Report-Lab2

The one big issue I have had with the assignment is getting to implement 'loops' in the state chart via the implementation of the documented features. First, when the user answers a question with something that is not in the relevant grammar, the “notInGrammar” action is carried out correctly, however, the initial question is not raised again and is seemingly skipped to continue back to the listening state. Setting the transition back to the question to 'reenter: true' did not appear to change this behaviour, though I do not understand why.
A similar problem occurs when the 'ASR_NOINPUT' timeout event is raised: In case this event is raised in the 'PromptAndAsk' state, I would like the machine to notify the user accordingly (“Sorry, I didn’t hear you.”) and then transition back to the last active state as well as re-raise the last question. To my knowledge this should have been possible with a history state on 'PromptAndAsk'. But similarly to the previous issue, the transition does not happen as expected (reenter: true not affecting this behaviour, assuming I tried implementing it correctly), rather it always return back to 'AskWho'. Again, this does not render the program dysfunctional, nevertheless, a time-out in asking for an appointment confirmation leading to the questions restarting is tedious. Neither having the history state on the same level as all the other states in 'PromptAndAsk', nor sectioning some of the questions (person-related and day/time-related) into nested states, or switching between shallow/deep history states seemed to make a difference in this regard. I did inquire about the history states in class as well, where we ran into the same issues. As far as I can tell my implementation matches the documentation on Stately, so I am not sure how go about fixing it. Maybe there is some relation to what event SpeechState last received, but that is the only guess I can come up with.

Another limitation, is that the program can only handle one-word replies to questions. Though it is not troublesome as such and does not impact functionality, a potential user would need to be fully aware of the fact and it does lead to a dialogue that feels less than natural.
Furthermore, days, times, and names are all stored in the same grammar constant. While the system checks whether a given input is contained in this grammar, it does not (as of yet) cross-cross-reference whether the type of information suits the question ('Who are you meeting with?' - 'Monday', is accepted due to being part of the grammar). The guard/function to check a given input could additionally compare the type of needed information to the kind of attribute the input has in the internal grammar. This aspect I later fixed by making such specialised guards for each type of input, to insure the actual input matches the type of required information required. The previously simple 'Sorry, [input] is not in the grammar' message is now also specific to the type of information required. Moreover, this message could be modified to tell the user what legible inputs would be available, if they failed to deliver one after x tries.
Additionally, if more than one word of input can be parsed, users would also be enabled to over-answer questions to speed up the process (i.e. 'Who are you meeting with?' - 'I’m meeting Chris on Monday' -> person: Chris, day: Monday). Perhaps the input utterance could somehow be integrated over and each word evaluated on whether it is in the grammar and matches a certain type of input.
When keeping the vocabulary as small as it is here, another idea would be to push the ASR system to more easily recognise especially the less common names used here. If one of the few names is 'Rasmus' for example, the system should strongly favour this interpretation over its currently common guesses. ('Rasmuss', 'Erasmus', 'Christmas', …).
Finally, in a real-world use scenario, the grammar could be expanded to resolve time anaphoras like 'today', 'tomorrow', or 'in an hour'. Additionally, legible times should be freely combinable with minute specifications.