

テストデータは、問題文から copy&paste して解答例のようにプログラムに記述しておくことで打ち間違いを少なくしてください。

質問等について

プログラムが未完成だった場合、どこまで取り組めてどこがわからないかコメントを使って記述してください。完成できた場合でも質問があれば書いてください。個別に教えます。

時間をかけてききたい場合は TA office hour を利用するなどしてください。

コメントの書き方 (... の部分がコメントになる)

% ... 単一行内でのコメント

/* ... */ 複数行にわたるコメント

日本語を使用する場合はプログラムの冒頭に次の一文をいれておくと error, warning は出ません。

```
:- encoding(utf8).
```

節の実行順と if 文の関係

例: r6(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。項 T1 に出現するアトム A をすべて項 B に書き換えた結果が項 T2 である

(\mathcal{G}_2)

```
Term ::= Alphabet | Digit | f(Term) | g(Term,Term)
```

```
Alphabet ::= a | b
```

```
Digit ::= 0 | 1
```

第 3 引数 T1 にはどのような形のものがくるか。上から if...else if ... else の文を書いていることに相当する。

```
is_atom(a). is_atom(b). is_atom(0). is_atom(1).
```

[program 1] OK

```
subst_atom(A,B,A,B).
```

```
subst_atom(A,B,f(T1),f(T2)) :- subst_atom(A,B,T1,T2).
```

```
subst_atom(A,B,g(X1,Y1),g(X2,Y2))
```

```
:- subst_atom(A,B,X1,X2), subst_atom(A,B,Y1,Y2).
```

```
subst_atom(_,_,T1,T1).
```

[program 2] NG -- Why?

```
subst_atom2(A,B,A,B).
```

```
subst_atom2(_,_,T1,T1).
```

```
subst_atom2(A,B,f(T1),f(T2)) :- subst_atom2(A,B,T1,T2).
```

```
subst_atom2(A,B,g(X1,Y1),g(X2,Y2))
```

```
:- subst_atom2(A,B,X1,X2), subst_atom2(A,B,Y1,Y2).
```

```
[program 3] OK -- Why?
subst_atom3(A,B,A,B).
subst_atom3( _,_,T1,T1) :- atomic(T1).
subst_atom3(A,B,f(T1),f(T2)) :- subst_atom3(A,B,T1,T2).
subst_atom3(A,B,g(X1,Y1),g(X2,Y2))
    :- subst_atom3(A,B,X1,X2), subst_atom3(A,B,Y1,Y2).
```

テストの手法

[corrected 2026/05/15 16:50]

```
t1(X) :- subst_atom( a, g(b,b), a, X)
t2(X) :- subst_atom( a, g(b,b), 1, X)
t3(X) :- subst_atom( a, g(b,b), f(1), X)
t4(X) :- subst_atom( a, g(b,b), g( a, f(1) ), X)
```

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
```

- 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))`, `g(a), f(g(a,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 .

- (1) 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(f(0), b)` について動作確認せよ .

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

- (2) 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 .
- (3) 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 . (まず a を 0 に置き換えその結果に対して b を 1 に置き換えると考えよ .)
- (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 .
- (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 Term に含まれるファンクタ g の数が N であるという関係を表す述語 `n_of_g(Term,N)` のプログラムを作成せよ . 次に , 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 `n_of_f` および `n_of_g` を使用して作成せよ . 同数の場合は , `F=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 .
- (6)* `r7(3)` の関係を表す述語 `subst_ab_2(T1,T2)` のプログラムを `subst_atom` を使用せずに作成せよ . (再帰的に記述する際に置き換えを同時に行うと考える .)
- (7) 今回の演習問題 r7 (1)-(6)(論理的意味と実行過程を含む) の解答にあたって生成 AI を少しでも使用したものの番号をすべて記述せよ .