CISC/CMPE 204: Modelling Project
{GRP ID: 23}

Project Summary

Short summary of the project setting.

Our project strives to model the popular and fun game three card poker. The model determines which hand wins the round.

The model also provides suggestions on if the player should play the hand or fold according to their hand.

The game involves two players: the user and the dealer

Rules:

Three cards are dealt to the user and three cards are dealt to the user

The user can only see their three cards, they must decide whether they want to play the game or fold. If they play the round, the dealer must also play the round. If they fold, the dealer wins the pot.

The winner of the round is the player with the highest ranking. The order of the rankings is straight flush, three of a kind, straight, flush, pair, and three odd cards (high card).

Propositions

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

-**Hand:** Let x1, x2, x3, be the three cards in the user's hand

-N2 is true if x1 is a 2, …., N11 is true if x1 is a jack, N12 is true if x2 is a queen, N13 is true if x1 is a king, N14 is true if x14 is an ace

-SS is true if x1's suit is a spade, SC is true if x1 is a club, SD is true if x1 is a diamond, and SH is true if x1 is a heart

-Similarly, let y1, y2, y3 be the three cards in the dealer's hand, which follow the same Numbering and Suit principal as the user's hand.

¡**Hand Rankings**: SF is true if the hand is a straight flush, TK is true if the hand is a three of a kind, S is true if the hand is a straight, F is true if the hand has a flush, P is true if the hand has a pair, H is true if the hand is a high card **CHANGED THREE OF A KIND TO TK IN SOME PLACES

¡Would also need to know if two players have a high card hand, who's hand is higher, etc.

¡For example, if the user has at least a queen as their highest card, at least a 6 as their second highest card, and at least a 4 as their third highest card then they should play the hand

¡**Win/Lose**: W is true if user wins the hand, ~W if user loses the hand

**CREATE A PROPOSITION FOR HIGH CARD

Constraints

List of constraint types used in the model and their (English) interpretation. You only need to provide one example for each constraint type: e.g., if you have constraints saying "cars have one colour assigned" in a car configuration setting, then you only need to show the constraints for a single car. Essentially, we want to see the pattern for all of the types of constraints, and not every constraint enumerated.

Hand Constraints: Each hand must consist of exactly three cards, the rank of the hand is determined by standard three card poker hand rankings (Ex. $N1(x1)_U \wedge N1(x2)_U \wedge \sim N1(x3)_U \rightarrow P$ demonstrates that the user has a pair of aces) *Note: Subscript u means user

¡*REMINDER: No duplicate cards within the dealers three cards and the users three cards (ex. Only one King of Spades)*

¡*Ensure that all cards are valid card in a standard 52 card deck*

¡*A hand is only valid if it has exactly three cards:* $HAND_U \leftrightarrow (x1 \wedge x2 \wedge x3)$

***HAND IS ONLY ASSIGNED ONE HAND RANKING*

Winning Constraints: The player with the highest ranking hand wins, if the hands have equal value, then the dealer wins the tie

¡*Ex.* $(SF(x1, x2, x3) \wedge \sim SF(y1, y2, y3)) \vee (T(x1, x2, x3) \wedge \sim T(y1, y2, y3)) \vee (S(x1, x2, x3) \wedge \sim S(y1, y2, y3)) \vee (F(x1, x2, x3) \wedge \sim F(y1, y2, y3)) \vee (P(x1, x2, x3) \wedge \sim P(y1, y2, y3)) \vee (H(x1, x2, x3) \wedge \sim H(y1, y2, y3)) \rightarrow W_U$

***NOTE: ACE IS HIGH*

Determining Hand Rankings

**USER CAN HAVE FLUSH AND PAIR (CONSIDER THIS WHEN DETERMINE HAND RANKINGS). DETERMINE HAND RANKINGS THEN COMPARE?

ORDER GOES SF, T, S, F, P, H

Check in order so selects highest hand as hand ranking (ex. Pair and three of a kind selects three of a kind)

Will have to check for both dealer and user

Also, need to save high card value somehow (could be at a different step)

**Could also order it as if not higher rankings and ranking is true than imply that ranking!!

****MUST ORDER FIRST .sort() (Let x1 be lowest card, then x2, then x3 highest)**

DETERMINE HIGH CARD AND ORDER NUMBERS:

$N14(x1) \mid N14(x2) \mid N14(x3) \rightarrow HC14$

$N13(x1) \mid N13(x2) \mid N13(x3) \wedge \sim(N14(x1) \mid N14(x2) \mid N14(x3)) \rightarrow HC13$

$N12(x1) \mid N12(x2) \mid N12(x3) \wedge \sim(N14(x1) \mid N14(x2) \mid N14(x3)) \wedge \sim(N13(x1) \mid N13(x2) \mid N13(x3)) \rightarrow HC12$

$N11(x1) \mid N11(x2) \mid N11(x3) \wedge \sim(N14(x1) \mid N14(x2) \mid N14(x3)) \wedge \sim(N13(x1) \mid N13(x2) \mid N13(x3) \wedge \sim(N12(x1) \mid N12(x2) \mid N12(x3)) \rightarrow HC11$

**CONTINUE DOWN TO 2

******NOT SURE HOW TO DO THIS USING PROPOSITIONAL LOGIC*

INSTEAD OF IF AND ELSE USE NOT RANKINGS THAT ARE HIGHER AND THAT RANKING

Step 1: Check if a straight flush (three numbers in a row and all cards same suit)

***MUST ORDER CARDS BY NUMBER BEFORE CHECK IF IN A ROW (AND DETERMINE HC) .sort()*

Case 1: Three Spades in a row

$(SS(X1) \wedge SS(X2) \wedge SS(X3)) \wedge (N(X1) + 1 = N(X2) \wedge N(X2) + 1 = N(X3)) \rightarrow SF$

Case 2: Three Clubs in a row

$(SC(X1) \wedge SC(X2) \wedge SC(X3)) \wedge (N(X1) + 1 = N(X2) \wedge N(X2) + 1 = N(X3)) \rightarrow SF$

Case 3: Three diamonds in a row

$(SD(X1) \wedge SD(X2) \wedge SD(X3)) \wedge (N(X1) + 1 = N(X2) \wedge N(X2) + 1 = N(X3)) \rightarrow SF$

Case 4: Four hearts in a row

$(SH(X1) \wedge SH(X2) \wedge SH(X3)) \wedge (N(X1) + 1 = N(X2) \wedge N(X2) + 1 = N(X3)) \rightarrow SF$

Step 2: Else check if three of a kind (three of the same number)

Case 1 -Three two's: $(\sim SF \wedge N2(X1) \wedge N2(X2) \wedge N2(X3)) \rightarrow TK$

Case 2 - Three three's: $(\sim SF \wedge N3(X1) \wedge N3(X2) \wedge N3(X3)) \rightarrow TK$

Case 3: $(\sim SF \wedge N4(X1) \wedge N4(X2) \wedge N4(X3)) \rightarrow TK$

Case 4: $(\sim SF \wedge N5(X1) \wedge N5(X2) \wedge N5(X3)) \rightarrow TK$

Case 5: $(\sim SF \wedge N6(X1) \wedge N6(X2) \wedge N6(X3)) \rightarrow TK$

Case 6: $(\sim SF \wedge N7(X1) \wedge N7(X2) \wedge N7(X3)) \rightarrow TK$

Case 7: $(\sim SF \wedge N8(X1) \wedge N8(X2) \wedge N8(X3)) \rightarrow TK$

Case 8: $(\sim SF \wedge N9(X1) \wedge N9(X2) \wedge N9(X3)) \rightarrow TK$

Case 9 - Three tens: $(\sim SF \wedge N10(X1) \wedge N10(X2) \wedge N10(X3)) \rightarrow TK$

Case 10 - Three jacks$(\sim SF \wedge N11(X1) \wedge N11(X2) \wedge N11(X3)) \rightarrow TK$

Case 11 - Three queens$(\sim SF \wedge N12(X1) \wedge N12(X2) \wedge N12(X3)) \rightarrow TK$

Case 12 - Three kings$(\sim SF \wedge N13(X1) \wedge N13(X2) \wedge N13(X3)) \rightarrow TK$

Case 13 - Three aces$(\sim SF \wedge N14(X1) \wedge N14(X2) \wedge N14(X3)) \rightarrow TK$

Step 3: Else check if straight (three numbers in a row) **MUST ORDER FIRST .sort()

$(\sim SF \wedge \sim TK \wedge N(X1) + 1 = N(X2) \wedge N(X2) + 1 = N(X3)) \rightarrow S$

Step 4: Else check if flush (three of a suit)

Case 1: Three Spades

$(\sim SF \wedge \sim TK \wedge \sim S \wedge SS(X1) \wedge SS(X2) \wedge SS(X3)) \rightarrow F$

Case 2: Three Clubs

$(\sim SF \wedge \sim TK \wedge \sim S \wedge SC(X1) \wedge SC(X2) \wedge SC(X3)) \rightarrow F$

Case 3: Three diamonds

$(\sim SF \wedge \sim TK \wedge \sim S \wedge SD(X1) \wedge SD(X2) \wedge SD(X3)) \rightarrow F$

Case 4: Four hearts

$(\sim SF \wedge \sim TK \wedge \sim S \wedge SH(X1) \wedge SH(X2) \wedge SH(X3)) \rightarrow F$

Step 5: Else check if pair (two of a number)

***ORDER THE CARDS AND THEN PAIR IS ALWAYS MIDDLE*

$(\sim SF \wedge \sim TK \wedge \sim S \wedge \sim F \wedge N1(x1)_U \wedge N1(x2)_U \wedge \sim N1(x3)_U) \rightarrow P$

$(\sim SF \wedge \sim TK \wedge \sim S \wedge \sim F \wedge \sim N1(x1)_U \wedge N1(x2)_U \wedge N1(x3)_U) \rightarrow P$

Step 6: Else set the hand ranking to high card

$(\sim SF \wedge \sim TK \wedge \sim S \wedge \sim F \wedge \sim P) \rightarrow H$

**PLAY OR FOLD RECOMMENDATION

"The usual strategy for when to make the Play wager is to bet whenever you have Queen-6-4 or better, including any time your high card is an Ace or a King, no matter how high your other two cards are, and also anytime your hand is Q-7 or better, regardless of your third card."

**CONTINUE TO USE THE ORDERED CARDS (x1 low, x2 middle, x3 high)

Let RP be true if recommend play, ~RP if recommend play

High Card is ace or king: $N14(x1) \mid N13(x1) \rightarrow RP$

Hand is queen-6-4 or better: (N14(x1) | N13(x1) | N12(x1)) ∧ (N14(x2) | N13(x2) | N12(x2) | N11(x2) | N10(x2) | N9(x2) | N8(x2) | N7(x2) | N6(x2)) ∧ (~N2(x3) ∧ ~N3(x3)) → RP

Hand is Queen-7 or better: (N14(x1) | N13(x1) | N12(x1)) ∧ (N14(x2) | N13(x2) | N12(x2) | N11(x2) | N10(x2) | N9(x2) | N8(x2) | N7(x2)) → RP

**An interesting proof would be why play when high card is ace or king

Determine Winner

IF HANDS ARE SAME RANK, CHECK HIGH CARD

TRACK HIGH CARD USER (HCU) and High card dealer (HCD)

*COULD SIMPLIFY FROM H(x1, x2, x3) to H(x)

Condition: User has a straight flush

(SF(x1, x2, x3) ∧ ~SF(y1, y2, y3)) → W_U

(SF(x1, x2, x3) ∧ SF(y1, y2, y3) ^ (HCU > HCD) → W_U

SF(x1, x2, x3) ∧ SF(y1, y2, y3) ^ (HCU < HCD) → L_U

Condition: User has a three of a kind

(T(x1, x2, x3) ∧ ~T(y1, y2, y3) ∧ ~SF(y1, y2, y3)) → W_U

(T(x1, x2, x3) ∧ T(y1, y2, y3) ∧ ~SF(y1, y2, y3) ^ (HCU > HCD)) → W_U **MAY NOT NEED SF BECAUSE CANT HAVE BOTH

(T(x1, x2, x3) ∧ T(y1, y2, y3) ∧ ~SF(y1, y2, y3) ^ (HCU < HCD)) → LU

Condition: User has a straight

(S(x1, x2, x3) ∧ ~S(y1, y2, y3) ∧ ~SF(y1, y2, y3) ∧ ~T(y1, y2, y3)) → W_U

(S(x1, x2, x3) ∧ S(y1, y2, y3) ∧ ~SF(y1, y2, y3) ∧ ~T(y1, y2, y3) ^ (HCU > HCD)) → W_U

(S(x1, x2, x3) ∧ S(y1, y2, y3) ∧ ~SF(y1, y2, y3) ∧ ~T(y1, y2, y3) ^ (HCU < HCD)) → L_U

Condition: User has a flush (3 cards same suit)

(F(x1, x2, x3) ∧ ~F(y1, y2, y3) ∧ ~S(y1, y2, y3) ∧ ~SF(y1, y2, y3) ∧ ~T(y1, y2, y3)) → W_U

(F(x1, x2, x3) ∧ ~F(y1, y2, y3) ∧ ~S(y1, y2, y3) ∧ ~SF(y1, y2, y3) ∧ ~T(y1, y2, y3) ^ (HCU > HCD)) → W_U

(F(x1, x2, x3) ∧ ~F(y1, y2, y3) ∧ ~S(y1, y2, y3) ∧ ~SF(y1, y2, y3) ∧ ~T(y1, y2, y3) ^ (HCU < HCD)) → L_U

Condition: User has a pair

**USER CAN HAVE FLUSH AND PAIR (CONSIDER THIS WHEN DETERMINE HAND RANKINGS). DETERMINE HAND RANKINGS THEN COMPARE?

(P(x1, x2, x3) ∧ ~P(y1, y2, y3) ∧ ~F(y1, y2, y3) ∧ ~S(y1, y2, y3) ∧ ~SF(y1, y2, y3) ∧ ~T(y1, y2, y3)) → W_U

(P(x1, x2, x3) ∧ ~P(y1, y2, y3) ∧ ~F(y1, y2, y3) ∧ ~S(y1, y2, y3) ∧ ~SF(y1, y2, y3) ∧ ~T(y1, y2, y3) ^ (HCU > HCD)) → W_U

(P(x1, x2, x3) ∧ ~P(y1, y2, y3) ∧ ~F(y1, y2, y3) ∧ ~S(y1, y2, y3) ∧ ~SF(y1, y2, y3) ∧ ~T(y1, y2, y3) ^ (HCU < HCD)) → L_U

Condition: User has a high card
(H(x1, x2, x3) ∧ ~P(y1, y2, y3) ∧ ~F(y1, y2, y3) ∧ ~S(y1, y2, y3) ∧ ~SF(y1, y2, y3) ∧ ~T(y1, y2, y3)) → W_U

THIS IS SAME AS:
(H(x1, x2, x3) ∧ H(y1, y2, y3) ^ (HCU > HCD)) → W_U
(H(x1, x2, x3) ∧ H(y1, y2, y3) ^ (HCU < HCD)) → L_U

**DO SAME BUT REVERSE FOR DEALER WIN

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.

We have considered how to determine the hand rankings using propositional logic. Then, we have used our hand rankings to determine the player with the best hand using propositional logic. We have also created recommendations for when the player should play the hand or fold the hand. We have considered using JAPE proofs to demonstrate why you should play or fold on a hand. Furthermore, we have considered multiple ways in which we could represent a hand using propositions. For example, P can be true if the hand has a pair. Additionally, 8C could be true if a card is the 8 of clubs. We have not completed the code for our game so we have not been able to determine the accuracy of the model at determining the winner or the accuracy of the model for recommending whether the user should fold or play the hand.

Jape Proof Ideas

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

Requested Feedback

Insight on how to implement high card value/ranking using propositional logic.

1)Are we coding our propositions and constraints in the right way? Should we have a proposition to represent each card in the decl (ex. 2D true if card is two of diamonds, KH true if card is two of hearts, etc.)?

2) What should the final product output? What should output look like at the end of runtime?
Are we seeking a specific goal of what should be outputted? Overall, what is our goal! For example, should we turn it into a game that the user can play and make the decision if they want to play or not based on our recommendation?

3) Are we thinking too narrowly, should we be taking into account cards of potential other players / dealer.
Right now we base our decision off 1 user's hand. Should we implement other users? Are we following the appropriate path? The user currently only has their three cards to make their decision, does this make our logic too simple?

3i) are we able to meet virtually to discuss in further detail?

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!

We can extend our model to a predicate logic setting. First, we can examine how predicate logic would change our winning constraints. For example, we could say that for all hands where the user has a flush there exists some hand that beats this hand. Let x be a hand, let H be the user, let F be a flush, let y be the dealers hand This would look like $\forall x[(H(x) \land H(F)) \rightarrow \exists y(y > x)]$. Extending the model to predicate logic would also change our propositions. For example, we currently have let P be true if the user has a pair. We could think of this as if the user has some card rank x then there exists some other card y where y and x are the same card and the user has a pair. $\forall x[U(x) \rightarrow \exists y(y = x)]$. However, we must also consider that the dealer may have the cards that create the pair.

Useful Notation

Feel free to copy/paste the symbols here and remove this section before submitting.

$\land$
$\lor$
$\neg$
$\rightarrow$
$\forall$
$\exists$