Magic Hexagon and Constraints

M. Anton Ertl, TU Wien

Magic Hexagon

```
  A B C
 D E F G
H I J K L
 M N O P
  Q R S
```

- A–S all have different values $\in \{1...19\}$
- Each row and diagonal has sum 38
- Eliminate rotational and mirror symmetries

```
: mhex ( -- )
 occupationmap 20 erase
 try< {: A :}
  try< {: C :} A C < if
   38 A - C - occupy< {: B :}
    try< {: L :} A L < if
     38 C - L - occupy< {: G :}
      try< {: S :} A S < if
       38 L - S - occupy< {: P :}
        try< {: Q :} A Q < if
         38 S - Q - occupy< {: R :}
          try< {: H :} A H < if C H < if
           38 Q - H - occupy< {: M :}
            38 H - A - occupy< {: D :}
             try< {: E :}
              38 D - E - G - occupy< {: F :}
               38 B - F - P - occupy< {: K :}
                38 G - K - R - occupy< {: O :}
                 38 P - O - M - occupy< {: N :}
                  38 R - N - D - occupy< {: I :}
                   38 M - I - B - E = if
                    38 A - E - O - S - occupy< {: J :}
                     H I + J + K + L + 38 = if
                      C F + J + N + Q + 38 = if
                       ... print hexagon ...
... close all the open control structures ...
```

```
  A B C
 D E F G
H I J K L
 M N O P
  Q R S
```

```
: mhex ( -- )
 ... open control structures, check constraints ...
                                cr ."     " A .. B .. C ..
                                cr ."   " D .. E .. F .. G ..
: .. 4 .r ;                     cr      H .. I .. J .. K .. L ..
                                cr ."   " M .. N .. O .. P ..
                                cr ."     " Q .. R .. S .. cr
 ... close control structures ...
;
```

```
create occupationmap 20 allot
\ 0 if free, non-0 if occupied

: occupation! ( f u -- )
    occupationmap + c! ;

: occupy< ( u -- u )
    ]] dup >r occupationmap + c@ 0= if
       true r@ occupation! r@ [[ ; immediate
: >occupy ( -- )
    ]] false r@ occupation! then
       rdrop [[ ; immediate

: try< ( run-time: -- u )
    ]] 20 1 do i occupy< [[ ; immediate
: >try ( run-time: -- )
    ]] >occupy loop [[ ; immediate
```

```
: mhex ( -- )
 occupationmap 20 erase
 try< {: A :}
  try< {: C :} A C < if
   38 A - C - occupy< {: B :}
    try< {: L :} A L < if
     38 C - L - occupy< {: G :}
      try< {: S :} A S < if
       38 L - S - occupy< {: P :}
         ... inner code ...
        >occupy
       then >try
      >occupy
     then >try
    >occupy
   then >try
 >try ;
```