The following code will produce small arrows positioned at a given location with a given length and optional shift.

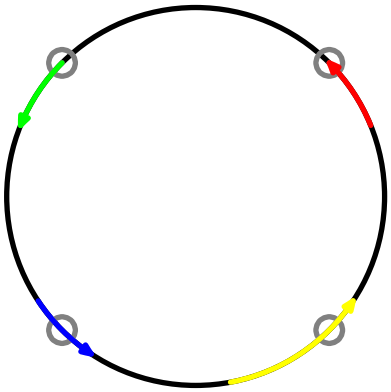
```
\startMPcode
  pickup pencircle scaled 2pt;

  path p ; p := fullcircle scaled 5cm ; draw p ;

  for i=1,3,5,7 :
    drawdot point i of p withpen pencircle scaled 10 withcolor .5white ;
  endfor ;

  drawarrow leftarrow   (p,point 1 of p,1cm)      withcolor red    ;
  drawarrow rightarrow  (p,point 3 of p,1cm)      withcolor green  ;
  drawarrow centerarrow (p,point 5 of p,1cm)      withcolor blue   ;
  drawarrow pointarrow  (p,point 7 of p,2cm,.5cm) withcolor yellow ;
\stopMPcode
```

The shift of .5cm in the last example (the yellow arrow) is a backward shift. This may be fuzzy, but negative values would be even more confusing. By default the arrow is centered around the given point.



The code that produces the arrow is (here) defined as inclusion, but may end up it in the MetaFun macro collection.

```
\startMPinclusions
vardef pointarrow (expr pat, loc, len, off) =
  save l, r, s ; path l, r ; numeric s ;
% draw loc withpen pencircle scaled 10 withcolor .5white ;
  s := len/2 - off ; if s<=0 : s := 0 elseif s>len : s := len fi ;
  r := pat cutbefore loc ;
```

```
  r := (r cutafter point (arctime s of r) of r) ;
  s := len/2 + off ; if s<=0 : s := 0 elseif s>len : s := len fi ;
  l := reverse (pat cutafter loc) ;
  l := (reverse (l cutafter point (arctime s of l) of l)) ;
  (l..r)
enddef ;

def rightarrow  (expr pat,tim,len) = pointarrow(pat,tim,len,-len) enddef ;
def leftarrow   (expr pat,tim,len) = pointarrow(pat,tim,len,+len) enddef ;
def centerarrow (expr pat,tim,len) = pointarrow(pat,tim,len,   0) enddef ;
\stopMPinclusions
```