<span class="hljs-comment">(\* type variables \*)</span>

<span class="hljs-keyword">type</span> <span class="hljs-symbol">'a</span> t = <span class="hljs-symbol">'a</span> <span class="hljs-built\_in">list</span>

<span class="hljs-keyword">let</span> f (a : <span class="hljs-symbol">'a</span> <span class="hljs-built\_in">list</span>) : <span class="hljs-symbol">'a</span> = <span class="hljs-type">List</span>.hd a

<span class="hljs-comment">(\* polymorphic variants \*)</span>

<span class="hljs-keyword">type</span> t = [ <span class="hljs-type">`A</span> | <span class="hljs-type">`B</span> ]

<span class="hljs-comment">(\* variants \*)</span>

<span class="hljs-keyword">type</span> result = <span class="hljs-type">Sat</span> | <span class="hljs-type">Unsat</span> | <span class="hljs-type">Unknown</span>

<span class="hljs-comment">(\* module and module types \*)</span>

<span class="hljs-keyword">module</span> <span class="hljs-keyword">type</span> <span class="hljs-type">S</span> = <span class="hljs-keyword">sig</span>

<span class="hljs-keyword">val</span> compute : <span class="hljs-built\_in">unit</span> -&gt; <span class="hljs-built\_in">unit</span>

<span class="hljs-keyword">end</span>

<span class="hljs-keyword">module</span> <span class="hljs-type">Impl</span> : <span class="hljs-type">S</span> = <span class="hljs-keyword">struct</span>

<span class="hljs-keyword">let</span> compute <span class="hljs-literal">()</span> = <span class="hljs-literal">()</span>

<span class="hljs-keyword">end</span>