Indentation and Priorities

Härmel Nestra

Institute of Computer Science
University of Tartu
e-mail: harmel.nestra@ut.ee

Purposes

- Introduce a recently proposed extension of CFG/PEG for specifying programming language syntax that relies on indentation (like in Haskell, Python).
- Demonstrate the ability of the approach to be used for other purposes, most notably infix operator priorities.
- Prove a theorem about limitations of this approach.
- Outline some connections between this approach and attribute grammars.

Extending CFG/PEG for Specifying Indentation

Methods of syntax description

- Context-free grammars (CFG).
- Parsing expression grammars (PEG).

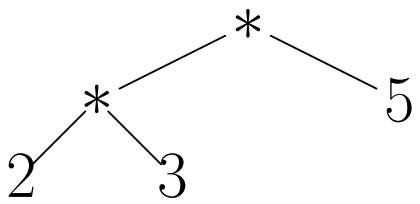
Example: CFG vs PEG

- Consider the grammar

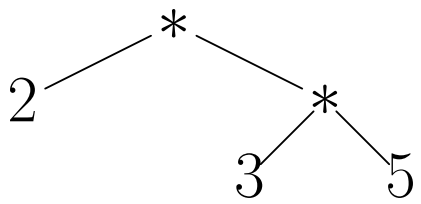
$$E ::= N \mid (E) \mid E + E \mid E * E$$
$$N ::= 0 \mid 1 \mid 2 \mid \ldots$$

How does $2 * 3 * 5$ parse?

(a) CFG only:



(b) Both CFG and PEG:



Indentation and alignment

- Pure CFG and PEG are unable to specify indentation and alignment.
- CFG/PEG is extended by constructs p^{ϱ} and $\lVert p \rVert$ (Adams 2013; Adams & Ağacan 2014):
  - p^{ϱ}, where $\varrho \subseteq \mathbb{N} \times \mathbb{N}$, means block p with indentation in relation ϱ to the current indentation;
  - $\lVert p \rVert$ means block p with the first token aligned.

Example: do expressions in Haskell

- The extent of a do expression can be specified either via indentation or braces and semicolons:

```
do                              do {
   c <- getChar                 c <- getChar; s <- getLine;
   s <- getLine          vs        putStrLn (c : s)
   putStrLn (c : s)             }
```

- Specification in this extension:

$$
\begin{aligned}
&\mathit{doexp} ::= \mathbf{do}^{>}\ \ (\mathit{istmts} \mid \mathit{stmts}) \\
&\mathit{istmts} ::= (\text{¦}\mathit{stmt}\text{¦}^{+})^{>} \\
&\mathit{stmts}\ \ ::= \mathbf{\{}^{>}(\mathit{stmt}(\mathbf{;}\,\mathit{stmt})^{*}[\mathbf{;}]\mathbf{\}})^{\circledast}
\end{aligned}
$$

Infix Operators and Priorities

Priorities

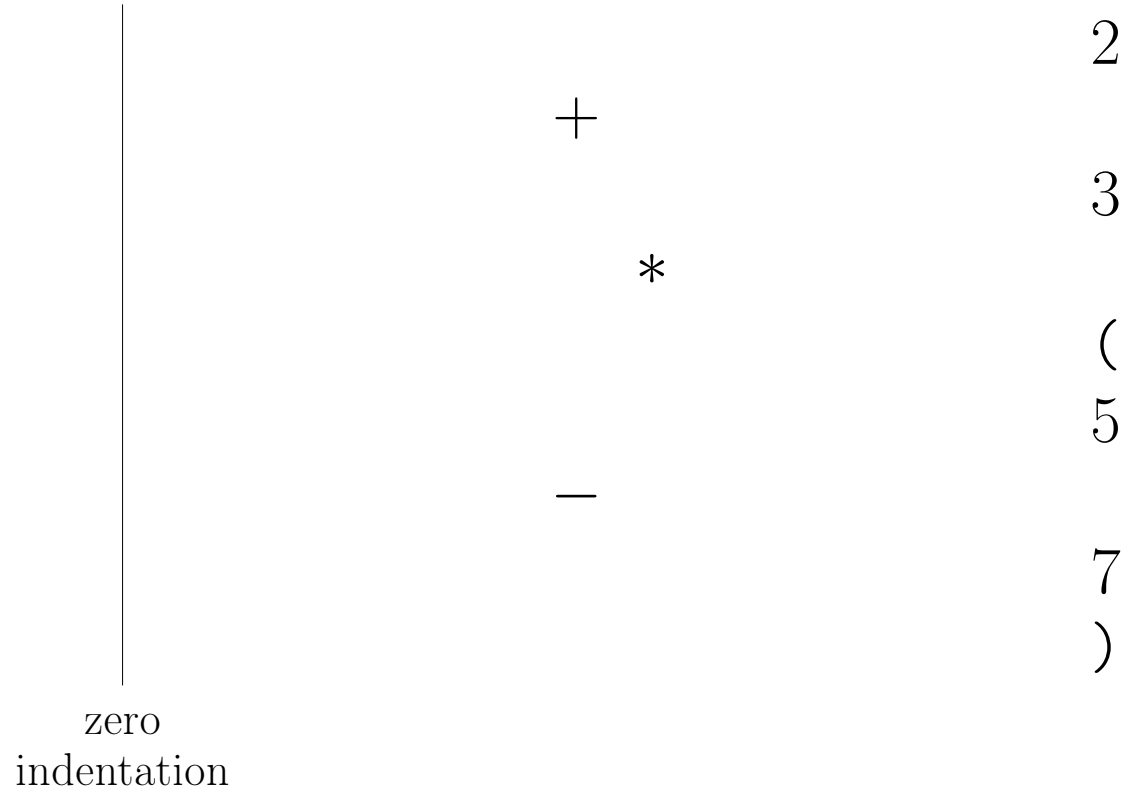
- Priorities of infix operators settle the right parsing of expressions in the case of omitted parentheses.
- Usually, higher priority means stronger attraction of the neigbourhood. For example, $2 + 3 * (5 - 7)$ parses as $2 + (3 * (5 - 7))$ since $*$ has higher priority.

Specification of expressions

- Specifying expressions involving infix operators of different priorities by a CFG/PEG?
  - Cumbersome...
  - Possible only in the case of a finite number of allowed priorities.
- The grammar extension developed for indentation naturally applies:
  - Priority plays the role of actual indentation;
  - Atoms have the highest priority;
  - Parentheses relax the expected priority in the subexpression surrounded;
  - Alignment carries the operator priority over to the whole expression.

Illustration: Priorities as indentation

- Expression $2 + 3 * (5 - 7)$, with priorities depicted as indentation:



Example: Grammars for arithmetic expressions

- The grammar of arithmetic expressions can be given as follows:

$$E ::= N \mid E^{\geq} \,!\oplus!\, E^{>} \mid (E^{\circledast})$$
$$N ::= 0 \mid 1 \mid \ldots$$
$$\oplus ::= + \mid * \mid \ldots$$

- An equivalent grammar:

$$E ::= N \mid E^{>} \,!\oplus!\, E^{>} \,!* \mid (E^{\circledast})$$
$$N ::= 0 \mid 1 \mid \ldots$$
$$\oplus ::= + \mid * \mid \ldots$$

More about Relations

Parsing process

- At each stage of parsing, partial information known about the current indentation is kept in the form of the set of all natural numbers allowed.
- The real indentation of tokens found during parsing refines the knowledge about indentation.
- A real indentation outside the set of allowed indentations leads to parse error.
- Restrict allowable sets of indentations to (possibly infinite) intervals of consecutive natural numbers.

Indention and dedention as allowable set transformers

- Relation $\varrho \subseteq \mathbb{N} \times \mathbb{N}$ as function: for any $n \in \mathbb{N}$,

$$\varrho(n) = \{m \in \mathbb{N} \ : \ (m, n) \in \varrho\}\,.$$

- Indention: for any $S \subseteq \mathbb{N}$,

$$\mathrm{indent}_\varrho(S) = \bigcup_{n \in S} \varrho(n).$$

- Dedention: for any $S, T \subseteq \mathbb{N}$,

$$\mathrm{dedent}_\varrho(S, T) = \{n \in S \ : \ \varrho(n) \cap T \neq \varnothing\}\,.$$

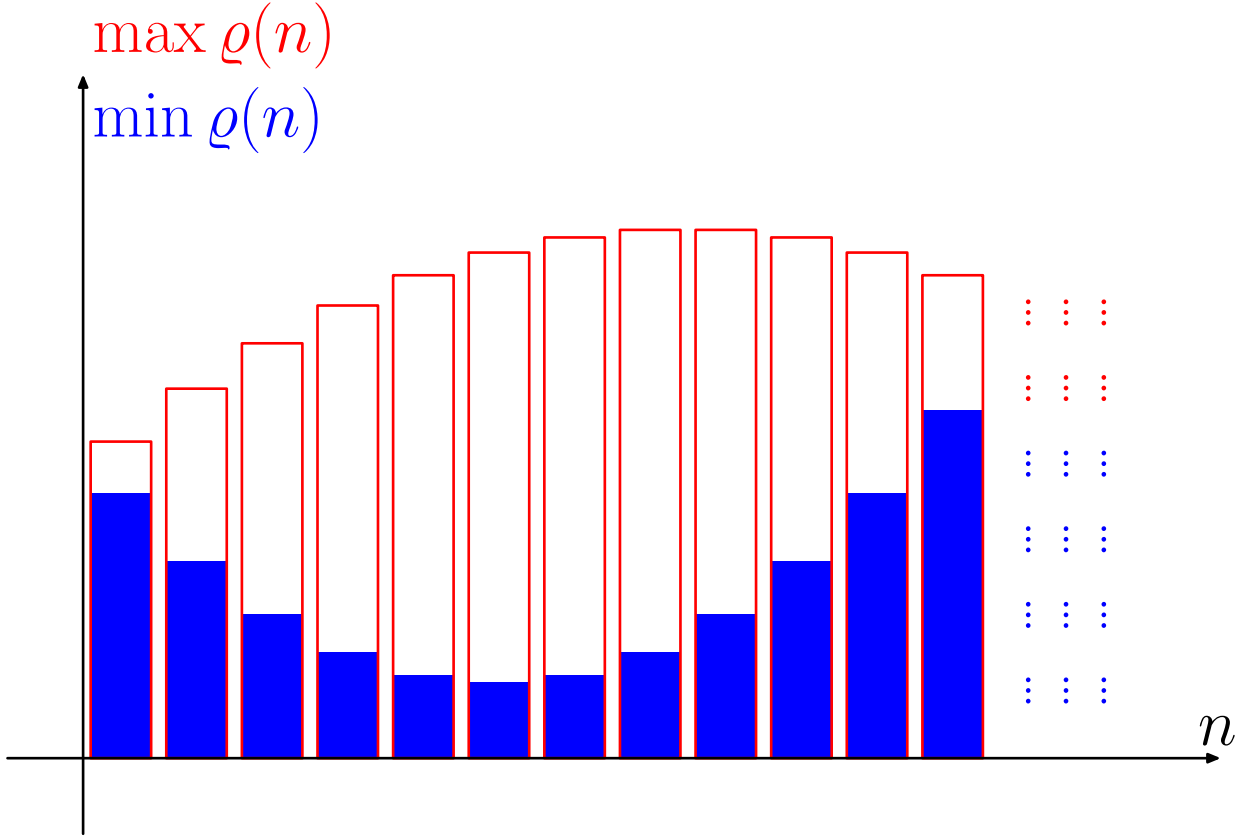
- The relations $\geq$, $>$, $=$ and $\circledast$ (i.e., $\mathbb{N} \times \mathbb{N}$) keep the allowable sets representable as intervals of consecutive numbers (Adams & Ağacan 2014).

Theorem

- For any relation $\varrho \subseteq \mathbb{N} \times \mathbb{N}$, indention and dedention of intevals always result in intervals of natural numbers iff the following conditions are met:
  - For every natural number n, $\varrho(n)$ is an interval of natural numbers;
  - For every two consecutive natural numbers n_1 and n_2, $\min \varrho(n_1) \leqslant \max \varrho(n_2) + 1$;
  - The function $f(n) = \max \varrho(n)$ is either non-decreasing or weakly unimodal; the function $g(n) = \min \varrho(n)$ satisfies the same condition w.r.t. the inverse order.

Illustration: The shape of admissible relations

- An example of an admissible relation ϱ:



Computing of indention and dedention

- Provided $\min \varrho(n)$, $\max \varrho(n)$ and their extremum points are computed in constant time:
  - The effect of indention can be computed in constant time;
  - The effect of dedention can be computed in logarithmic time.

Connections with Attribute Grammars

Example: Simple English grammar

- Here is a simple classic example CFG for a fragment of natural language:

  <sentence> ::= *<subject>* *<predicate>*
  <subject> ::= *<noun>*
  <predicate> ::= *<verb>* *<object>*
  <object> ::= *<noun>*
  <noun> ::= Alice | Bob | children | parents | . . .
  <verb> ::= like | help | . . . | likes | helps | . . .

- As it is, it enables incorrect sentences like “Alice like Bob”.

Example: Simple English grammar (ctd.)

- Assign two numbers 1 and 2 (or other values) to all nouns and verbs in their singular and plural form, respectively. The grammar can be corrected using the indentation extension:

  $$
  \begin{array}{lll}
  \langle sentence \rangle & ::= & \text{¦}\langle subject \rangle\text{¦}\ \text{¦}\langle predicate \rangle\text{¦} \\
  \langle subject \rangle & ::= & \langle noun \rangle \\
  \langle predicate \rangle & ::= & \langle verb \rangle\ \langle object \rangle^{\circledast} \\
  \langle object \rangle & ::= & \langle noun \rangle \\
  \langle noun \rangle & ::= & \text{John} \mid \text{Mary} \mid \text{children} \mid \text{parents} \mid \ldots \\
  \langle verb \rangle & ::= & \text{like} \mid \text{help} \mid \ldots \mid \text{likes} \mid \text{helps} \mid \ldots
  \end{array}
  $$

- Usually, this correction is done via attributes.

Example: Priority as an attribute

- Priority can be handled as a synthesized attribute:

$$
\begin{array}{ll}
E ::= N & [E.pr = N.pr] \\
E ::= E_1 \oplus E_2 & [E.pr = \oplus.pr \text{ if } E_1.pr > \oplus.pr \text{ and } E_2.pr > \oplus.pr] \\
E ::= (E_1) & [E.pr = \mathrm{maxpr}] \\
N ::= 0 \mid 1 \mid \ldots & [N.pr = \mathrm{maxpr}] \\
\oplus ::= + & [\oplus.pr = 6] \\
\oplus ::= * & [\oplus.pr = 7]
\end{array}
$$

. . . .

Unbalanced priorities

- OO dot (in Scala, for instance) binds strongerly to the right than function application binds to the left, while the dot binds weakerly to the left than function application binds to the right:

  `1.to(n).contains(10)` ≡ `(((1.to)(n)).contains)(10)`

- Lambda dot binds extremely strongly to the left and extremely weakly to the right.
- Fixities (associativities).

Unbalanced priorities as a quadruple of attributes

- The meaning of unbalanced priorities can be defined in terms of synthesized attributes $X.lpr$, $X.rpr$ and inherited attributes $X.lcxt$, $X.rcxt$.

$$
\begin{array}{ll}
S ::= E & [E.lcxt = \text{minpr}, E.rcxt = \text{minpr}] \\
E ::= N & [E.lpr = N.lpr, E.rpr = N.rpr] \\
E ::= E_1 \oplus E_2 & [E_1.lcxt = E.lcxt, E_1.rcxt = \oplus.lpr] \\
 & [E_2.lcxt = \oplus.rpr, E_2.rcxt = E.rcxt] \\
 & \left[\begin{array}{l} E.lpr = \oplus.lpr \text{ if } E_1.lpr > E.lcxt \text{ and } E_1.rpr > \oplus.lpr \\ E.rpr = \oplus.rpr \text{ if } E_2.lpr > \oplus.rpr \text{ and } E_2.rpr > E.rcxt \end{array}\right] \\
E ::= (E_1) & [E_1.lcxt = \text{minpr}, E_1.rcxt = \text{minpr}] \\
 & [E.lpr = \text{maxpr}, E.rpr = \text{maxpr}] \\
N ::= 0 \mid \ldots & [N.lpr = \text{maxpr}, N.rpr = \text{maxpr}] \\
\oplus ::= + & [\oplus.lpr = 59, \oplus.rpr = 60] \\
\oplus ::= * & [\oplus.lpr = 69, \oplus.rpr = 70]
\end{array}
$$

.....

Unbalanced priorities via the indentation framework

- Let x and y be quadruples with projections lpr, rpr, lcxt, rcxt. Define

$$y \overset{l}{\succ} x \rightleftharpoons \mathrm{lpr}\, y > \mathrm{lcxt}\, x \wedge \mathrm{rpr}\, y > \mathrm{lpr}\, x \wedge \mathrm{lcxt}\, y = \mathrm{lcxt}\, x \wedge \mathrm{rcxt}\, y = \mathrm{lpr}\, x$$

$$y \overset{r}{\succ} x \rightleftharpoons \mathrm{lpr}\, y > \mathrm{rpr}\, x \wedge \mathrm{rpr}\, y > \mathrm{rcxt}\, x \wedge \mathrm{lcxt}\, y = \mathrm{rpr}\, x \wedge \mathrm{rcxt}\, y = \mathrm{rcxt}\, x$$

$$y \circledast x \rightleftharpoons \mathrm{lcxt}\, y = \mathrm{rcxt}\, y = \mathrm{minprior}$$

- Grammar for expressions, assuming that literals and parenthized expressions have maximum priorities:

$$E ::= N \mid E \overset{l}{\succ} \mathbin{!} \oplus \mathbin{!} E \overset{r}{\succ} \mid (\, E^{\circledast} \,)$$
$$N ::= 0 \mid 1 \mid \ldots$$
$$\oplus ::= + \mid * \mid \ldots$$

Indentation extension and attribute grammars

- Inherited attributes that inherit from parents only can be trivially represented in the indentation extension.
- An rule f of a synthesized attribute that depends on the values of the same attribute of n children can be represented in the indentation extension iff there exist binary relations $\varrho_1, \ldots, \varrho_n$ such that, for every $a_1, \ldots, a_n, b$,

$$b = f(a_1, \ldots, a_n) \iff (a_1, b) \in \varrho_1 \wedge \ldots \wedge (a_n, b) \in \varrho_n.$$

- Attributes inherited from siblings or synthesized from other attributes? ...
- Indentation usually cannot be expressed as an inherited or synthesized attribute.

Conclusion

Contributions

- We have analyzed the limitations of the indentation extension of CFG/PEG in terms of the efficiently expressible/computable indentation sets.
- We have drawn connections between this extension and attribute grammars. It shows that neither the indentation extension nor attribute grammars is more powerful. Still, both these approaches can describe, e.g., expressions with operators with unbalanced priorities.