

# MPCal Syntax Highlighting

We use `minted` package for syntax highlighting in the  $\text{\LaTeX}$  documents. Figure 1 shows an MPCal code block that uses `mpcal` lexer. It's useful when you want to show a block of MPCal code. Figure 2 depicts a  $\text{TLA}^+$  code block using `tla+` lexer. It supports PlusCal and MPCal languages in the comments.

```
1 mapping macro TCPChannel {
2   read {
3     await Len($variable) > 0;
4     with (msg = Head($variable)) {
5       $variable := Tail($variable);
6       yield msg;
7     };
8   }
9
10  write {
11    await Len($variable) < BUFFER_SIZE;
12    yield Append($variable, $value);
13  }
14 }
```

Figure 1: MPCal code block.

```

1  ----- MODULE demo -----
2  EXTENDS Naturals, Sequences, TLC
3  (*****
4  --mpcal Bug {
5      archetype AFoo() {
6          c1: print("Hello");
7      }
8
9      fair process (Foo = 1) == instance AFoo();
10 }
11
12 \* BEGIN PLUSCAL TRANSLATION
13 --algorithm Bug {
14     fair process (Foo = 1) {
15         c1:
16             print "Hello";
17
18     }
19 }
20 \* END PLUSCAL TRANSLATION
21 (*****)
22 \* BEGIN TRANSLATION (chksum(pcal) = "30725eec" /\ chksum(tla) = "ab8ae8f8")
23 VARIABLE pc
24
25 vars == << pc >>
26
27 ProcSet == {1}
28
29 Init == /\ pc = [self \in ProcSet |-> "c1"]
30
31 c1 == /\ pc[1] = "c1"
32       /\ PrintT("Hello")
33       /\ pc' = [pc EXCEPT ![1] = "Done"]
34
35 Foo == c1
36
37 Terminating == /\ \A self \in ProcSet: pc[self] = "Done"
38                 /\ UNCHANGED vars
39
40 Next == Foo
41         \/ Terminating
42
43 Spec == /\ Init /\ [] [Next]_vars
44         /\ WF_vars(Foo)
45
46 Termination == <>(\A self \in ProcSet: pc[self] = "Done")
47
48 \* END TRANSLATION
49 =====

```

Figure 2: TLA<sup>+</sup> code block.