

3.1 問題（依存型プログラミング的な表現）

引数として以下の 3 つ

1. リストのリスト `lst`
2. 整数 `i`
3. 整数 `j`

を受け取ったら、`lst` の `i` 番目のリストの `j` 番目の要素を削除したものを返す関数 `remove-lst` を作成せよ。その際、以下の 3 つの条件

1. `lst` の `i` 番目のリストが存在する。
2. `i` 番目のリストの `j` 番目の要素が存在する。
3. 返されるリストたちの長さは、`i` 番目のみ 1 短くなっており、他は変わらない。

が保証されるようにせよ。（ここで、先頭はいずれも 0 番目とする。）

3.2 問題（論理的な表現）

以下の 3 つのデータがある。

1. リストのリスト `lst`
2. 整数 `i`
3. 整数 `j`

このとき、`lst` の `i` 番目のリストの `j` 番目の要素を取れるならば、以下を満たすリストが（唯一）存在することを示せ。

`lst` の中の

- `i` 番目のリストについては、その `j` 番目の要素を取り除き、
- それ以外のリストについては、もとのまま

であるようなリストで、返ってくるリストたちの長さは、

- `i` 番目のリストについては、もとのリストより 1 短くなっており、

- それ以外については、もとのリストと同じになっている。（ここで、先頭はいずれも 0 番目とする。）

3.3 OCaml によるプログラム

この問題（の依存型プログラミング表現の方）を（3つの条件を満たさない形で）OCaml を使って書くと [remove.ml](#) のようになります。（プログラム中のテストが、プログラムの入出力の具体例になっています。）このプログラム中、2か所に現れる `raise CannotHappen` となるケースを、上手に引数に条件を加えることで、起こり得ないようになるとともに、返ってくるリストの長さが所定のところのみ 1 小さくなっていることを示してくださいと言うのが問題の趣旨です。

2 か所の例外は、どちらもリストの長さを考慮すれば簡単に削除できそうに見えます。`lst` の長さを n 、`lst` 中の各リストの長さを順に m_0, \dots, m_{n-1} としましょう。すると i は 0 から $n-1$ までの整数、 j は 0 から m_i-1 までの整数とすれば良いでしょう。ここで問題となるのは j の範囲が m_i （したがって i ）に依存しているところです。それをうまく書けるかどうかが知りたいです。

3.4 問題の背景

項の单一化を行う際には、メタ変数に項を代入します。例えば SK 論理の式 $SXK(YKZ)K$ があったとしましょう。ここで X, Y, Z はメタ変数です。この式のメタ変数 Y に S を代入すると $SXK(SKZ)K$ が得られます。この代入の操作の際、メタ変数は $[X; Y; Z]$ の 3 つから $[X; Z]$ のふたつに減っています。この操作が、上の問題で `lst` の長さが 1 の場合に対応しています。つまり、`lst` の 0 番目のリストの 1 番目の Y を削除し、残ったメタ変数のリストを返しています。

この例では「SK 論理の式」というひとつの概念のみを扱っていたので、メタ変数もそれを表す 1 種類のみでした。上の問題は、項の種類を複数に拡張し、相互再帰している場合に対応しています。今、項が n 種類からなり、それらが相互再帰しているとします。 n 種類の項に対応したメタ変数がそれぞれ m_0, \dots, m_{n-1} 個あったとしましょう。この状態で、 i 番目の項の j 番目のメタ変数に代入を行うと、代入後のメタ変数は、そのひとつだけが削除された形になります。

現在、このような形でメタ変数の数を管理しながら項の单一化のプログラムを書こう

としているのですが、相互再帰してメタ変数の種類が複数になった場合、どのようにして良いのかわからなくなり、皆さんのお知恵を借りたいと思った次第です。