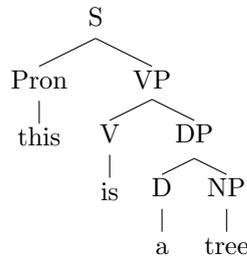# 1 Simple trees

A simple tree can just be any bracketed structure, introduced by the command \Tree:

```
                        S
                  Pron      VP
                   |
                  this   V      DP
                         |
                         is  D    NP
                             |     |
                             a    tree
```

Remember to include a space before every closing bracket, or the tree will not compile properly. The bracketed sentence can written linearly, or tabbed for easier viewing; both of the following bracketed sentences would produce the same tree:

```
[.S [.Pron this ] [.VP [.V is ] [.DP [.D a ] [.NP tree ] ]
] ]
```

```
[.S
    [.Pron this ]
    [.VP
        [.V is ]
        [.DP
            [.D a ]
            [.NP tree ]
        ]
    ]
]
```

Make sure there are spaces between the words in the sentence and the closing brackets! If there are no spaces (e.g. `[.DP a]`) you'll get an error and the document won't compile!

2 Triangles

Triangles can be added easily using the \qroof{} command, putting the content inside the brackets and the node name after the brackets:

```
                    VP
            read              DP

                      those five plays by Shakespeare
```
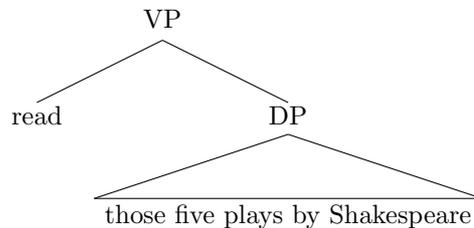
```
[.VP read \qroof{those five plays by Shakespeare}.DP ]
```
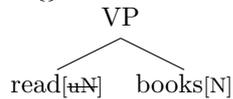
3   More complicated trees

Node names can include just about anything, including special formatting. Thus, trees can have not only VP, NP, TP, but also fancier things like vP, $b\breve{a}P$, etc.:

```
              vP

          v       VP

              read   books
```
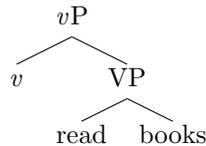
```
[.\textit{v}P \textit{v} [.VP read books ] ]
```

To add strikeout (for instance, in feature checking), you can use the `ulem` package, which includes an `\sout{}` command:

```
             VP

        read[~~uN~~]   books[N]
```

```
[.VP read[\sout{uN}] books[N] ]
```

4   Movement arrows

You can use the package `tree-dvips` to add movement arrows to trees. To do this, you must find both the node where the arrow begins and the node where the arrow ends, and label them with names using the \node{} command. Then, below the tree, use the command \anodecurve[bl]{BEGINNODE}[bl]{ENDNODE}{1in} to draw the arrow:

```
[.\textit{v}P [.\textit{v} \textit{v} \node{end}{read} ] [.VP
\node{start}{$\langle$read$\rangle$} books ] ]
\anodecurve[br]{start}[bl]{end}{1in}
```

```
                  vP

           v            VP

        v    read   ⟨read⟩   books
```
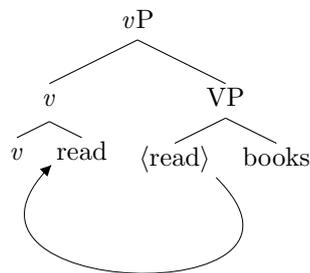
Note that, for this to work, you can't directly build LaTeX => PDF with tree-dvips; for some reason that package doesn't work with the pdflatex tool. Instead, you have to first build PostScript, and build PDF from that. If you're using the TeXnicCenter or a similar editing application, you can do this simply by selecting the LaTeX => PS => PDF option in the Build menu.