

syntaxdi

L^AT_EX-Paket zum Setzen von Syntaxdiagrammen in Form von
Railroaddiagrammen

0.8.2 2020-10-16

Paketdokumentation

Johannes PIEPER Johannes KUHAUPT Ludger HUMBERT
André HILBIG Adrian SALAMON Daniel SPITTANK

<https://ddi.uni-wuppertal.de/material/schulepaket.html>

schulepaket@johpie.de

Mit diesem Paket werden Styles für TikZ bereit gestellt, um spezielle Syntaxdiagramme erstellen zu können. Diese sind auch als Railroaddiagramme bekannt und wurden von Niklaus Wirth zur Dokumentation der Syntax von Pascal eingeführt. In der Dokumentation des Pakets `pgf` wird die Erstellung in dem Tutorium **Diagrams as Simple Graphs** (S. 66–74) Schritt für Schritt beschrieben. Das Paket war vormals Teil des Schulepakets.

Table of Contents

1 Syntaxdiagramme	1	1.2 Beispiel	2
1.1 TikZ-Stile	1	2 Index	4

1 Syntaxdiagramme

Mit den Paketen `syntaxdi` und `tikz` ist es möglich, spezielle Syntaxdiagramme als Railroaddiagramme zu erstellen. Dazu sind Elemente definiert worden, die automatisch durch Pfeile miteinander verbunden werden.

Hierzu definiert das Paket `syntaxdi` einige TikZ-Stile, die einfach genutzt werden können.

1.1 TikZ-Stile

`nonterminal`

definiert ein Non-Terminal.

1 Syntaxdiagramme

terminal

definiert ein Terminal.

fnonterminal

definiert ein Non-Terminal ohne automatische Verzweigung.

fterminal

definiert ein Terminal ohne automatische Verzweigung.

point

definiert einen Punkt, der ohne ankommenden Pfeil gezeichnet wird.

endpoint

definiert einen Punkt, der mit ankommenden Pfeil gezeichnet wird.

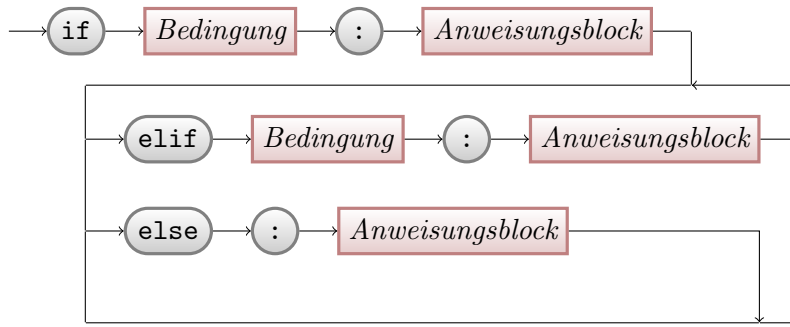
1.2 Beispiel

Damit kann z. B. das Syntaxdiagramm für eine Mehrfachverzweigung in der Programmiersprache Python in Form eines Railroaddiagramms dargestellt werden.

```
1 \begin{tikzpicture}[syntaxdiagramm]
2   \node [] {};
3   \node [terminal] {if};
4   \node [nonterminal] {Bedingung};
5   \node [terminal] {:};
6   \node [nonterminal] {Anweisungsblock};
7   \node (ersteReiheEnde) [point] {};
8   \node (ersteReiheEndeUnten) [point, below=of ersteReiheEnde] {};
9   \node (zweiteReiheStartOben) [point, left=of ersteReiheEndeUnten,
10     xshift=-75mm] {};
11  \node (zweiteReiheStart) [point, below=of zweiteReiheStartOben] {};
12  {
13    [start chain=elif going right]
14    \chainin (zweiteReiheStart);
15    \node [terminal] {elif};
16    \node [nonterminal] {Bedingung};
17    \node [terminal] {:};
18    \node [nonterminal] {Anweisungsblock};
19    \node (elifEnde) [point] {};
20    \node (elifEndeOben) [point, above=of elifEnde] {};
21    \draw[->,left] (elifEndeOben) -- (ersteReiheEndeUnten);
22  }
23  \node (dritteReiheStart) [point, below=of zweiteReiheStart,
24    yshift=-5mm] {};
25  \node (vierteReiheStart) [point, below=of dritteReiheStart,
26    yshift=-5mm] {};
27  \node (vierteReiheEnde) [point, xshift=84mm] {};
```

1 Syntaxdiagramme

```
28 {
29   [start chain=else going right]
30   \chainin (dritteReiheStart);
31   \node [terminal] {else};
32   \node [terminal] {:};
33   \node (elseEnde) [nonterminal] {Anweisungsblock};
34   \draw[->] (elseEnde) -| (vierteReiheEnde);
35 }
36 \node (ende) [endpoint] {};
37 \end{tikzpicture}
```



2 Index

E		P	
<code>endpoint</code>	2	<code>pgf</code> (package)	1
		<code>point</code>	2
F		S	
<code>fnonterminal</code>	2	<code>syntaxdi</code> (package)	1
<code>fterminal</code>	2		
N		T	
<code>nonterminal</code>	1	<code>terminal</code>	2
		<code>tikz</code> (package)	1