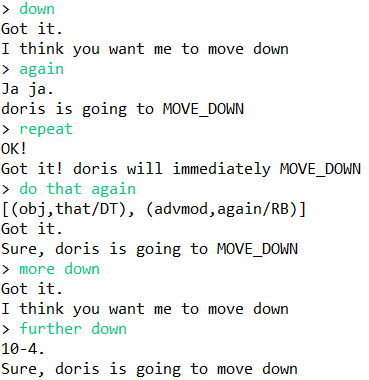
**CSSE 413: Artificial Intelligence**

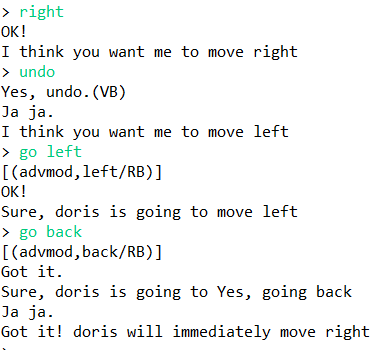
Dialog NLP Lab Manual

Name: \_\_\_\_\_Doris Chen\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

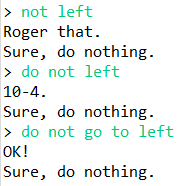
1. **Parsing**.
   1. [8 pts] The ability to tell the robot to repeat a command, such as: "again," "do that again," "more left," "further left."



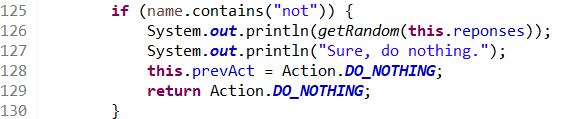
* 1. [8 pts] The ability to tell the robot to undo the prior move command, such as: "undo," "go back."



* 1. [10 pts] The ability to deal with negation. If the user inputs "not left," or "do not go left," etc. the robot should not go left. Explain how you implemented this feature.



I simply used a String.contains() method at where the input is a string instead of a semantic graph, and I checked if there is “not” in the command. If there is, then do nothing.



* 1. [30 pts] Your software should be able to parse any sentence in which the VB is "clean" or "move", but only those sentences that make sense. Please explain how you implemented this feature.

I looked up the graph structure when I typed these sentences:

> Please move to the right space.

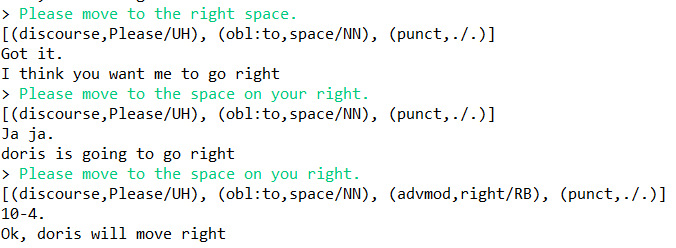
> Please move to the space on your right.

> Please move to the space on you right.

“clean” or “move” is either in the pair of the root, or in the pair of children of root.

For 1) “clean” or “move” is in the pair of the root, I looked for “clean” and “move” in looping through second one of childPairs(root)

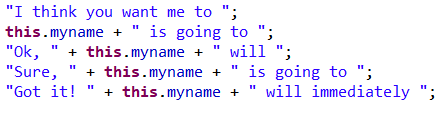
For 2) “clean” or “move” is in the pair of the one of root’s children, I did: for each child of root, check “clean” or “move” of the second one of child’s pair(childPairs(pair.second)), using a nested for loop.



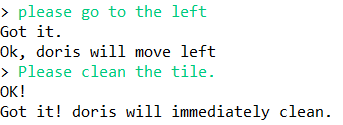
1. [20 pts] **Keyword Search**. List the *five* variations that your system uses in case of a successful keyword search.

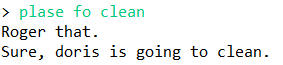
Even I typed sth wrong in the sentence, the robot will find the keyword which is “clear” or “move”

this.myname is robot’s name the user gave



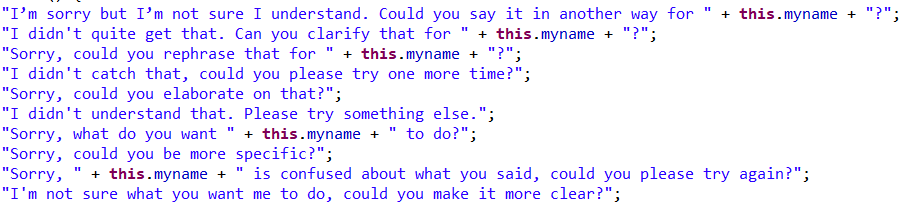
Ex:

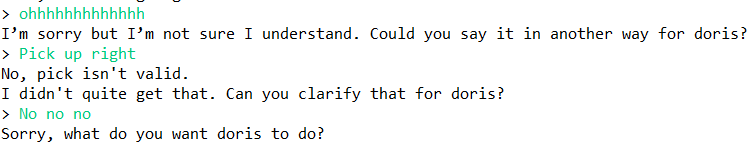




1. [10 pts] **Random Responses**. List the *ten* phrases that your system uses to request clarification.

this.myname is robot’s name the user gave





1. [14 pts] **Human-robot Interaction (HRI).** 
   1. Does your robot have a name and can they report it?

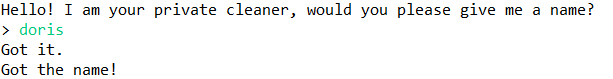
Yes. My robot will have a name which user gives, and they can report the name when user ask.





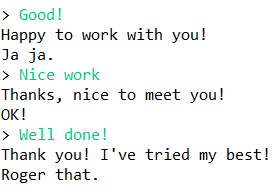
* 1. Can your system name the robot and can the robot remember it?

Yes. My robot doesn’t have name at the beginning, and it will ask for a name once the system started.



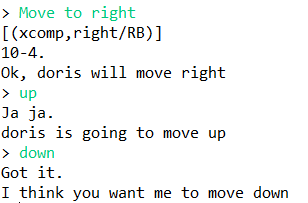
* 1. Please list the (at least three) different praise phrases.

1. Good!
2. Nice work!
3. I like your work
4. Well done!
5. Nice!



* 1. Please list the (at least five) different acknowledgement phrases:

1. Got it.
2. Roger that.
3. 10-4
4. Ja ja.
5. OK!



1. [10 pts] **Extra credit**: Add a speech-to-text component.

Did you implement this component? \_No\_\_

1. [10 pts] **Extra credit:** Add a text-to-speech component.

Did you implement this component? \_No\_\_

Paste an annotated sample dialog here. It should demonstrate your systems capabilities as specified above. Ensure that the dialog covers at least one phrase from each of the items (1) – (4). Please use a different font (or bold face or color) to explain which of the features a dialog item demonstrates. Feel free to err on the side of plenty.