Over time for single monte carlo trials

Alec and I tried to make some toy examples for me to create the visuals with, however when we did not understand the output of r2u2.

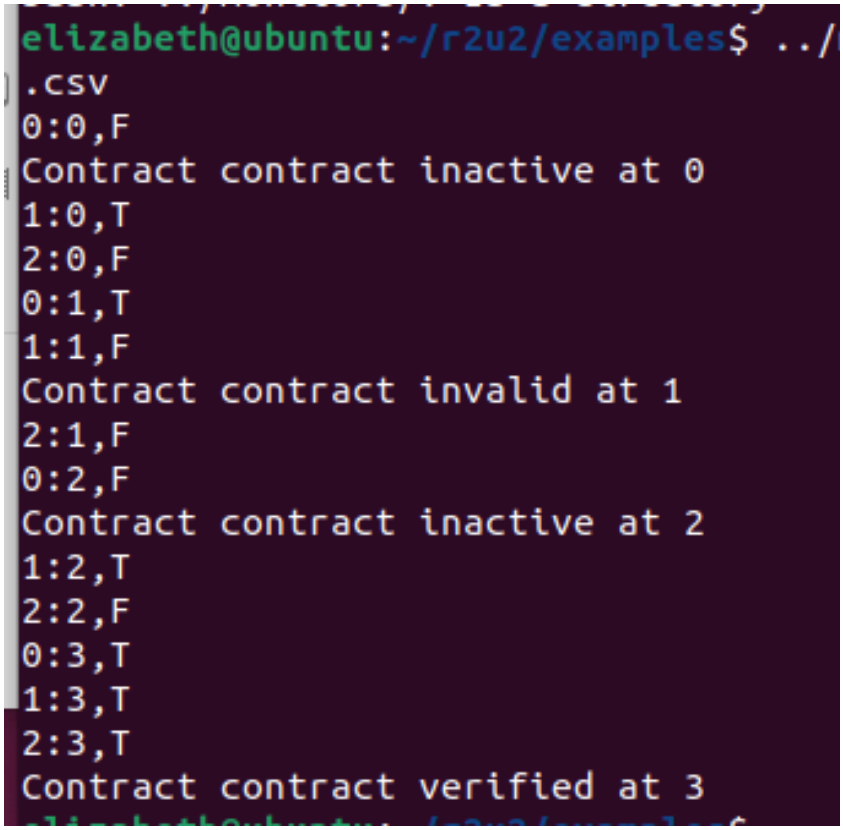
We used an example of the assume guarantee contract that is built into r2u2 and got outputs.

However, the outputs are in a form which is different from what I was expecting the output to be. We asked Chris (the person who's been working on the examples) what the outputs mean, but it's still unclear.

My plan for next week is to understand what the outputs mean and how they relate to the understandability of the system's outputs and in the meantime, create my own outputs in the format that I was expecting the outputs to be in, and testing if the visuals will work.

```
1 INPUT
2     b0,b1: bool;
3
4 FTSPEC
5     contract: b0 => b1;
```

	Standard	Standard
1	# b0	b1
2	0	0
3	1	0
4	0	1
5	1	1



a contract can be “inactive”, “verified”, or “invalid”. A contract like “A => G” is “inactive” if “A” is F, “verified” if both “A” and “G” are T, and “invalid” if “A” is T but “G” is F. This is just a fancy way of interpreting the semantics of implies:

```
A | G | A -> G  | status
------------------------
F | F |   T    | inactive
F | T |   T    | inactive
T | F |   F    | invalid
T | T |   T    | verified
```

To get this to work in R2U2, we break “A => G” into three formulas:

f0: “A”
f1: “A -> G”
f2: “A & G”

If f0 is F we report “inactive", else if f1 is F we report “invalid", else if f2 is T we report “verified".

So in the output you’re seeing, each line gives <formula ID> : <timestamp> , <verdict>. At time 0 you have that f0 is F, f1 is T and f2 is F, so we report "inactive”.

1. **Design and implement a back-end** that takes as input R2U2 time-verdict sequences, or interfaces directly with R2U2, to post-process sets of verdict sequences for subsequent analysis and review. The back-end product could be text, graphical, or a combination

   Given our Aggregate data, we can say that it always fails at this one spot, and from there we can ask why and how, and we can learn more about our system.

   For Example: we have **G[0,5](a | b | c)**
   and 50 runs of this, we always fail at time 3, or 4 where all variables are false.
   Range of trace values across the timeframe.

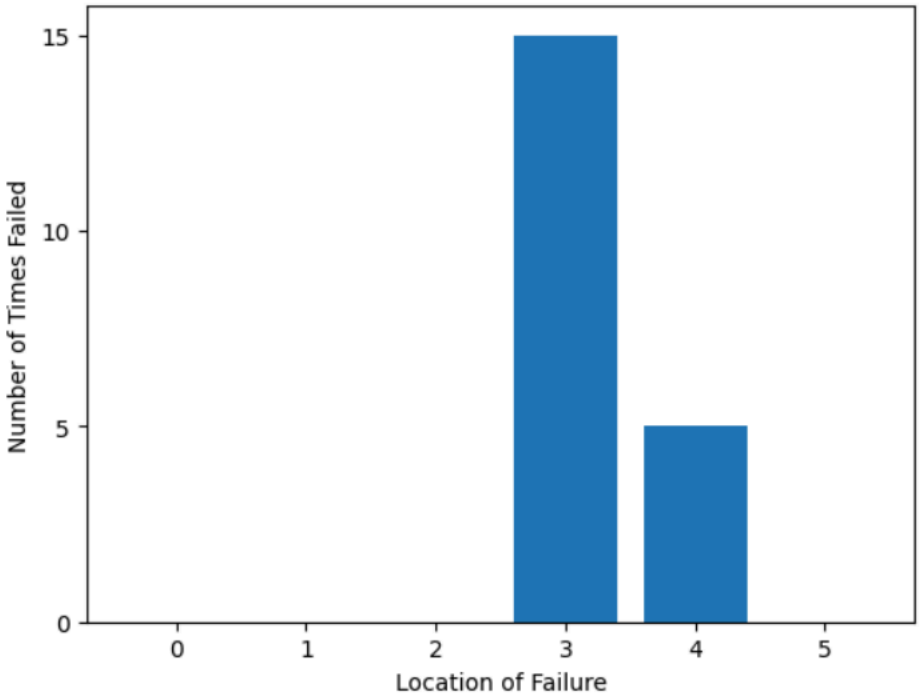
   a b c over 0-5

   Visual 1:
   1 particular run

a	F	F	F	F	F	F
b	F	T	F	F	F	T
c	T	T	T	F	T	T
	0	1	2	3	4	5

   Data for 50 runs:
   # of Failures: 20
   Failure Locations:
   - 15 at 3
   - 5 at 4

   Graph of failure Locations



Visual 2: Aggregate heat map of failure locations

	Number of Failures						Total
	0	0	0	15	5	0	20
a				F	F		
b				F	F		
c				F	F		
t	0	1	2	3	4	5	

*Create a marker when files fail

Thoughts: some contracts may be 'global' (not associated with specific task, e.g. resource predictions). Some contracts may be 'task-centric' (precondition of task instances). Is it valuable to distinguish up-front between such contract failures?

Why could this be useful: contracts for sin... by Frank, Jeremy D. (ARC-TI) (Unverified)Frank, Jeremy D. (ARC-TI) (Unverified)11:02 AM

Why could this be useful: contracts for single tasks are likely to fail only once. 'global' contracts might fail often at different times.

has context menu

-
- The most common failure trace is listed at each time step
- Another Example: **G[0,5](a & b)**

  Data for 50 runs:
  # of runs: 50
  # of Failures: 20
  Failure Locations:
  - 15 at 3
  - 5 at 4

	Number of Failures						Total
Prop	0	0	0	15	5	0	20
a				T	F		
b				F	F		
t	0	1	2	3	4	5	

Aggregate Timeline Data							Total
Time Step	0	1	2	3	4	5	6
Number of Failures	0	0	0	15	5	0	20
system_1				T	F		
system_2				F	F		

Additionally:
Number of times which variable fails to cause overall failure gets heat mapped

	Number of Failures						Total
prop	0	0	0	15	5	0	20
a				5	2		
b				10	3		Runs

event name	A	B	C	D	E	F	50

Lookup table: reference (event based, not time)
Future Work:

What went wrong rather than where it went wrong.
Magnitude of failure? - Some appear on CNN
Weight assume-guarantee violation for multiple contracts?

Daveism: You can't guide your ship to harbor if you don't know where the harbor is.

Micky: quickly identify what events are happening and where
what groupings of tasks interact that cause sometimes a fault
Layout why we need the visualizations (requirements for the work)

Add a fine point to the

Possible future work: solution to enumerating possible failure modes

Theoretical Contribution - Cores of a Formula: Minimum unsatisfiable cores for LTL

- read work, this would be a publishable result - a way of explaining why

TODO: Look for related work.

Document that. Make a bibtek file, write a related work part of the thesis. 2 sentences per paper. What they did. How is it different from me? ~75 citations

Can I learn form or build upon

2. **Design and implement a back-end** that takes as input R2U2 time-verdict sequences, or interfaces directly with R2U2, to post-process sets of verdict sequences for subsequent analysis and review. The back-end product could be text, graphical, or a combination

    - Simulation aggregation examples, how does that apply to r2u2 outputs in the visual form.
    - Some minimum influential subformula here

3. **Provide a mechanism to display the outputs from R2U2** from analyzing a large set of the timeline representation data, aiding identification of off-nominal or "interesting" executions-at-atomic-event-sequences (the inputs to R2U2).
    a. Make it more interpretable
    b. Ways to draw it on a timeline (make it pretty~)
    c. Relate it to displaying a counterexample

Ideally this will be expanded to include some visual representation that domain users find intuitive to interact with.

4. **Create a capability for two-way traceability between the timeline executions and the R2U2 verdicts.**
    a. Evaluate partial formula, find unsatisfiable core

    Means min influential sum formula here

    Given an R2U2 verdict, show the connection back to the original timeline execution in a way that enables users to make changes to the timeline execution (e.g., through re-planning) to change the corresponding R2U2 analysis that checks the timeline execution against an assume-guarantee contract. This capability will enable a straightforward way to fix any timeline executions found to violate assume-guarantee contract requirements.

    - TImeline provides input trace
    - What caused it to say false
    - timeline execution = input trace
    - r2u2 verdict = output

Agreement Points

1. **Design and implement a back-end** that takes as input R2U2 time-verdict sequences, or interfaces directly with R2U2, to post-process sets of verdict sequences for subsequent analysis and review. The back-end product could be text, graphical, or a combination
2. **Provide a mechanism to display the outputs from R2U2** from analyzing a large set of the timeline representation data, aiding identification of off-nominal or “interesting” executions-at-atomic-event-sequences (the inputs to R2U2).
   Ideally this will be expanded to include some visual representation that domain users find intuitive to interact with.

3. **Create a capability for two-way traceability between the timeline executions and the R2U2 verdicts.**

    Given an R2U2 verdict, show the connection back to the original timeline execution in a way that enables users to make changes to the timeline execution (e.g., through re-planning) to change the corresponding R2U2 analysis that checks the timeline execution against an assume-guarantee contract. This capability will enable a straightforward way to fix any timeline executions found to violate assume-guarantee contract requirements.

MOUS5 Information Architecture

Main screen:

- Open Project
    - File Explorer
        - Use selects folder containing C2PO, MAP, Traces folder
            user opens project folder with exactly one of each
            gives error if **files are not found** or **if more than one is present internally:**
            MOUS5 sets the location of the files to the files that r2u2 will run
            run r2u2 on files
    - **Project Explorer**

- Create New project
    - File creator (similar to this site https://codepen.io/pen/)
        - C2PO Text Editor
            - Save C2P0 File
        - MAP Text Editor
            - Save Map File
        - .CSV Text Editor
            - Save .csv in trace folder
            Can also be created visually*

Project Explorer

- Hide Successful
- Hide Failed
- Show 0 Instance
- Show All
- View File Editors
    - c2po, map, traces
- Sort Columns
-