

### Sample Extended BNF: Haskell expression fragment

This BNF for Haskell expressions is taken directly from the Haskell 98 Report, but extracts only selected parts.

$exp \square exp^0$	
$exp^i \square exp^{i+1} [qop^{(n,i)} exp^{i+1}] \mid lexp^i \mid rexp^i$	$(0 \leq i \leq 9)$
$lexp^i \square (lexp^i \mid exp^{i+1}) qop^{(l,i)} exp^{i+1}$	$(0 \leq i \leq 9, \text{left-assoc})$
$rexp^i \square exp^{i+1} qop^{(r,i)} (rexp^i \mid exp^{i+1})$	$(0 \leq i \leq 9, \text{right-assoc})$
$qop^{(n,4)} \square < \mid \leq \mid == \mid /= \mid \geq \mid >$	$(\text{non-assoc, prec 4})$
$qop^{(l,6)} \square + \mid -$	$(\text{left-assoc, prec 6})$
$qop^{(l,7)} \square * \mid /$	$(\text{left-assoc, prec 7})$
$qop^{(r,8)} \square ** \mid ^^ \mid ^$	$(\text{right-assoc, prec 8})$
$qop^{(r,9)} \square .$	$(\text{right-assoc, prec 9})$
$exp^{10} \square \backslash \text{apat}_1 \dots \text{apat}_n \rightarrow exp$	$(\text{lambda abstraction, } n \geq 1)$
$\mid \text{let decls in exp}$	$(\text{let expression})$
$\mid \text{if exp then exp else exp}$	$(\text{conditional})$
$\mid \text{case exp of \{ alts \}}$	$(\text{case expression})$
$\mid \text{do \{ stmts \}}$	$(\text{do expression})$
$\mid \text{fexp}$	
$\text{fexp} \square [ \text{fexp} ] \text{aexp}$	$(\text{optional function application})$
$\text{aexp} \square \text{qvar}$	$(\text{qualified variable})$
$\mid \text{gcon}$	$(\text{general constructor})$
$\mid \text{literal}$	$(\text{number, string, etc.})$
$\mid ( \text{exp} )$	$(\text{parenthesized expression})$
$\mid (\text{exp}_1, \dots, \text{exp}_n)$	$(\text{tuple, } n \geq 2)$
$\mid [\text{exp}_1, \dots, \text{exp}_n]$	$(\text{list, } n \geq 0)$
$\text{qvar} \square \text{identifier}$	
$\text{literal} \square \text{numeral} \mid \text{charconst} \mid \text{stringconst} \mid \text{boolconst}$	

#### partial EBNF for Haskell expressions

Operations and expressions are categorized as left/right/non-associative, and the syntax categories provided reflect both this and the precedence of operations. A

more complete table of Haskell operators appears below. Haskell syntax for variables and numbers is not included here, but is not significantly different from other programming languages. Note that a number of symbols (e.g., `|`, `[`, `]`) are used both as Haskell characters *and* BNF markup, and the Haskell BNF distinguishes their use by different fonts — this is a serious potential source of confusion. The BNF instances of these symbols are written in bold in the BNF above.

The Haskell operators, their associativity and precedence, are summarized in the following table:

prec	left-assoc	non-assoc	right-assoc
9	!!		.
8			<b>**</b> , <b>^^</b> , <b>^</b>
7	<b>*</b> , <b>/</b> , <b>%</b> , <b>`div`</b> , <b>`mod`</b> , <b>`rem`</b> , <b>`quot`</b>		
6	<b>+</b> , <b>-</b>		
5			<b>:</b> , <b>++</b>
4		<b>/=</b> , <b>&lt;</b> , <b>&lt;=</b> , <b>==</b> , <b>&gt;</b> , <b>&gt;=</b> , <b>`elem`</b> , <b>`notElem`</b>	
3			<b>&amp;&amp;</b>
2			<b>  </b>
1		<b>&lt;-</b>	
0			<b>\$</b> , <b>`seq`</b>