

練習問題

- 以下の文法 (\mathcal{G}_1) が定義されているとき、与えられた表現が項 (Term) か否かを判定する `isTm1(Term)` のプログラムを作成せよ。 `isTm1(f(a))`, `isTm1(f(f(b)))` は成功し, `isTm1(c)` は失敗することを確認せよ。 \mathcal{G}_1 はプログラムが対象とする言語であり、プログラミング言語と混同しないように注意。(\mathcal{G}_1 そのものをプログラムの一部として記述するのは間違いである。)

(\mathcal{G}_1)

```
Term ::= Alphabet | f(Term)
Alphabet ::= a | b
```

- この記法は BNF 記法 (Backus-Naur Form または Backus-Normal Form) と呼ばれる代表的な文法記述方法に基づく。
 - 大文字からはじまるものは非終端記号 (定義がある), 小文字からはじまるものは定数およびファンクタ (与えられたもの)
- 以下の文法 (\mathcal{G}_2) が定義されているとき、与えられた表現が項 (Term) か否かを判定する `isTm2(Term)` のプログラムを作成せよ。データ `f(a)`, `g(0,f(b))`, `f(c)`, `g(a)` について動作確認せよ。

(\mathcal{G}_2)

```
Term ::= Alphabet | Digit | f(Term) | g(Term,Term)
Alphabet ::= a | b
Digit ::= 0 | 1
```

演習問題 (r7)

* のついている問題はオプションなのでできる者のみ解答せよ。

- 以下の文法 (\mathcal{G}_3) が定義されているとき、与えられた表現が項 (Term) か否かを判定する `isTm3(Term)` のプログラムを作成せよ。データ `g(f(a),0)`, `g(0,f(b))`, `g(b,f(0))` について動作確認せよ。

```
Term ::= Alphabet | Digit | f(Term) | g(Alphabet,Term)
Alphabet ::= a | b
Digit ::= 0 | 1
```

- 文法 (\mathcal{G}_2) になかったもののみがデータとして与えられるとする。与えられた項 Term に Alphabet が出現する回数が C であるという関係を表す述語 `count_alphabet(Term,C)` のプログラムを r6(2) で作成した述語 `count_ocr` を使用して作成せよ。たとえば `count_alphabet(g(0,f(g(b,f(a))))),C)` は C=2 となって成功する。
- 文法 (\mathcal{G}_2) になかったもののみがデータとして与えられるとする。与えられた項 Term に含まれるファンクタ f の数が N であるという関係を表す述語 `n_of_f(Term,N)` のプログラムを作成せよ。たとえば, `n_of_f(g(a,g(f(0),1)),N)` は N=1 となって成功する。

- (4) 文法 (\mathcal{G}_2) になかったもののみがデータとして与えられるとする。与えられた項 Term に出現するファンクタ f の数と, g の数の比較をし, 多い方が F であるという関係を表す述語 `compare_fnumbers(Term,F)` のプログラムを作成せよ。同数の場合は, $F = \text{same}$ とせよ。必要な述語は自分で定義せよ。たとえば, `compare_fnumbers(g(a,g(f(0),1)),F)` は $F = g$ となって成功する (r7(3) と同様に `n_of_g` を定義し, `n_of_f` の結果と比較するのが基本的な手法だが, f,g の数を求める共通の述語を生成する方が望ましい。できる人はこちらでやってみてください。)
- (5) 文法 (\mathcal{G}_2) になかったもののみがデータとして与えられるとする。与えられた項 T1 に出現する a,b をそれぞれ 0,1 に置き換えた結果が T2 であるという関係を表す述語 `subst_ab(T1,T2)` のプログラムを r6(1) で作成した `subst_atom` を使用して作成せよ。たとえば `subst_ab(g(g(0,a),f(g(b,a))),T)` は $T = g(g(0,0),f(g(1,0)))$ となって成功する。
- (6)* 論理式が以下のように定義されたとする。与えられた表現 E が論理式であるかどうかを判定する述語 `isFormula(E)` のプログラムを作成せよ。たとえば, `isFormula(and(p,neg(q)))`, `isFormula(imp(and(p,q),p))` は成功し, `isFormula(or(and(p,q)))` は失敗する。

```
Fml ::= Atm | neg(Fml) | and(Fml,Fml) | or(Fml,Fml) | imp(Fml,Fml)
Atm ::= p | q
```

- (7) r7 の練習問題 2 についてレポートせよ。(i) プログラムの各節の論理的意味 (命題の形になっていること), (ii) `?- isTm2(g(0,f(b)))`. を実行したときの動作 (トレースを貼り付けてはいけない! 「ゴール」「実行」「単一化 (ユニフィケーション)」という用語をすべて用いてどのゴールとどの節のヘッドが単一化されて変数がどう書き換わり, どのゴールが呼ばれるなどを段階的に記述すること)。