Fall 2021

Linear Temporal Logic and Timeline Visualization

Christopher Pohlen

Human Computer Interaction Creative Component

Linear Temporal Logic and Timeline Visualization

Masters of Science in HCI

Committee Members:

Kristin-Yvonne Rozier

Jonathan Kelly

Christopher Pohlen

November 24, 2021

Introduction

In the field of formal methods, two things tend to be accepted as true: the formal methods methodology is difficult for people to learn and is error prone. Despite the benefits from formal methods on requirement specification and other areas of interest, these two things remain relatively constant. Furthermore, the specificity and unintuitiveness of formal methods can be confusing and lead to errors even for seasoned formal methods practitioners. The situation is improving, however. Tools and frameworks have recently made the transition between natural language and more formalized semantics easier. An example of this is the FRET tool from NASA [1] and the R2U2 runtime verification framework [2]. Despite such advances, there is no tool that can directly translate between Linear Temporal Logic (LTL) and an equivalent useful timeline visualization.

This work explores the literature surrounding timelines and temporal information visualizations, and is an exploratory effort to find visualizations or types of visualizations that would be conducive to the creation of a computerized tool. This tool would provide an effective means for teaching and performing work in formal methods using LTL. A more effective timeline visualization tool would help teach formal methods because changing a timeline (where the learner can see precisely the changes they are making) would have an exact corresponding change to an LTL formula. The tool would also help reduce the error rate because a mistake in the LTL would be much more visible in the corresponding timeline.

Related Work

NASA's FRET tool is an effective tool for simple requirements specification [1]. It uses a close-to-English syntax to make the specification process easier and less error-prone. The simulation tool is of particular interest to this research. It allows the FRET user to test if the specification variables work as the user expects. There is more than one improvement that can be made, however. First, not all LTL formulas are capable of being represented in FRET. Giannakopoulou et al. (2021), showed that FRET could be converted to LTL [3], but the reverse is not necessarily true. Nested temporal operators or formulas with multiple temporal operators cannot be represented in a single FRET requirement; its syntax simply does not allow it. Secondly, the timeline visualizations FRET provides are prefabricated to fit the timing phases used and do not include the variables or the formula itself. This means that without written context generated by FRET the timelines are not comprehensible. Other than including FRET's visualizations as an option in the temporal format preference questions shown later, FRET results could not be included in this work's survey because many of the LTL formulas that we utilized contain nested temporal operators or multiple temporal operators.

In order to help create methods of describing temporal information for use in computer science, Allen (1983) worked to create an algebraic system for describing intervals in time to potentially use in artificial intelligence and database systems [4]. Allen's work featured several tree structures for referencing interval information. Such a tree structure makes potential visualizations easier to create rules for. Similar to Allen's work, the Panda tool parses a LTL formula and creates a binary tree format [5]. A branch node in the tree would contain a temporal or propositional operator and leaf nodes would contain propositional variables. This binary system lends itself to a recursive system to visualize the formula. The code for Panda is open source, and will eventually be used to create the temporal visualization tool this work is exploring towards.

Brehmer et al. (2017) investigated how timelines are structured for storytelling [6]. They found several dimensions that form different timeline formats. As LTL is temporal information, the timeline formats it could theoretically be displayed in are varied. Brehmer et al.'s (2017) work explored these potential timeline formats. The paper shows how temporal information can be displayed in many different ways in timelines or timeline-like forms. Studies by Gigante et al. [7] and Harrison et el. [8] importantly show how the expressiveness of timelines can be utilized for temporal data visualization. Gigante et al. showed how both quantitative and qualitative information can be expressed, while Harrison et al. included proofs that showed how overlapping events can be expressed on timelines and remain efficacious.

Ideally, the computer tool created from this work would be useful for requirements specification or other formal methods tasks. Several other research projects had tools which utilized timelines or other functionality that are useful in an LTL visualization tool. Karam's (1994) work provided insight on displaying timelines in a tool, but the paper used timing diagrams instead of explicit LTL formatting of any kind [9]. Dachselt's and Weiland's (2006) Timezoom introduced the idea of "zooming" into a bigger and more general timeline to see detailed internal information [10]. As LTL formulas can involve complex interactions between variables, it is an interesting idea to succinctly show complex information while still having the detailed minutiae available. Smith et al. (2001) created a tool for generating automata from a timeline. However, the tool did not use LTL itself [11].

Several other studies introduce other potential ways of condensing or presenting information in creative ways. Belden et al. (2018) showed how a collapsible box structure can help medical practitioners find exactly the temporal information necessary to correctly treat a patient [12]. Their tool had segmented information sections for easy navigation and to reduce screen clutter. Bjarnason et al. (2012) created a timeline system for project management and retrospectives [13]. It was designed to help keep track of previous meetings and decisions and its format is useful as a template. A timeline format by Gao and Hunter (2011) contributes an interesting additional feature: a way to connect related information across various timelines and time periods by creating visual connecting lines between related events [14]. Swan and Allan (2000) created a tool for automatically generating timelines based on topics and terms [15]. Theirs could be a useful idea for filtering large numbers of variables and their timelines.

In summary, this work's research direction is important because the related work is focused on either the overall capability of timelines and other methods of expressing temporal information, LTL requirement specification, but not both. This research looks towards using both ideas to make formal methods an easier to learn and more effective field of study.

Method

Participants for this exploratory work were essentially pilot testers for the survey. They were relatively new formal methods practitioners from an introductory formal methods class. 13 total responses were received by the target audience but only 4 fully completed the survey. Small numbers of participants were expected for this exploratory work, but the follow on work will gather many more participants. The target audience in that case is any formal methods practitioners of varying formal methods experience levels from any background, age, profession, etc. They will be gathered through personal contacts and colleagues, online formal methods forums, and any other method that is likely to gather partricipants with the requisite formal methods experience.

In order to gather data on the types of visualizations that would be useful for an LTL and timeline visualization tool, we devised a survey. Since the survey is for finding a Human Computer Interaction (HCI) solution to a problem that can be very technical in nature, we preferred to sample those with only the requisite formal methods knowledge to take the survey. The survey is designed with 3 major parts in mind. Questions about specific timelines are multiple choice and have a particular correct answer, and questions about preferences are ranked ordering questions and are entirely subjective. The survey is structured as follows:

1. An opening section
   - Anonymous demographic information
   - Some questions to prime participants to the basics of the propositional visualizations on the timelines, shown in Figure 1
   - Questions for getting the participants' preferred propositional and temporal visualizations in Figure 2 and Figure 3, respectively, in the Appendix
2. A section testing the participant's ability to correctly ascribe LTL formulas to timelines using their choices of propositional and temporal operators; a second set of questions used a random and entirely different combination of operators from the participant's choices (Figures 4 through 15)
3. A section for questions about opinions and comments

Since FRET is the most closely related work, a FRET option was added to the temporal preference questions using screenshots from the FRET program [1]. If any of the questions have the FRET option selected when the participant goes to the next screen, two things

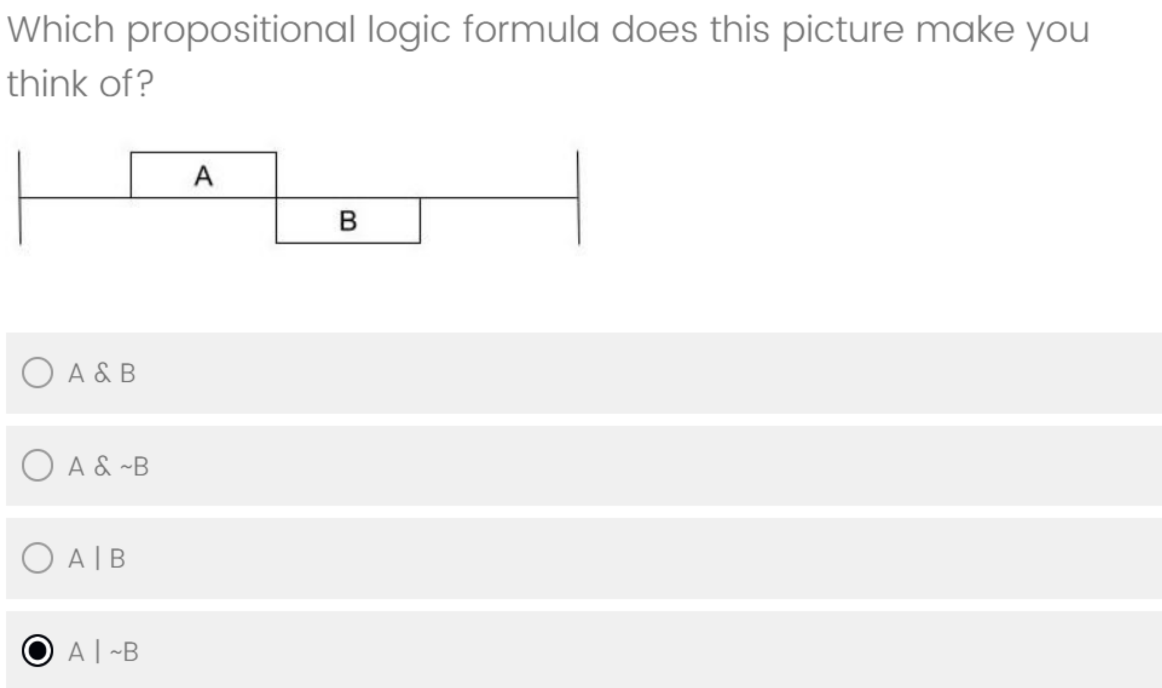


Figure 1: A question to help prime participants to how the formulas will appear.

happen. First, a follow up free-response question appears asking why they selected one of those options. Second, instead of having a section of questions based on the propositional and temporal choices of the participant (a full section of LTL questions corresponding to FRET options could not be generated), the participant will get two random combinations of propositional and temporal visualization formats.

The preference questions are asked in a particular way. The propositional and temporal visualization formats (shown later) were created with varying degrees of complexity and explicitness. As such, the preference questions display the various visualizations from most simple/least explicit to most complex/most explicit. When developing these visualization formats, it made sense to stick to formats that may be easier in terms of software development. Simple boxes in relatively simple layouts would be easier to create. The formats also tended to appear most like traditional timelines. The format that would be most experimental would be the String format due to the fact that the layout would be more complicated.

The LTL formulas used in the testing section are listed below (grouped by temporal operators used). Figures for each of the equations appear in the appendix using one of the LTL visualizations from the survey.

Next time:
(A & X B) | $\sim$ C (Figure 4)

(A | X ~ B) & C (Figure 5)

Eventually:
(A & F ~B) & C (Figure 6)
(F A | B) &F C (Figure 7)

Always:
A & G(B | C) (Figure 8)
G A | (B & G C) (Figure 9)

Until:
A & (B U C) (Figure 10)
(A | B) U ~C (Figure 11)

Release:
(A R B) | C (Figure 12)
A & (B R ~C) (Figure 13)

Mixed:
~ G(XXX q | ~p) or F(p & XXX ~q) (Figure 14)
G (~P | (~Q U S)) (Figure 15)

LTL Visualization Formats:

There are 18 total LTL formats using a combination of one of the six propositional formats and one of the 3 temporal formats. Each format follows a set of rules that work recursively and all rules involve placement of boxes, connections, size, etc. Every LTL visualization format has one rule in particular that it follows: variables and operators are encased in boxes to delineate them. To ease explanations of the all the visualization formats, the operators ~, *Always*, *Eventually*, and *Next Time* are called 'singular operators', while &, |, *Until*, and *Release* are refered to as 'pairing operators'.

Propositional Format Rules:

There are six propositional visualization formats created for this study whose names describe their visualization scheme: Basic, Multi-colored Simple, Multi-colored Complex, Boxed Symbols, Boxed Words, and String. This ordering is from simplest and least explicit to most complex and most explicit.
The first three formats follow a similar initial pattern:

- stacking boxes on top of one another denotes the propositional & operator
- placing boxes side by side to one another denotes the propositional | operator

The first two also share the property that boxes will change in size to correspond with other boxes connected to it by other pairing operators.

Basic:
Having the negated box(es) be placed beneath the horizontal line of the timeline denotes the propositional operator $\sim$.

Multi-colored Simple:
Having the negated box(es) colored red denotes the propositional operator $\sim$.

Multi-colored Complex:
In addition to the previous format, red colored box(es) denote the propositional operator $\sim$. An extra box will encase any paired boxes if more than two total boxes are present. Paired boxes also share direct contact.

The next two formats follow largely the same pattern. Paired boxes are preceded by a box containing the temporal and/or propositional operator(s) acting on the box(es). Temporal operators in these formats can share an operator box with a propositional operator as long as the temporal operator is the parent of the other operator.

Boxed Symbols:
This formats operators use the symbolic version of the propositional operators: &, |, and $\sim$.

Boxed Words:
This formats operators use the name version of the propositional operators: AND, OR and NOT.

String:
For this format, a black line connects all operator and variable boxes together in the order they appear in the formula or an equivalent. The operator that is furthest to the outside of a formula or the "root" of the binary tree is placed directly on the timeline. Any operator boxes that come next by rank order and all other boxes connected to it by extension move upwards away from the timeline.

There are two exceptions to this move up rule. The first is that two operators can instead share a box if the operators are functionally adjacent in the formula, one of the operators is temporal, and that operator is the parent of the propositional one. The second is that the propositional variable boxes can share the same level as their parent operators unless it is a double operator box from the first exception.

Temporal Format Rules:

Because of the uncertain nature of some temporal operators, a solid box denotes an operator that is definitely present in that time unit. A dotted line box denotes any operator or variable box that *could* be present at that time unit. The placement of boxes and certain other visual features sometimes depends on the propositional format being used and will be explained later. These rules for the temporal operators are true for all three of the temporal formats.

Next Time:
The subformula affected by this operator is placed into the next time unit on the timeline. For the Basic and Multi-colored propositional formats, when pairing operators have their variables or subformulas separated across time units by *Next Time*, lines extend from the paired boxes to connect them. For boxes paired with &, the line extends from the bottom of one box to the top of the other (maintaining the stacked nature of &). For |, the line extends from the sides of the paired boxes (mainting the side by side nature of |).
For the other propositional formats, the boxes or string will extend over the time unit boundary.

Always:
This operator sees the subformula it encompasses repeated for at least two time units before the timeline ends and an ellipsis is placed at the end of the timeline.

Eventually:
This operator causes the subformula it encompasses to repeat dotted boxes for at least two time units, an ellipsis is placed mid-timeline, and the time unit after the ellipsis has the solid boxes of the subformula.

Until:
This operator visualization appears similar to the *Eventually* operator, but uses solid boxes instead of dotted boxes for the left-side subformula. In addition, after the ellipsis, the left side subformula is negated and the right side subformula is placed in the time unit.

Release:
For this operator, the left and right subformals both follow the *Eventually* pattern, but the left subformula always appears as solid boxes while the right side appears as dotted boxes until after the ellipse when it turns solid as well.

Other than the Basic format, the temporal visualization formats make slight changes to the expression of the temporal operator visualizations above.

Basic:
This format uses no explicit symbols or words.

Nested Symbols:
This format uses the symbols of the temporal operators (X, $\Box$, $\Diamond$, U, and R, respectively). The symbols are placed in the boxes of the variables or operators which are terminal for the *Next Time, Until, and Release* operators. For *Always* and *Eventually*, however, the symbol is maintained through the repeated boxes to show a bit more continuity.

Nested Words:
This format uses the word names of the different operators. The names generally have their own box preceding the subformula unless another rule allows them to collapse with another operator. In addition, the operator boxes go in front of the first part of the subformula rather than the last part subformula in the final timestep.

Results

As a result of so few participants being involved in the pilot there are few trends that can be gleaned. However, one interesting result that should be watched for in the future is that the more explicit or complex a particular propositional format was, the more likely it appeared in the top half of the preferences. 5 participants got far enough for this trend to be seen in the survey. For the three most explicit propositional formats, they were collectively chosen 80% of the time as first choices, 80% of the time as second choices, and 60% of the time as third choices. If this result holds true in a larger study, it may suggest that more explicitness or complexity may be preferred. However, it would be a stronger design decision if those preferences were compared to the percentage of correct answers in the second part of the survey. No trend could be noted from the temporal format preferences. Several analyses are proposed for a similar future survey with more participants.

- Regression between preferred LTL combination and number of correct answers
- Chi square between formal methods experience and LTL combination preference
- Regression between formals methods experience and number of correct answers
- Percentages of which LTL combinations were preferred
- Chi square between LTL combination preference and desire to change choices of propositional and temporal operators
- Percentage of participants that believe there is a some valid LTL combination
- Percentage of participants that prefer GUIs/text-based interfaces for formal methods tools

- Chi square between age of participants and preference for GUIs/text-based interfaces (we chose to use ranges of ages in a dropdown box for this pilot survey)

- Any qualitative trends in the suggestion questions

Discussion and Future Work

The purpose of this research was to explore ways of visualizing temporal information in a way that would inform later efforts to create a formal methods tool for improving work and learning in the field. We hope to see a number of results. First, we would like to see any trends in the preferences for certain visualization formats. Seeing trends would give insight into the types of things the participants would want to see from a LTL and timeline visualization tool. In addition, seeing if certain formats allow the participants to pick more correct answers would imply a greater level of intuitiveness.

There are some limitations to this work. At the beginning of the survey, we asked for preferences between the different propositional and temporal operator formats. The questions purposefully list the options from least complex and least explicit to the most complex and most explicit formats. This was done both to show the contrast between the various formats, but to also reduce survey complexity. This allows for the participants to see the contrasts between them more easily. However, this could lead to an order effect where particular options more chosen more often than others based on positioning of the options rather than the participant's unencumbered choice.

Another limitation of this survey is the boxed nature of all the LTL visualization formats. We chose the boxed nature for this initial exploratory work because it is relatively easy to implement programmatically. However, as mentioned in the discussion of [6], there are many potential ways to show temporal information on timelines or timeline-like formats that may be better at expressing the temporal information important for formal methods work.

A current limitation on further research in this subject is the current lack of formalized proofs for the various LTL visualization formats in the survey to prove that the LTL formula and the visualization share a one-to-one relationship. While the rules for the visualization formats are closely based on the binary tree structure provided by [5], truly formal proofs for each of the formats have not been created. While the proofs may not be necessary to explore simple preferences for the visualizations, they will definitely be necessary for future work in creating the tool.

There are a number of future directions this work can take. The most direct and important one would be to create the LTL timeline visualization tool for which this work is exploring. The tool will likely be a large undertaking on its own. More research may be desired during the tool's creation as well. For other directions, further research forcus on developing more visualization formats. Brehmer's et al. [6] work showed that there are at least 14 different general formats for timelines. All the formats in this survey are only related to a small subset of those 14. Given the fact that there could be dozens of ways to show propositional and temporal operators in those other formats, the implications for this kind of temporal data visualization research are vast.

References

[1] D. Giannakopoulou, A. Mavridou, J. Rhein, T. Pressburger, J. Schumann, and N. Shi, "Formal requirements elicitation with fret," 2020.

[2] P. Moosbrugger, K. Y. Rozier, and J. Schumann, "R2u2: monitoring and diagnosis of security threats for unmanned aerial systems," *Aerospace Engineering Publications*, p. 146, 2017.

[3] D. Giannakopoulou, T. Pressburger, A. Mavridou, and J. Schumann, "Automated formalization of structured natural language requirements," *Information and Software Technology*, vol. 137, p. 106590, 2021.

[4] J. F. Allen, "Maintaining knowledge about temporal intervals," *Commun. ACM*, vol. 26, no. 11, pp. 832–843, 1983.

[5] K. Y. Rozier and M. Y. Vardi, "A multi-encoding approach for ltl symbolic satisfiability checking," *17th International Symposium on Formal Methods (FM2011)*, vol. Lecture Notes in Computer Science (LNCS), no. 6664, pp. 417–431, 2011.

[6] M. Brehmer, B. Lee, B. Bach, N. H. Riche, and T. Munzner, "Timelines revisited: A design space and considerations for expressive storytelling," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 9, pp. 2151–2164, 2017.

[7] N. Gigante, A. Montanari, M. C. Mayer, and A. Orlandini, "Timelines are expressive enough to capture action-based temporal planning," in *2016 23rd International Symposium on Temporal Representation and Reasoning (TIME)*, pp. 100–109, 2016.

[8] B. Harrison, R. Owen, and R. M. Baecker, "Timelines: An interactive system for the collection and visualization of temporal data," 1994.

[9] G. M. Karam, "Visualization using timelines," *ACM SIGSOFT*, vol. ISSTA, pp. 125–137, 1994.

[10] R. Dachselt and M. Weiland, "Timezoom: A flexible detail and context timeline," p. 682–687, 2006.

[11] M. H. Smith, G. J. Holzmann, and K. Etessami, "Events and constraints: A graphical editor for capturing logic requirements of programs," in *Proceedings Fifth IEEE International Symposium on Requirements Engineering*, pp. 14–22, IEEE, 2001.

[12] J. L. Belden, P. Wegier, J. Patel, A. Hutson, C. Plaisant, J. L. Moore, N. J. Lowrance, S. A. Boren, and R. J. Koopman, "Designing a medication timeline for patients and physicians," *Journal of the American Medical Informatics Association*, vol. 26, pp. 95–105, 12 2018.

[13] E. Bjarnason, R. Berntsson Svensson, and B. Regnell, "Evidence-based timelines for project retrospectives — a method for assessing requirements engineering in context," in *2012 Second IEEE International Workshop on Empirical Requirements Engineering (EmpiRE)*, pp. 17–24, 2012.

[14] L. Gao and J. Hunter, "Publishing, linking and annotating events via interactive timelines: an earth sciences case study," in *CEUR workshop proceedings*, vol. 779, pp. 11–21, Citeseer, 2011.

[15] R. Swan and J. Allan, "Automatic generation of overview timelines," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, (New York, NY, USA), p. 49–56, Association for Computing Machinery, 2000.

Appendix

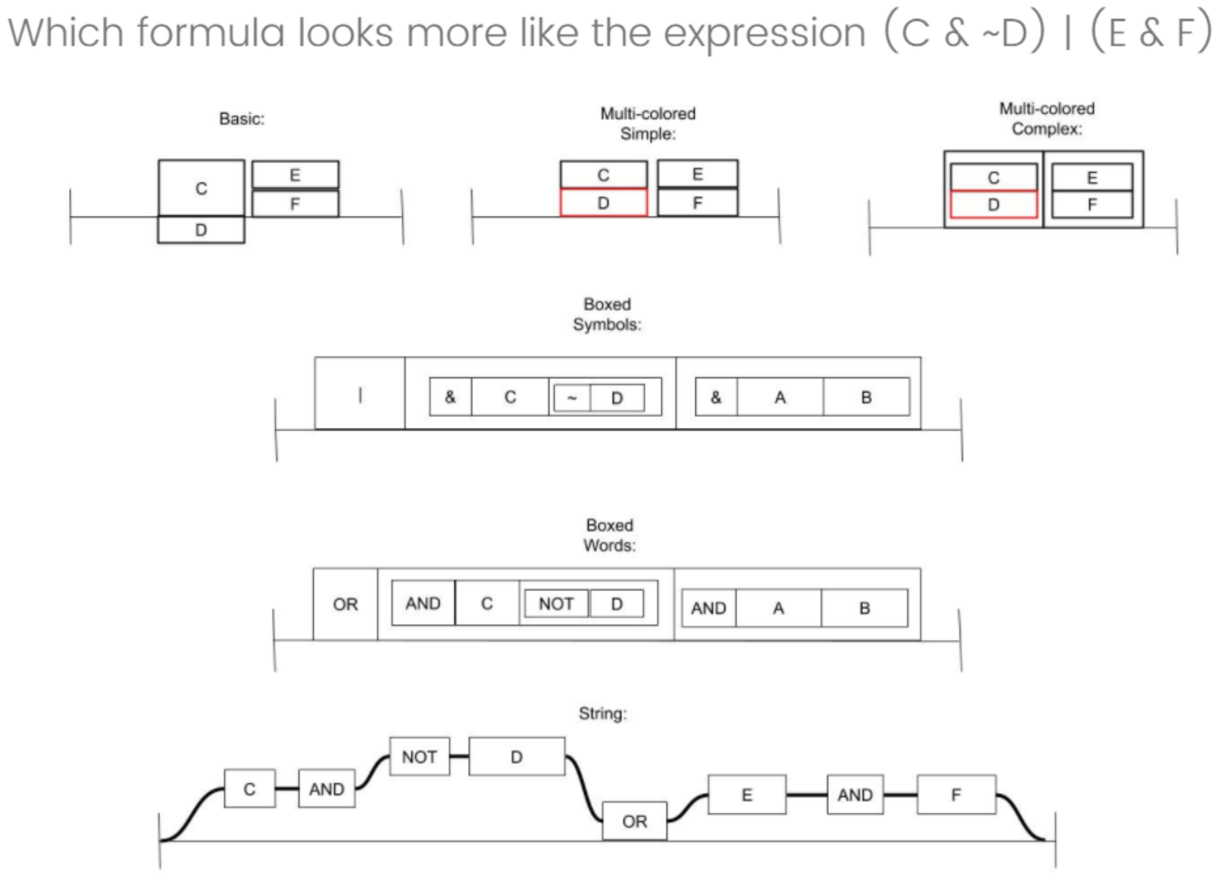


Figure 2: A question to determine the participant's preference for propositional operator visualizations.

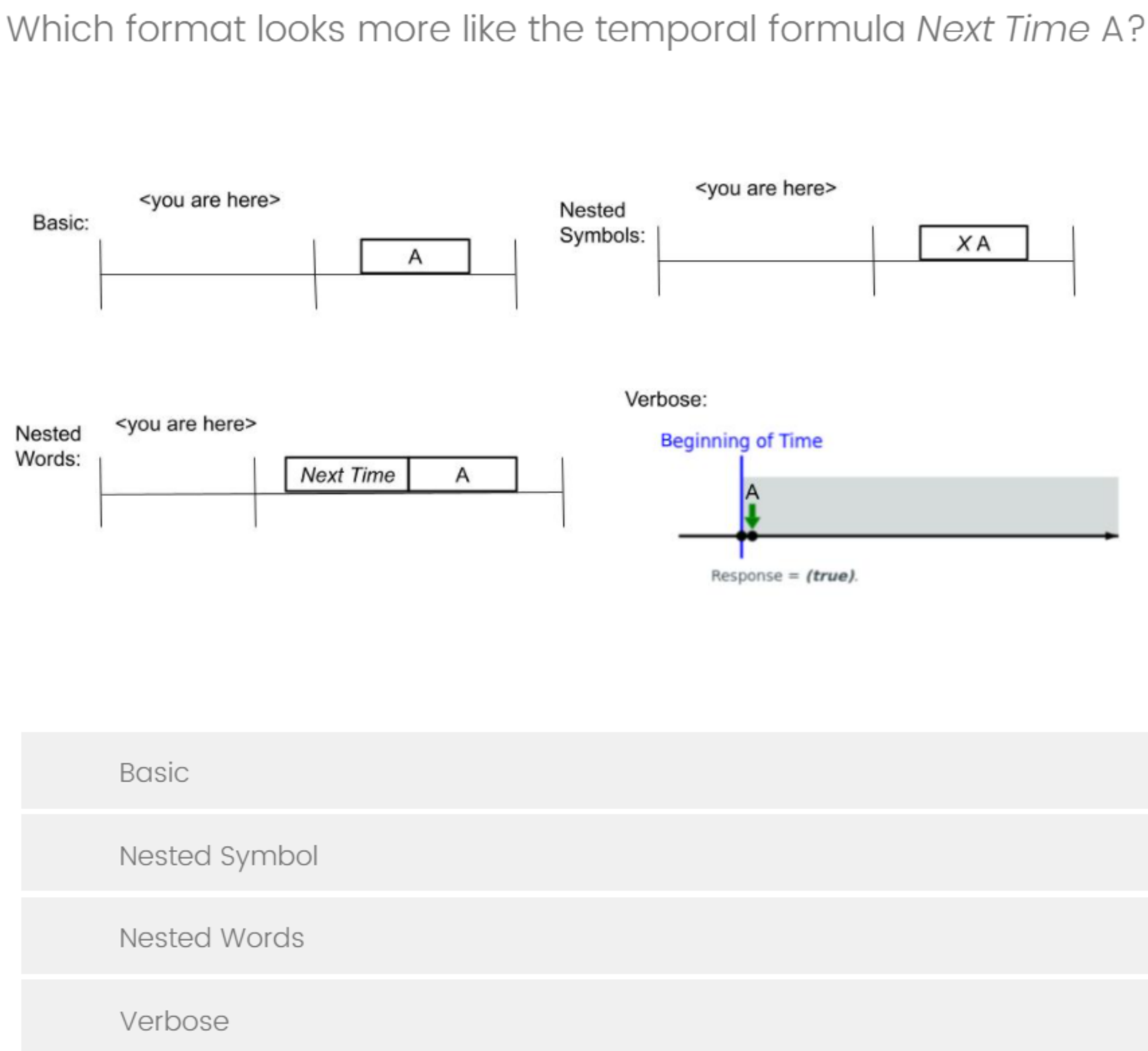
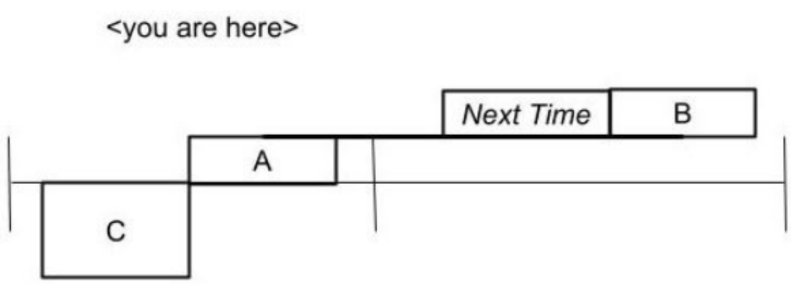


Figure 3: A question to determine the participant's preference for temporal operator visualizations.

Which formula does this timeline most look like?



- *X* B | (A & C)
- (A | ~C) & *X* B
- (A & *X* B) | ~C

Figure 4: A question asking about the *Next Time* operator using propositional format, Basic, and temporal format, Nested Words.

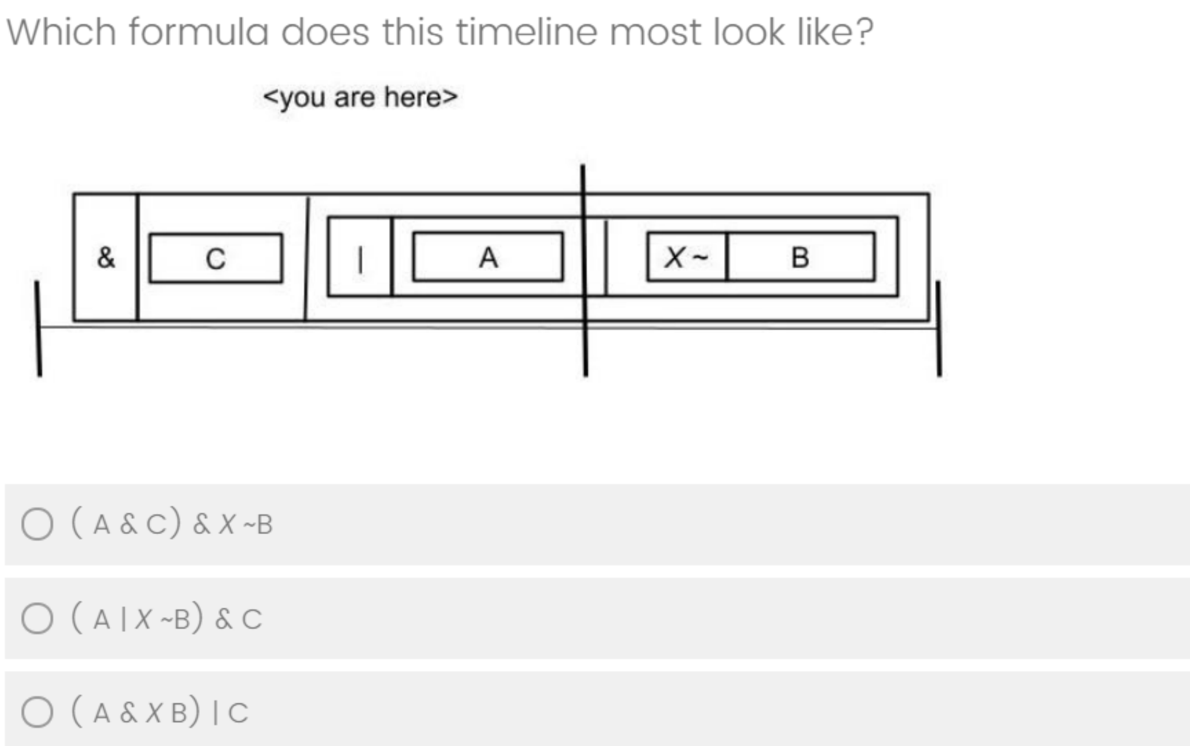


Figure 5: A question asking about the *Next Time* operator using propositional format, Boxed Symbols, and temporal format, Nested Symbols.

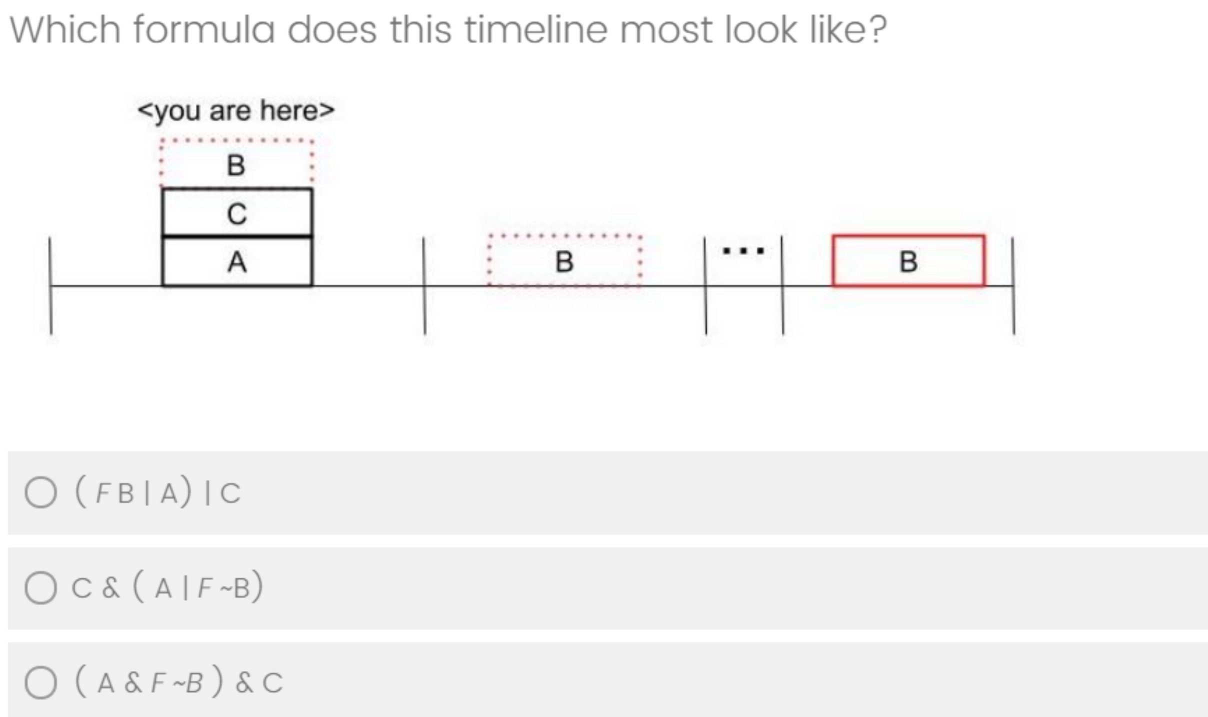
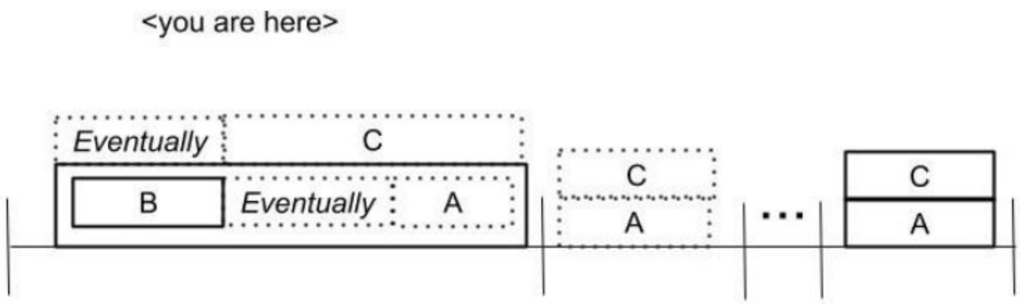


Figure 6: A question asking about the *Eventually* operator using propositional format, Multi-colored Simple, and temporal format, Basic.

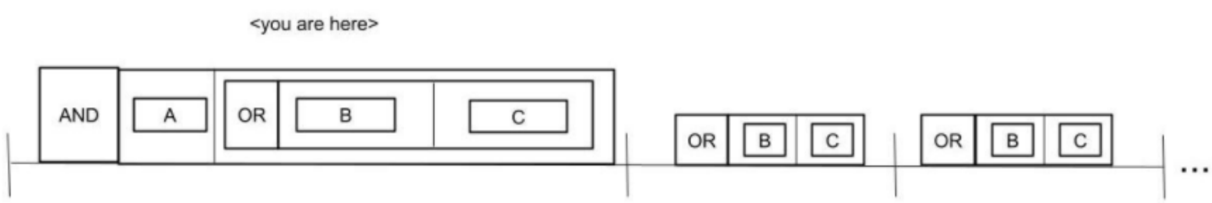
Which formula does this timeline most look like?



- (*F* A | B) & *F* C
- (*F* B | A) | C
- (*F* C & *F* A) | B

Figure 7: A question asking about the *Eventually* operator using propositional format, Multi-colored Complex, and temporal format, Nested Words.

Which formula does this timeline most look like?



- *G* (A & (B | C))
- A & *G* (B | C)
- ~A | (*G* B & *G* C)

Figure 8: A question asking about the *Always* operator using propositional format, Boxed Words, and temporal format, Basic.

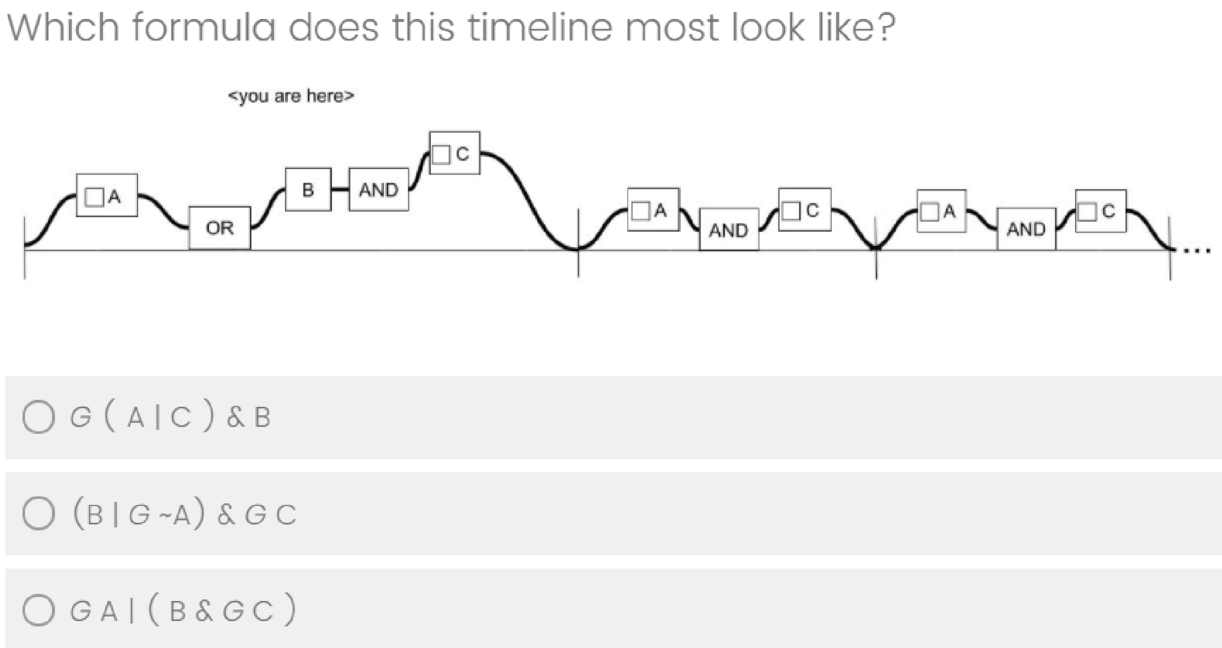


Figure 9: A question asking about the *Always* operator using propositional format, String, and temporal format, Nested Symbols.

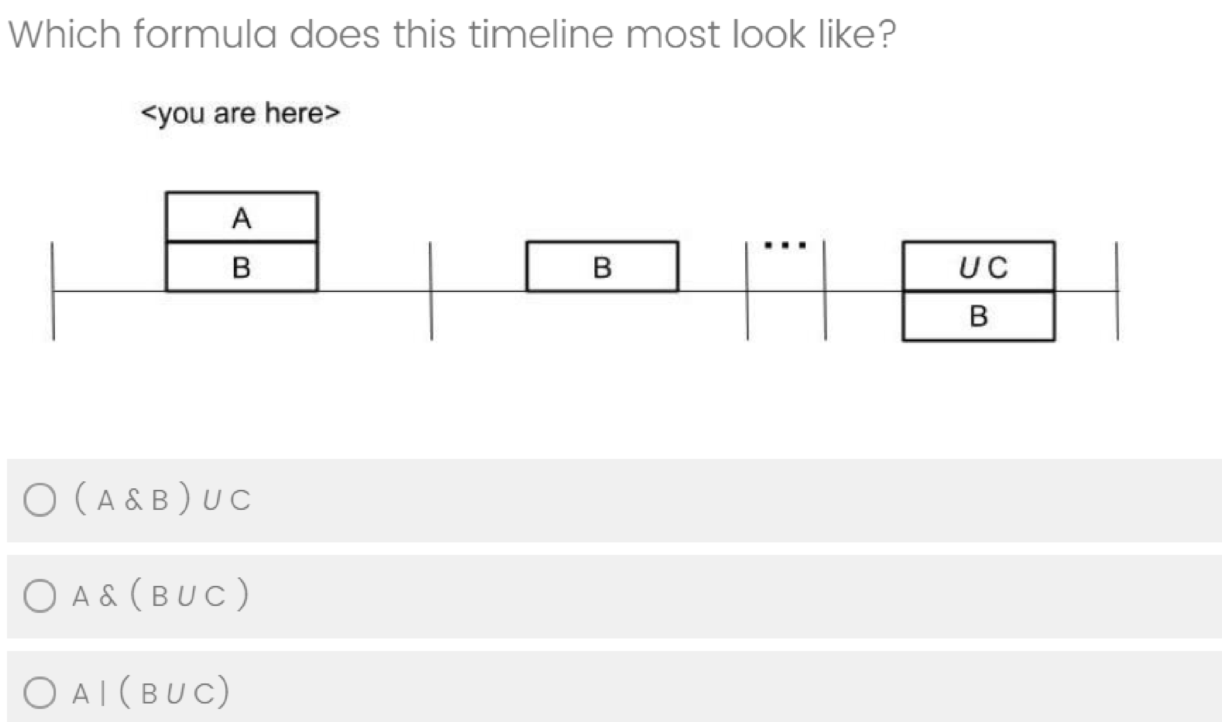


Figure 10: A question asking about the *Until* operator using propositional format, Basic, and temporal format, Nested Symbols.

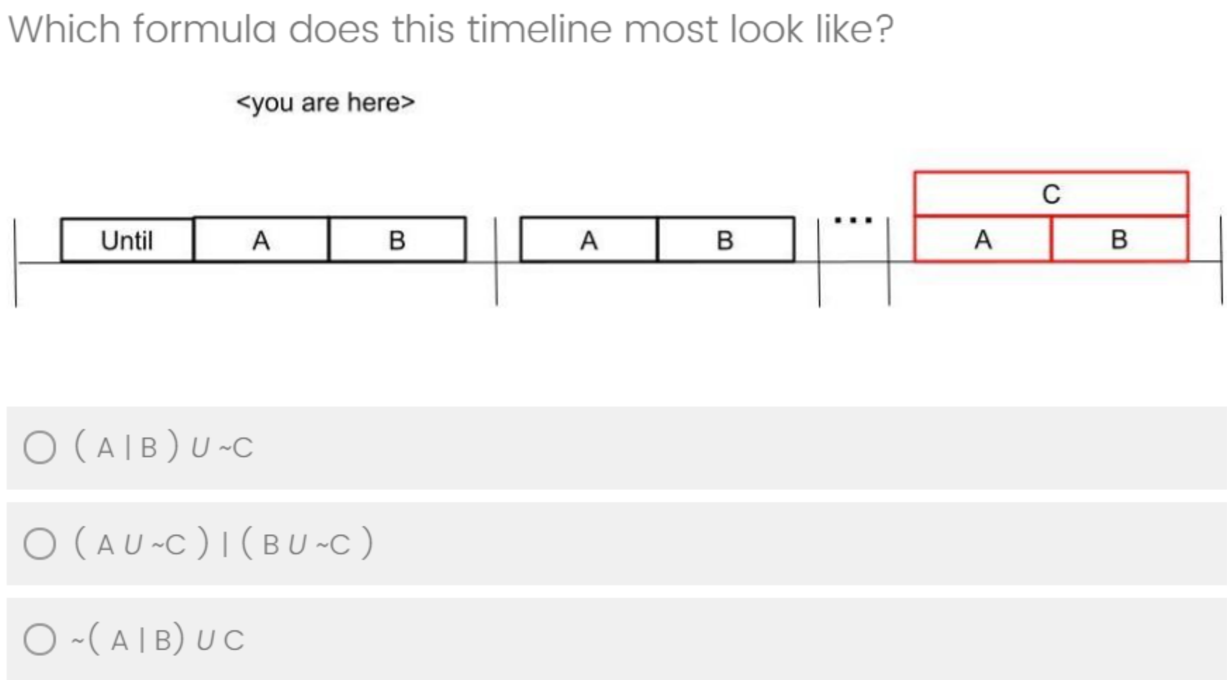


Figure 11: A question asking about the *Until* operator using propositional format, Multi-colored Simple, and temporal format, Nested Words.

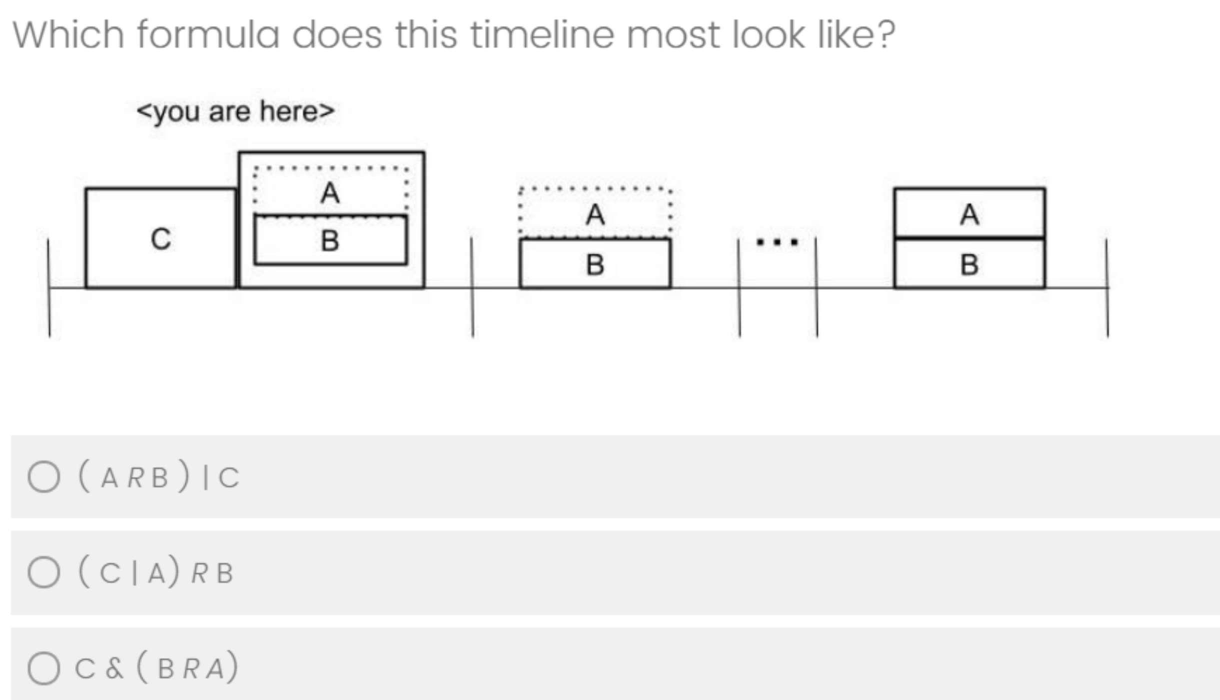


Figure 12: A question asking about the *Release* operator using propositional format, Multi-colored Complex, and temporal format, Basic.

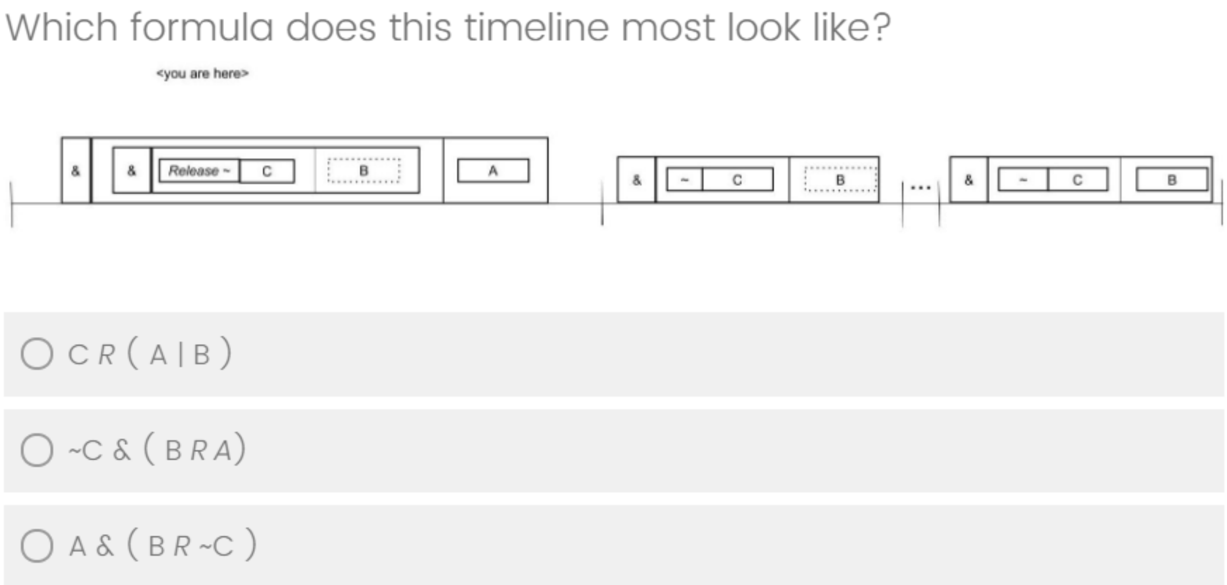


Figure 13: A question asking about the *Release* operator using propositional format, Boxed Symbols, and temporal format, Nested Words.

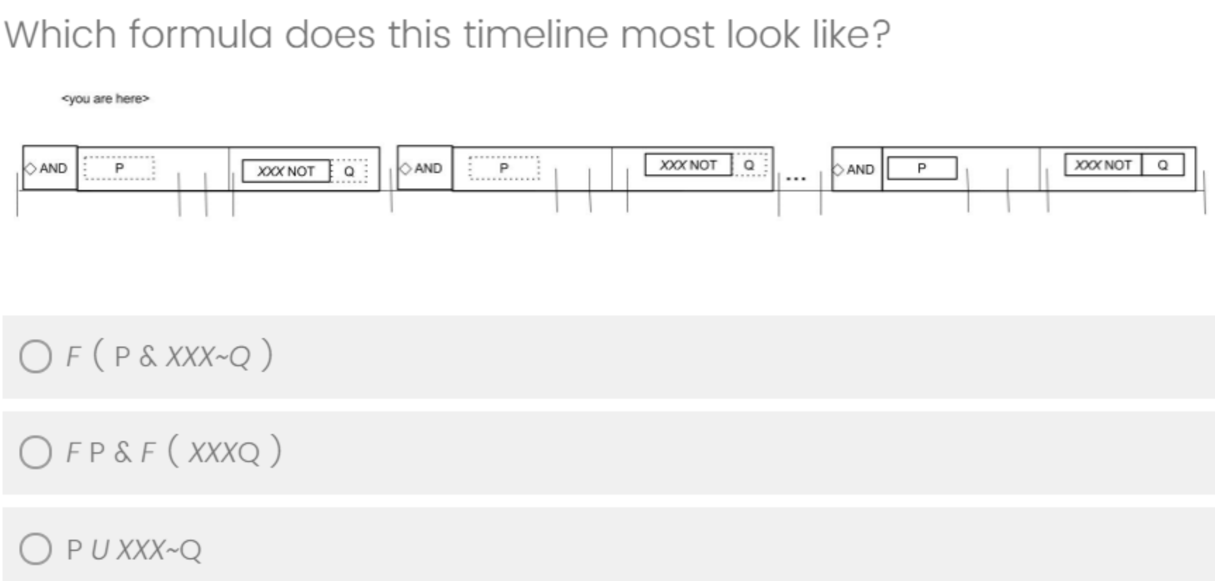
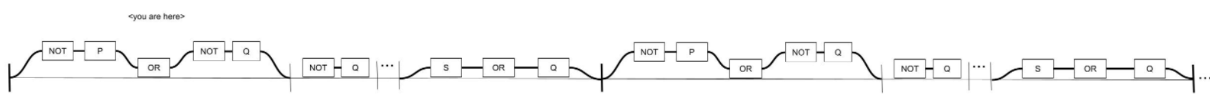


Figure 14: A question asking about the mixed formula using propositional format, Boxed Words, and temporal format, Nested Symbols.

Which formula does this timeline most look like?



○ *G* ~P | (~Q *U* S)

○ *G* (~P | (~Q *U* S))

○ G (P & (S *R* Q))

Figure 15: A question asking about the mixed formula operator using propositional format, String, and temporal format, Basic.