

1 Prolog for AI

2002

Give a simple definition of the Prolog predicate `dfx` that can perform symbolic differentiation with respect to the variable `x` of expressions composed of integers (e.g. 0, 1, ...), symbolic constants (e.g. a, b, ...), symbolic variables (e.g. x, y, ...) and the operators +, - and *, for addition, subtraction and multiplication. The first argument of `dfx` is the expression to differentiate and the second argument is the result. Your definition need not perform any simplification of the result. [6 marks]

Trace the execution of the call: `dfx(x*x-2, R)`. [2 marks]

Now modify your definition so that it simplifies the result by the applications of rewriting rules such as: $1*x \Rightarrow x$ and $x-0 \Rightarrow x$. [8 marks]

Discuss to what extent, if any, either of your predicates could be used to integrate an expression. [4 marks]

ANSWER:

First part:

```
dfx(x, 1) :- !.
dfx(X+Y, DX+DY) :- !, dfx(X,x,DX), dfx(Y,x,DY).
dfx(X-Y, DX-DY) :- !, dfx(X,x,DX), dfx(Y,x,DY).
dfx(X*Y, DX*Y+X*DY) :- !, dfx(X,x,DX), dfx(Y,x,DY).
dfx(_, 0).
```

trace:

```
dfx(x*x-2, R) =>
  R = DX-DY where
    dfx(x*x,DX) and dfx(2,DY)
  dfx(x*x, DX*x+x*DY) :- !, dfx(x,DX), dfx(x,DY)
    => DX=1 and DY=1
    so dfx(x*x, 1*x+x*1)
  dfx(2, DY) =>DY=0
    so dfx(2, 0)
  so R + (1*x+x*1)-0
```

second part:

```
dfx(x, 1) :- !.
dfx(X+Y, RES) :- !, dfx(X,DX), dfx(Y,DY), simplify(DX+DY,RES).
dfx(X-Y, RES) :- !, dfx(X,DX), dfx(Y,DY), simplify(DX-DY,RES).
dfx(X*Y, RES) :- !, dfx(X,DX), dfx(Y,DY),
    simplify(Y,YY), simplify(DX*YY, L),
    simplify(X,XX), simplify(XX*DY, R),
    simplify(L+R,RES).
```

```
dfx(_, 0).
```

```
simplify(X*Y, 0) :- simplify(X, 0).
simplify(X*Y, 0) :- simplify(Y, 0).
simplify(X*Y, YY) :- simplify(X, 1), simplify(Y,YY).
simplify(X*Y, XX) :- simplify(Y, 1), simplify(X,XX).

simplify(X+Y, YY) :- simplify(X, 0), simplify(Y,YY).
```

```
simplify(X+Y, XX) :- simplify(Y, 0), simplify(X,XX).  
simplify(X-Y, XX) :- simplify(Y, 0), simplify(X,XX).  
simplify(X-Y, YY) :- simplify(X, 0), simplify(-Y,YY).
```

```
simplify(-(-X), XX) :- simplify(X,XX).  
simplify(-0, 0).  
simplify(+X,XX) :- simplify(X,XX).
```

last part:

No because '!' is non reversible. In any case it is could only integrate expressions that were identical to ones that could be the result of df. There are an infinite number of terms that could differentiate to 0. It may loop trying to find on.

The situation is even worse the the simplifying version of df.