Task 3. Improvements – Report

I found my app had some limitations which could be improved, and in this report I will describe them and the solution I applied to fix them.

The biggest issue I had to deal with was in the state “checkGrammar”, since the app had to be able to backtrack to the previous state if the user’s input was not in the grammar, but to the next if it was. I have three states that make use of “checkGrammar”, so I defined a boolean variable in the context for each of those states and in “checkGrammar” I included six conditions that used these variables and the result of isInGrammar() to move forward or backward in the dialogue. These variables are assigned to “true” inside the state handling the listening, and then to “false” in “checkGrammar” if the condition applies and the app moves forward, to avoid the condition being true later in the dialogue and therefore going back to the wrong state.

One other error I faced stemmed from the fact that if the system could not hear any input the app crashed. I therefore made sure to handle the ASR_NOINPUT message in every state, and to assign the value “null” to “lastResult” if this is the message the app receives after listening. I then made sure that the directions to move on to the next state were guarded by a condition that checked if “lastResult” was not null, and if the condition did not apply the target was instead precisely the “NoInput” child state.

Since I added the days “today” and “tomorrow” to the grammar, I had to implement a function isTodayTomorrow() to select the right preposition between “for” (for “today” and “tomorrow”) and “on” (for any other day) where the app summarizes the details of the appointment in the states “confirmFullDay” and “confirmMeetingTime” depending on the day selected by the user.

I also found that to end the dialogue the user had to forcedly make an appointment, so I created another state called “exit” where the app is redirected if the user says “exit” at any point in the dialogue (when the app is in its listening state). To inform the user of this feature, I made sure to include it in the initial prompt.

I modified the initial code for the “checkGrammar” state to have the app saying “OK” if the user’s input is in the grammar, instead of always repeating it and letting them know if it was or not. I kept this kind of response for the cases when the input is not in the grammar.

Lastly, I solved mispronunciation of the time format like “11:00” which was being read as “eleven hundred hours” by adding “am” and “pm” to the numeric strings.