1 Conditional inclusion of code

From previous examples the `%<*guard>...%</guard>` syntax should be a little familiar. This example demonstrates conditional extraction of such elements in the source document into multiple files. This technique makes it easy, say, to maintain a 'debug' version of a package without polluting the public source with code for testing.

```
 1 ⟨A⟩   code in 'A'
 2 ⟨B⟩   code in 'B'
 3 ⟨!A⟩   code not in 'A'
 4 ⟨!B⟩   code not in 'B'
 5 ⟨A & B⟩   code in 'A' and 'B'
 6 ⟨A | B⟩   code in 'A' or 'B'
 7 ⟨(A | B)&!(A & B)⟩   code in 'A' xor 'B'
```

Note the change in the typeset source when guards are nested:
(this is equivalent to `%<A&B>`)

```
 8 ⟨∗A⟩
 9 ⟨B⟩   'B' nested inside 'A'
10 ⟨/A⟩
```

2 Verbatim files that are produced

2.1 Generated from 'A'

```
code in 'A'
code not in 'B'
code in 'A' or 'B'
code in 'A' xor 'B'
```

2.2 Generated from 'B'

```
code in 'B'
code not in 'A'
code in 'A' or 'B'
code in 'A' xor 'B'
```

2.3 Generated from 'A' and 'B'

```
code in 'A'
code in 'B'
code in 'A' and 'B'
code in 'A' or 'B'
'B' nested inside 'A'
```