Beispiel: Auswertung arithmetischer Ausdrücke

Linksrekursives Übersetzungsschema:

$$
\begin{array}{lcll}
E & \to & E_1 \texttt{+} T & \{E.\mathrm{val} = E_1.\mathrm{val} + T.\mathrm{val}\} \\
E & \to & E_1 \texttt{-} T & \{E.\mathrm{val} = E_1.\mathrm{val} - T.\mathrm{val}\} \\
E & \to & T & \{E.\mathrm{val} = T.\mathrm{val}\} \\
T & \to & \texttt{(}\ E\ \texttt{)} & \{T.\mathrm{val} = E.\mathrm{val}\} \\
T & \to & \mathbf{digit} & \{T.\mathrm{val} = \mathbf{digit}.\mathrm{val}\}
\end{array}
$$

Nach Elimination der Linksrekursion:

$$
\begin{array}{lcl}
E & \to & T \\
 & & \{R.\mathrm{val} = T.\mathrm{val}\} \\
 & & R \\
 & & \{E.\mathrm{val} = R.\mathrm{s}\}
\end{array}
$$

$$
\begin{array}{lcl}
R & \to & \texttt{+}\, T \\
 & & \{R_1.\mathrm{v} = R.\mathrm{v} + T.\mathrm{val}\} \\
 & & R_1 \\
 & & \{R.\mathrm{s} = R_1.\mathrm{s}\}
\end{array}
$$

$$
\begin{array}{lcl}
R & \to & \texttt{-}\, T \\
 & & \{R_1.\mathrm{v} = R.\mathrm{v} - T.\mathrm{val}\} \\
 & & R_1 \\
 & & \{R.\mathrm{s} = R_1.\mathrm{s}\}
\end{array}
$$

$$
\begin{array}{lcl}
R & \to & \varepsilon \\
 & & \{R.\mathrm{s} = R_1.\mathrm{v}\}
\end{array}
$$

$$
\begin{array}{lcl}
T & \to & \texttt{(}\ E\ \texttt{)} \\
 & & \{T.\mathrm{val} = E.\mathrm{val}\}
\end{array}
$$

$$
\begin{array}{lcl}
T & \to & \mathbf{digit} \\
 & & \{T.\mathrm{val} = \mathbf{digit}.\mathrm{val}\}
\end{array}
$$

Prädiktiver Übersetzer (Prinzip)

```
int
E()
{
  int Tval, Eval, Rv, Rs;

  if (lookahead=='(' || isdigit(lookahead)) {
    Tval = T();
    Rv = Tval;
    Rs = R(Rv);
    Eval = Rs;
    return Eval;
  }
  else {
    error();
  }

}/*E*/

int
T()
{
  int Tval, Eval, numval;

  if (lookahead=='(') {
    match('(');
    Eval = E();
    match(')');
    Tval = Eval;
    return Tval;
  }
  else if (isdigit(lookahead)) {
    numval = lookahead;
    match(digit);
    Tval = numval;
    return Tval;
  }
  else {
    error();
  }

}/*T*/
```

```
int
R(int Rv)
{
  int Tval, R1v, R1s, Rs;

  if (lookahead=='+') {
    match('+');
    Tval = T();
    R1v = Rv + Tval;
    R1s = R(R1v);
    Rs = R1s;
    return Rs;
  }
  else if (lookahead=='-') {
    match('-');
    Tval = T();
    R1v = Rv - Tval;
    R1s = R(R1v);
    Rs = R1s;
    return Rs;
  }
  else if (lookahead=='$' || lookahead==')') {
    Rs = Rv;
    return Rs;
  }
  else {
    error();
  }

}/*R*/
```