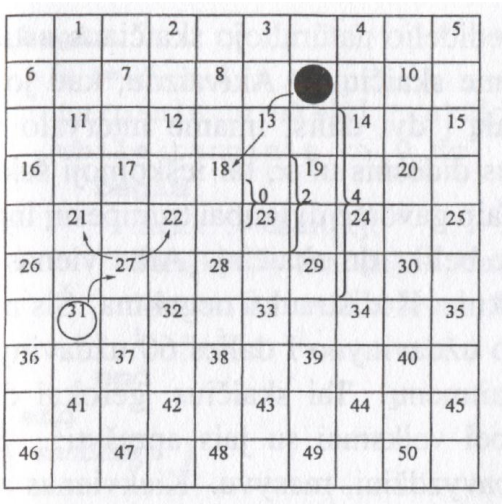
Šaškės: viena prieš vieną

Panagrinėkime, kada baltoji gali nukirsti juodąją.

Baltoji šaškė gali kirsti juodąją tik tada, kai abi šaškės stovi gretimose eilutėse, t. y. kai šaškes skiria 0 tuščių eilučių.

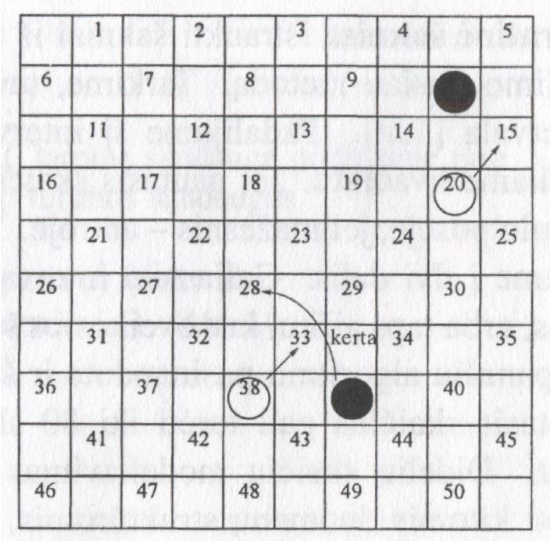
1) Tarkime, kad abi šaškes skiria lyginis eilučių skaičius, pavyzdžiui 4. Atlikus baltajai šaškei vieną ėjimą, jas skirs 3 eilutės, atlikus juodajai šaškei ėjimą – dvi. T y. kiekvieną kartą, kai ateis baltosios šaškės eilė, jas skirs lyginis eilučių skaičius: 4, 2, 0. Juodoji šaškė niekada neturės galimybės numušti baltosios. Taigi, tokiu atveju baltoji šaškė pasieks pergalę arba lygiąsias:



2) Jei abi šaškes skiria nelyginis eilučių skaičius, baltoji šaškė niekaip negalės numušti juodosios. Šiuo atveju ji gali tik pasiekti lygiąsias arba pralaimėti.

Pagal šiuos samprotavimus ir sukursime strategiją: jei šaškes skiria lyginis eilučių skaičius, baltoji šaškė stengiasi artėti prie juodosios, jei nelyginis – nutolti.

Reikia nepamiršti poros atskirų atvejų. Jei baltoji šaškė stovi kraštiniame stulpelyje, ji turi tik vieną galimą ėjimą. Jei abi šaškės stovi toje pačioje eilutėje, ir jas skiria vienas langelis, baltoji turi bėgti nuo juodosios šaškės, nes toji gali kirsti atgal:



Turint šią strategiją pats sunkiausias dalykas, surasti sąryšį tarp langelio numerio ir eilutės bei stulpelio, kuriame jis yra. Algoritme laikėme, kad eilutės numeruojamos iš viršaus į apačią, stulpeliai – iš kairės į dešinę.

```
function ėjimas (b, j: integer): integer;
  var b_eil, b_st, j_eil, j_st: integer;
begin   { apskaičiuojame abiejų šaškių eilutes ir stulpelius }
  b_eil := (b - 1) div 5 + 1;
  if (b - 1) mod 10 < 5
     then b_st := b mod 5 * 2
     else b_st := 2 * (b mod 5 - 1) + 1;
  j_eil := (j - 1) div 5 + 1;
  if (j - 1) mod 10 < 5
     then j_st := j mod 5 * 2
     else j_st := 2 * (j mod 5 - 1) + 1;
  { rasime ėjimą }
  if b_st = 1   { jei baltoji šaškė kraštiniame stulpelyje }
       then b_st := b_st + 1
       else if b_st = 10
               then b_st := b_st - 1
  else if b_eil = j_eil   { abi šaškės vienoje eilutėje }
          then if b_st < j_st          { baltoji kairiau juodosios }
                  then b_st := b_st - 1      { tolstama }
                  else b_st := b_st + 1
```

```
  else if abs(b_eil-j_eil-1) mod 2 = 0  { lyginis eilučių skaičius }
          then if b_st < j_st
                  then b_st := b_st + 1        { artėjama }
                  else b_st := b_st -1
          else if b_st < j_st            { nelyginis eilučių skaičius }
                  then b_st := b_st - 1
                  else b_st := b_st + 1;    { artėjama }
  b_eil := b_eil - 1;
  if b_st mod 2 = 0
  then ėjimas := (b_eil-1) div 2 * 10 + b_st div 2
  else ėjimas := (b_eil-1)div 2 * 10 + 5 + b_st div 2 + 1
end;  { ėjimas }
```