Reescrita de termos em Prolog

Renan Thiago da S. Rosa [1] **, Renick Muller Teixeira Costa** [1]

[1]Instituto de Ciências Exatas e Naturais (ICEN) – Universidade Federal do Pará (UFPA)
Belém – PA – Brasil

{renickmillerteixeiracosta,renannojosa}@gmail.com

1. Introdução

1.1. Transformações geométricas

Pode-se utilizar diferentes sistemas de coordenadas para descrever os objetos modelados em um sistema 2D. O Sistema de Coordenadas serve para nos dar uma referência em termos de medidas do tamanho e posição dos objetos dentro da nossa área de trabalho. Uma vez que aplicações gráficas frequentemente requerem transformações de um sistema de coordenadas para outro, estas as transformações geométricas podem ser representadas na forma de equações modeladas em forma de matriz.

$$Soma: \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 4 & 6 \\ 6 & 4 & 2 \\ 2 & 4 & 4 \end{bmatrix}$$

Este exemplo faz uso do produto interno e da transposição. Uma matriz é representada aqui como uma lista de listas. O código relevante é reproduzido na Listagem 1. Neste trecho de código, o predicado de transposição recebe uma matriz de tamanho qualquer e transforma as suas linhas em colunas. Ao final do processo, espera-se como resultado algo parecido com a matriz de transposição.

```
transpose([[]|_], []).
transpose(M,[Ci|Cn]) :- columns(M, Ci, R),
                        transpose(R, Cn).
columns([], [], []).
columns([[Cii|Cin]|C], [Cii|X], [Cin|V]) :- columns(C, X, V).
```

Figura 1. Predicado de transposição

$$Transpostas: A \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix} = A^t \begin{bmatrix} 1 & 3 & 2 \\ 2 & 2 & 1 \\ 3 & 1 & 2 \end{bmatrix}$$

A seguir, para efetuar a multiplicação entre matrizes, o número de colunas da primeira matriz deve ser igual ao número de linhas da segunda matriz. Uma vez que esta pré-condição é verdadeira, multiplica-se a primeira linha da primeira matriz pela primeira e segunda coluna da segunda matriz e soma-se os resultados dessa multiplicação. A matriz abaixo demonstra a operação.

$$Multiplicação: \begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix} * \begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix} = \begin{bmatrix} 1*2+2*3 & 1*2+2*2 \\ 3*1+2*3 & 3*2+2*2 \end{bmatrix} = \begin{bmatrix} 7 & 6 \\ 9 & 10 \end{bmatrix}$$

```
mmt([],_,[]).
mmt([Ai|An],B,[Ci|Cn]):- mmc(Ai,B,Ci),
                          mmt(An,B,Cn).
mmc(_,[],[]).
mmc(A,[Bi|Bn],[Ci|Cn]):- ip(A,Bi,Ci),
                          mmc(A,Bn,Cn).

ip([],[],0).
ip([Ai|An],[Bi|Bn],(X + Ai * Bi)):- ip(An, Bn, X).
```

Figura 2. Predicado de multiplicação.

1.2. Transformações lineares

Quando se trata de gerar imagens em duas dimensões, apenas a criação de primitivas e aplicação de atributos não é suficiente. É absolutamente necessário que sejam feitas certas transformações. Facilidade de aplicação de transformações em primitivas, para depois gerar as formas mais complexas. Transformações mais comuns: translação, rotação e escala. Ao expressarmos posições em coordenadas homogêneas, as equações de transformações geométricas ficam reduzidas a multiplicação de matrizes 3x3 elementos.

$$ex(a,= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(phi) & sin(phi) & 0 \\ 0 & -sin(phi) & cos(phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix})$$

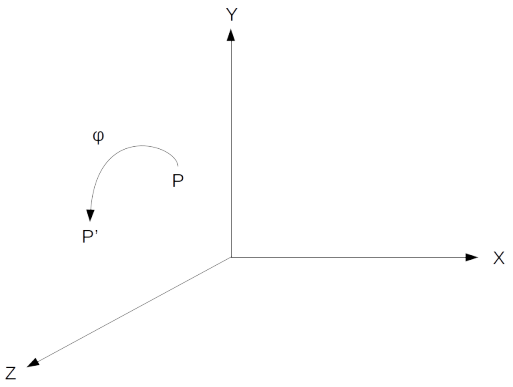

Figura 3. Representação da transformação sobre o eixo X

$$ex(b, = \begin{bmatrix} cos(theta) & 0 & -sin(theta) & 0 \\ 0 & 1 & 0 & 0 \\ sin(theta) & 0 & cos(theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix})$$

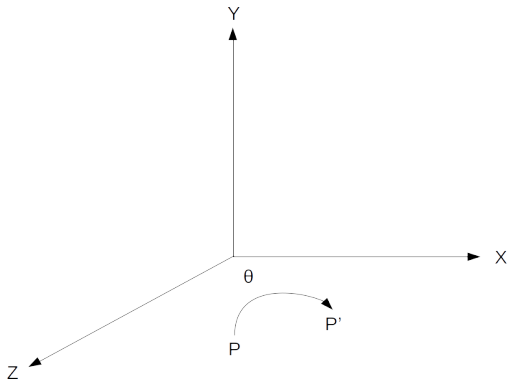

Figura 4. Representação da transformação sobre o eixo Y

$$ex(c, = \begin{bmatrix} cos(psi) & sin(psi) & 0 & 0 \\ -sin(psi) & cos(psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix})$$

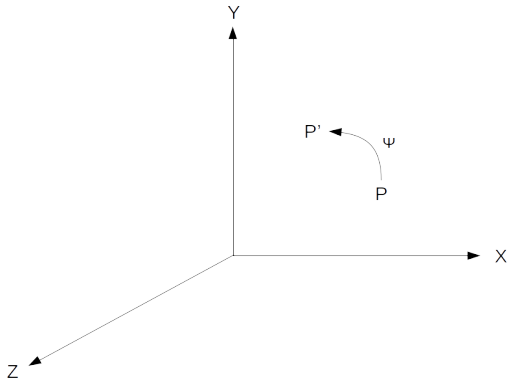

Figura 5. Representação da transformação sobre o eixo Z

1.3. Simplificação

Quando a matriz b é multiplicada pela matriz a obtemos uma matriz que é o resultado da composição das transformações sobre o eixo X seguido pela rotação sobre o eixo Y. O resultado é mostrado a seguir.

?- ex(a, A), ex(b, B), mm(A, B, P). A = [[cos(theta), 0, -sin(theta), 0], [0, 1, 0, 0] [sin(theta), 0, cos(theta), 0], [0, 0, 0, 1]] B = [[1, 0, 0, 0], [0, 1, cos(phi), sin(phi)], [0, 0, -sin(phi), cos(phi)], [0, 0, 0, 0]] P = [[0+0*0+ -sin(theta)*0+0*0+cos(theta)*1, 0+0*0+ -sin(theta)*0+0*1+cos(theta)*0, 0+0*0+ -sin(...)* -sin(...)+0*cos(phi)+cos(theta)*0, 0+ ... * ... + - ... * cos(...)+0*sin(phi)+cos(theta)*0], [0+0*0+0*0+1*0+0*1, 0+0*0+0*0+1*1+0*0, 0+ ... * ... + 0* - ... + 1*cos(phi)+0*0, ... + ... + ... * ... + 1*sin(...)+0*0], [0+0*0+cos(theta)*0+0*0+sin(theta)*1, 0+ ... * ... + cos(...)*0+0*1+sin(theta)*0, ... + ... + ... * ... + 0*cos(...)+sin(theta)*0, ... + ... + ... * ... + sin(...)*0], [0+ ... * ... + 0*0+0*0+0*1, ... + ... + ... * ... + 0*1+0*0, ... + ... + ... * ... + 0*0, ... + ... + ... * ...]] .

Este resultado, apesar de correto, está aquém do desejado, uma vez que a resposta está nublada por termos desnecessários. Os próximos predicados tomam como entrada a resposta anterios e a simplificam.

```
s(X, X).
s(A+B, C):- !, s(A, A1), s(B, B1), op(A1+B1, C).
s(A-B, C):- !, s(A, A1), s(B, B1), op(A1-B1, C).
s(A*B, C):- !, s(A, A1), s(B, B1), op(A1*B1, C).
op(A+B, C) :- integer(A), integer(B),!, C is A+B.
op(0+A, A) :- !.
op(A+0, A) :- !.
op(1*A, A) :- !.
op(0*_, 0) :- !.
op(A*1, A) :- !.
op(_*0, 0) :- !.
op(A-0, A) :- !.
op(A-A, 0) :- !.
op(X, X).
```

Figura 6. Predicado de simplificação aritmética.

O predicado exposto na Figura 6 realiza a simplificação mais direta. Esta consiste em retiar redundancias aritméticas, como multiplicações por 0 e 1. A última, na Figura 7, simplificação utiliza as leis de De Morgan.

```
dn(-(-(A)), 8) :- !, dn(A, 8).
dn(-(A+8), U+V) :- !, dn(-(A), U), dn(-(8), V).
dn(-(A*8), U*V) :- !, dn(-(A), U), dn(8, V).
dn(A+8, U+V) :- !, dn(A, U), dn(8, V).
dn(A*8, U*V) :- !, dn(A, U), dn(8, V).
dn(A, A).
```

Figura 7. Predicado de simplificação usando De Morgan.

Abaixo ambos os predicados de simplificação são combinados em umachamada recursiva a fim de simplificar onde for possível na matriz resultante da transformação geométrica incial. Após a chamada do predicado de simplificação, espera-se algo como na Figura 8.

```
/* simplify an expression */
simp(X,Y):- dn(X,A), s(A,Y).

/* simplify each element of a matrix */
simplist([], []).
simplist([[H|T]|Z],[R|S]) :- !, simplist([H|T], R), simplist(Z, S).
simplist([H|T], [R|S]) :- simp(H,R), simplist(T, S).
```

$$D, = \begin{bmatrix} cos(theta) & 0 & (-sin(theta) * -sin(phi)) & (-sin(theta) * cos(phi)) \\ 0 & 1 & cos(phi) & sin(phi) \\ sin(theta) & 0 & (cos(theta) * -sin(phi)) & (cos(theta) * cos(phi)) \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Figura 8. Predicado de simplificação e matriz resultante.

D=[[cos(theta), 0, (-sin(theta)*-sin(phi)), (-sin(theta)*cos(phi))], [0, 1, cos(phi), sin(phi)], [sin(theta), 0, (cos(theta)*-sin(phi)), (cos(theta)*cos(phi))], [0, 0, 0, 0]]