Documenting mapping clauses

Sail source

```
default Order dec
$include <prelude.sail>

scattered union Instr

val encdec : Instr <-> bits(32)

union clause Instr = Add : (bits(5), bits(5), bits(5))

mapping clause encdec =
    Add(rd, rx, ry) <-> 0xFFFF @ rd : bits(5) @ 0b1 @ rx : bits(5) @ ry : bits(5)
```

Result

Sail allows us to specify bi-directional functions, called *mappings*. These can be broken into multiple scattered functions in the same way that functions can (see the `clauses.adoc` example for details). However in this case, we can select the clause we want to document by matching on either the left or the right pattern of the mapping.

```
sail::encdec[left-clause="Add(_, _, _)",type=mapping]
```

which produces:

```
mapping clause encdec =
    Add(rd, rx, ry) <-> 0xFFFF @ rd : bits(5) @ 0b1 @ rx : bits(5) @ ry : bits(5)
```

NOTE	The matching language in the `left-clause` and `right-clause` attributes is a subset of the Sail pattern language that includes constructor patterns, identifiers, wildcards, and binary literals.