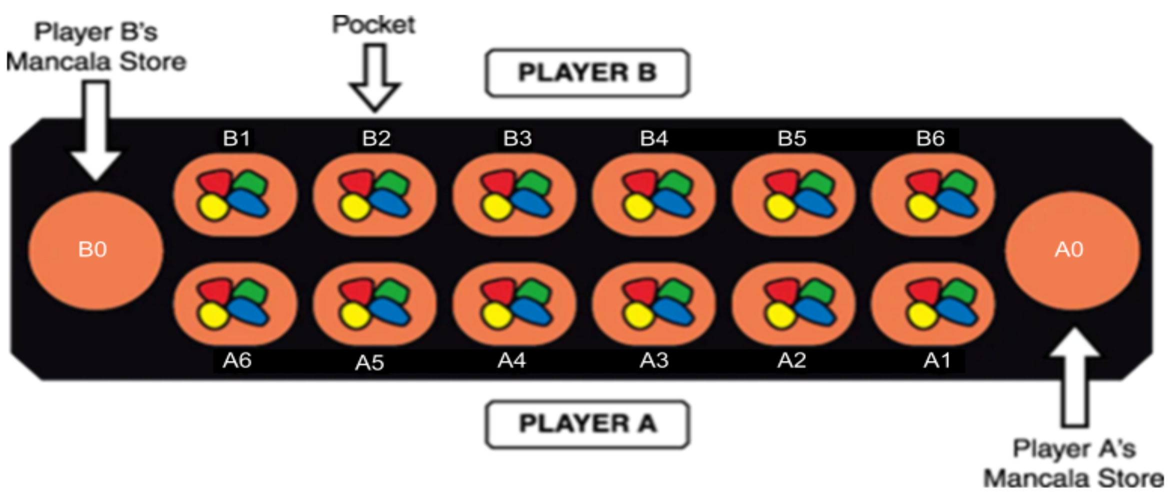
Hi Group 14, this is your TA! You can see my comments below :)

Feel free to reach out if you have any questions, or just want to run some ideas by me!

Project Summary

Mancala Description: Mancala is an ancient boardgame where you strategically 'sow' and 'capture' until you run out of pieces. Played by two people, each player tries to get a higher score by strategically picking and placing game pieces (traditionally stones, but any small object works) around the playing area, which is made up of 12 pits or pockets (six per side) and two stores (larger pockets at the end of each board). To set the game up, four pieces are placed in each of the 12 pits on the board. Once it has been decided who goes first, that player picks up all the pieces from one of the pockets on their side of the board and moving one place to the right, begins depositing a piece in each pocket until they have no more pieces left in their hand. If the player's last deposit lands in their store, they get another turn. Pieces are not deposited in the opponent's store. Similarly, if the player's last deposit lands in an empty pit on their side of the board, they claim all the pieces in pit directly opposite theirs. The game ends when either of the players has six empty pits on their side, winner is determined by who has the most pieces in their respective store.



For our project, we will be modelling the optimal next move for player A, given a randomized board state.

Propositions

Notation: Position is denoted by the variables 'r' and 'c' where 'r' represents the side of the board (A or B) and 'c' represents the columns (0 through 6 where 0 is the player's store and 6 is the furthest pit from the respective store). The number of gems in a selected Finally, x represents a player (A or B).

This is good - might I suggest changing the format to make these explanations more concise?

Ex. c: represents the columns (0-6) such that 0 <= c <= 6.

x: represents which player is currently playing, such that x must equal either "A" or "B".

Modelling plausible game state

- How many gems are in a hand and what pit a gem will be dropped in
    - $H_{g, r, c}$ = True if there are > 1 gems in player A's hand

You should also account for the other constraints on this variable -> it wouldn't be true if c = 7.

- The side of the selected pit
    - S_r = True if pit is on player A's side

If 'S' indicates that the pit is on player "A's" side, then the presence of 'r' is not required as it adds complexity. What I mean by this is, you can represent the pit being on player "A" or "B's" side using S and ~S.

- Position of the pit and how many gems it contains
    - $P_{g, r, c}$ = True if there are gems in the selected pocket

Account for other constraints.

- Identifying which player gets another turn
    - T_x = True if player x gets another turn
- Determining if the player collects gems
    - $C_{g, r, c}$ = True if player A collects gems from the indicated pit

Account for other constraints.

- Determining if two pits are opposite of each other
    - O_c = True if the columns in row A and B are the same

Constraints

Board specific constraints

- Player A can only move gems on their side of the board
    - $H_{g, a, c} \land S_r \rightarrow P_{g, A, c}$

I get what you mean. However, this applies to both Player A and B. You might want to generalize this proposition slightly.

- Player A cannot drop a gem in Player B's store
    - $H_{g, A, c} \rightarrow \neg S_A$
- A pocket must have seeds in it

- $P_{g,\, r,\, c} \rightarrow H_{g,\, r,\, c}$

<u>Game constraints</u>

- Dropping a gem in a pit
  - $H_{g,\, r,\, c} \rightarrow H_{g-1,\, r,\, c-1} \land P_{g+1,\, r,\, c-1}$
- If the final gem is dropped in player A's store, then player A gets another turn
  - $H_{0,\, A,\, 0} \rightarrow T_A$
- Player A collects seeds from player B's pit if Player A's last seed is dropped in the opposite pit
  - $(H_{0,\, A,\, c} \land P_{1,\, A,\, c}) \land \neg P_{0,\, B,7-c} \rightarrow C_{g,\, B,\, c}$

Model Exploration

Mancala is a game that requires you to anticipate your opponent's moves and set up future plays for yourself accordingly. Since we limited the scope of our model to a single step play, it became difficult for us to figure out how to add in this anticipation element. To combat this, we've designed our model to analyse the following four strategies (listed in order of importance):

1. Putting the final seed in the player's store
2. Block the opponent from putting a seed in their store
3. Put the player's stone into the opponent's empty pit
4. Empty the player's pit
5. *If all else fails, play the right most pit*

How are these going to be measured in your model in order to garner a conclusion? Will there be a points system?

Given a randomized board state, the model will run through those four propositions. The proposition with the highest truth value will be selected as the optimal next move.

Jape Proof Ideas

Our first proof is to see if selecting a certain pit for example pit a3 with 3 gems in it would lead to the player getting another turn. For the current game state, we assume no other pit other than the one we mentioned would contain at least 1 gem. We model it using this:

Sa4, Sa4→H3a3, H3a3→H2a2∧P1a2, H2a2→H1a1∧P1a1, H1a1→H0a0∧P1a0, H0a0→A ⊢Sa3→A

Our second proof is the same as before, but if selecting pit a6 with 1 gem in it would lead to collecting the gems from pit b2. This is modelled with this proof:

Sa6, Sa6 →H1a6, H1a6→H0a5∧P1a5,H0a5∧P1a5∧¬P0b2→ C⊢Sa1→C

Our final proof is proving that selecting pit a1 (which would have 4 gems) would prevent the other player from selecting pit b6 (which would have 2 initially) to collect from a3 (which would have 2). Like the other proofs, we assume no other pit other than the ones we mentioned would contain a gem. This is the proof:

Sa1,Sa1→H4a1,H4a1→H3a0∧P1a0, H3a0→H2b6∧P3b6, H2b6→H1b5∧P1b5, H1b5→H0b4∧P1b4, Sb6, Sb6→H3b6, H3b6→H2b5∧P2b5,H2b5→H1b4∧P2b4,H1b4→H0b3∧P1b3, (H0b4 ∧ P1b4)→C ⊢Sb6→¬C

Requested Feedback

1. As this project stands, the logic model only counts if there is (or isn't) a seed in hand. I.e.: if there is at least one seed in hand the proposition is true, else false. Does this run into any perceivable logical errors?

I am not sure that I am interpreting the question correctly. Since you have a proposition that holds the number of gems that a player collects, I believe it is sufficient.

2. Following this project's line of logic, do the four constraints seem complete and thorough? Is there any areas where the constraints may be lacking?

As touched on slightly in the constraints section, your constraints would be more accurate if the were generalized - not specifically cover Player A.

3. Overall, on the topic of project complexity; does this project seem to be in the correct scope for this assignment?

If you can find a way to measure the optimization of a single turn, then I believe the scope of this project is great!

First-Order Extension

The inclusion of predicate logic settings would be used to show the ideal move that would be achieved, as above. This would be by stating that there is a seed that exists that fulfills the most important strategy. For example, we could extend the first-order to say: there is a pocket that will result in the final seed being dropped in the player store, therefore satisfying strategy one. Or, all pockets would be ineffective in satisfying strategy one, and strategy two, strategy three, strategy four, therefore the rightmost pit would be played. This would not require big changes in the constraints, but the propositions would have to include the strategies as listed above.