall(t, int^int^t → int^t)`
7. `forall(t1, forall(t2, t1^list(t1)^t2 → int^t2))`
8. `forall(t1, forall(t2, forall(t3, t1^t2^t3 → t3)))`
9. `forall(t, t → string^t)`
10. `forall(t1, forall(t2, t1^t2 → list(t1)^t2))`

2 Raisonner

Soit le morceau de code suivant, où f est une fonction quelconque que l'on ne connaît pas (e.g. définie dans un autre fichier pas encore écrit) et où le langage utilise une syntaxe de style C:

```
int x = 16;
void foo (callback_type f, char *s) {
    int y = 64;
    s[0] = 'E';
    char *st = strcpy (s); /* Renvoie une copie de la chaîne. */
    char *si = st + 1;
    si[4] = 'a';
    {
        char *s4 = st + 1;
        int x = y + strlen (si);
        f (si, x);
    }
    printf ("Done!\n");
    return;
}
```

On aimerait savoir (par exemple pour décider de la validité de certaines optimisations) si certaines conditions sont nécessairement toujours vraies juste avant le `return`.

On s'intéresse plus particulièrement aux conditions suivantes:

<code>st[0] == 'E'</code>	<code>x == 16</code>
<code>s[0] == 'E'</code>	<code>y == 64</code>
<code>s4[4] == 'a'</code>	<code>si == st + 1</code>

Dans chacun des cas suivants, indiquer lesquelles de ces 7 conditions sont nécessairement vraies et si non, justifier pourquoi pas:

1. Le langage est exactement comme C: portée statique, passage d'arguments par valeur.
2. Le langage est comme C sauf que les arguments sont passés par référence.
3. Le langage est comme C mais avec portée dynamique.

3 Objets géométriques

Soit le langage hypothétique Claskell qui combine Haskell avec un système orienté objet à base de classes traditionnelles.

```
class Drawable { method draw :: Image → Image; };
class Square extends Drawable
{ size :: Float; method draw(d :: Image) {...}; };
class Circle extends Drawable
{ radius :: Float; method draw(d :: Image) {...}; };
```

Sachant donc que $Circle \subseteq Drawable$ et $Square \subseteq Drawable$, indiquer quelles relations de sous-typage seraient valides:

1. $Square \subseteq Square$
2. $Circle \rightarrow Circle \subseteq Circle \rightarrow Drawable$
3. $Circle \rightarrow Circle \subseteq Drawable \rightarrow Drawable$
4. $List\ Square \subseteq List\ Drawable$
5. $(Square \rightarrow Drawable) \rightarrow Circle \subseteq (Square \rightarrow Circle) \rightarrow Drawable$

Soit un objet $c :: Circle$ et soit une fonction *zoom* qui prend un facteur d'agrandissement et un objet de la classe *Drawable*, et renvoie un objet équivalent mais agrandi. Donner le type de:

6. *zoom*
7. *zoom* 2.0 *c*

Finalement, un programmeur tout aussi hypothétique décide de traduire le code vers Haskell (et ses classes de type):

```
class Drawable a with draw :: Image → Image
data Square = Square { size :: Float }
data Circle = Circle { radius :: Float }
instance Drawable Square with
  draw d = ...
instance Drawable Circle with
  draw d = ...
c :: Circle = ...
```

Donner le type Haskell qui en résulte pour:

6. *zoom*
7. *zoom* 2.0 *c*

4 Tri inefficace en Prolog

1. Définir en Prolog la relation *sorted(+L)* qui vérifie que L est triée.
2. Définir la relation *takeone(+L₁, -X, -L₂)* qui énumère les éléments X de L_1 et renvoie dans L_2 le reste de la liste L_1 (e.g. *takeone*([1, 2, 3], X, L) renverrait les trois réponses suivantes: $X = 1, L = [2, 3]$; $X = 2, L = [1, 3]$; et $X = 3, L = [1, 2]$).
3. En utilisant *takeone*, définir la relation *permute(+L₁, -L₂)* qui énumère toutes les listes L_2 , permutations de la liste L_1 .
4. En utilisant la technique de *generate&test* et les relations ci-dessus, définir une relation *sort(+L₁, -L₂)* qui trie la liste L_1 et renvoie le résultat dans L_2 .

Notes:

- Chacune de ces questions est indépendante des autres, donc vous pouvez par exemple répondre à la dernière en utilisant les relations des questions précédentes, même si vous ne savez pas comment définir ces relations.
- Vous pouvez définir des relations auxiliaires si vous en ressentez le besoin.

5 Gestion mémoire

Soit le code d'une bibliothèque C ci-dessous:

```
typedef struct list_elem list_elem;
struct list_elem {
    char value;
    list_elem *next;
}
typedef struct list list;
struct list { list_elem *head; }

list *list_alloc (void);
void list_insert (list *l, char v);
char list_remove (list *l, int n);
list *list_concat (list *l1, list *l2);
...
```

1. Combien d'opérations (complexité algorithmique) va faire `list_remove` pour une valeur `N` donnée et une liste de longueur `M`?
2. `list_remove` devrait-elle libérer l'espace mémoire occupé par le `list_elem`? Pourquoi?
3. `list_remove` devrait-elle libérer l'espace mémoire occupé par la valeur enlevée (i.e. la N ème valeur)? Pourquoi?
4. Si un `char` a une taille de 1B, que les pointeurs ont une taille de 8B, et que ces valeurs doivent obéir l'alignement naturel, quelle doit être la taille totale d'un objet de type `list_elem`?
5. Sachant que la fonction `list_concat` ne modifie aucun de ses arguments, et que `l1` est de longueur N_1 et `l2` est de longueur N_2 , combien de bytes de mémoire doit-elle allouer?