A Misconception-Driven Adaptive Tutor for Linear Temporal Logic

Siddhartha Prasad
Tim Nelson
Shriram Krishnamurthi

BROWN

Ben Greenman

THE UNIVERSITY OF UTAH

Siddhartha Prasad

PhD student at Brown

Q. Anyone teach LTL?

Q. Anyone teach LTL?

LTL = Linear Temporal Logic

G ~ *Always*

F ~ *Eventually*

U ~ *Until*

X ~ *in Next state*

LTL = Linear Temporal Logic

G ~ *Always* | **F** ~ *Eventually* | **U** ~ *Until* | **X** ~ *in Next state*

"The blue light turns on exactly once"

LTL = Linear Temporal Logic

G ~ *Always* | **F** ~ *Eventually* | **U** ~ *Until* | **X** ~ *in Next state*

"The blue light turns on exactly once"

```
!blue U (blue & X(G(!blue)))
```

LTL = Linear Temporal Logic

G ~ *Always* | **F** ~ *Eventually* | **U** ~ *Until* | **X** ~ *in Next state*

"The blue light turns on exactly once"

`!blue U (blue & X(G(!blue)))`

`F(blue & X(G(!blue)))`

Near miss!

LTL = Linear Temporal Logic

G ~ *Always*

F ~ *Eventually*

U ~ *Until*

X ~ *in Next state*

"Green is on for 3 ticks"

LTL = Linear Temporal Logic

G ~ *Always* | **F** ~ *Eventually* | **U** ~ *Until* | **X** ~ *in Next state*

"Green is on for 3 ticks"

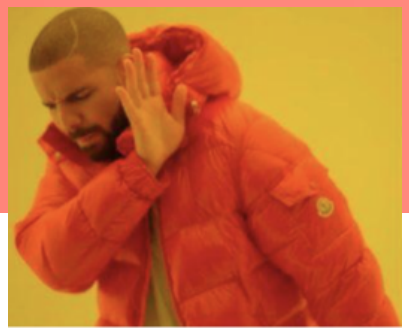
```
green & X(green) & X(X(green))
```

LTL = Linear Temporal Logic

G ~ *Always* | **F** ~ *Eventually* | **U** ~ *Until* | **X** ~ *in Next state*

"Green is on for 3 ticks"

✓ `green & X(green) & X(X(green))`



✗ `X(X(X(green)))`


5 years of study

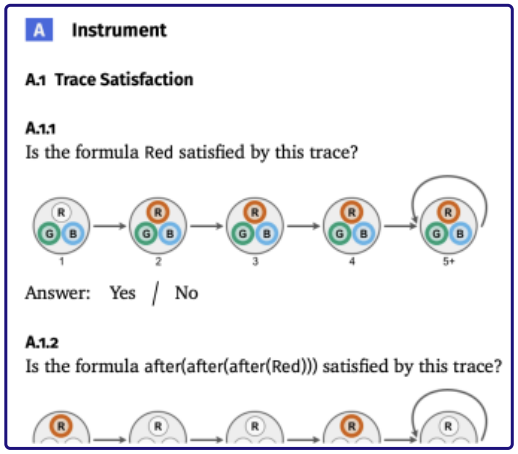
Bad Prop	Bad State Index	Cycle G
Bad State Quant.	Exclusive U	Implicit F
Implicit G	Implicit Prefix	Other Implicit
Spreading X	Trace Split U	Weak U

Catalog of Misconceptions

Bad Prop	Bad State Index	Cycle G
Bad State Quant.	Exclusive U	Implicit F
Implicit G	Implicit Prefix	Other Implicit
Spreading X	Trace Split U	Weak U

Catalog of Misconceptions

Survey Instruments



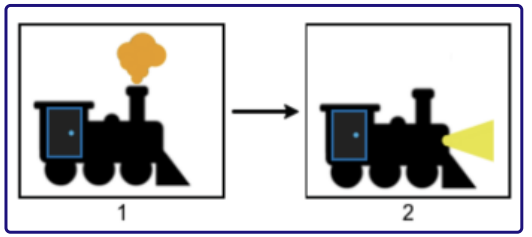
A Instrument

A.1 Trace Satisfaction

A.1.1
Is the formula Red satisfied by this trace?

Answer: Yes / No

A.1.2
Is the formula after(after(after(Red))) satisfied by this trace?



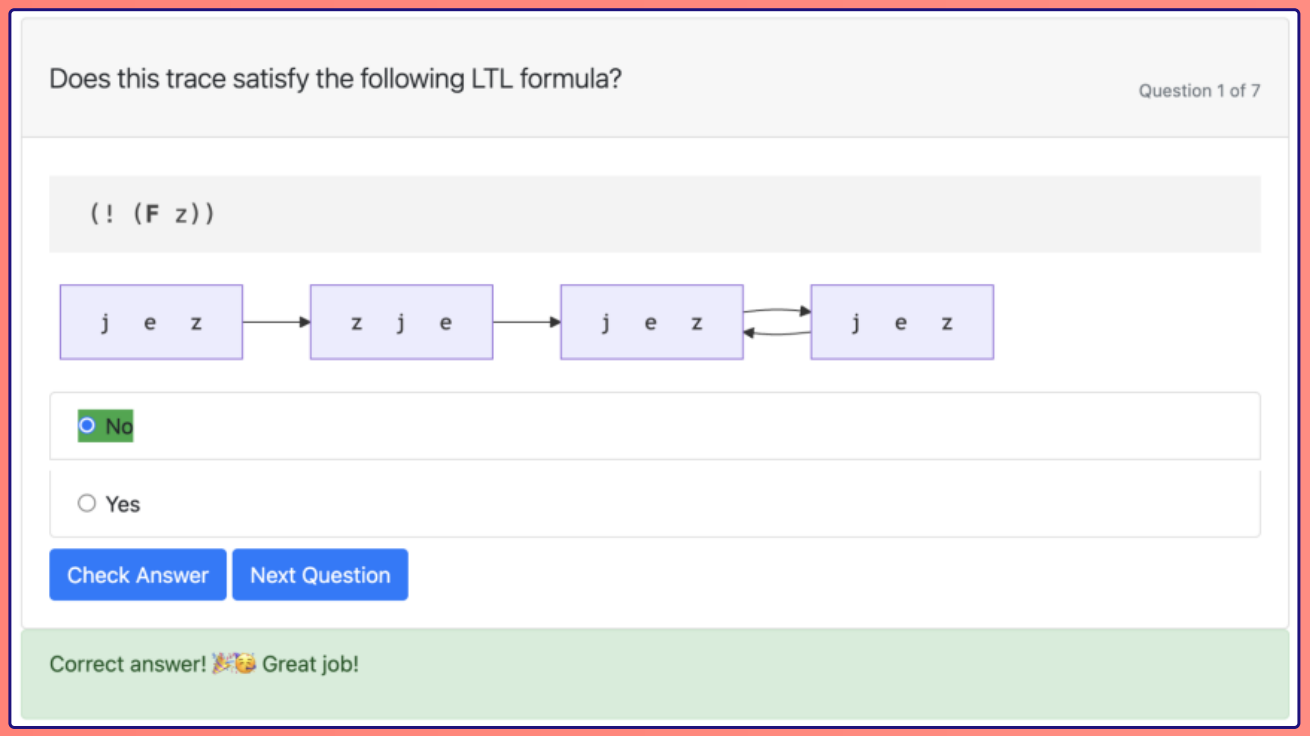
Part 3 of 3: Translate English to formulas

Translate the following English sentences to LTL formulas.

- If you do not know how to express an idea in LTL, write "I

Today:

Adaptive LTL Tutor



Lots of Examples

Adaptive

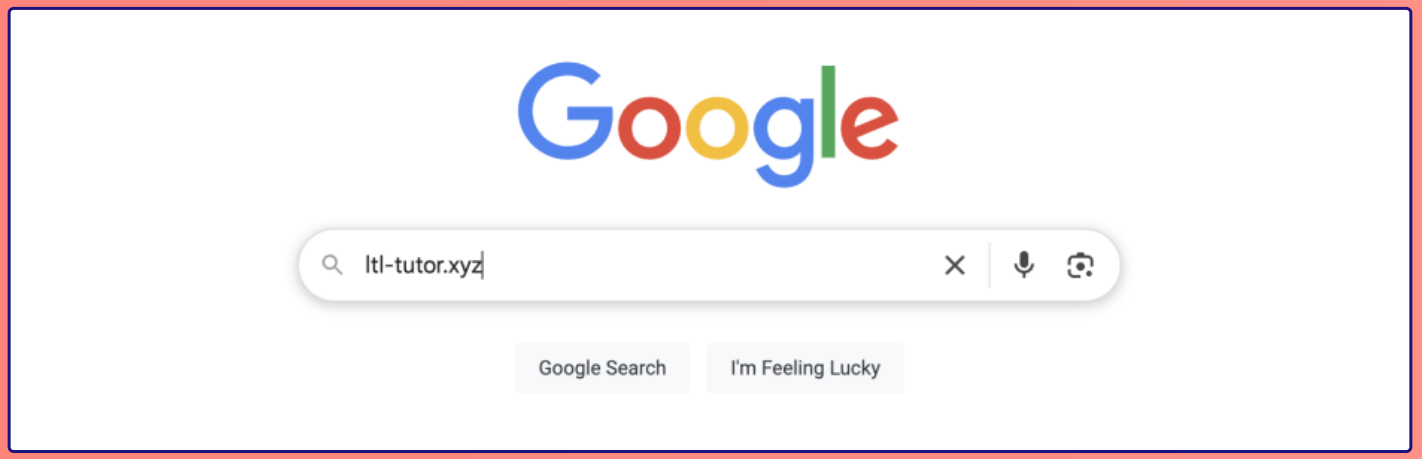
Instant Feedback

Curriculum-Agnostic

Self-Hostable

yes, GDPR

Demo


Google
ltl-tutor.xyz
Google Search
I'm Feeling Lucky

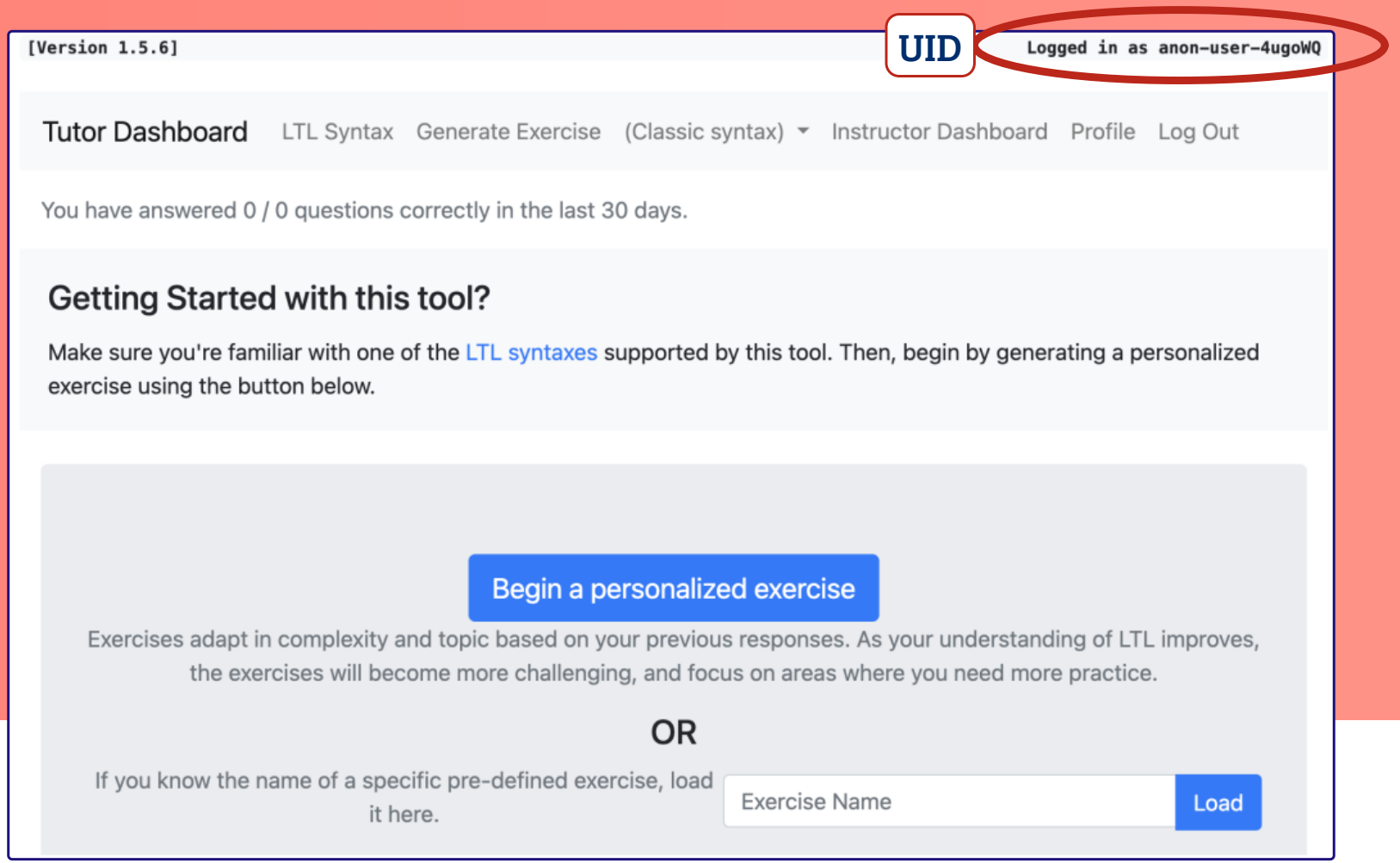
LTLTutor is built and maintained by the PLT Group at Brown University. We may use your responses to exercises for research purposes, but will not share responses with any third parties. By using this tool, you consent to this use of your responses. If you do not consent, please close this page, and do not use the LTL Tutor.

Quick Start

'Quick Start' lets you use the LTL tutor without creating an account. However, you will not have access to your account on other devices and browsers.

Have a course code? ▶

Instructor Login ▶

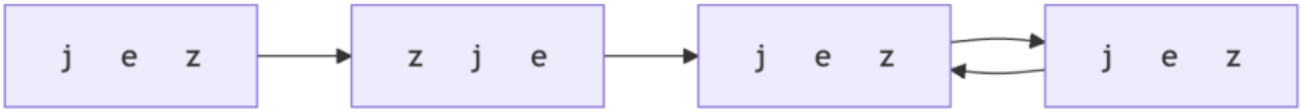

[Version 1.5.6]
UID
Logged in as anon-user-4ugoWQ
Tutor Dashboard
LTL Syntax
Generate Exercise
(Classic syntax)
Instructor Dashboard
Profile
Log Out
You have answered 0 / 0 questions correctly in the last 30 days.
Getting Started with this tool?
Make sure you're familiar with one of the LTL syntaxes supported by this tool. Then, begin by generating a personalized exercise using the button below.
Begin a personalized exercise
Exercises adapt in complexity and topic based on your previous responses. As your understanding of LTL improves, the exercises will become more challenging, and focus on areas where you need more practice.
OR
If you know the name of a specific pre-defined exercise, load it here.
Exercise Name
Load

Correct Answer

Does this trace satisfy the following LTL formula?

Question 1 of 7

```
(! (F z))
```



- (●) No
- ( ) Yes

Check Answer | Next Question

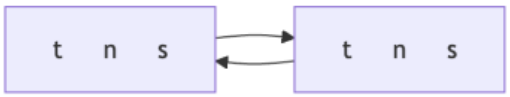
Correct answer! 🎉🥳 Great job!

Incorrect, Trace-Match

Does this trace satisfy the following LTL formula?

Question 3 of 5

```
(F (! (G t)))
```



- No
- Yes

Check Answer | Next Question

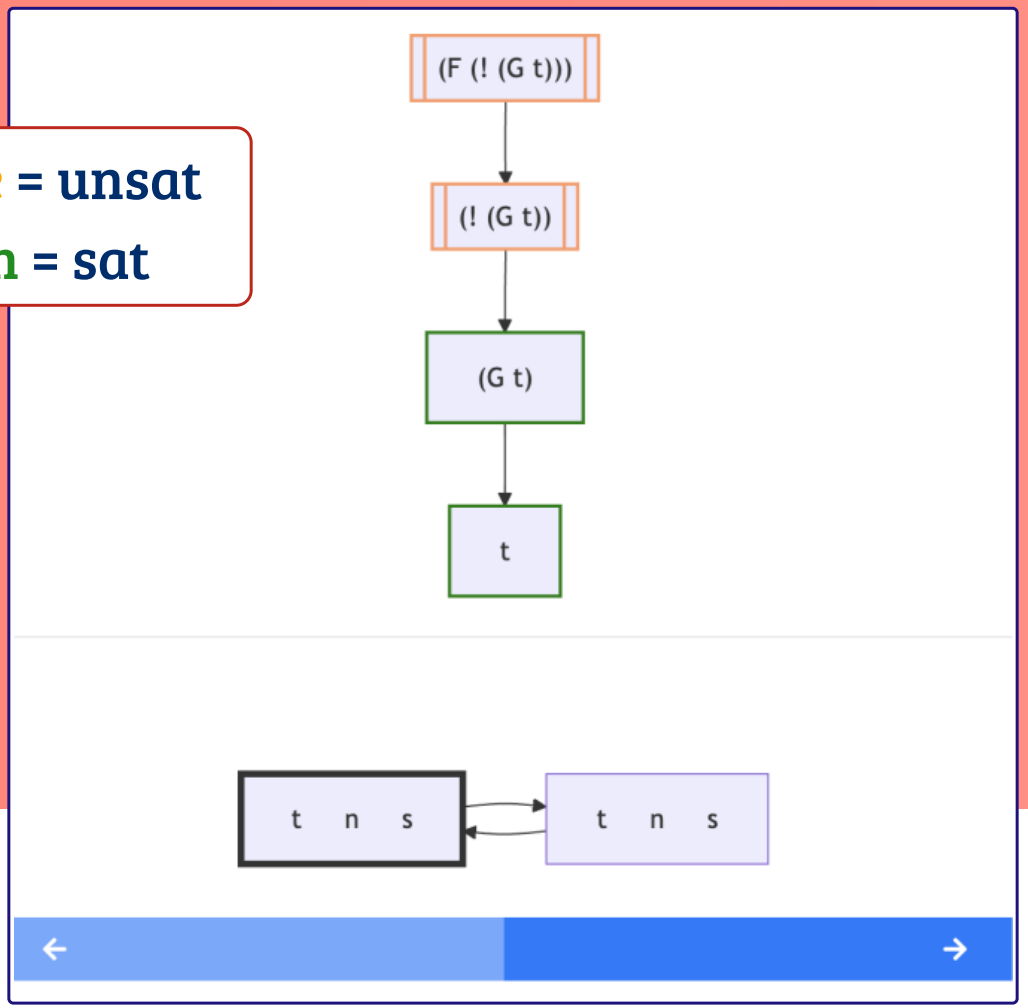
That's not correct 😕 Don't worry, keep trying! The correct answer is highlighted in green (i.e: No)

Step through the trace and your answer.

The trace is accepted by the formula `(F (G (G t)))`, but not by the formula `(F (! (G t)))`.

Orange = unsat

Green = sat



Incorrect Eng-to-LTL

Which of the following LTL formulae best represents this English sentence?

Question 2 of 8

whenever 's' holds, eventually 'c' holds

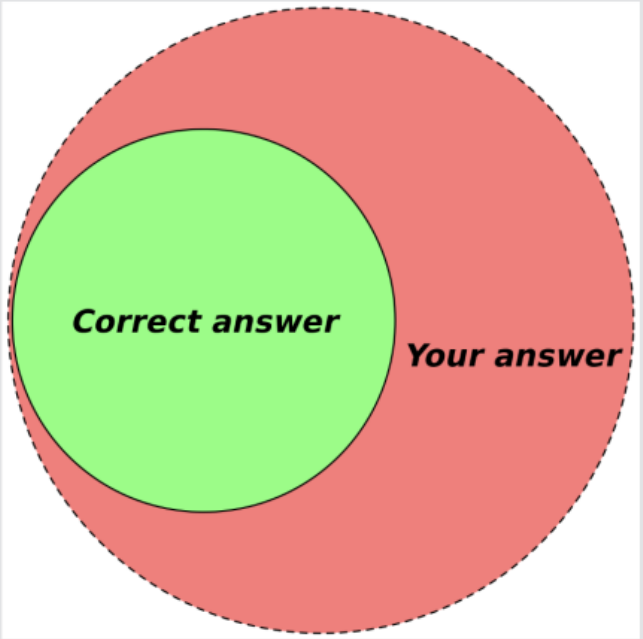
Is this English sentence confusing or unclear?

- `(G (s -> (F c)))`
- `(F (s -> (F c)))`
- `(s -> (F c))`

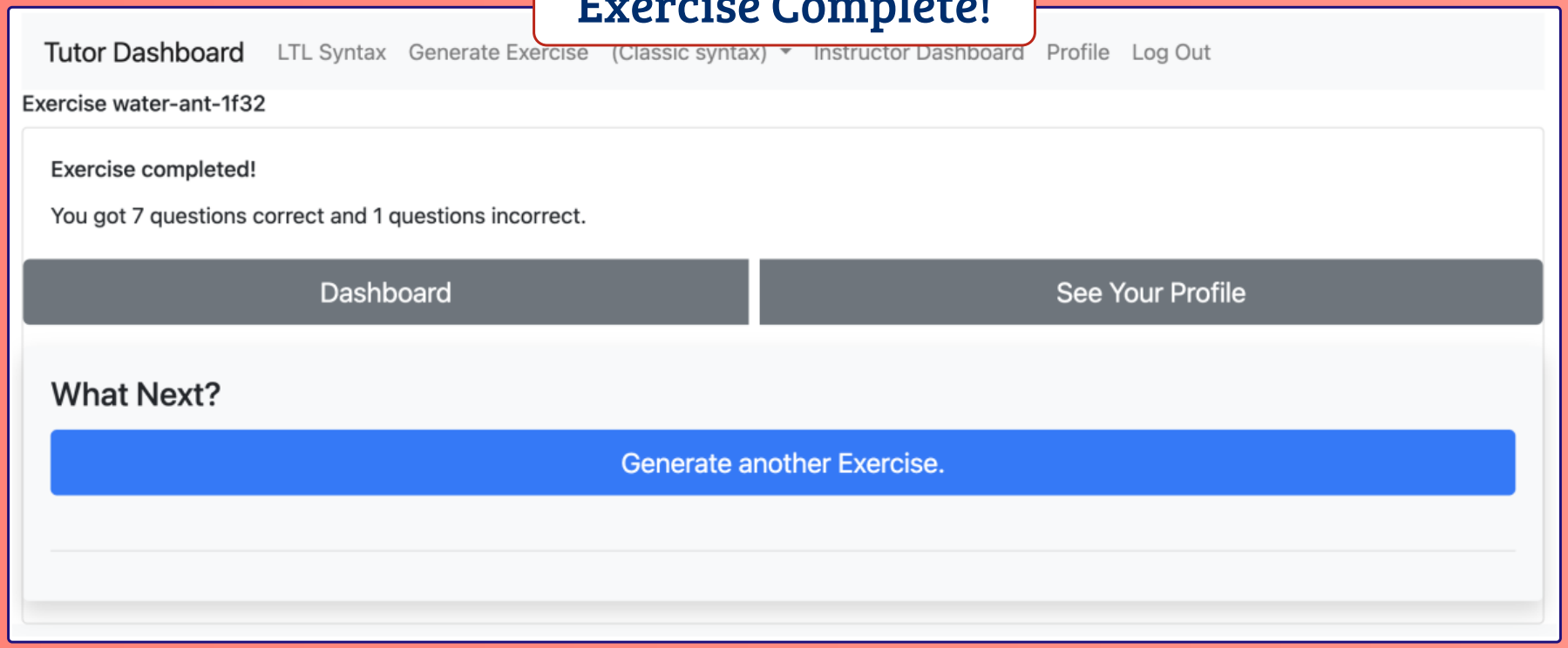
That's not correct 😕 Don't worry, keep trying! The correct answer is highlighted in green (i.e: `(G (s -> (F c)))`)

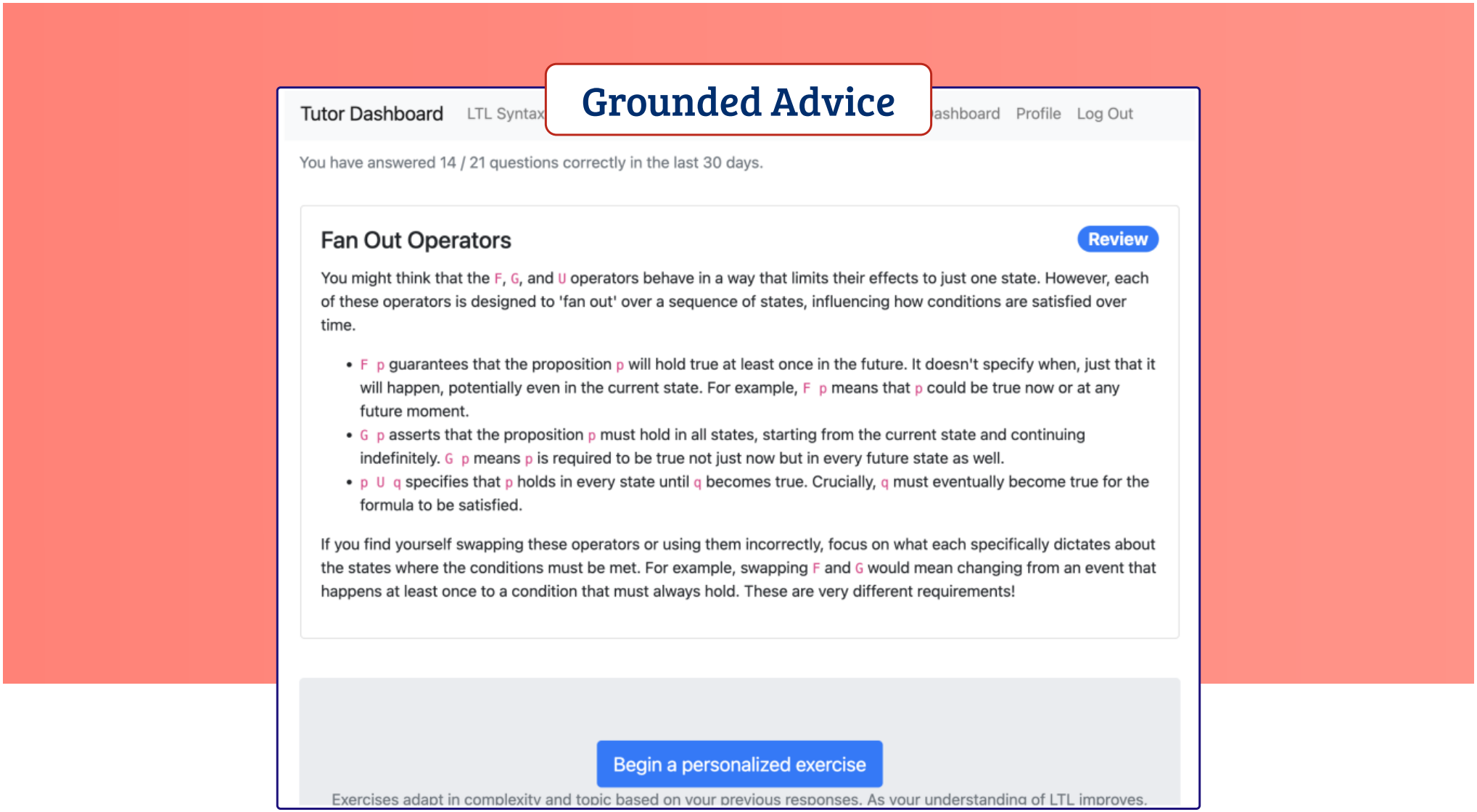
Your selection is more permissive than the correct answer. Here is a trace that satisfies your selection, but not the correct answer:

Alt Trace: c;! c & s;cycle{! c;! c}



Exercise Complete!



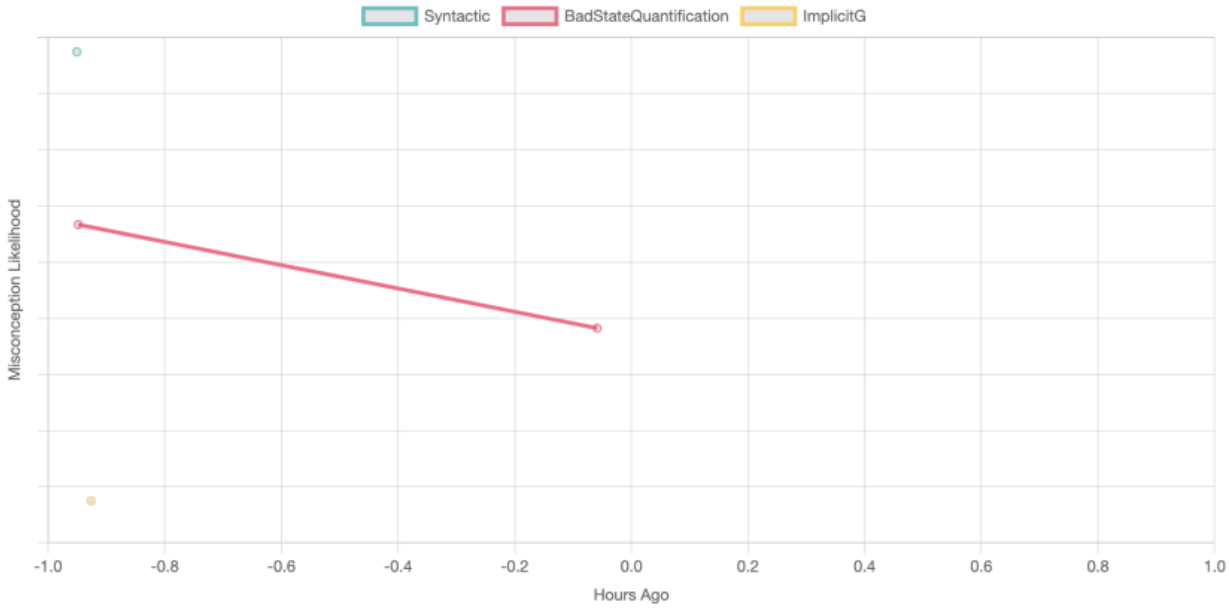

Grounded Advice
Tutor Dashboard
LTL Syntax
ashboard
Profile
Log Out
You have answered 14 / 21 questions correctly in the last 30 days.
Fan Out Operators
Review
You might think that the F, G, and U operators behave in a way that limits their effects to just one state. However, each of these operators is designed to 'fan out' over a sequence of states, influencing how conditions are satisfied over time.
F p guarantees that the proposition p will hold true at least once in the future. It doesn't specify when, just that it will happen, potentially even in the current state. For example, F p means that p could be true now or at any future moment.
G p asserts that the proposition p must hold in all states, starting from the current state and continuing indefinitely. G p means p is required to be true not just now but in every future state as well.
p U q specifies that p holds in every state until q becomes true. Crucially, q must eventually become true for the formula to be satisfied.
If you find yourself swapping these operators or using them incorrectly, focus on what each specifically dictates about the states where the conditions must be met. For example, swapping F and G would mean changing from an event that happens at least once to a condition that must always hold. These are very different requirements!
Begin a personalized exercise
Exercises adapt in complexity and topic based on your previous responses. As your understanding of LTL improves.

Your Profile

In the last 365 days, you have answered **10 out of 14** question(s) correctly.

View Logs | View Generated Exercises

Estimated Misconceptions Over Time

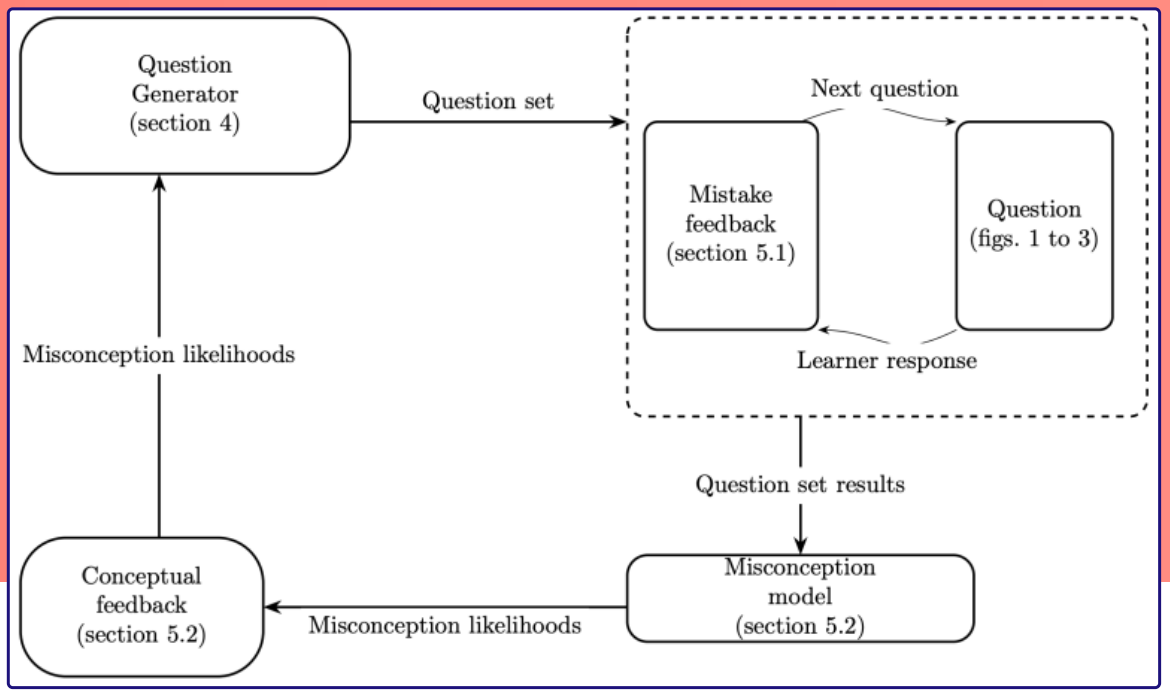


End of Demo

More in Paper:

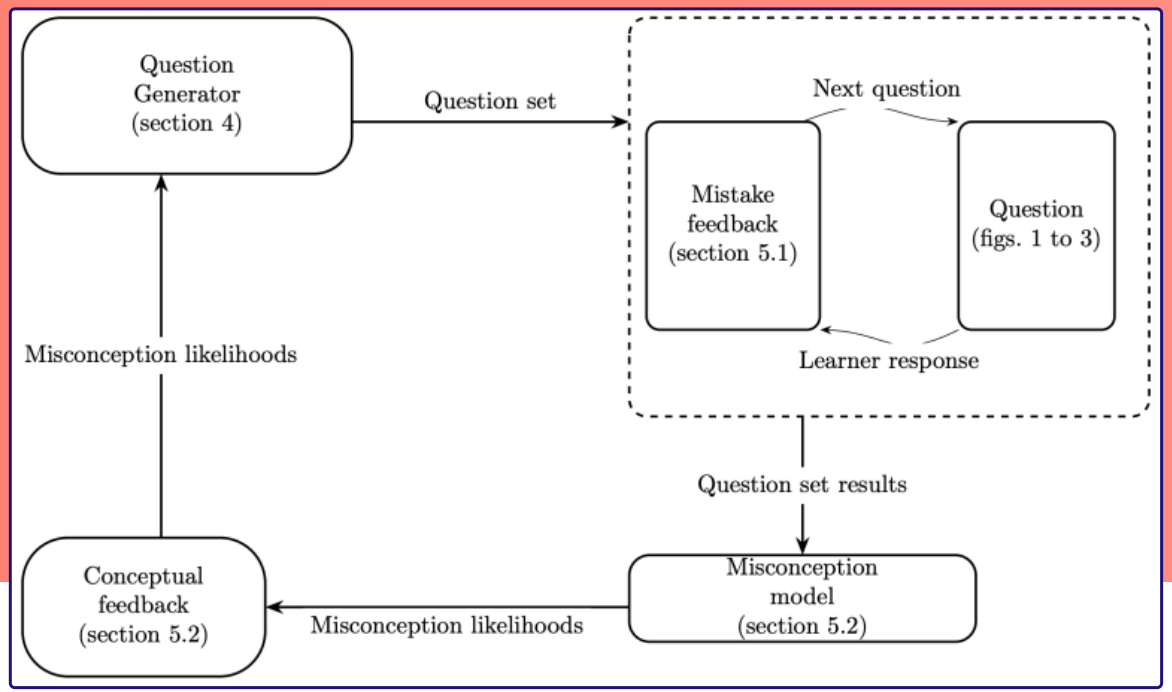
More in Paper:

Design



More in Paper:

Design



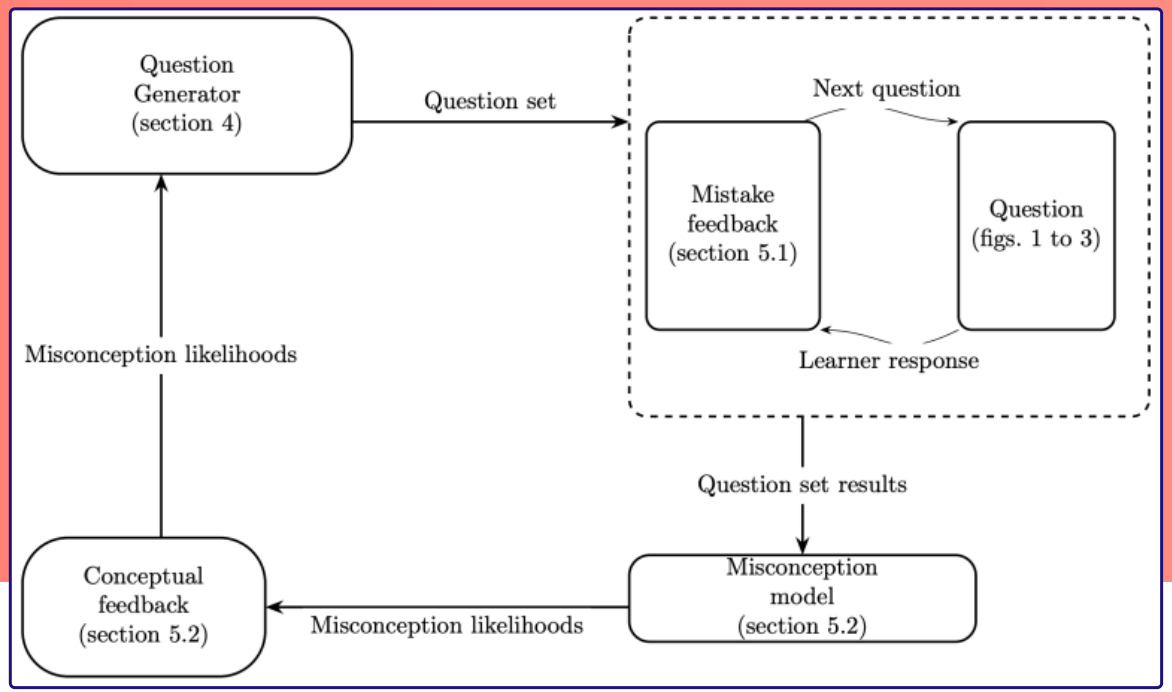
Concept-Based Mutation

A **U** (!A & B) ⟶ A **U** B

Exclusive U

More in Paper:

Design



Concept-Based Mutation

A **U** (!A & B) ⟶ A **U** B

Exclusive U

Correct Translations

(no LLMs)

$\mathbf{X}^{N}(A)$

"In N steps, A will hold"

https://ltl-tutor.xyz

Open Source

Adaptive

Lightweight

Try it!