LaTeX - A Book of Drawings
An ever evolving document

Geraldo Xexéo

2023-02-13 22:31:18

Contents

List of Figures

Chapter 1

Introduction

This is a book of drawings made in LaTeX with Tikz[1] and other useful packages. Some drawings are in Portuguese.

Although most drawings in this book of examples use Tikz[2], there are some easier solutions for some specific drawings. Moreover, Tikz has multiple libraries that must be included, and I don't kept control of it, I just added everyone.

For example, chessboard[3] is a useful package for drawing chess boards. I enjoy that it uses a very practical notation that is known to chess players.

The following code generates the image in Figure 1.1.

Listing 1.1: Code for a Chess board

```
\chessboard[addfen={bnrbnkrq/%
pppppppp/%
8/8/8/8/%
PPPPPPPP/BNRBNKRQ},showmover=false]
```

[1] Pronounced "tics"

[2] https://ctan.org/pkg/pgf?lang=en

[3] https://ctan.org/pkg/chessboard?lang=en

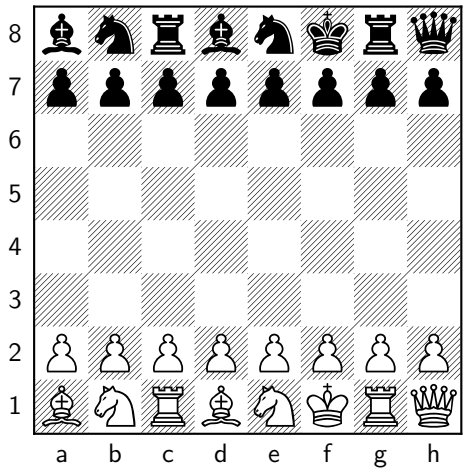


Figure 1.1: A position form Fischer's Random Chess

#648fff	#785ef0	#dc267f	#fe6100	#ffb000	#000000	#ffffff
						100%
						95%
						90%
						85%
						80%
						75%
						70%
						65%
						60%
						55%
						50%
						45%
						40%
						35%
						30%
						25%
						20%
						15%
						10%
						5%
						0%

Figure 1.2: Cores da paleta sugerida pela IBM para daltonismo, com atenuação na cor para branco.

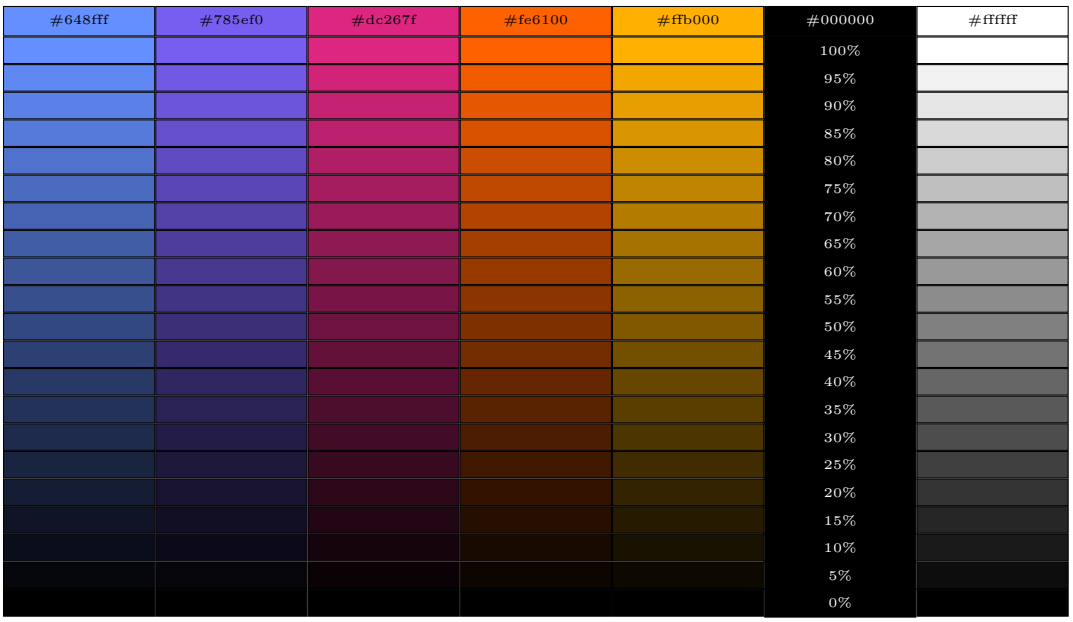


Figure 1.3: Cores da paleta sugerida pela IBM para daltonismo, com atenuação na cor para preto.

```
\begin{tikzpicture}
\node[draw,text width=3cm] at (0,0) {A
\begin{itemize}
    \item item um
    \item item dois
\end{itemize}
};
\end{tikzpicture}
```

A

- item um
- item dois

Figure 1.4: Itemize no nó precisa transformar o nó de mbox para minipage, e o text width faz isso. `https://tex.stackexchange.com/questions/213662/enumerate-within-tikz-node`

Chapter 2

3D and Fake 3D

Some times I had to build some 3D drawings based on boxes, for example, to describe a Data Warehouse Cube. There is an easy solution that is to develop a basic cube subroutine and use them to build more complex figures.

O próximo código constrói uma caixa calculando os pontos, como uma projeção de 3D em 3D.

Listing 2.1: Cubo azul em Fake 3D

```
\newcommand{\drawbox}[5]{
    \pgfmathsetmacro \angle {30}
    \pgfmathsetmacro \xd {{2/3*cos(\angle)*#5}}
    \pgfmathsetmacro \yd {{2/3*sin(\angle)*#5}}
    \pgfmathsetmacro \x {{#1-#5+(#2-#5)*(\xd)*#5}}
    \pgfmathsetmacro \y {{#3-#5+(#2-#5)*(\yd)*#5}}

    \draw[fill=#4] (\x,\y) --
    (\x+#5,\y) -- (\x+#5,\y+#5) --
    (\x,\y+#5) -- cycle;

    \draw[fill=#4] (\x,\y+#5) --
    (\x+\xd,\y+#5+\yd) --
    (\x+#5+\xd,\y+#5+\yd) --
    (\x+#5,\y+#5) -- cycle;

    \draw[fill=#4] (\x+#5,\y+#5) --
    (\x+#5+\xd,\y+#5+\yd) --
    (\x+#5+\xd,\y+\yd) --
    (\x+#5,\y) -- cycle;

\drawbox{}{1}{1}{blue}{1}
}
```

A simple blue cube, Figure 2.1, can be easily drawn with:

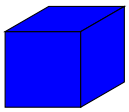


Figure 2.1: A fake 3D cube

And a composite figure can use the order of drawing to build a cube made of cubes, as in Figure 2.2.

Listing 2.2: A cube made of cubes

```
\pgfmathsetmacro{\profX}{5}

  \drawbox{1}{\profX}{1}{green}{.5}
  \drawbox{1.5}{\profX}{1}{green}{.5}
  \drawbox{2}{\profX}{1}{green}{.5}

  \drawbox{1}{\profX}{1.5}{green!50}{.5}
  \drawbox{1.5}{\profX}{1.5}{green!50}{.5}
  \drawbox{2}{\profX}{1.5}{green!50}{.5}

  \drawbox{1}{\profX}{2}{green!25}{.5}
  \drawbox{1.5}{\profX}{2}{green!25}{.5}
  \drawbox{2}{\profX}{2}{green!25}{.5}

  \pgfmathsetmacro{\profX}{3}

  \drawbox{1}{\profX}{1}{red}{.5}
  \drawbox{1.5}{\profX}{1}{red}{.5}
  \drawbox{2}{\profX}{1}{red}{.5}

  \drawbox{1}{\profX}{1.5}{red!50}{.5}
  \drawbox{1.5}{\profX}{1.5}{red!50}{.5}
  \drawbox{2}{\profX}{1.5}{red!50}{.5}

  \drawbox{1}{\profX}{2}{red!25}{.5}
  \drawbox{1.5}{\profX}{2}{red!25}{.5}
  \drawbox{2}{\profX}{2}{red!25}{.5}

  \drawbox{1}{1}{1}{blue}{.5}
  \drawbox{1.5}{1}{1}{blue}{.5}
  \drawbox{2}{1}{1}{blue}{.5}

  \drawbox{1}{1}{1.5}{blue!50}{.5}
```

```
  \drawbox{1.5}{1}{1.5}{blue!50}{.5}
  \drawbox{2}{1}{1.5}{blue!50}{.5}

  \drawbox{1}{1}{2}{blue!25}{.5}
  \drawbox{1.5}{1}{2}{blue!25}{.5}
  \drawbox{2}{1}{2}{blue!25}{.5}
```

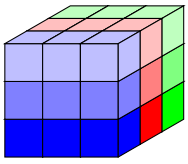


Figure 2.2: A fake 3D cube of cubes

It is also possible to use fixed coordinates, such as the following code, that result in Figure 2.3.

Listing 2.3: A cube with fixed 2d Coordinates

```
\begin{tikzpicture}[scale=0.33] %[x={10.0pt},y={10.0pt}]
\draw[line width=2pt] (0,0) -- (0,10);
\draw[line width=2pt] (0,0) -- (10,0);
\draw[line width=2pt] (0,10) -- (10,10);
\draw[line width=2pt] (0,10) -- (10,10);
\draw[line width=2pt] (10,0) -- (10,10);
\draw[line width=2pt] (0,0) -- (3,5);
\draw[line width=2pt] (10,0) -- (13,5);
\draw[line width=2pt] (0,10) -- (3,15);
\draw[line width=2pt] (10,10) -- (13,15);
\draw[line width=2pt] (3,5) -- (13,5);
\draw[line width=2pt] (3,5) -- (3,15);
\draw[line width=2pt] (13,5) -- (13,15);
\draw[line width=2pt] (3,15) -- (13,15);
\draw[line width=1pt] (5,0) -- (5,10);
\draw[line width=1pt] (5,10) -- (8,15);
\draw[line width=1pt] (5,0) -- (8,5);
\draw[line width=1pt] (8,5) -- (8,15);
\draw[black, fill=blue,fill opacity=0.5] (5,0) -- (5,10) -- (8,15) -- (8,5) -- cy
\end{tikzpicture}% pic 1
\qquad % <----------------- SPACE BETWEEN PICTURES
% ou
%\hspace{3cm}
\begin{tikzpicture}[scale=0.33] %[x={10.0pt},y={10.0pt}]
```

```
\draw[line width=2pt] (0,0) -- (0,10);
\draw[line width=2pt] (0,0) -- (10,0);
\draw[line width=2pt] (0,10) -- (10,10);
\draw[line width=2pt] (0,10) -- (10,10);
\draw[line width=2pt] (10,0) -- (10,10);
\draw[line width=2pt] (0,0) -- (3,5);
\draw[line width=2pt] (10,0) -- (13,5);
\draw[line width=2pt] (0,10) -- (3,15);
\draw[line width=2pt] (10,10) -- (13,15);
\draw[line width=2pt] (3,5) -- (13,5);
\draw[line width=2pt] (3,5) -- (3,15);
\draw[line width=2pt] (13,5) -- (13,15);
\draw[line width=2pt] (3,15) -- (13,15);
\draw[line  width=1pt] (0,10) -- (10,0);
\draw[line  width=1pt] (0,10) -- (3,5);
\draw[line  width=1pt] (3,5) -- (10,0);
\draw[black, fill=blue,fill opacity=0.5] (0,10) -- (10,0) -- (3,5) -- cy
\draw[line  width=1pt] (3,15) -- (10,10);
\draw[line  width=1pt] (3,15) -- (13,5);
\draw[line  width=1pt] (10,10) -- (13,5);
\draw[black, fill=blue,fill opacity=0.5] (3,15) -- (10,10) -- (13,5) --
\end{tikzpicture}% pic 2
```

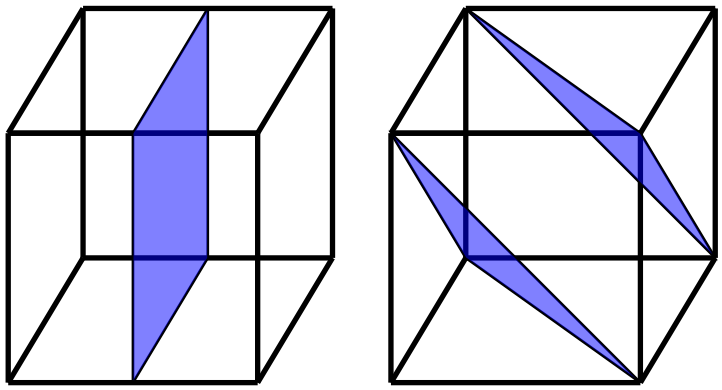


Figure 2.3: Two cubes and 3 planes.

Chapter 3

Desenhos de Negócios

You can draw a triangle by trial and error, as I did to find the best point for putting the top edge in Figure 3.1, with single command `\draw`, as shown in the following code.

```
\begin{tikzpicture}
  \draw (-2,0) node[anchor=north]{\Large Prazo}   -- (2,0) node[anchor=north]{\Large Custo}  -- (
\end{tikzpicture}
```

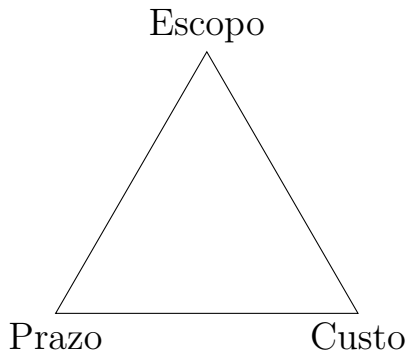


Figure 3.1: Triangle

Another figure done by deciding where the edges should be *a priori*, resulting in Figure 3.2. In this code, there is a good example of using `foreach` to achieve a result. Also, the use of `intersections` to calculate where a point is.

```
\begin{tikzpicture}

\coordinate (A) at (-3.5,0) {};
\coordinate (B) at ( 3.5,0) {};
\coordinate (C) at (0,6) {};
```

```
\path[name path=AC,draw=none] (A) -- (C);
\path[name path=BC,draw=none] (B) -- (C);

\filldraw[draw=black, ultra thick,fill=white]
(A) -- (B) -- (C) -- cycle ;

\foreach \y/\A in {0/Despejo Controlado,
                   1/Aterro ou Incineração,
                   2/Reciclagem,
                   3/Reuso,
                   4/\parbox{3cm}{\centering
                   Redução}}
    {
    \path[draw=none, very thick, dashed, name
    path=horiz] (A|-0,\y) -- (B|-0,\y);
    \draw[draw=black, very thick, dashed,
          name intersections={of=AC and horiz,by=P},
          name intersections={of=BC and horiz,by=Q}]
          (P) -- (Q)
          node[midway,above,font=\bfseries\scshape,
          color=black] {\A};
}

\node[single arrow,rotate=90,draw=black,minimum
height=6cm] at (-4,3) {melhor opção};

\end{tikzpicture}
```

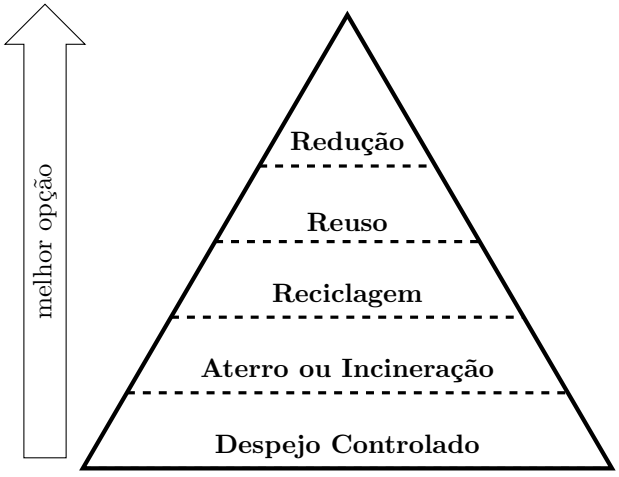


Figure 3.2: A tipical pyramid of concepts.

Chapter 4

Transparences, Shades

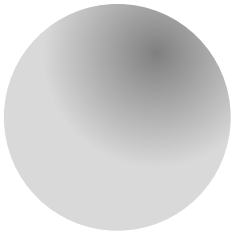


Figure 4.1: Shading, Fdding and Transform

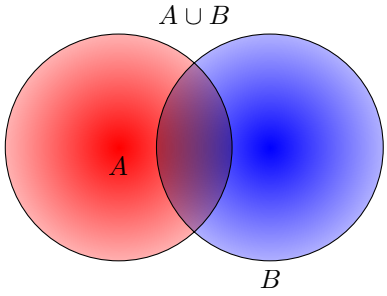


Figure 4.2: Multiplicando cores fading

Chapter 5

Graphs and Trees

In this chapter there are many exemples of using nodes and paths to draw diagrams.

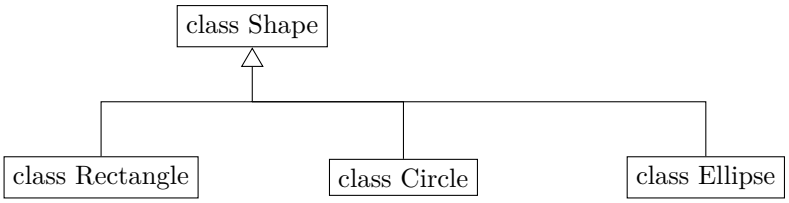


Figure 5.1: Caption

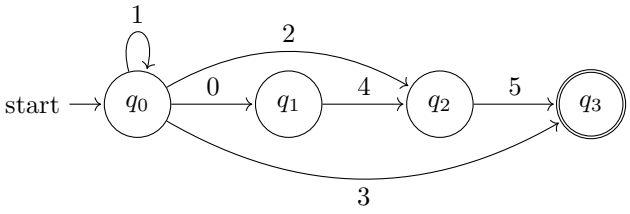


Figure 5.2: Based on the topathas manual.

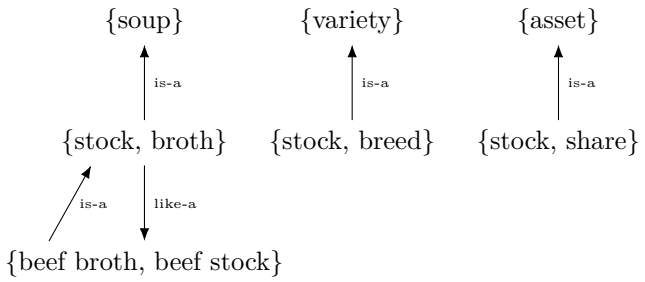


Figure 5.3: Relations from Wordnet

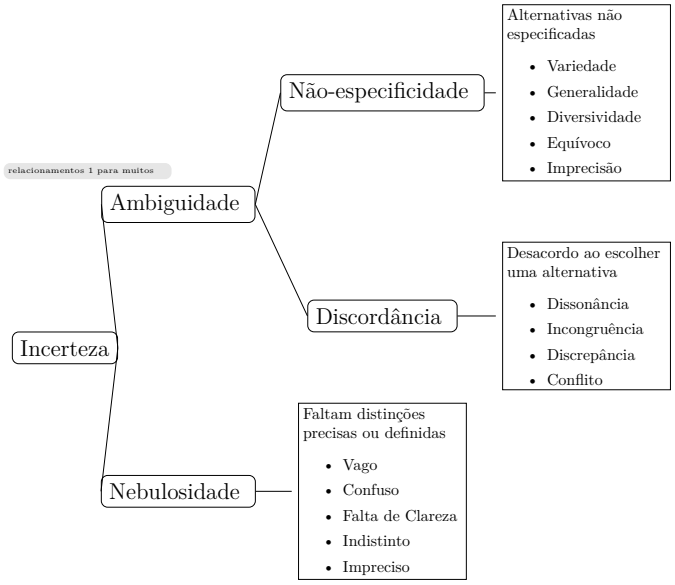


Figure 5.4: Exemplo de uso de Transform Canvas, mas que não deu certo no documento em que foi usado, pois gerou outras mudanças

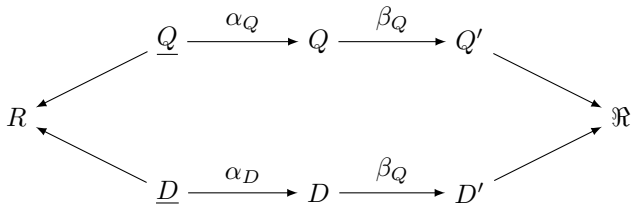


Figure 5.5: Figure from IR, first try, using calculated absolute positions

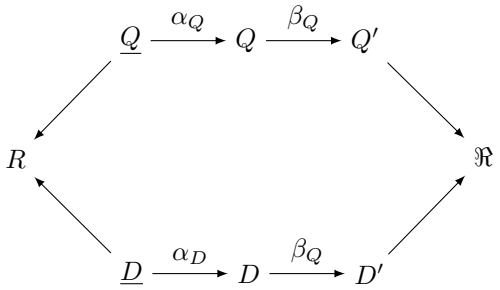


Figure 5.6: Figure from IR, second try, using relative positions.

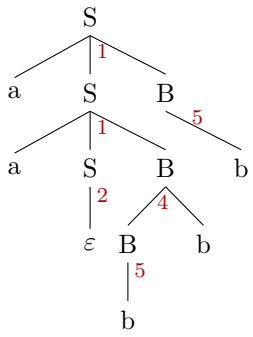
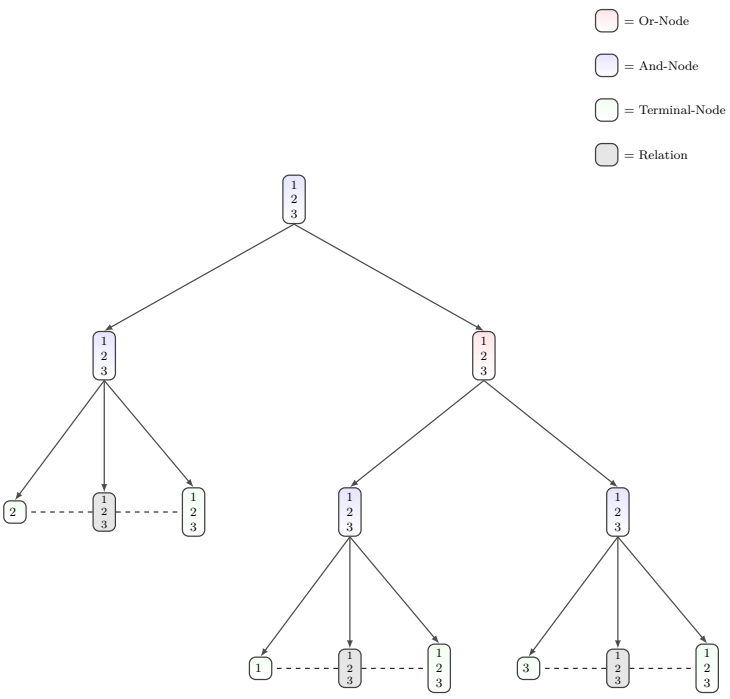


Figure 5.7: This tree is described in https://tex.stackexchange.com/questions/85112/drawing-a-syntax-tree-in-tikz



centering

Figure 5.8: This tree is described in `https://tex.stackexchange.com/questions/123212/tikz-tree-some-childs-without-arrows`

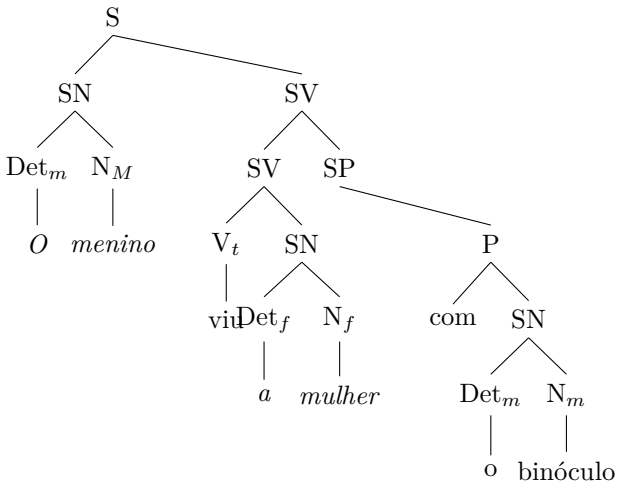


Figure 5.9: Primeira forma de interpretar sintaticamente a sentença “O menino viu a mulher de binóculo”, onde o menino tem o binóculo, baseado em ...

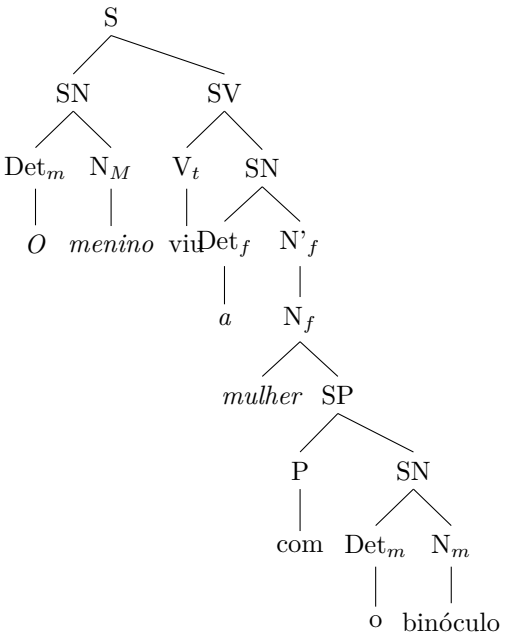


Figure 5.10: Segunda forma de interpretar sintaticamente a sentença “O menino viu a mulher de binóculo”, onde a mulher tem o binóculo, baseado em ...

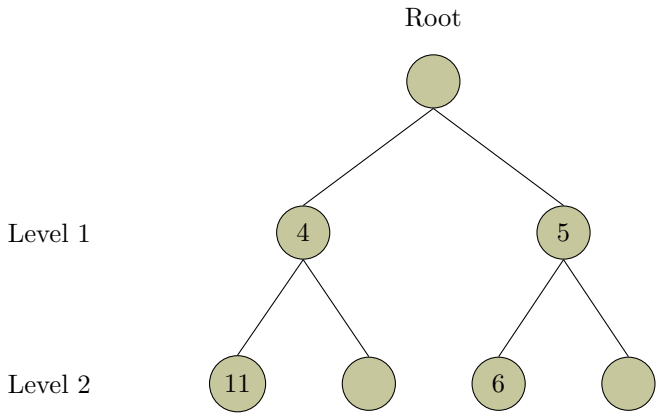


Figure 5.11: https://tex.stackexchange.com/questions/153598/how-to-draw-empty-nodes-in-tikz-qtree

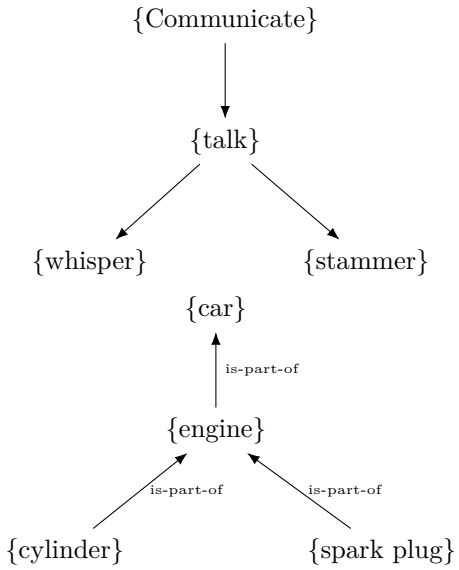


Figure 5.12: Wordnet

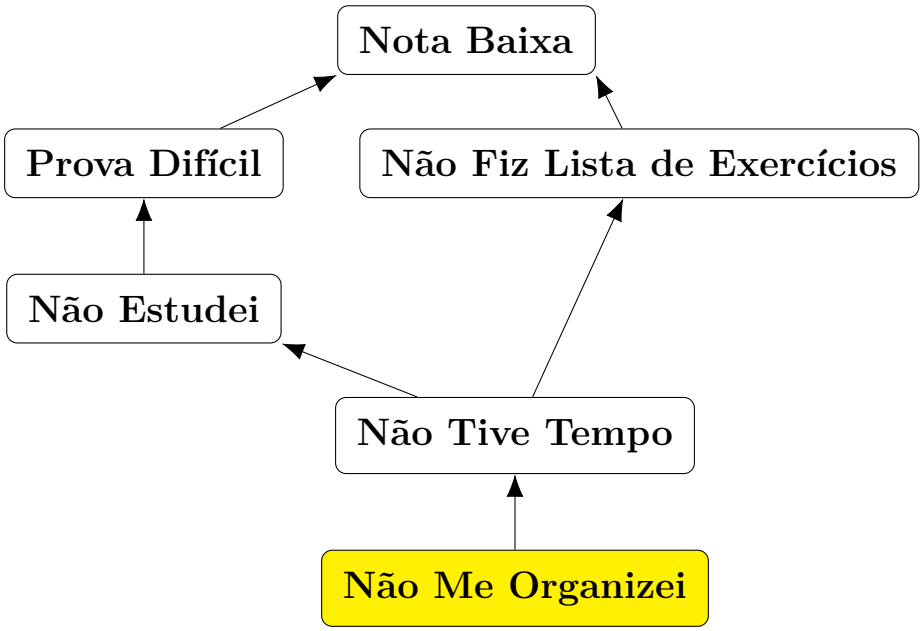


Figure 5.13: Grafo - Diagrama de Causas Raiz Vertical

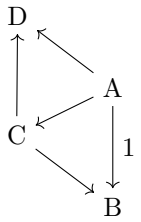


Figure 5.14: Auto layout

5.1 Nodes With Lists Inside

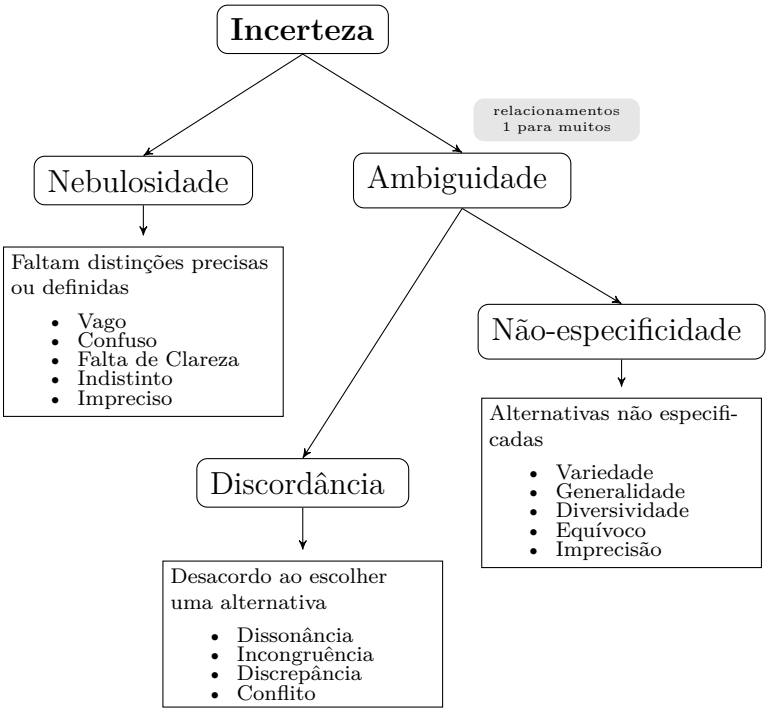


Figure 5.15: Graph Tree

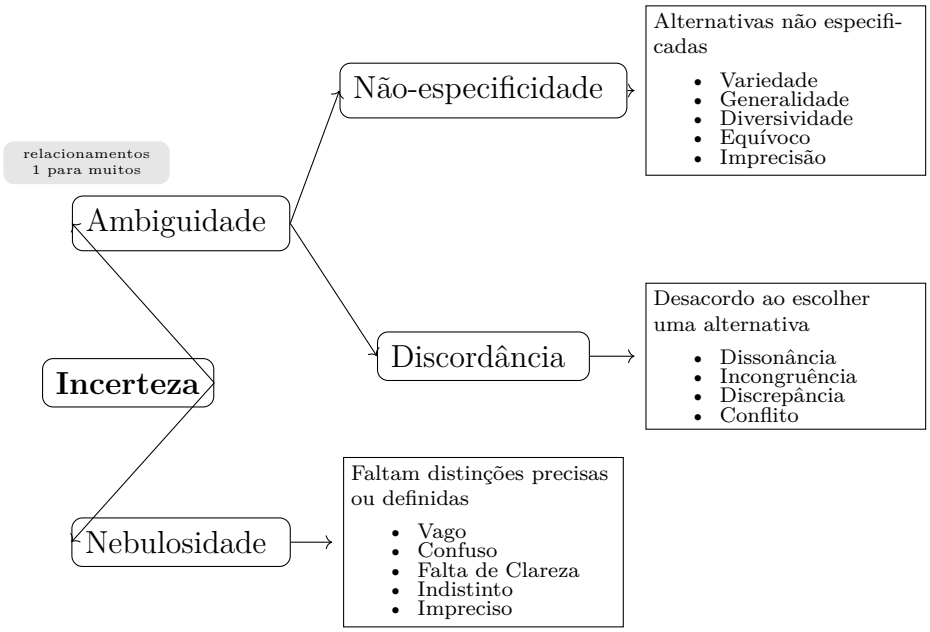


Figure 5.16: Caption

5.2 Positioning and TikzMath

5.3 Using arrays and foreach

The next figure describes a vector where each cell points to a linked list. It is first necessary to define the styles of cells and links:

```
\tikzset{
node of list/.style = {
             draw,
             fill=orange!20,
             minimum height=6mm,
             minimum width=6mm,
             node distance=6mm
   },
link/.style = {
     -stealth,
     shorten >=1pt
     },
array element/.style = {
    draw, fill=white,
    minimum width = 6mm,
    minimum height = 10mm
```

```
  }
}
```

Then, we will use a command that builds a linked list using `foreach`.

```
\def\LinkedList#1{%
  \foreach \element in \list {
     \node[node of list, right = of aux, name=ele] {\element};
     \node[node of list, name=aux2, anchor=west] at ([xshift=-.4pt] ele.east) {};
     \draw[link] (aux) -- (ele);
     \coordinate (aux) at (aux2);
   }
   \fill (aux) circle(2pt);
}
```

Finally, the following code results in Figure 5.17.

```
\begin{tikzpicture}
\foreach \index/\list in {
.2/{(3,11),(16,24),null},
.4/{(4,10),(17,23),null},
.6/{(5,9),(18,22),null},
.8/{(6,8),(19,21),null},
1/{7,20,null}} {
   \node[array element] (aux) at (0,-\index*5) {\index};
   \LinkedList{\list}
}
\end{tikzpicture}
```

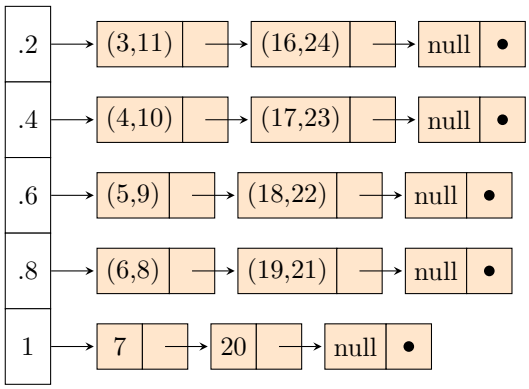


Figure 5.17: This figure uses a pre-defined commando to draw a linked list.

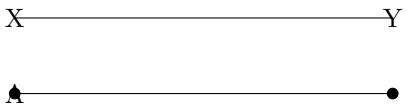


Figure 5.18: Exemplo de somar posições

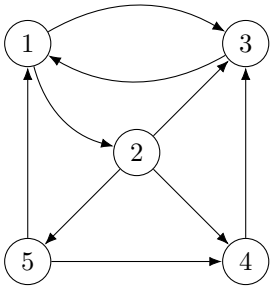


Figure 5.19: Grafo

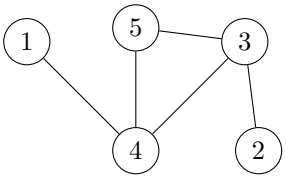


Figure 5.20: Grafo Co Citação

```
 \begin{figure}
\centering
\shorthandoff{"}
\tikz \graph {
a ->["x"] b ->["y"'] c ->["z" red] d;
};
    \caption{Bug na interação do BABEL com o TIKZ precisa desligar o aspas com shortha
    \label{fig:my_labeldasd}
\end{figure}
```

$$a \xrightarrow{x} b \xrightarrow[y]{} c \xrightarrow{z} d$$

Figure 5.21: Bug na interação do BABEL com o TIKZ precisa desligar o aspas com shorthandoff

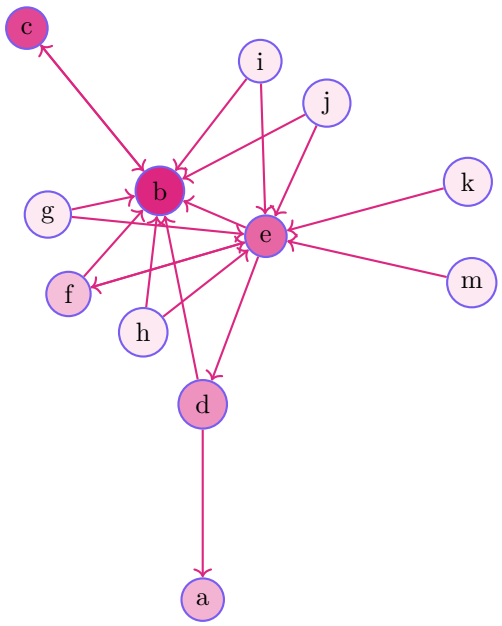


Figure 5.22: Grafo com força spring e variações elétricas

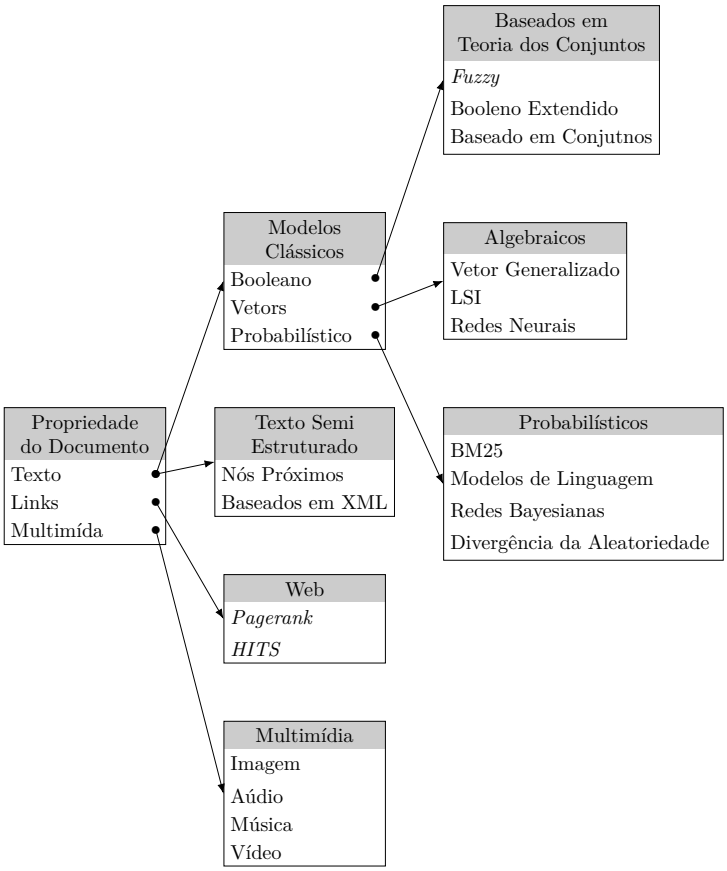


Figure 5.23: Taxonomia de Modelos de IR com posicionamento relativo

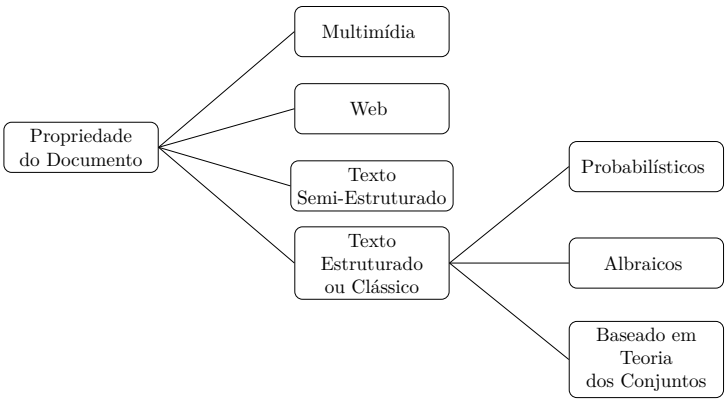


Figure 5.24: Taxonomia do Modelos de IR com child nodes

Chapter 6

Cartesian Coordinates

6.1 Grids

The following code allows for the creation of Figure 6.1 and Figure 6.2

```
\makeatletter
\def\grd@save@target#1{%
  \def\grd@target{#1}}
\def\grd@save@start#1{%
  \def\grd@start{#1}}
\tikzset{
  grid with coordinates/.style={
    to path={%
      \pgfextra{%

        \edef\grd@@target{(\tikztotarget)}%
        \tikz@scan@one@point\grd@save@target\grd@@target\relax
        \edef\grd@@start{(\tikztostart)}%
        \tikz@scan@one@point\grd@save@start\grd@@start\relax

       \draw[minor help lines] (\tikztostart) grid (\tikztotarget);

        \draw[major help lines] (\tikztostart) grid (\tikztotarget);

        \grd@start

        \pgfmathsetmacro{\grd@xa}{\the\pgf@x/1cm}
        \pgfmathsetmacro{\grd@ya}{\the\pgf@y/1cm}

        \grd@target

        \pgfmathsetmacro{\grd@xb}{\the\pgf@x/1cm}
```

```
        \pgfmathsetmacro{\grd@yb}{\the\pgf@y/1cm}

        \pgfmathsetmacro{\grd@xc}{\grd@xa + \pgfkeysvalueof{/tikz/grid with coordinates

        \pgfmathsetmacro{\grd@yc}{\grd@ya + \pgfkeysvalueof{/tikz/grid with coordinates

        \foreach \x in {\grd@xa,\grd@xc,...,\grd@xb}
        \node[anchor=north] at (\x,\grd@ya) {\pgfmathprintnumber{\x}};

        \foreach \y in {\grd@ya,\grd@yc,...,\grd@yb}
        \node[anchor=east] at (\grd@xa,\y) {\pgfmathprintnumber{\y}};
      }
    }
  },
  minor help lines/.style={
    help lines,
    step=\pgfkeysvalueof{/tikz/grid with coordinates/minor step}
  },
  major help lines/.style={
    help lines,
    line width=\pgfkeysvalueof{/tikz/grid with coordinates/major line width},
    step=\pgfkeysvalueof{/tikz/grid with coordinates/major step}
  },
  grid with coordinates/.cd,
  minor step/.initial=.2,
  major step/.initial=1,
  major line width/.initial=2pt,
}
\makeatother
```

The first example of grid construction generates Figure 6.1.

```
\begin{tikzpicture}

  \draw(-1,-1) to[grid with coordinates,grid with coordinates/major line width=1pt] (3

\end{tikzpicture}
```

The second exame uses the same macro defined before, changing line widths, resulting in Figure 6.2

```
\begin{tikzpicture}

  \draw(-2,-2) to[grid with coordinates,grid with
  coordinates/major line width=2pt,grid with
  coordinates/major step=.5,grid with
  coordinates/minor step=0.1] (3,3);
```

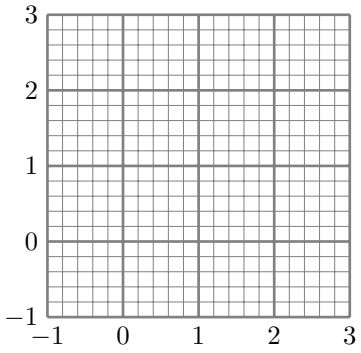


Figure 6.1: Grid with Coordinates

```
\end{tikzpicture}
```

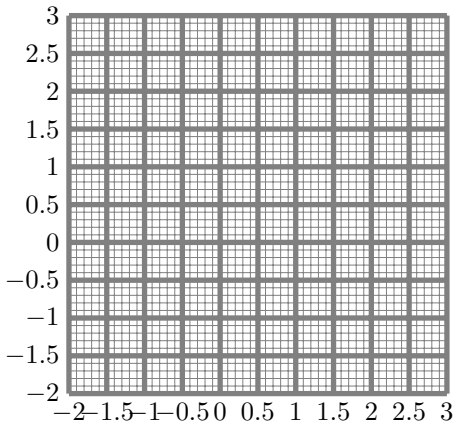


Figure 6.2: Grid with Coordinates, another use

6.2 Simple Figures

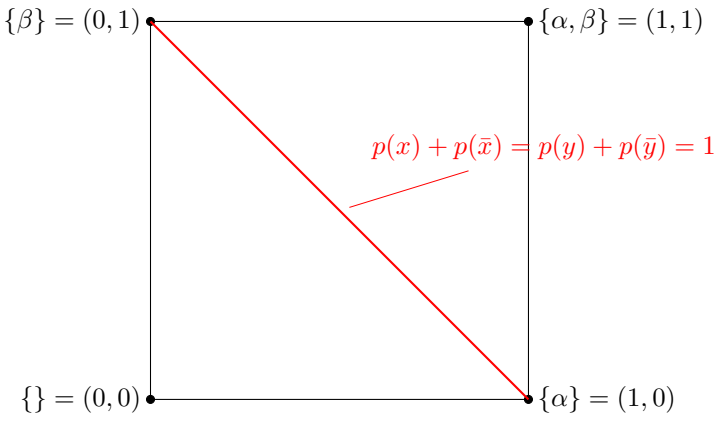


Figure 6.3: ERRADA!!!!!

6.3 Axis and Vectors

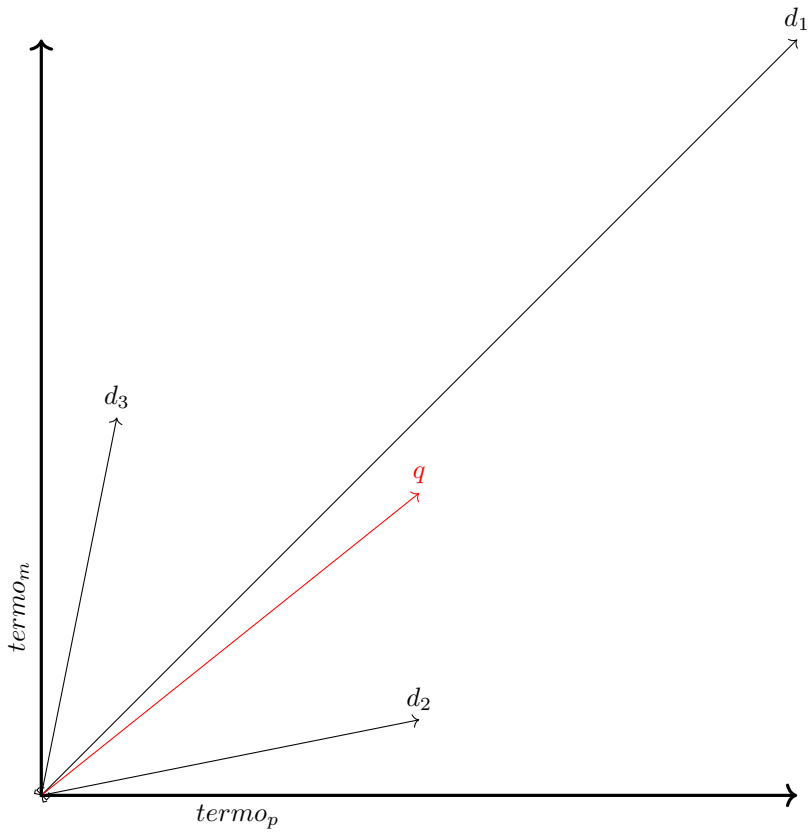


Figure 6.4: Regras fuzzy funcionam como especificação de pedaços das funções sendo agregadas

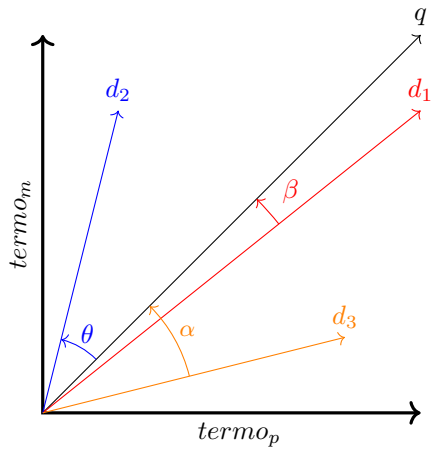


Figure 6.5: Usando o coseno dos vetores.

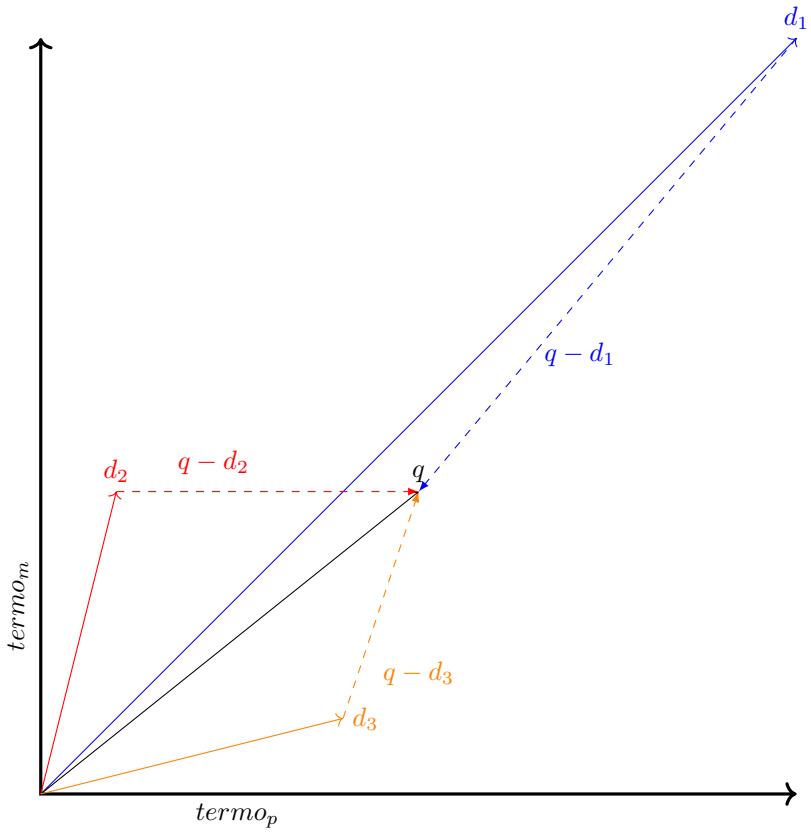


Figure 6.6: Exemplo de problema com o uso do tamanho dos vetores

6.4 Real 3D

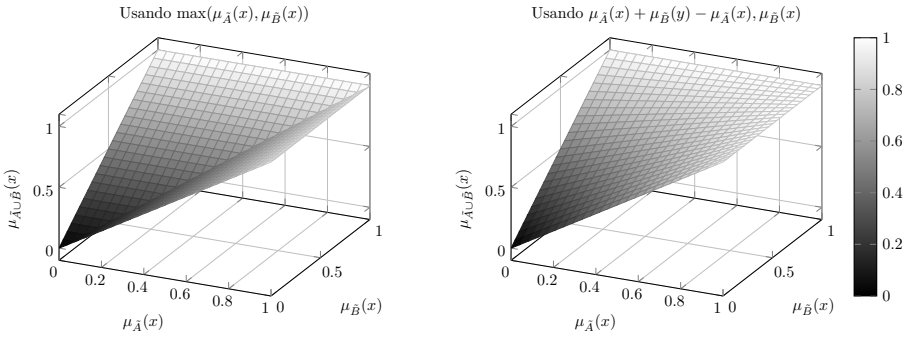


Figure 6.7: Desenho de 3D (real) a partir de funções.

6.5 More difficult drawing mixing nodes and curves

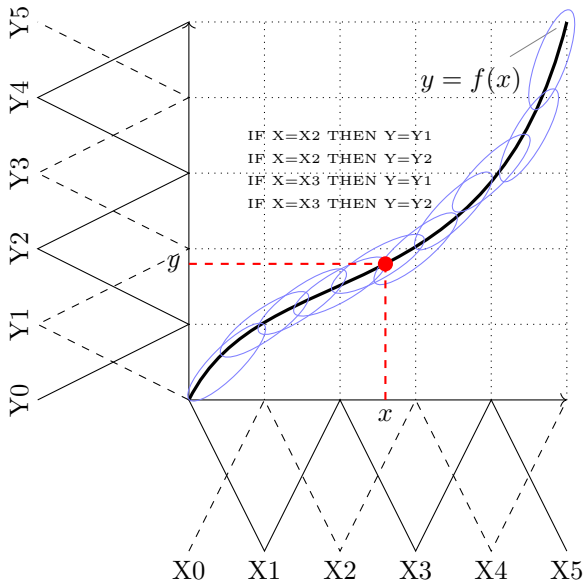


Figure 6.8: Regras fuzzy funcionam como especificação de pedaços das funções sendo agregadas

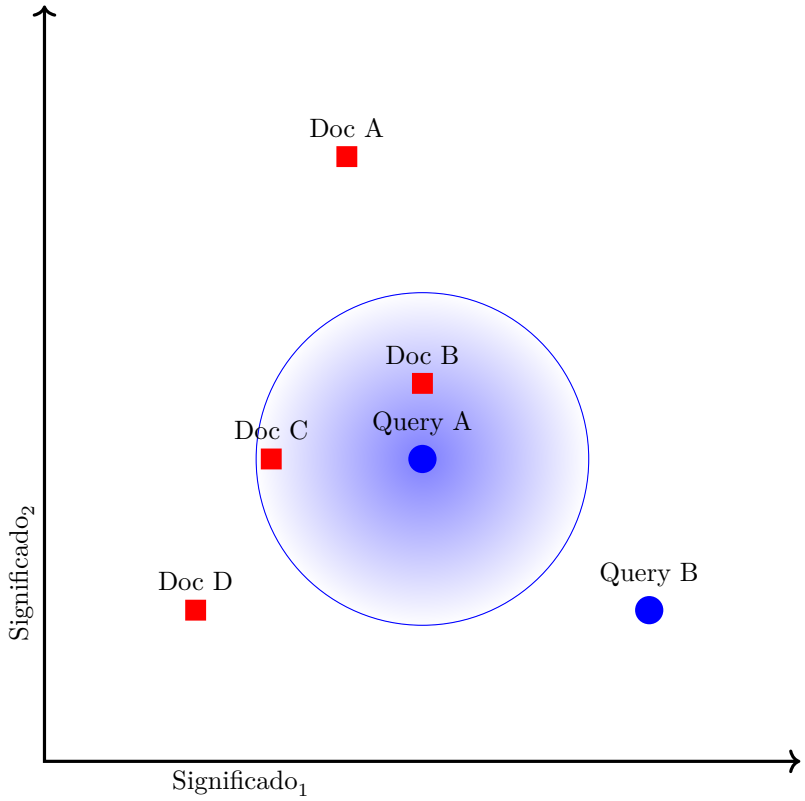


Figure 6.9: Ideia do Modelo Vetorial

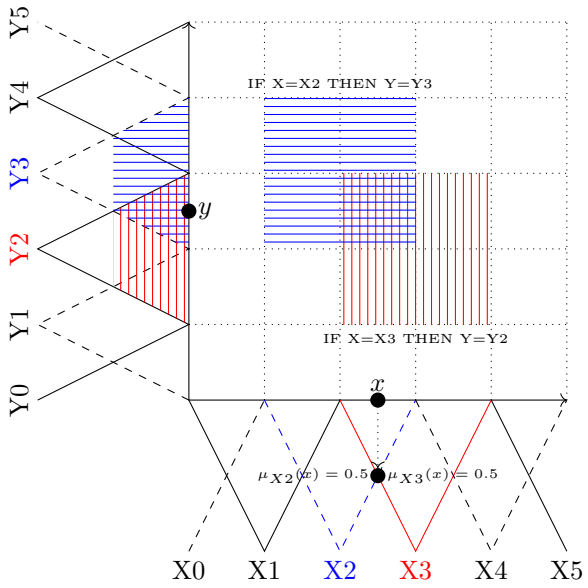


Figure 6.10: Duas regras ativadas simultaneamente de um conjunto de regras, a partir de uma entrada x, são agregadas e uma função de defuzzificação, como o centróide, é usada para determinar y

6.6 1D

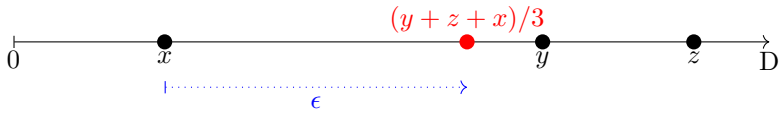


Figure 6.11: Visão gráfica da medida simples de concordância para três pontos, x basicamente em desacordo, considerando um contra a média de todos

Figure 6.12: Visão gráfica da medida simples de concordância para três pontos, x basicamente em acordo

6.7 Desenhos de Fuzzy

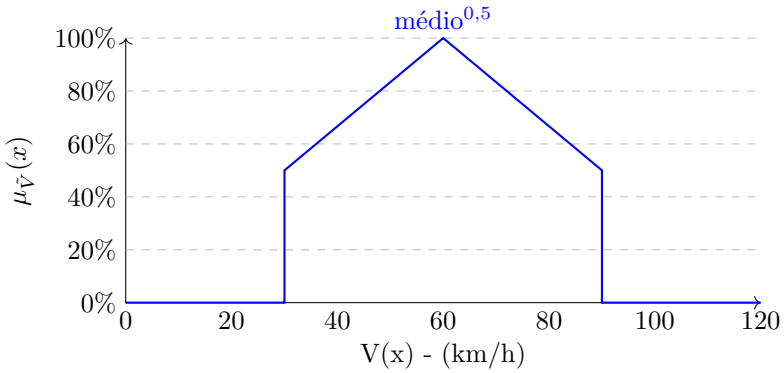


Figure 6.13: O conjunto referente ao corte-alfa de médio com $\alpha = 0,5$, ou seja médio^{0,5}.

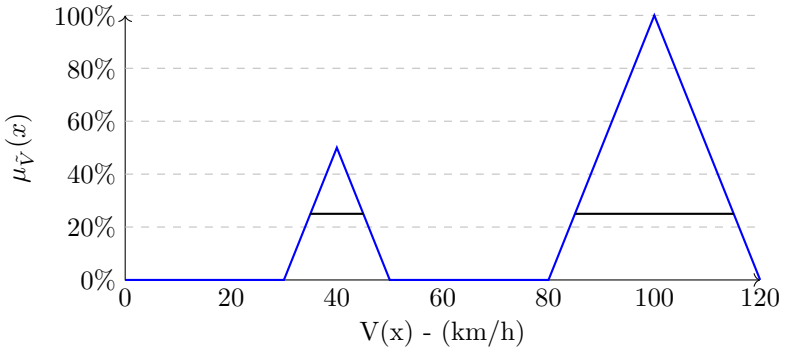


Figure 6.14: Exemplo de cortes-α

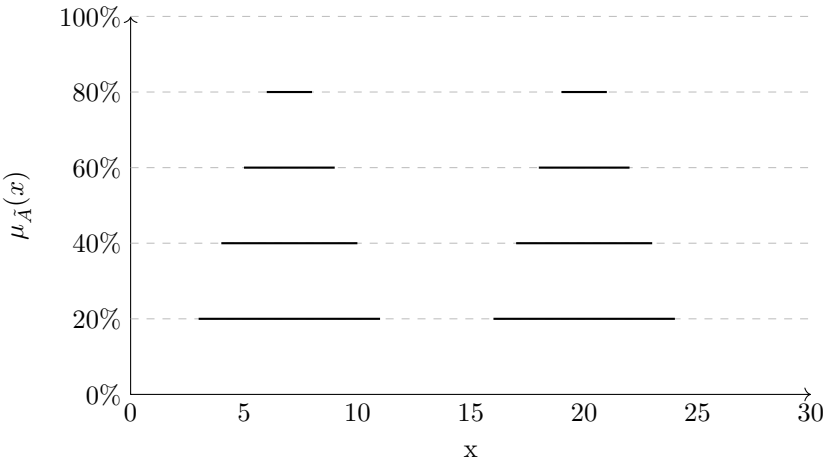


Figure 6.15: Representação dos cortes-α do conjunto $\tilde{A}$ dos números perto de 7 ou 20

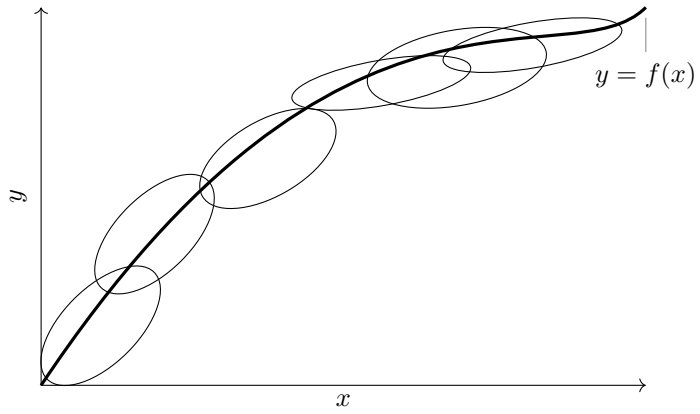


Figure 6.16: Grafico de $y = f(x)$

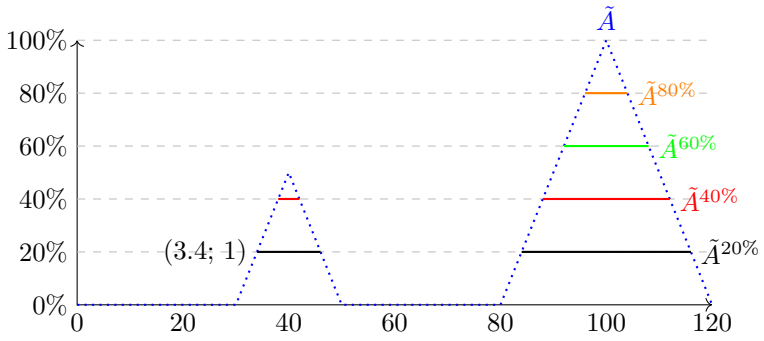
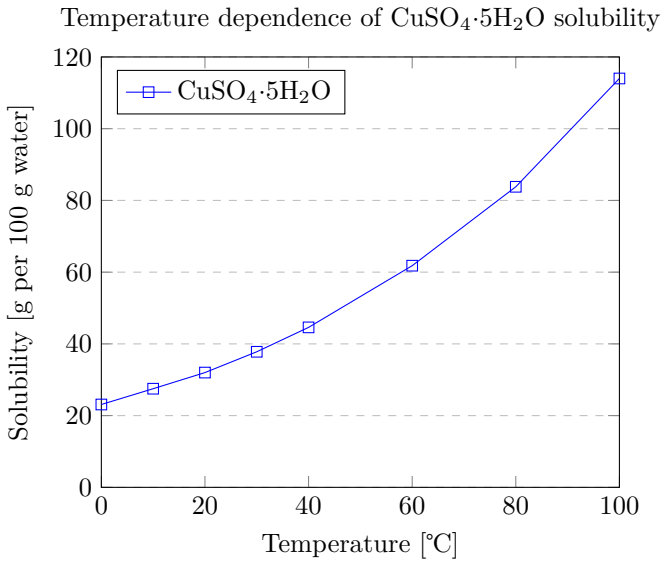
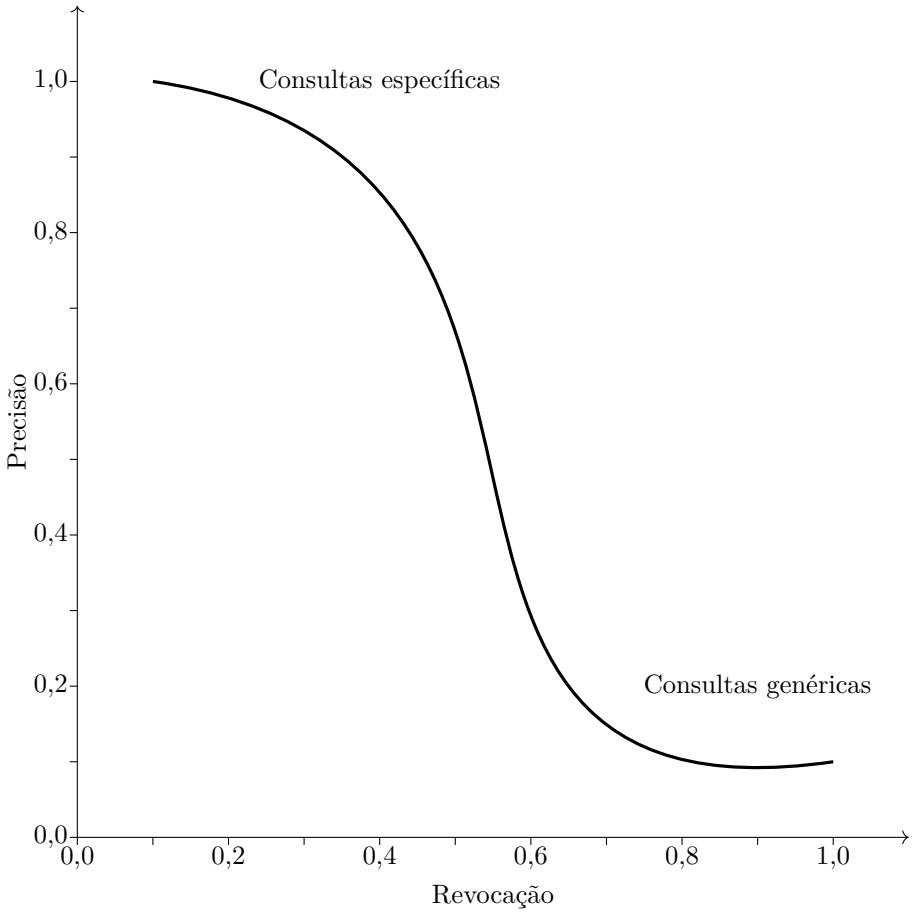


Figure 6.17: Exemplo de cortes-α, onde as linhas horizontais indicam os valores do eixo das abcissas que pertencem ao conjunto nítido correspondente

6.8 Gráficos a partir de números




Consultas específicas
Consultas genéricas
Precisão
Revocação
1,0
0,8
0,6
0,4
0,2
0,0
0,0
0,2
0,4
0,6
0,8
1,0

Chapter 7

Circle Magic

Pie charts are very easy!

```
\begin{tikzpicture}
\pie{24/SAP, 12/Oracle,
6/Sage, 6/Infor, 5/Microsoft,
47/Outros}
\end{tikzpicture}
```

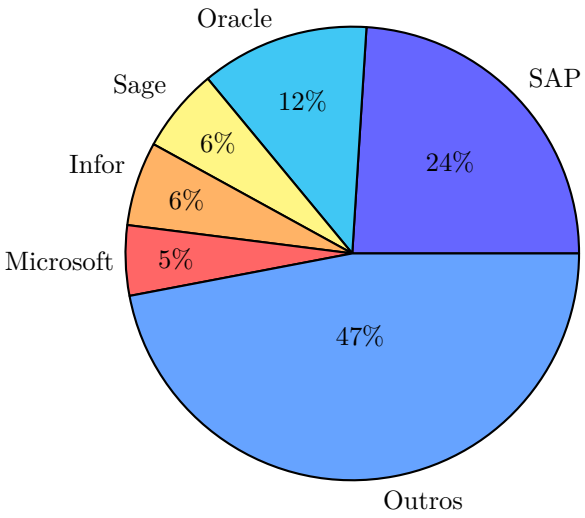


Figure 7.1: ERP Market in 2013

Using polar coordinates it is easy to draw a circle. The following code results in Figure 7.2.

```
\begin{tikzpicture}
```

```
\coordinate (center) at (1,2);
\def\radius{2.5cm}
% a circle
\draw[dotted] (center) circle[radius=\radius];

\fill[black] (center) ++(0:\radius)
circle[radius=4pt] node[black,right] {1} ;

\fill[red] (center) ++(36:\radius)
circle[radius=2pt] node[right] {2 Malala's Call};

\fill[red] (center) ++(2*36:\radius)
circle[radius=2pt] node[above] {3};

\fill[blue] (center) ++(3*36:\radius)
circle[radius=2pt] node[above] {4};

\fill[red] (center) ++(4*36:\radius)
circle[radius=2pt] node[left] {5};

\fill[red] (center) ++(5*36:\radius)
circle[radius=2pt] node[left] {6};

\fill[blue] (center) ++(6*36:\radius)
circle[radius=2pt] node[left] {7};

\fill[red] (center) ++(7*36:\radius)
circle[radius=2pt] node[below] {8};

\fill[red] (center) ++(8*36:\radius)
circle[radius=2pt] node[below] {9};

\fill[blue] (center) ++(9*36:\radius)
circle[radius=2pt] node[right] {10};

 \draw[-{>[scale=2.5,
        length=2,
        width=3]}]  (center)+(4*36:\radius) --
        +(9*36:\radius) ;
 \draw[-{>[scale=2.5,
        length=2,
        width=3]}]  (center)+(2*36:\radius) --
        +(9*36:\radius) ;
 \draw[-{>[scale=2.5,
        length=2,
        width=3]}]  (center)+(0:\radius) --
```

```
        +(9*36:\radius) node [midway, right] {$w$};

   \draw[-{>[scale=2.5,
        length=2,
        width=3]}]  (center)+(7*36:\radius) --
        +(3*36:\radius) ;
   \draw[-{>[scale=2.5,
        length=2,
        width=3]}]  (center)+(5*36:\radius) --
        +(3*36:\radius) ;
   \draw[-{>[scale=2.5,
        length=2,
        width=3]}]  (center)+(0:\radius) --
        +(3*36:\radius) node [midway, below] {$w$};

  \draw[-{>[scale=2.5,
        length=2,
        width=3]}]  (center)+(1*36:\radius) --
        +(6*36:\radius) ;
   \draw[-{>[scale=2.5,
        length=2,
        width=3]}]  (center)+(8*36:\radius) --
        +(6*36:\radius) ;
   \draw[-{>[scale=2.5,
        length=2,
        width=3]}]  (center)+(0:\radius) --
        +(6*36:\radius) node [midway, below] {$w$};


\end{tikzpicture}
```

Another example of using polar coordinates to draw a circle results in Figure 7.3.

```
\begin{tikzpicture}
% posicao central do circulo
  \coordinate (center) at (1,2);
% coloca o nome aqui
\def\nome{Campbell's Hero Journey}
% nome fica no centro
  \node[align=center,text width=4cm,anchor=center] at (center) {\baselineskip=16pt \Huge{\nome}\p
% raio do circulo
  \def\radius{4cm}
% numero de pontos
  \def\passos{10}
% tamanho em angulo graus do passo
```

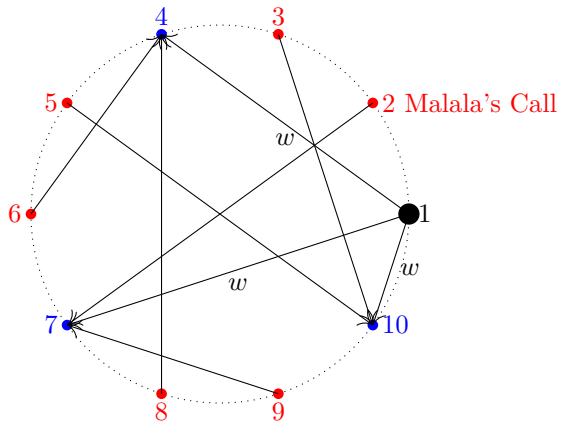


Figure 7.2: Points and chords in a circle.

```
  \def\passo{360/\passos}

% em vez de círculo, podemos usar um arco
% aqui tem um truque, que é usar o shift para
% o primeiro valor que você usar no nosso
% "loop aberto", no caso o 2*\passo
%

\draw[black] ([shift=(2*\passo:\radius)]center) arc (2*\passo:-7*\passo:\radius);

% cada ponto é um fill

% tem que acertar para cada ponto o multiplicador do passo
% isso deveria ser um for, mas é realmente melhor
% fazer na mão para controlar tudo

  \fill (center) ++(2*\passo:\radius)
   node[above,yshift=1em,xshift=2em] {\textbf{Primeiro Ato}};


  \fill[black] (center) ++(2*\passo:\radius)
  %circle[radius=4pt]
  node[regular polygon, regular polygon sides=3, fill,regular polygon rotate=-
90,minimum width = 11pt,inner sep =0] {}
  node[left,yshift=-.7em ] {1} node[black, right,xshift=.5em,yshift=.3em] {Call to Acti

  \fill[black] (center) ++(1*\passo:\radius)
  circle[radius=2pt] node[left] {2} node[right] {Malala's Call};
```

```
  \fill[black] (center) ++(0*\passo:\radius)
  circle[radius=2pt] node[left] {3} node[right] {Malala's Call};

% SEGUNDO ATO

  \fill (center) ++(-1*\passo:\radius)
   node[above,yshift=.5em,xshift=4.3em] {\textbf{Segundo Ato}};
  \fill[black] (center) ++(-1*\passo:\radius)
  circle[radius=2pt] node[below,right] {Malala's Call}
  node[left] {4};

  \fill[black] (center) ++(-2*\passo:\radius)
  circle[radius=2pt]  node[above] {5} node[below,xshift=3.5em] { Malala's Call};

% TERCEIRO ATO

    \fill (center) ++(-3*\passo:\radius)
   node[below,yshift=-.5em,xshift=-3em] {\textbf{Terceiro Ato}};

  \fill[black] (center) ++(-3*\passo:\radius)
  circle[radius=2pt]  node[above] {6} node[left,xshift=-1em] {Malala's Call};



  \fill[black] (center) ++(-4*\passo:\radius)
  circle[radius=2pt] node[right] {7} node[left] {Malala's Call};

  \fill[black] (center) ++(-5*\passo:\radius)
  circle[radius=2pt] node[right] {8} node[left] {Malala's Call};

  \fill[black] (center) ++(-6*\passo:\radius)
  circle[radius=2pt] node[right] {9} node[left]  {Malala's Call};

  \fill[black] (center) ++(-7*\passo:\radius)
  node[shape=rectangle,fill] {} node[right,yshift=-.5em] {10}  node[above,xshift=-
3.5em] {Malala's Call };

 \end{tikzpicture}
```

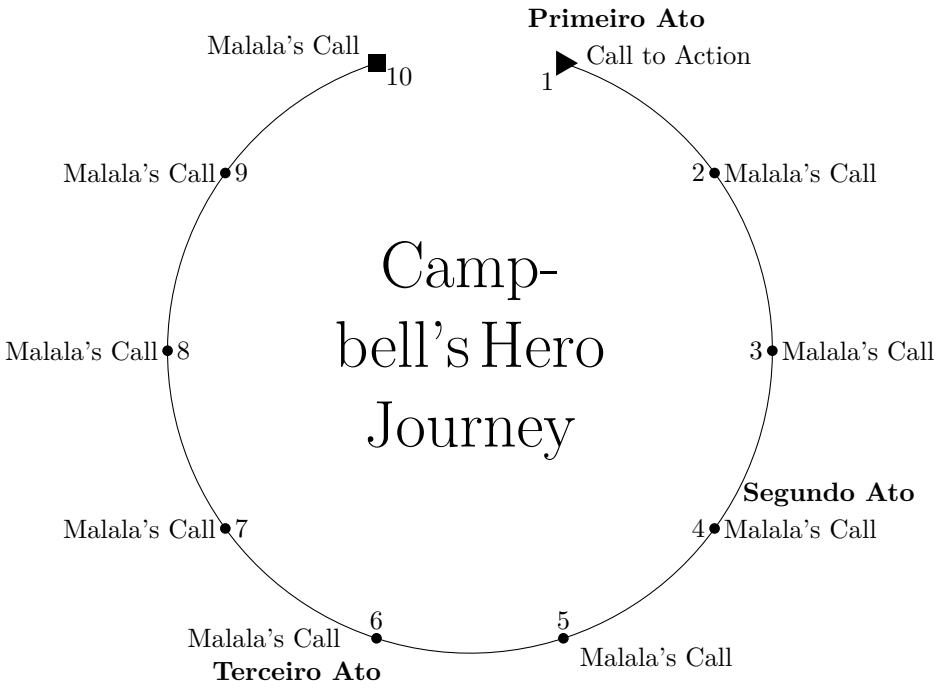


Figure 7.3: The Heroine Learner Journey

Chapter 8

Measurements

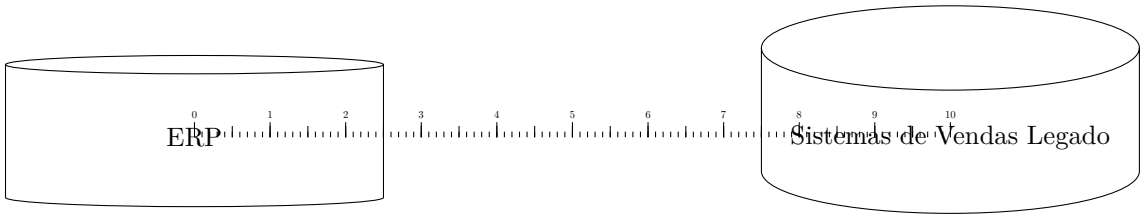


Figure 8.1: Tentando usar cilindros, mas varia com o texto dentro dele

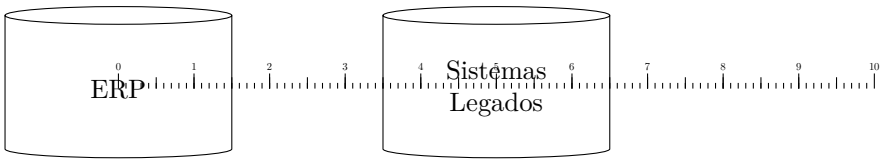


Figure 8.2: Tentando usar cilindros,texto de fora

Chapter 9

Other

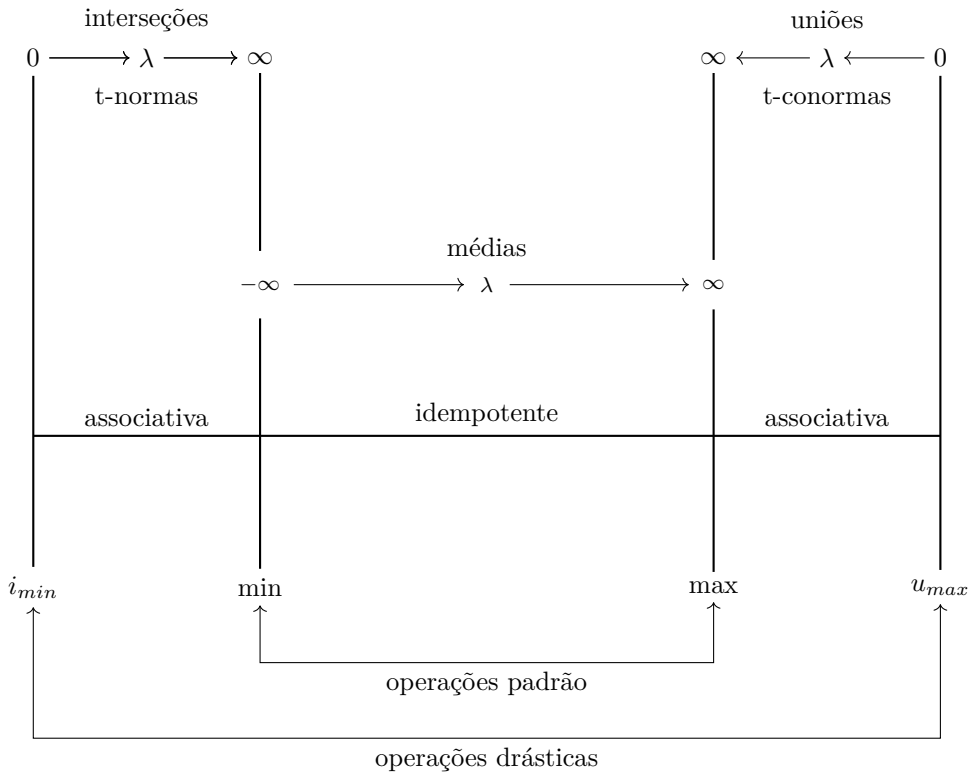


Figure 9.1: Variáveis e posições

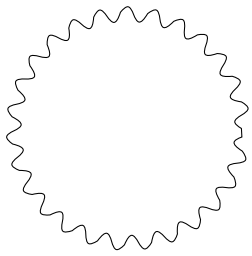


Figure 9.2: Decorations need many libraries (at least)

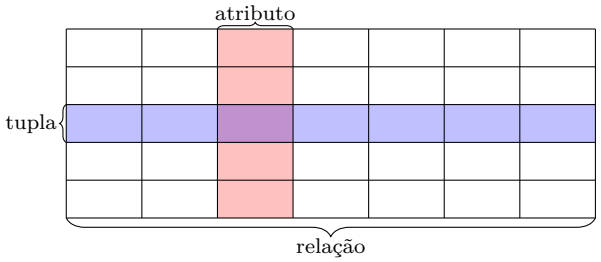


Figure 9.3: Modelo abstrato de uma relação como uma tabela - usando decorations de chaves

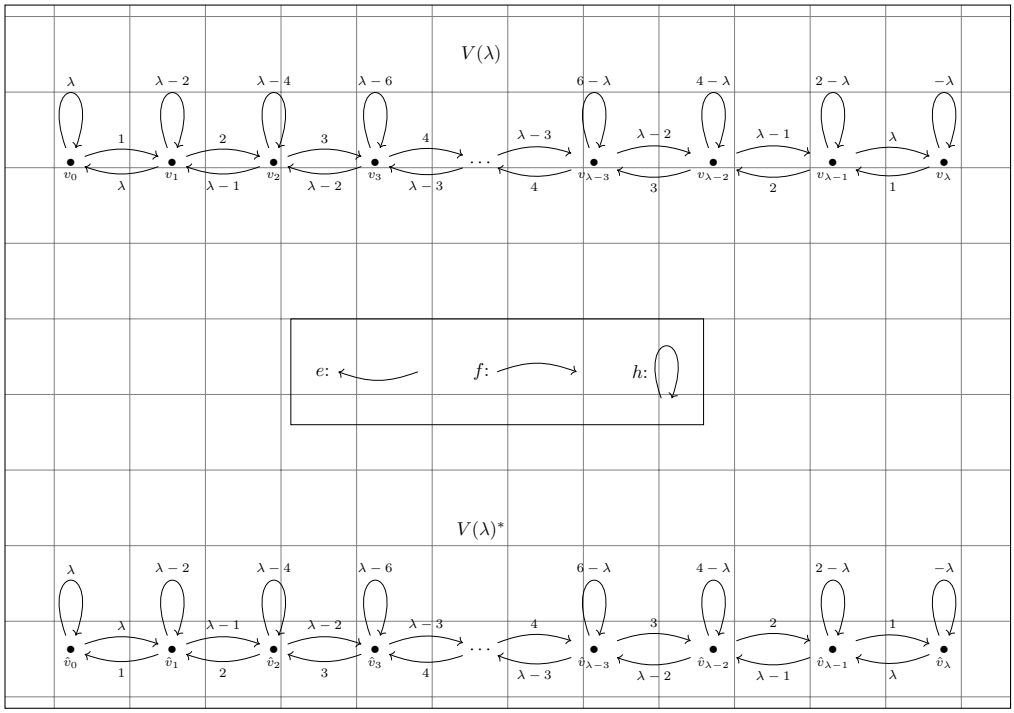


Figure 9.4: The manual states: Tracking of the picture size is (locally) switched off This means that the bounding box is lost, which needs to be specified manually via the useasboundingbox path (= path[use as bounding box]) which also needs to be outside of the scope that has transform canvas applied to. You might consider the necessarity to transform your whole picture (this also affects font-sizes!).

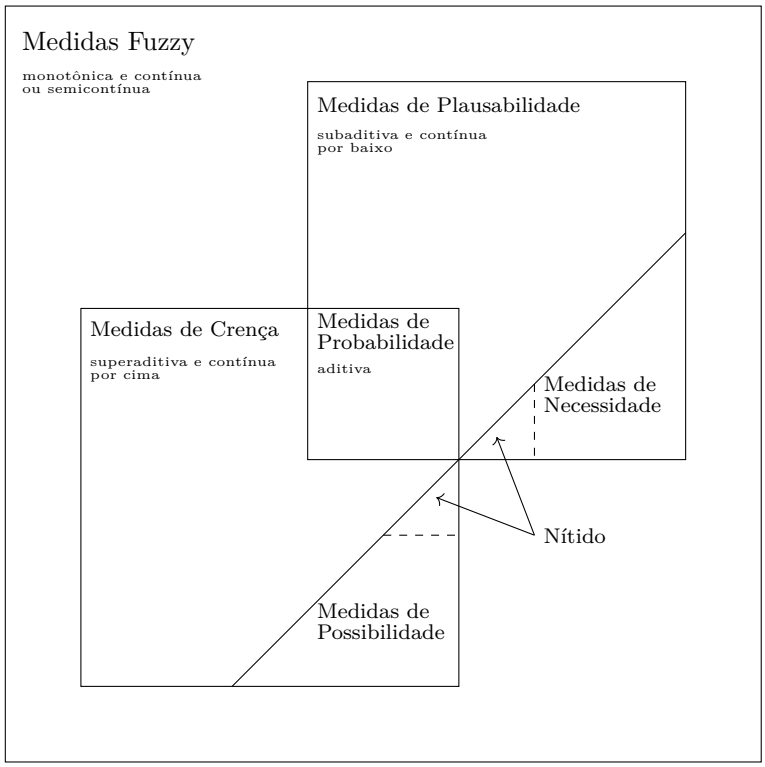


Figure 9.5: Klir e os tipos de medida

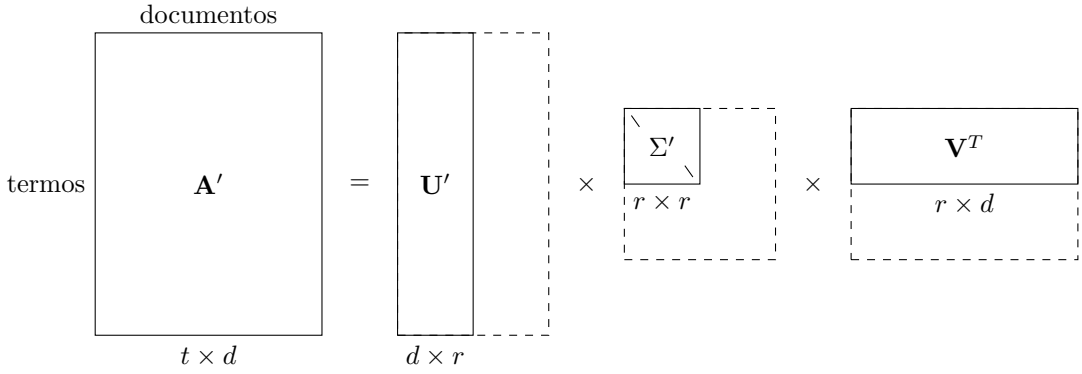


Figure 9.6: LSI

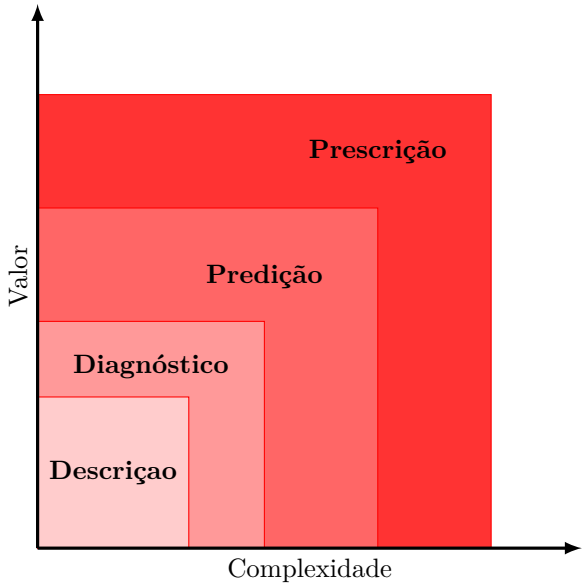


Figure 9.7: Caption

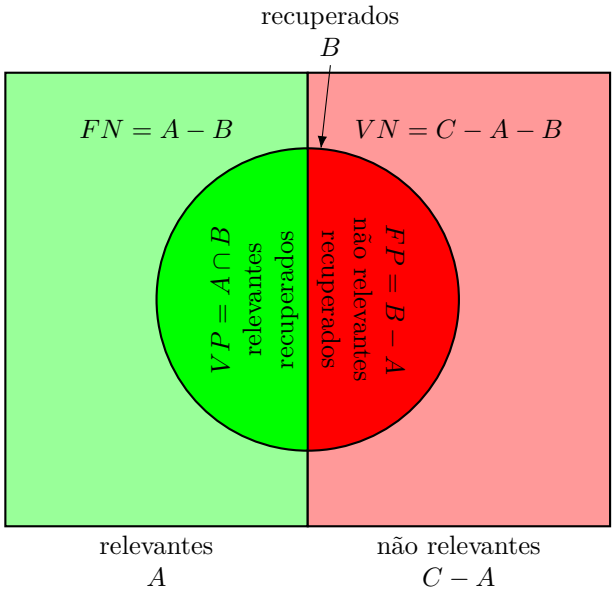


Figure 9.8: Information Retrieval

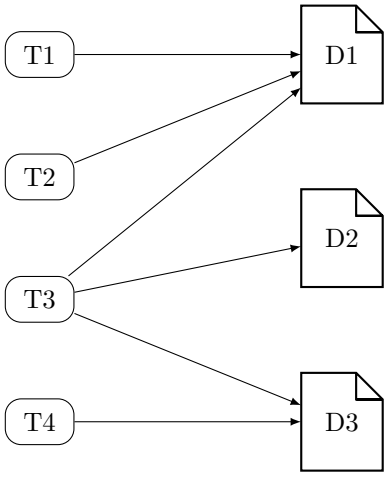


Figure 9.9: Representação de um índice

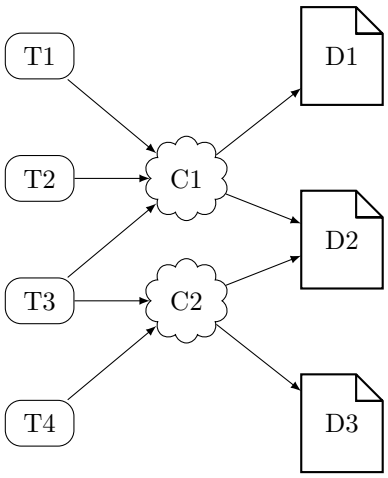


Figure 9.10: Representação do LSI/LSA

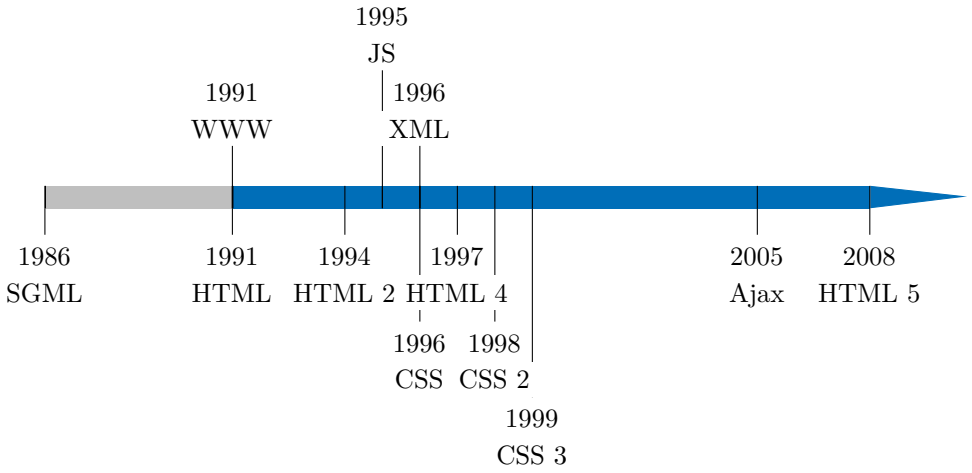


Figure 9.11: Cronologia

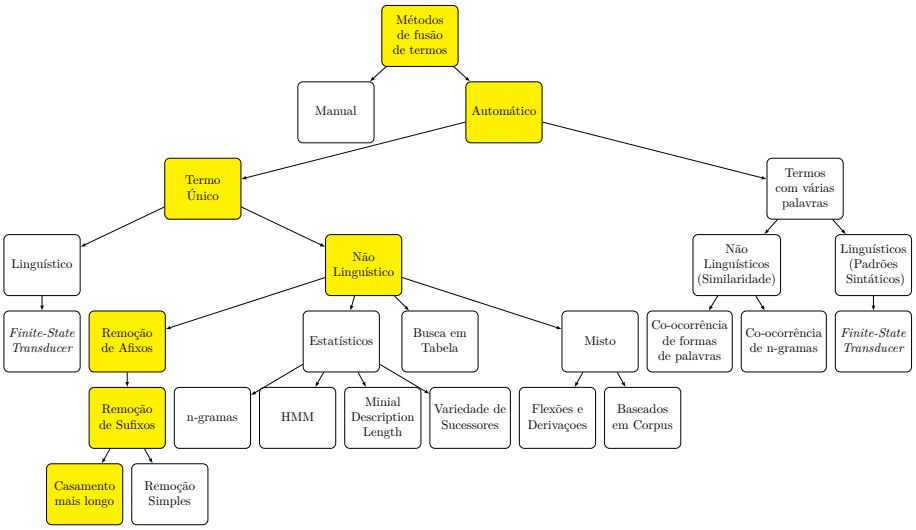


Figure 9.12: Stemmers

Chapter 10

Mais!

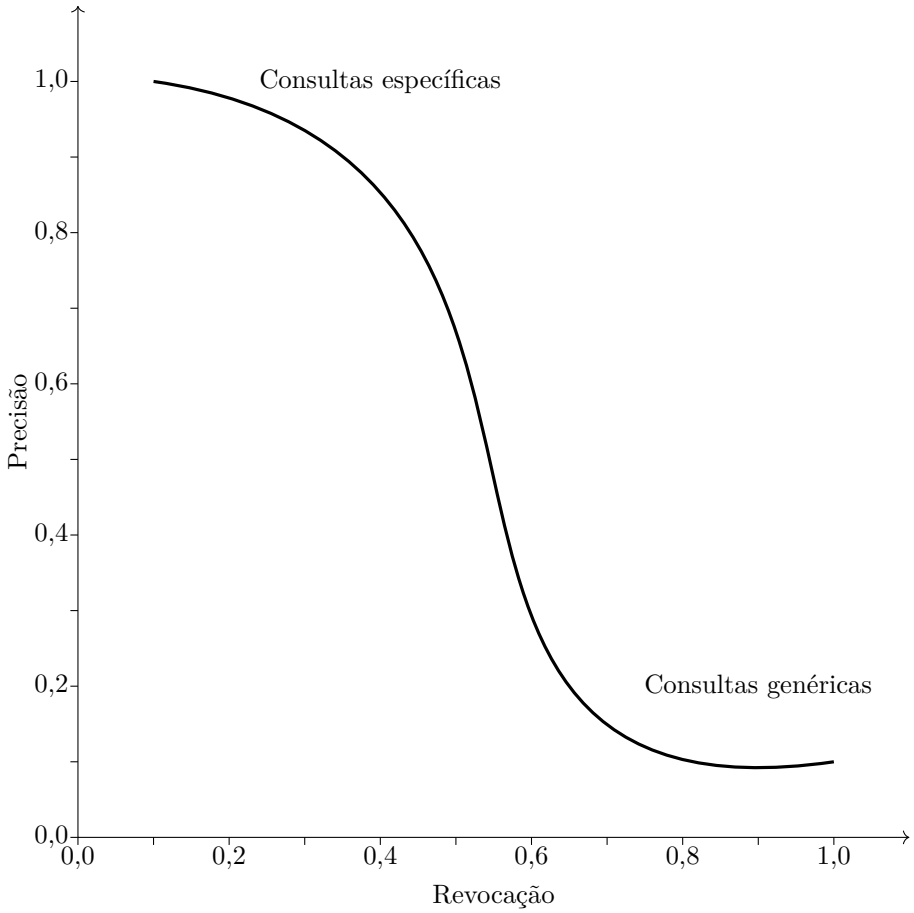


Figure 10.1: Caption

Chapter 11

Arquiteturas

Figure 11.4 uses different interesting commands, such as calculating point position by the intersections of a vertical and a a horizontal reference (using `(h-|v)` or `(v|-h)`, general and specific styles, style overwriting, y and x shifts in points, etc.

```
\begin{tikzpicture}%
[every node/.style={%
draw,%
black,%
align=center,%
node distance=1cm and 3cm,%
minimum height = 1.5cm,%
minimum width = 3cm,
},
registro/.style={%
minimum height = 18pt,%
minimum width = 4cm,
node distance=.5cm and 3cm,%
},
every path/.style={%
black,
Latex-Latex,
thick
}%
]

\node (UdC) at (0,0) {Unidade \\ de Controle};
\node (ALU) [above = of UdC]  {ALU};
\node (B) [above = of ALU]  {Buffer};

\node[registro] (Pilha) [right = of UdC] {Registro de Pilha};
```

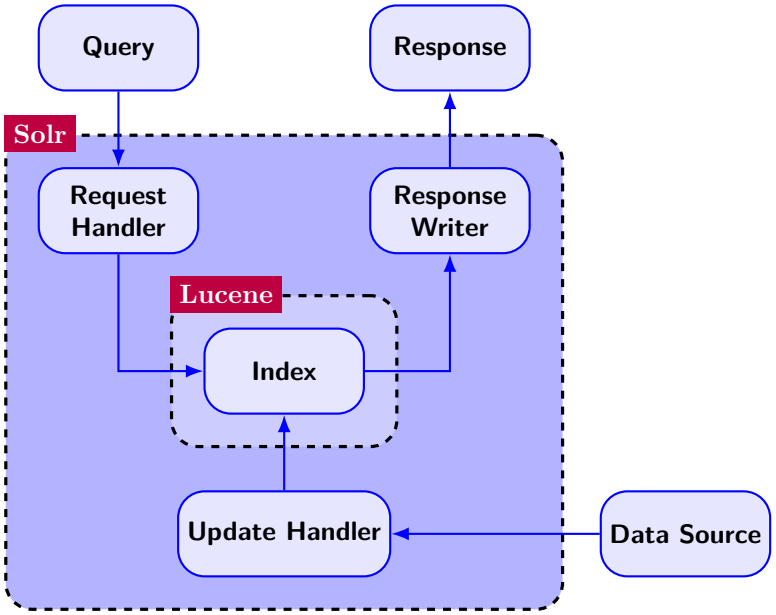


Figure 11.1: Solr Architecture

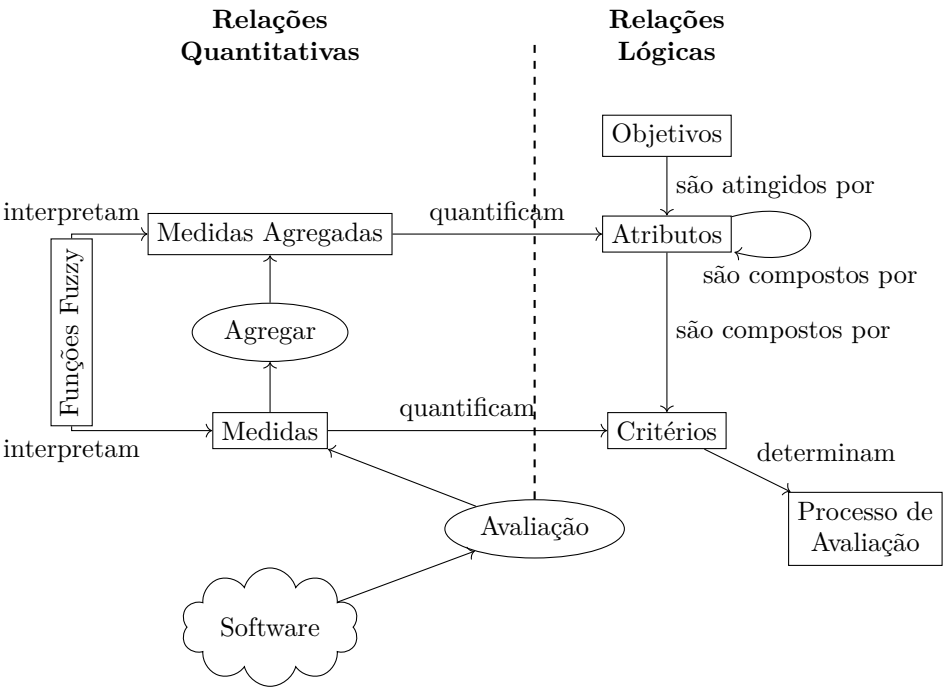


Figure 11.2: Modelo Fuzzy de Qualidade Rocha

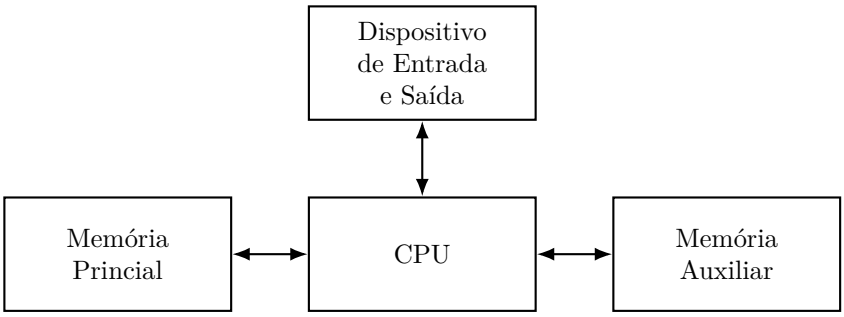


Figure 11.3: Computador Simples, usa estilos genéricos

```
\node[registro] (End) [above = of Pilha] {Registro de Endereço};
\node[registro] (Regn) [above = of End]  {Registro Geral N};
\node[registro,draw=none] (Regp) [above = of Regn]  {...};
\node[registro] (Reg2) [above = of Regp]  {Registro Geral 2};
\node[registro] (Reg1) [above = of Reg2]  {Registro Geral 1};

\node[draw=none] (ini) at  (2,-1) {};
\node[draw=none]  at (2,7) {};
 %($(UdC.south west)!.5!(Pilha.east)$)
\draw[-,ultra thick] ([yshift=-10pt]3,0|-Pilha.west) -- ([yshift=10pt]3,0|-
Reg1.west);
\draw (Pilha.west) -- (3,0|-Pilha.west);
\draw (Reg1.west) -- (3,0|-Reg1.west);
\draw (Reg2.west) -- (3,0|-Reg2.west);
\draw (Regn.west) -- (3,0|-Regn.west);
\draw (End.west) -- (3,0|-End.west);
\draw (ALU.east) -- (3,0|-ALU.east);
\draw (UdC.east) -- (3,0|-UdC.east);
\draw (B.east) -- (3,0|-B.east);

\draw[-,ultra thick] ([xshift=15pt,yshift=15pt]Reg1.north)
-- node [pos=0.5,draw=none,above] {Barramento
de Dados} ([xshift=-15pt,yshift=15pt]Reg1.north-|B.north);
\draw ([yshift=15pt]Reg1.north)
-- (Reg1.north) ;
\draw ([yshift=15pt]Reg1.north-|B.north)
-- (B.north) ;

\node[draw=none] (T1) at ([yshift=-10pt]Pilha.south) {Barramento Interno de Dados};
\draw[Latex-,dotted] ([yshift=-10pt]3,0|-Pilha.west) -- (T1.west) ;

\draw[ultra thick,-] ([yshift=15pt,xshift=-15pt]B.west) --
```

```
([yshift=-15pt,xshift=-15pt]UdC.south west) --
node [draw=none,below,pos=0.5] {Barramento de Controle}
([yshift=-15pt,xshift=15pt]Pilha.south east |-
UdC.south west) --
([yshift=15pt,xshift=15pt] Pilha.south east |- B.west)
;

\draw (Pilha.east) -- ([xshift=15pt]Pilha.east -| Pilha.south east);
\draw (Pilha.east) -- ([xshift=15pt]Pilha.east -| Pilha.south east);
\draw (Pilha.east) -- ([xshift=15pt]Pilha.east -| Pilha.south east);
\draw (End.east) -- ([xshift=15pt]End.east -| Pilha.south east);

\draw (UdC.west) -- ([xshift=-15pt]UdC.west -| UdC.south west);
\draw (ALU.west) -- ([xshift=-15pt]ALU.west -| UdC.south west);
\draw (B.west) -- ([xshift=-15pt]B.west -| UdC.south west);

\end{tikzpicture}
```

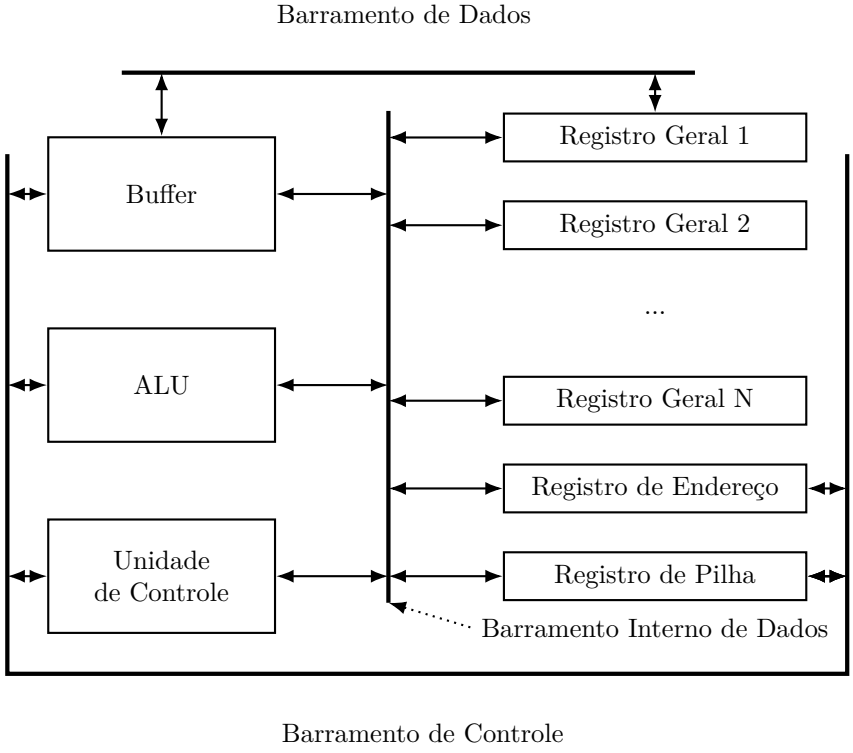


Figure 11.4: Abstração da CPU, usa referências com shift e cálculo de pontos por interseção de uma referência vertical com uma horizontal, estilos genéricos e específicos

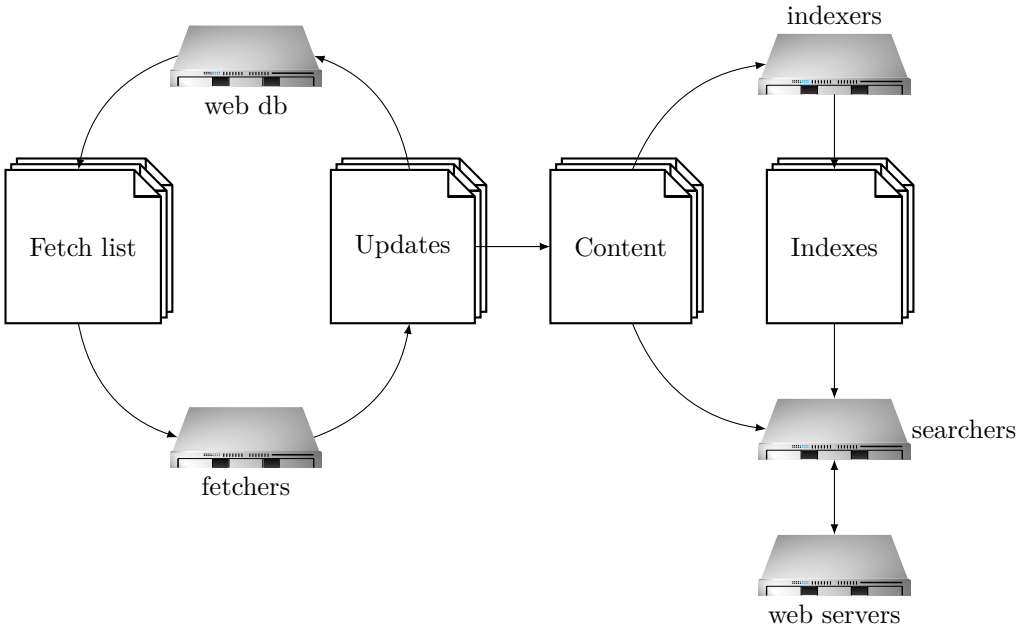


Figure 11.5: Lucene?

11.1 Mecanismos de Busca

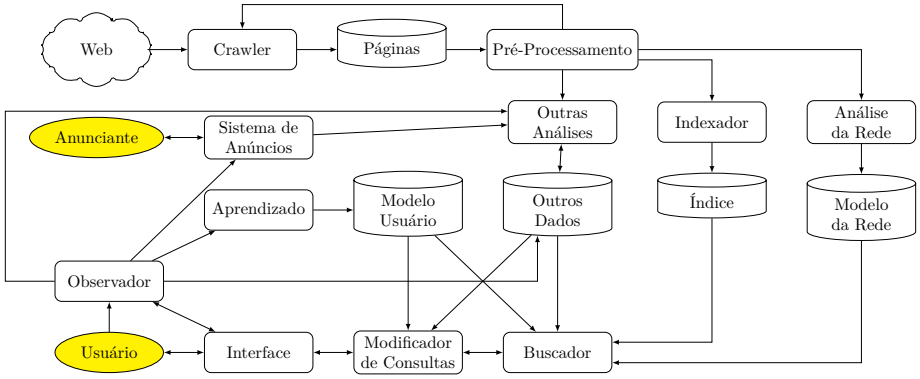


Figure 11.6: Modelo genérico de um mecanismo de busca moderno

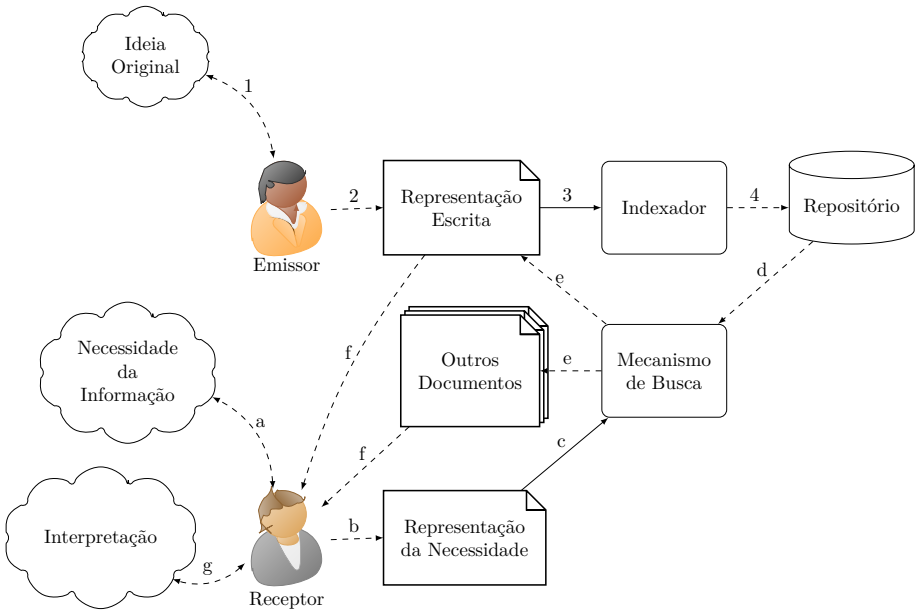


Figure 11.7: Caption Identificada 1

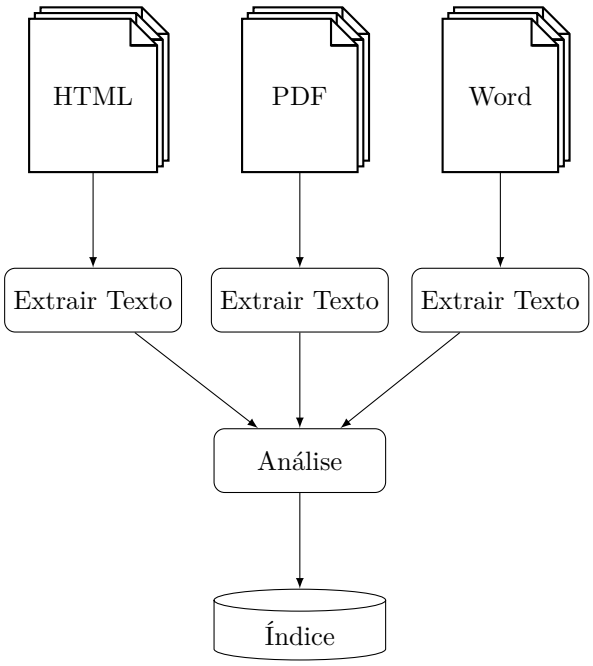


Figure 11.8: Indexar

Chapter 12

Neural Networks

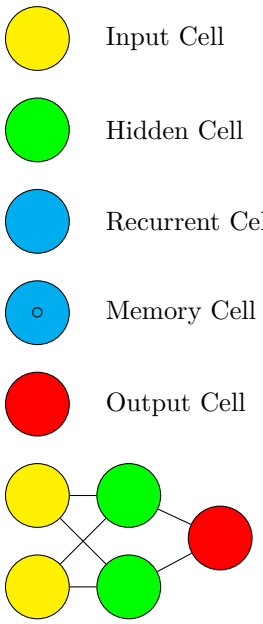


Figure 12.1: Tipo de NN

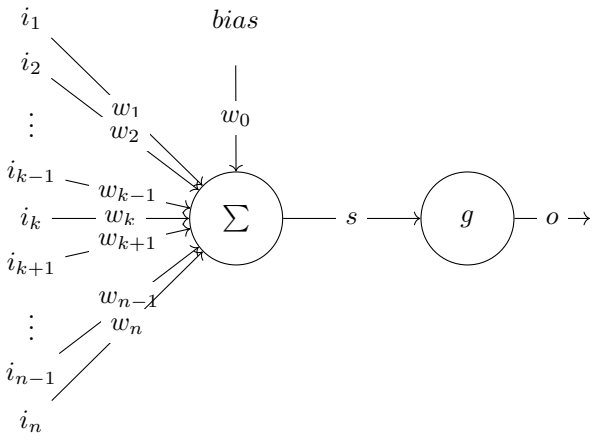


Figure 12.2: A Single Neuron in a Neural Network