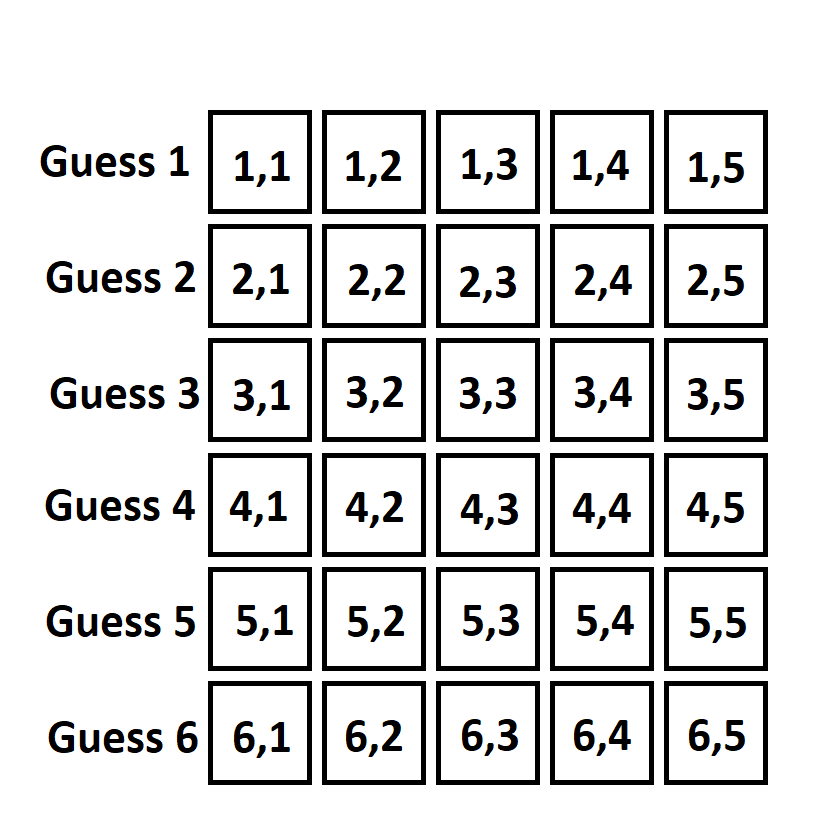
# Project Summary



Our goal is to find an optimal solution to wordle using logical propositions. We will have a randomly sized dictionary of words to choose a word and have our logical proposition find words to try out and using the clues the logical proposition will search for better words to guess.

# Propositions

Wordle is a 5 letter word guessing game with 6 tries which equates to 30 slots to put letters in. Each of the letters can be put into 4 different states . (x, y) denotes the location on the board.



ROW = [1, 2, 3, 4, 5, 6]

COL = [1, 2, 3, 4, 5]

STATUS = ["CORRECT", "INCORRECT", "PARTIAL", "EMPTY"]

LETTER = ["A”...”Z”, “BLANK”]

* Slots(r, c) Each row has a column
* SlotStatus(r, c, s) each slot has a status
* SlotLetter (r, c , l) each slot has a letter



# Constraints

Basic constraints

* Each row maps to every column
* Each slot would have exactly one status
* Each slot would have exactly one letter

Advanced Constraints

* If the letter is blank then the status must be empty and vice versa also the entire row is blank and empty

Solving Constraints

* If we run out of letters in letter bank allow duplicate letters
* If there are 5 partially correct and totally correct letters start trying to guess the word
* If the letter is not in the correct word, then all words in the word list with the incorrect letter should be removed.
* If the correct letter is in the right position, then remove all words without the correct letter from word list.

Good Guess – A guess that includes 5 different letters, and every guess after that includes different letters to determine all 5 letters necessary to solve said Wordle. Exclude words with letters that have si as true (meaning this chosen slot and letter are both incorrect and unnecessary to solve the Wordle). Exclude words with letters that have sp as true in that position and set words that have those letters aside into the candidate list however, a possible problem that comes into play with this is the possibility of repeating letters, however, a solution is to reduce the set of words in the word bank to those without repeating letters. Exclude words with letters that have sc as true and set words that have those letters in the position aside into candidate list.

Bad guesses – Repeated words or unnecessarily repeating letters / already determined partially correct letters. The goal is to not reuse characters that have already been deemed incorrect and to use the other characters left in the set of letters to determine a proper guess.

# Model Exploration

Luck based cases

When there are at least 5 sp and sc together as true it will switch strategies and start to attempt to guess the word. This is under the impression the letters and possible positions have somehow been found before the 6th guess, any guess from 1st-4th.

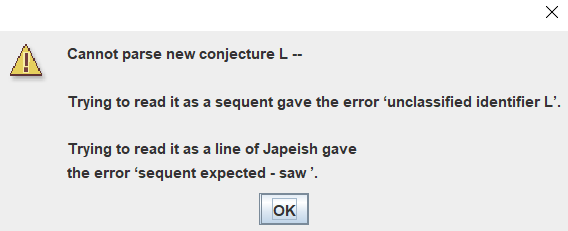
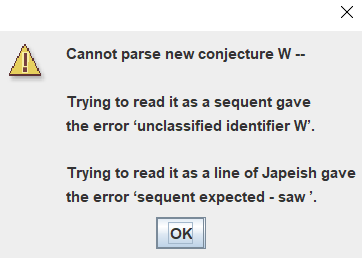
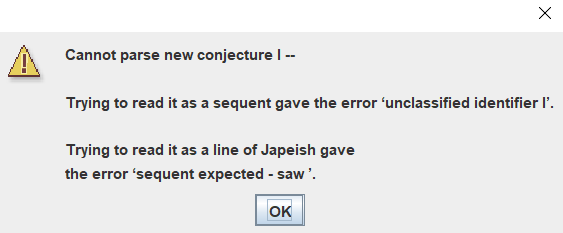
Last guess

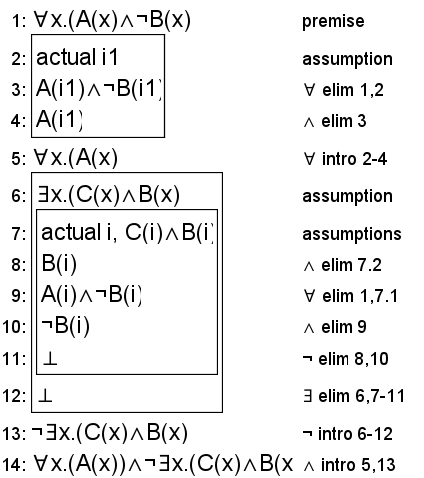
The last guess will always try to guess the word while previous guesses will try to gather information, the first 5 guesses will determine the 5 letters necessary to solve the wordle, whereas the sixth guess has a goal of using the letters and information found in pervious guesses to rearrange and solve the given wordle.

The running out case

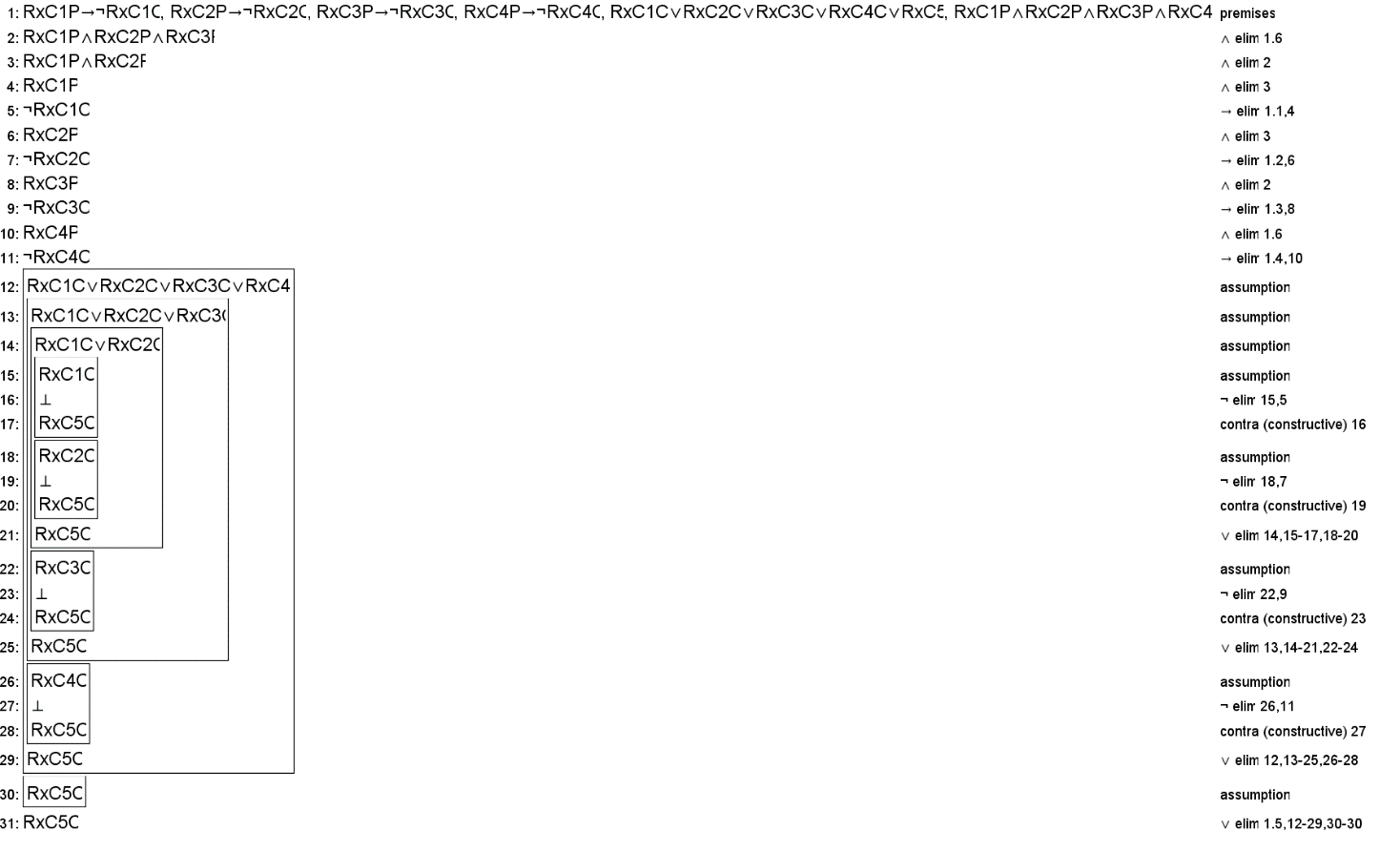
Since we are using a randomly generated list, we may run out of valid words to guess that have all different letters and such depending on the size of the list. So, we may not be able to guess a word with 5 distinct letters not including the letters that were already guessed. This means we would first introduce words with the partially correct letters in different placements along with the rest of the letters we haven't tried out.

# Jape Proof Ideas

* Proof 1 ∀x.(A(x)∧¬B(x))⊢∀x.(A(x))∧¬∃x.(C(x)∧B(x)): If we remove words with incorrect letters Then there will be no words with incorrect letters
  + A = Words
  + B = Letters
  + C = Incorrect letters
  + (W, L, I are illegal arguments for jape apparently)



* Proof 2, If a partially correct letter is in slot 1,2,3 an slot 4 in our guesses then it must be correct slot 5, RxC1P→¬RxC1C,RxC2P→¬RxC2C,RxC3P→¬RxC3C,RxC4P→¬RxC4C,RxC1C∨RxC2C∨RxC3C∨RxC4C∨RxC5C,RxC1P∧RxC2P∧RxC3P∧RxC4P⊢RxC5C
* RxC1P→¬RxC1C,RxC2P→¬RxC2C,RxC3P→¬RxC3C,RxC4P→¬RxC4C : If there is a partially correct slot at Row x col x then that slot cannot be correct.
* RxC1C∨RxC2C∨RxC3C∨RxC4C∨RxC5C : For the letter it could be correct in slot 1, 2, 3, 4, or 5
* RxC1P∧RxC2P∧RxC3P∧RxC4P: The letter has appeared partially correct in slot 1, 2, 3, and 4
* ⊢RxC5: Having 4 partially correct letters in different slots means the last slot is a correct letter.



* Proof 3: If there exist a correct letter and it is in the correct slot, then all other letters cannot go into that slot. ∃y.(B(1)∧RxCxC(x))⊢¬∀y.(B(y)∧¬B(1))∧RxCxC(x)

B= Letter

R= Row

C=Col

Second C = Correct

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

# 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:**

**E(s):** Means that the slot (s) within the guess is empty

**I(s):** Means that the slot (s) within the guess is incorrect (wrong letter)

**P(s):** Means that the slot (s) within the guess is partially correct (wrong slot, right letter)

**C(s):** Means that the slot (s) within the guess is correct (correct letter in correct slot)

**U(s):** Means that the slot (s) is a unique letter compared to the other slots inside the guess

**W(g):** Means that the guess (g) is the correct word from the word bank

**L(g):** Means that the guess (g) is a word inside the candidate list of possible words

**G(g):** Means that the guess (g) is a “good guess”

**B(g):** Means that the guess (g) is a “bad guess”

**R(g):** Means that the guess (g) is a guess that has already been made

**Updated Constraints:**

– There exists a correct word from the word bank (AKA a possible answer)

– There exists a guess where all slots are correct slots, implying that the guess is the correct one from the word bank

– If a guess is repeated, there are no correct slots or there are slots that are incorrect/empty/partially correct.

– For all slots, each slot is unique in comparison to the other slots inside the guess and the guess is not considered a “bad guess”. This implies it is a “good guess”.