Alternative computer architectures and programming languages report: Prolog

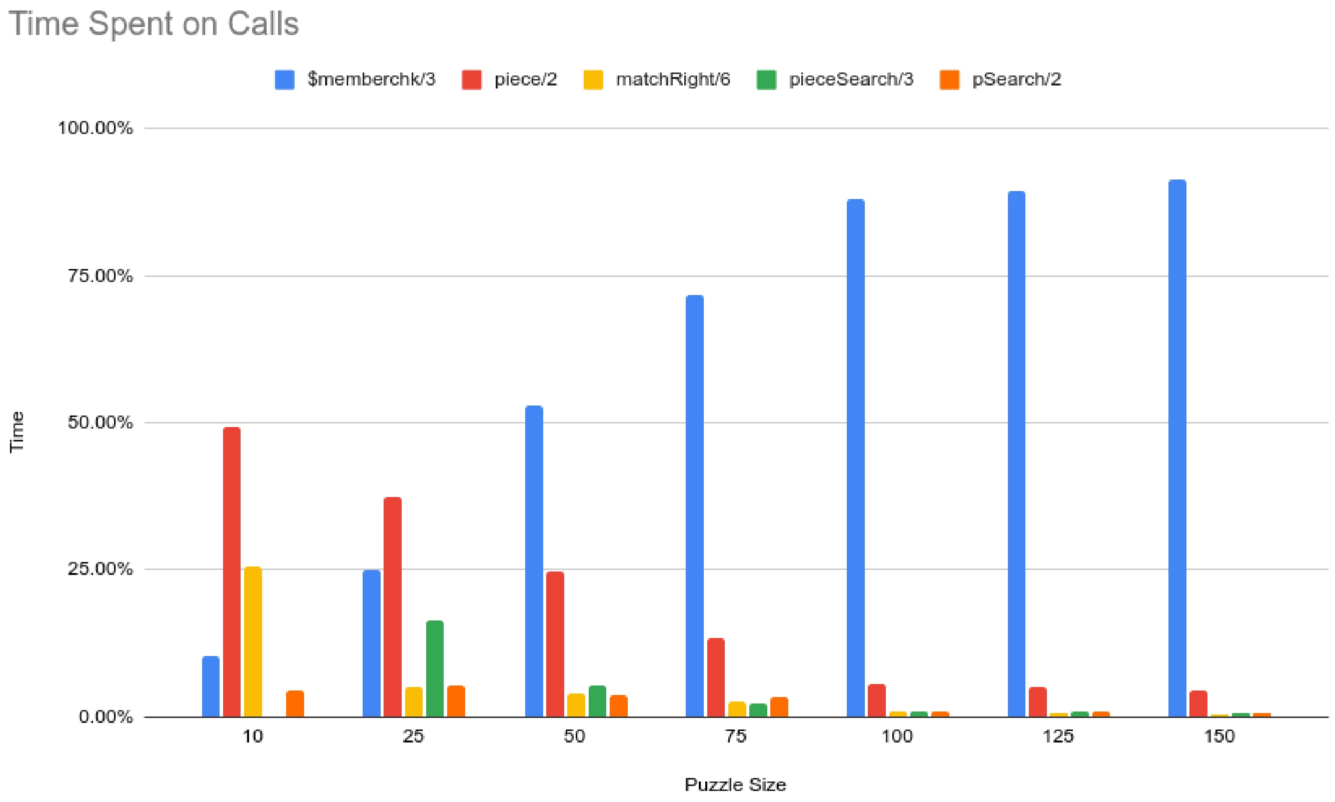
Jigsaw Puzzle Solver

Lu Knoblich

Table of contents

AMS Project Intro

References for Markdown



```
piece(p01, [s00,s01,s02n,s00]).
% (...)

piece(marginLeft,[null,smargin,null,null]).
piece(margin2d,[smargin,smargin,smargin,smargin]).
piece(margin2d,[s00,smargin,smargin,smargin]).
```

This is achieved through the predicate `matchingShapes/2`, which was kept as a separate predicate to avoid the endless loops that would occur if `shapeMatch(A,B):- shapeMatch(B,A).` had been used.

```
@startyaml
pSearch(Piece, [Upper, Right, Down, Left]):
    0 deg rot.: piece(Piece, [Upper, Right, Down, Left])
    90 deg rot.: piece(Piece, [Left, Upper, Right, Down])
    180 deg rot.: piece(Piece, [Down, Left, Upper, Right])
    270 deg rot.: piece(Piece, [Right, Down Left, Upper])

@endyaml
```

```
@startuml

:**solutionDownwards**;
    if (previous row __is__ last bottom row) then (no: continue)
        :- matchRight starting on left margin
          matching upper row shape
        - add pieces from new row
          to list of unavailable pieces
        - solutionDownwards get following row (recursion)
        **return** this row and following rows;
    else (yes: stop)
        : no more rows can be built (puzzle complete)
        **return**  empty list;
    endif
:- list containing current and following rows;
@startuml
```

```
@startuml

:**matchRight**;
    if (current piece == right margin) then (no: continue)
        :- get required Left and Upper shapes
        - search piece with matching shapes
        - check if piece is still available
        - matchRight on new piece (get following matches)(recursion)
        **return** this piece and following matches;
    else (yes: stop)
        :**return**  empty list;
    endif
:- list of pieces forming built row
- bottom shape of built row;
@startuml
```

Call \ PuzzleSize	10	25	50	75	100	125	150
$memberchk/3	10.40%	24.90%	53.00%	71.60%	88.00%	89.40%	91.30%
piece/2	49.30%	37.50%	24.60%	13.40%	5.70%	5.00%	4.50%
matchRight/6	25.40%	5.00%	4.00%	2.60%	0.90%	0.60%	0.40%
pieceSearch/3	0.00%	16.30%	5.40%	2.20%	0.80%	1.00%	0.60%
pSearch/2	4.50%	5.30%	3.50%	3.40%	1.00%	0.90%	0.70%