Table 1: The pattern names P, nodes V_p and node attributes $A(v)$, with edges E_p and edge attributes $A(e)$, and the rules.

P	v	$A(v)$	e	$A(e)$	**Rule**
P**_section_change:** Marks a transition to a new section using an edge with `section_change`=True.	C N	$A(C)$[**traversed**]=True; $A(N)$ [**traversed**]=False;	$C \rightarrow N$	$A(C,N)$[**section_change**]=True;	**Rule-Insert-Skipped:** Inserts skipped nodes from the skip graph into the main graph at the transition point into the next section.
P**_user_query:** User asks a question after a traversed node with no skip or abandon status.	C	$A(C)$[**traversed**]=True; $A(C)$[**skipped**]=False; $A(C)$[**abandoned**]=False; $A(C)$[**dialogue_act**]='user-query';			**Rule-Handle-User-Input:** Inserts a system response node after the current node.
P**_yes_answer:** User responds with a yes answer to a yes/no question.	C L R	$A(C)$[**response_required**]=True; $A(C)$[**traversed**]=False; $A(C)$[**abandoned**]=False; $A(C)$[**dialogue_act**]={'yes-answer', 'statement'};	$C \rightarrow L$ $C \rightarrow R$	$A(C,L)$[**label**]= 'no_path'; $A(C,R)$[**label**]= 'yes_path';	**Rule-Yes-Answer:** Removes non-relevant branches and proceeds along the yes path.

***P*_yn_other_answer:** User responds with a no answer to a yes/no question.	C L R	$A(C)$[**response_required**]=True; $A(C)$[**traversed**]=False; $A(C)$[**abandoned**]=False; $A(C)$[**dialogue_act**]= {'no-answer', 'other-answer', 'no-response'};	$C \to L$ $C \to R$	$A(C, L)$[**label**]= 'no_path'; $A(C, R)$[**label**]= 'yes_path';	**Rule-Other-Answer:** Default action if 'yes-answer' not encountered. Removes non-relevant branches and proceeds along the no path.
***P*_yes_answer:** User responds with a no answer to a yes/no question.	C L R	$A(C)$[**response_required**]=True; $A(C)$[**traversed**]=False; $A(C)$[**abandoned**]=False; $A(C)$[**dialogue_act**]='yes-answer';	$C \to L$ $C \to R$	$A(C, L)$[**label**]= 'no_path'; $A(C, R)$[**label**]= 'yes_path';	**Rule-No-Answer:** Removes non-relevant branches and proceeds along the no path.
***P*_stuck_broken:** User is stuck on a 'broken-down' type question.	C	$A(C)$[**response_required**]=True; $A(C)$[**traversed**]=False; $A(C)$[**abandoned**]=False; $A(C)$[**skipped**]=False; $A(C)$[**is_stuck**]=True; $A(C)$[**type**]='broken-down';			**Rule-Stuck-Abandon-Broken-Down:** Marks traversed current node as abandoned. Removes the follow-up nodes along the broken-down path and places them in the abandoned list.

P**_stuck_after_hint:** User remains stuck after a hint.	PP P C	$A(PP,P,C)$[**response_required**] =True; $A(C)$[**traversed**]=False; $A(PP,P)$[**traversed**]=True; $A(PP,P,C)$[**skipped**]=False; $A(PP,P,C)$[**is_stuck**]=True; $A(PP,P,C)$[**is_engaged**]=False; $A(PP,P,C)$[**is_stressed**]=False; $A(C)$[**type**]='hint'; $A(PP,P)$[**type**]='copy';	$PP \to P$ $P \to C$		**Rule-Stuck-Hint-Skip:** Moves the current question and it's followups to the skip graph.
P**_stuck_thrice:** User is stuck three times in a row.	PP P C	$A(PP,P,C)$[**response_required**] =True; $A(C)$[**traversed**]=False; $A(PP,P)$[**traversed**]=True; $A(PP,P,C)$[**skipped**]=False; $A(PP,P,C)$[**is_stuck**]=True; $A(PP,P,C)$[**is_engaged**]=False; $A(PP,P,C)$[**is_stressed**]=False; $A(PP,C)$[**type**]='copy'; $A(PP)$[**type**]='normal';	$PP \to P$ $P \to C$		**Rule-Stuck-Hint:** Adds a hint node after the current node.
P**_stuck_twice:** User is stuck again on a rephrased question.	P C	$A(P,C)$[**response_required**] =True; $A(C)$[**traversed**]=False; $A(P)$[**traversed**]=True; $A(P,C)$[**skipped**]=False; $A(P,C)$[**is_stuck**]=True; $A(P,C)$[**is_engaged**]=False; $A(P,C)$[**is_stressed**]=False; $A(C)$[**type**]='copy'; $A(P)$[**type**]='normal';	$P \to C$		**Rule-Stuck-Repeat-Again:** Inserts another rephrased version of the question.

P**_stuck_once:** User is stuck first time on the current question.	C	$A(C)$[**response_required**]=True; $A(C)$[**traversed**]=False; $A(C)$[**is_stuck**]=True; $A(C)$[**is_engaged**]=False; $A(C)$[**is_stressed**]=False;	C		**Rule-Stuck-Repeat:** Inserts a rephrased version of the question.
P**_stressed_thrice_alt:** User has been stressed for three turns and an alternate path is present.	PP P C N	$A(PP,P,C)$[**response_required**] =True; $A(C)$[**traversed**]=False; $A(PP,P)$[**traversed**]=True; $A(PP,P,C)$[**skipped**]=False; $A(PP,P,C)$[**is_stuck**]=False; $A(PP)$[**type**]='normal' $A(PP,P,C)$[**is_stressed**]=True;	$PP \rightarrow P$ $P \rightarrow C$ $C \rightarrow N$	$A(C,N)$[**label**] ='alternate'	**Rule-Stressed-Alternate:** Removes the default path and it's followups while maintaining the alternate path
P**_stressed_alt_combo:** User has been stressed for three turns and no alternate path is present.	PP P C	$A(PP,P,C)$[**response_required**] =True; $A(C)$[**traversed**]=False; $A(PP,P)$[**traversed**]=True; $A(PP,P,C)$[**skipped**]=False; $A(PP)$[**type**]='normal' $A(PP,P,C)$[**is_stressed**]=True;	$PP \rightarrow P$ $P \rightarrow C$		**Rule-Stressed-Break-Down:** Inserts simpler sub-questions (e.g., who, what, why)

P**_engaged_stuck:** User is stuck but engaged.	C	$A(C)$[**response_required**] =True; $A(C)$[**traversed**]=False; $A(C)$[**skipped**]=False; $A(C)$[**type**]='normal' $A(C)$[**is_stressed**]=False; $A(C)$[**is_stuck**]=True; $A(C)$[**is_engaged**]=True;			**Rule-Engaged-Stuck-Hint:** Inserts a hint node to support the user.
P**_engaged_stuck_copy:** User is stuck on a rephrased question.	C	$A(C)$[**response_required**] =True; $A(C)$[**traversed**]=False; $A(C)$[**type**]='copy' $A(C)$[**is_stressed**]=False; $A(C)$[**is_stuck**]=True; $A(C)$[**is_engaged**]=True;			**Rule-Engaged-Stuck-Copy-Hint:** Inserts a hint node to the original question instead of a rephrased question.
P**_engaged_stuck2_alt:** User has been engaged and stuck for the last two turns, and an alternate path is present.	P C N	$A(P,C)$[**response_required**] =True; $A(P)$[**traversed**]=True; $A(C)$[**traversed**]=False; $A(C)$[**type**]='hint' $A(P,C)$[**is_stressed**]=False; $A(P,C)$[**is_stuck**]=True; $A(P,C)$[**is_engaged**]=True; $A(P,C)$[**skipped**]=False;	$P \to C$ $C \to N$	$A(C,N)$[**label**] ='alternate'	**Rule-Engaged-Stuck-Twice-Alternate:** Removes the default path and maintains alternate path after repeated difficulty.

P**_engaged_stuck2:** User has been engaged and stuck for the last two turns.	P C	$A(P,C)$[**response_required**] =True; $A(P)$[**traversed**]=True; $A(C)$[**traversed**]=False; $A(C)$[**type**]='hint' $A(P,C)$[**is_stressed**]=False; $A(P,C)$[**is_stuck**]=True; $A(P,C)$[**is_engaged**]=True; $A(P,C)$[**skipped**]=False;	$P \rightarrow C$		**Rule-Engaged-Stuck-Twice:** Inserts a sequence of follow-up questions.
P**_other_answer_alt:** User has been engaged and stuck for the last two turns.	C N	$A(C,N)$[**response_required**] =True; $A(C,N)$[**traversed**]=False; $A(C)$[**dialogue_act**]='other-answer'	$C \rightarrow N$	$A(C,N)$[**label**] ='alternate'	**Rule-Other-Answer-Alternate:** Removes the non-alternate path.
P**_other_answer:** User does not know the answer.	C N	$A(C,N)$[**response_required**] =True; $A(C,N)$[**traversed**]=False; $A(C)$[**dialogue_act**]='other-answer'			**Rule-Abandon-Other-Answer:** Marks the node as abandoned and puts any followups in the abandoned list.

P**_off_topic_3_alt:** User has been off topic for the last three turns, and an alternate path has been provided.	PP P C N	$A(PP, P, C)$[**response_required**] =True; $A(C)$[**traversed**]=False; $A(PP, P)$[**traversed**]=True; $A(PP, P, C)$[**skipped**]=False; $A(PP, P, C)$[**is_engaged**]=True; $A(PP)$[**type**]='normal' $A(P, C)$[**type**]='prompt' $A(PP, P, C)$[**is_offtopic**]=True;	$PP \rightarrow P$ $P \rightarrow C$ $C \rightarrow N$	$A(C, N)$[**label**] ='alternate'	**Rule-OffTopic-Thrice-Alternate:** Removes the default path and it's followups while maintaining the alternate path.
P**_off_topic_thrice:** User has been off topic for the last three turns.	PP P C	$A(PP, P, C)$[**response_required**] =True; $A(C)$[**traversed**]=False; $A(PP, P)$[**traversed**]=True; $A(PP, P, C)$[**skipped**]=False; $A(PP, P, C)$[**is_engaged**]=True; $A(PP)$[**type**]='normal' $A(P, C)$[**type**]='prompt' $A(PP, P, C)$[**is_offtopic**]=True;	$PP \rightarrow P$ $P \rightarrow C$		**Rule-OffTopic-Thrice-Skip:** Skips the question and it's follow-sups and moves it to the skip graph.
P**_off_topic_twice:** User has been off topic for the last two turns.	P C	$A(P, C)$[**response_required**] =True; $A(C)$[**traversed**]=False; $A(P)$[**traversed**]=True; $A(P, C)$[**skipped**]=False; $A(P, C)$[**is_engaged**]=True; $A(P)$[**type**]='normal' $A(C)$[**type**]='prompt' $A(P, C)$[**is_offtopic**]=True;			**Rule-OffTopic-Twice-Insert-Prompt:** Inserts a prompt node to redirect the user back to the topic.

P**_off_topic_once:** User has been off topic the current turn.	C	$A(C)$[**response_required**] =True; $A(C)$[**traversed**]=False; $A(P,C)$[**skipped**]=False; $A(P,C)$[**is_engaged**]=True; $A(C)$[**type**]='normal' $A(C)$[**is_offtopic**]=True;			∞ **Rule-OffTopic-Once-Prompt:** Inserts a prompt node to redirect the user back to the topic.