# A screenshot of a game Description automatically generatedProject Summary

The symbols our project concerns are black and white squares that must be separated such that no two of the same colour are in the same “section” as denoted by the drawn line.

The drawn line must start on the designated start point, and end on the designated end point. The playing grid is limited to 3x3, and there is always at most one blank space in the grid. We will not model these constraints.

For our model, we will use a 2D grid defined on a 3x3 game grid, where 0 ≤ x ≤ 6 and 0 ≤ y ≤ 6. This grid is only defined for discrete values of x and y. Example: On (1,5) there is a white square. On (2,1) there is a line segment. Note that line segments do not need to be defined on intersections or corners.

Image courtesy of witnesspuzzles.com

# Propositions

*List of the propositions used in the model, and their (English) interpretation.*

* lx,y: This is true when a line is on the grid space (x,y)
* wx,y: This is true when space (x,y) on the grid contains a white square
* bx,y: This is true when space (x,y) on the grid contains a black square
* tx,y: This is true when space (x,y) on the grid contains neither a white or black square (empty)
* jx,y: This is true when a black square is touching the space at (x,y)
* kx,y: This is true when a white square is touching the space at(x,y)
* ix,y: This is true when an empty tile (x,y) is touching a black square and a white square
* px,y: This is true when a white square at (x,y) is touching a black square
* sx,y: This is true when space (x,y) on the grid is the starting point of the line
* fx,y: This is true when space (x,y) on the grid is the ending point of the line
* q: This is only true when the drawn line is a solved solution.

# Constraints

*List of constraint types used in the model and their (English) interpretation.*

* A square can be exclusively black or white, not both.
  + (wx,y→¬ bx,y)∧ (bx,y→¬ wx,y)
* A square is “touching” a black square only when a black square is two x or y coordinates away (no diagonals), and on the x or y coordinate between them there is no line segment.
  + jx,y ↔ (bx,y+2 ∧ ¬lx,y+1) ∨ (bx+2,y ∧ ¬lx+1,y) ∨ (bx,y-2 ∧ ¬lx,y-1) ∨ (bx-2,y ∧ ¬lx-1,y)
  + Same idea applies for white squares (kx,y).
* An empty space is only touching a black square and a white square when the point has neither white nor black on it and the previous constraints for touching black and white squares are met.
  + ix,y ↔ ¬ bx,y ∧ ¬ wx,y ∧ jx,y ∧ kx,y
* Similar idea applies for white squares touching black squares.
  + px,y ↔ ¬ bx,y ∧ wx,y ∧ jx,y
* A solution is only solved when no empty spaces are touching black and white squares, and no white squares are touching black squares:
  + q ↔ ¬(i1,1 ∨ i1,3 ∨ i1,5 ∨… ∨ i5,5 ) ∧ ¬(p1,1 ∨ p1,3 ∨ p1,5 ∨… ∨ p5,5 )
* There can only ever be one starting point.
  + (s0,0∧¬ (s2,0∨ s4,0∨…∨ s6,6)) ∨ (s2,0∧¬ (s0,0∨ s4,0∨…∨ s6,6)) ∨ …
  + The same applies to the ending point.
  + The same applies to there being only one empty space.
* The starting point must be on an "intersection" of the line area and cannot be within a tile.
  + ¬s0,1 ∧ ¬s0,3 ∧ ¬s0,5 ∧ ¬s1,0 ∧ ¬s1,2 ∧ ¬s1,4 ∧ ¬s1,6 ∧ … ∧ ¬s6,3 ∧ ¬s6,5
  + Same applies for the ending point.
* A grid space can’t be both the starting point and ending point.
  + ¬sx,y ∨ ¬ex,y
* There must always be a line segment at the starting point and ending point.
  + sx,y → lx,y
  + ex,y → lx,y
* A line segment can never be on the inside of a tile:
  + ¬l1,1 ∧ ¬l1,3 ∧ ¬l1,5 ∧ ¬l3,1 ∧ ¬l3,3 ∧ ¬l3,5 ∧ ¬l5,1 ∧ ¬l5,3 ∧ ¬l5,5
* A grid space can only contain a white square if it is the inside of a tile:
  + ¬w0,0 ∧ ¬w0,1 ∧ … ∧ ¬w0,6 ∧ ¬w1,0 ∧ ¬w1,2 ∧ ¬w1,4 ∧ ¬w1,6 ∧ … ∧ ¬w6,6
  + The same applies for black and empty squares.
* Any point on the line that isn’t the start or end point must be connected to **two** other points of the line. In other words, the line must be a single continuous line from start to end without branching paths.
  + lx,y → ((sx,y ∨ ex,y) ∧ ((lx+1,y ∧ ¬lx-1,y ∧ ¬lx,y+1 ∧ ¬lx,y-1) ∨ … ∨ (¬lx+1,y ∧ ¬lx-1,y ∧ ¬lx,y+1 ∧ lx,y-1))) ∨ (¬sx,y ∧ ¬ex,y ∧ ((lx+1,y ∧ lx-1,y ∧ ¬lx,y+1 ∧ ¬lx,y-1) ∨ (¬lx+1,y ∧ ¬lx-1,y ∧ lx,y+1 ∧ lx,y-1) ∨ … ))

**EXPERIMENTAL** (Not implemented properly, might not work as intended):

* Any point on the line must either be on the starting position or be on a valid grid space for a line segment.
  + lx,y → sx,y ∨ cx,y
* A grid space is a valid route for the line to take if the line has somewhere to go after being added to the grid space.
  + cx,y ↔ ¬lx+1,y ∨ ¬lx,y+1 ∨ ¬lx-1,y ∨ ¬lx,y-1

# Model Exploration

*List all the ways that you have explored your model – not only the final version, but intermediate versions as well. See (C3) in the project description for ideas.*

* Explored how to determine if given a completed grid, whether or not the grid is a solved solution to the puzzle.

# Jape Proof Ideas

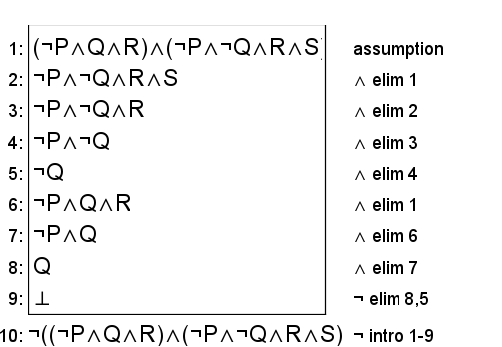
*List the ideas you have to build sequents & proofs that relate to your project.*

First proof:

Relevant constraints:

* An empty space is only touching a black square and a white square when the point has neither white nor black on it and the previous constraints for touching black and white squares are met.
  + ex,y ↔ ¬ bx,y ∧ ¬ wx,y ∧ jx,y ∧ kx,y
* Similar idea applies for white squares touching black squares.
  + px,y ↔ ¬ bx,y ∧ wx,y ∧ jx,y

Proof that it is impossible to have an empty space touching a black square and white square while having that same space be a white square:

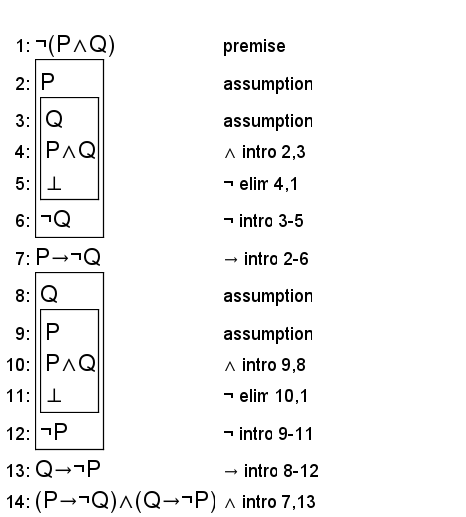


Second proof:

Relevant constraint:

* A grid space can’t be both the starting point and ending point.
  + ¬(sx,y ∧ ex,y)

Proof that a given grid space being the starting point implies that it is not the ending point, and vice versa:

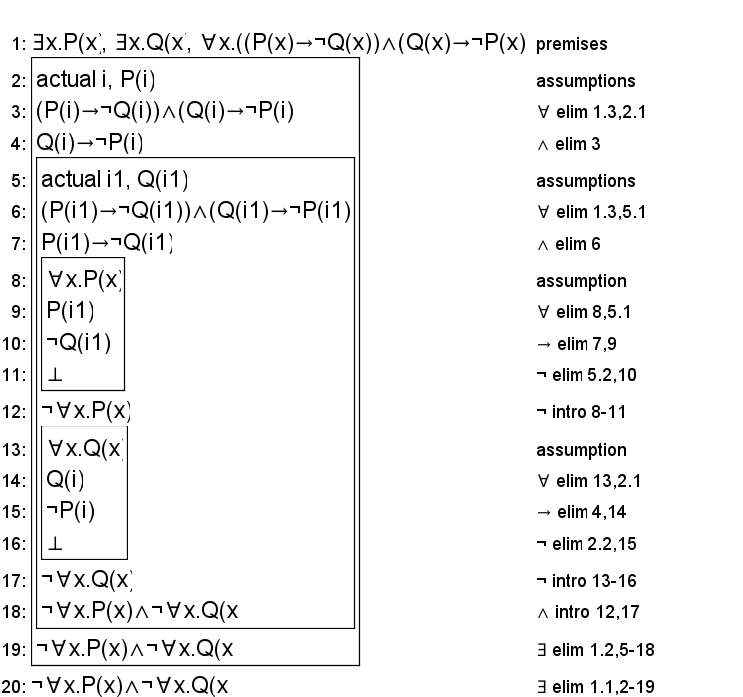


Third proof:

Relevant constraint:

* A square can be exclusively black or white, not both.
  + (wx,y→ ¬bx,y)∧ (bx,y→ ¬wx,y)

Proof that if there exists a white square and a black square (P(x) and Q(x) in this proof)) and that a square being white implies it is not black, and vice versa (shown through the constraint), that it is not possible for all squares to be black and it is not possible for all squares to be white.



# First-Order Extension

*Describe how you might extend your model to a predicate logic setting, including how both the propositions and constraints would be updated.* ***There is no need to implement this extension!***

## Updated Propositions

* L(x,y): This is true when a line is on the grid space (x,y)
* W(x,y): This is true when space (x,y) on the grid contains a white square
* B(x,y): This is true when space (x,y) on the grid contains a black square
* T(x,y): This is true when space (x,y) on the grid contains neither a white or black square (empty)
* J(x,y): This is true when a black square is touching the space at (x,y)
* K(x,y): This is true when a white square is touching the space at(x,y)
* I(x,y): This is true when an empty tile (x,y) is touching a black square and a white square
* P(x,y): This is true when a white square at (x,y) is touching a black square
* S(x,y): This is true when space (x,y) on the grid is the starting point of the line
* F(x,y): This is true when space (x,y) on the grid is the ending point of the line
* Q: This is only true when the drawn line is a solved solution.

## Updated Constraints

* A square can be exclusively black or white, not both.
  + ¬∃(x,y).(W(x,y) ∧ B(x,y))
* A square is “touching” a black square only when a black square is two x or y coordinates away (no diagonals), and on the x or y coordinate between them there is no line segment.
  + ∀(x,y).(J(x,y) ↔ ∃(x,y).((B(x,y+2) ∧ ¬ L(x,y+1)) ∨ (B(x+2,y) ∧ ¬ L(x+1,y)) ∨ (B(x,y-2) ∧ ¬ L(x,y-1)) ∨ (B(x-2,y) ∧ ¬ L(x-1,y))))
  + Same idea applies for white squares K(x,y).
* An empty space is only touching a black square and a white square when the point has neither white nor black on it and the previous constraints for touching black and white squares are met.
  + ∀(x,y).(I(x,y) ↔ ∃(x,y).(¬B(x,y) ∧ ¬W(x,y) ∧ J(x,y) ∧ K(x,y)))
* Similar idea applies for white squares touching black squares.
  + ∀(x,y).(P(x,y)↔ ∃(x,y). (¬ B(x,y) ∧ W(x,y) ∧ J(x,y)))
* A solution is only solved when no empty spaces are touching black and white squares, and no white squares are touching black squares:
  + Q ↔ ¬∃(x,y).I(x,y) ∧ ¬∃(x,y).P(x,y)
* There can only ever be one starting point.
  + ¬∃(x1,y1),(x2,y2).(S(x1,y1) ∧ S(x2,y2)) ∧ ∃(x,y).S(x,y)
  + The same applies to the ending point.
  + The same applies to there being only one empty space.
* The starting point must be on an "intersection" of the line area and cannot be within a tile.
  + ¬∃(x,y).(S(1,y) ∨ S(3,y) ∨ S(5,y) ∨ S(x,1) ∨ S(x,3) ∨ S(x,5))
  + Same applies for the ending point.
* A grid space can’t be both the starting point and ending point.
  + ¬∃(x,y).(S(x,y) ∧ F(x,y))
* There must always be a line segment at the starting point and ending point.
  + ¬∃(x,y).(S(x,y) ∧ ¬ L(x,y))
  + ¬∃(x,y).(F(x,y) ∧ ¬ L(x,y))
* A line segment can never be on the inside of a tile:
  + ¬L(1,1)∧ ¬L(1,3)∧ ¬L(1,5)∧ ¬L(3,1)∧ ¬L(3,3)∧ ¬L(3,5) ∧ ¬L(5,1)∧ ¬L(5,3)∧ ¬L(5,5)
* A grid space can only contain a white square if it is the inside of a tile:
  + ¬∃(x,y).(W(0,y) ∨ W(2,y) ∨ W(4,y) ∨ W(6,y) ∨ W(x,0) ∨ W(x,2) ∨ W(x,4) ∨ W(x,6))
  + The same applies for black and empty squares.
* Any point on the line that isn’t the start or end point must be connected to **two** other points of the line. In other words, the line must be a single continuous line from start to end without branching paths.
  + ∀(x,y).(L(x,y) → ((S(x,y) ∨ F(x,y)) ∧ ((L(x+1,y) ∧ ¬ L(x-1,y) ∧ ¬ L(x,y+1) ∧ ¬ L(x,y-1)) ∨ … ∨ (¬ L(x+1,y) ∧ ¬ L(x-1,y) ∧ ¬ L(x,y+1) ∧ L(x,y-1)))) ∨ (¬ S(x,y) ∧ ¬ F(x,y) ∧ ((L(x+1,y) ∧ L(x-1,y) ∧ ¬ L(x,y+1) ∧ ¬ L(x,y-1)) ∨ (¬ L(x+1,y) ∧ ¬ L(x-1,y) ∧ L(x,y+1) ∧ L(x,y-1)) ∨ … )))
* A grid space is a valid route for the line to take if the line has somewhere to go after being added to the grid space.
  + ∀(x,y).(C(x,y) ↔ ¬L(x+1,y) ∨ ¬L(x,y+1) ∨ ¬L(x-1,y) ∨ ¬L(x,y-1))
* Any point on the line must either be on the starting position or be on a valid grid space for a line segment.
  + ∀(x,y).(L(x,y) → S(x,y) ∨ C(x,y))

# Requested Feedback

*Provide 2-3 questions you’d like the TA’s and other students to comment on.*

1. One big problem we are having right now is making sure that the line is only a single continuous line from start to finish. Our constraints are allowing for closed loops of line segments that should not be possible given our intended rules of the model. How could we develop a constraint that removes the possibility of closed loops and forces the line to be a single continuous line using logic?
2. How would we go about exploring the model in different ways. Currently, our model generates an entire grid that is always solved. How would we model our problem in different ways?
3. Currently, the Jape proofs do not provide very useful results. What are some other properties that could arise from our model that would be good to prove in Jape?