MPCal Syntax Highlighting

We use minted package for syntax highlighting in the LaTeXdocuments. 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 TLA$^+$ code block using `tla+` lexer. It supports PlusCal and MPCal languages in the comments.

```
mapping macro TCPChannel {
    read {
        await Len($variable) > 0;
        with (msg = Head($variable)) {
            $variable := Tail($variable);
            yield msg;
        };
    }

    write {
        await Len($variable) < BUFFER_SIZE;
        yield Append($variable, $value);
    }
}
```

Figure 1: MPCal code block.

```
-------------------------------- MODULE demo --------------------------------
EXTENDS Naturals, Sequences, TLC
(*****************
--mpcal Bug {
    archetype AFoo() {
        c1: print("Hello");
    }

    fair process (Foo = 1) == instance AFoo();
}

\* BEGIN PLUSCAL TRANSLATION
--algorithm Bug {
    fair process (Foo = 1) {
        c1:
            print "Hello";

    }
}
\* END PLUSCAL TRANSLATION
****************)
\* BEGIN TRANSLATION (chksum(pcal) = "30725eec" /\ chksum(tla) = "ab8ae8f8")
VARIABLE pc

vars == << pc >>

ProcSet == {1}

Init == /\ pc = [self \in ProcSet |-> "c1"]

c1 == /\ pc[1] = "c1"
      /\ PrintT("Hello")
      /\ pc' = [pc EXCEPT ![1] = "Done"]

Foo == c1

Terminating == /\ \A self \in ProcSet: pc[self] = "Done"
               /\ UNCHANGED vars

Next == Foo
           \/ Terminating

Spec == /\ Init /\ [][Next]_vars
        /\ WF_vars(Foo)

Termination == <>(\A self \in ProcSet: pc[self] = "Done")

\* END TRANSLATION
=============================================================================
```

Figure 2: TLA$^+$ code block.