Experiments in runtime monitoring via probabilistic session types

@GSSI
C. Bartolo Bulò, A. Francalanza, C. Trubiani, E. Tuosto

@UoH
A. Scalas

@DTU

IT-Matters 12/4/2021

To be presented @ COORDINATION 2021
What I'm going to talk about:

- Binary session types in a nutshell

- A probabilistic variant

- Algorithmic derivation of (passive) monitors
  - are oblivious of participants' actual implementation
  - approximate participants' probabilistic observable behaviour
  - emit revocable judgements based on confidence intervals

- Ideas for future work
"It is unanimously agreed that statistics depends somehow on probability. But, as to what probability is and how it is connected with statistics, there has seldom been such complete disagreement and breakdown of communication since the Tower of Babel."

Binary session types in a nutshell

S runs a "lottery" game

C has to guess a number $1 \leq n \leq 100$ secretly chosen by S

seek for help
quit the game
Binary session types in a nutshell

S runs a "lottery" game

C has to guess a number $1 \leq n \leq 100$ secretly chosen by S

seek for help
quit the game

\[ \begin{aligned}
&C \rightarrow S: \text{Quit} \\
&C \rightarrow S: \text{Guess} \\
&S \rightarrow C: \text{Correct} & S \rightarrow C: \text{Incorrect} \\
&S \rightarrow C: \text{Help} \\
&S \rightarrow C: \text{Hint} \\
\end{aligned} \]
Binary session types in a nutshell

S runs a "lottery" game
C has to guess a number $1 \leq n \leq 100$ secretly chosen by S
seek for help
quit the game

```
S = rec X. & ?Guess (ht),
  (+) !Correct X,
  !Incorrect X

?Help. !Hint (str). X,
?Quit. end
```

```
C = rec X. (+) ?Guess (ht),
  & ?Correct X,
  !Incorrect X

!Help. !Hint (str). X,
!Quit. end
```
Binary session types in a nutshell

S runs a "lottery" game
C has to guess a number $1 \leq n \leq 100$ secretly chosen by S
seek for help
quit the game

```
S = rec X. & { ?Guess (ht),
                  + \{ !Correct X,
                        !Incorrect X
                  \}
                  \{ ?Help !Hint (str). X,
                    ?Quit. end
                  \}

C = rec X. (+)?Guess (ht),
    & { ?Correct X,
        !Incorrect X
      \}
    \{ !Help !Hint (str). X,
      !Quit. end
    \}
```
A probabilistic variant

\[ S = \text{rec } X. \ & \ {\text{?Guess (int)}[.25], \}
\quad \oplus \{ \begin{array}{l}
\text{Correct [.01]. } X, \\
\text{Incorrect [.99]. } X
\end{array} \}
\}
\]

\[ T ::= \ & \ {\text{?l}_{i}(s_{i}) \ [p_{i}]. T_{i}} \]
\quad \oplus \{ \begin{array}{l}
\text{!l}_{i}(s_{i}) \ [p_{i}]. T_{i}
\end{array} \}
\]

\text{rec } X. T

\text{end}

\forall i : 0 < p_{i} < 1

\sum_{i \in \mathbb{I}} p_{i} = 1
A probabilistic variant

\[ S = \text{rec } X. \& \begin{cases} \text{Guess (int)}[.25], \\ \oplus & \text{Correct [.01], } X, \\ \text{! Incorrect [.99], } X \\ \end{cases} \]

? Help [.2], ? Hint (str), X

? Quit [.05], end

What is \( S \) actually specifying?
A probabilistic variant

\[
S = \text{rec } X, \& \{ \text{Guess (int)}[.75],
    \{\text{Correct} [.01], X,
     \text{incorrect} [.997], X
    \}
\}
\]

?Help [.2], !Hint (str). X

?Quit [.05], end

What is S actually specifying?

smartArse() ->
receive
    \{ guess, G, C \} -> C ! incorrect,
    smartArse();
    help -> C ! "I wish I could help",
    smartArse();
    quit -> io:format("another sucker got it!")
end;

T ::= \& \{ \text{t}_i (\text{int}) [\text{p}_i], T_i \}_{i \in \mathbb{N}}

\text{rec } X, T

\text{end}
"on-line" monitoring of probability

Let's consider a simple frequentist approach to estimate probabilities...
"On-line" monitoring of probability

Let's consider a simple frequentist approach to estimate probabilities.

Our monitors:

- have a parameter: confidence level $0 \leq \alpha \leq 100$
"on-line" monitoring of probability

Let's consider a simple frequentist approach to estimate probabilities.

Our monitors:

- have a parameter: confidence level $0 \leq c \leq 100$

- keep estimating probabilities:

$$\hat{p}_{ij} = \frac{\# 	ext{branch } i}{\# 	ext{choices } i}$$

\[ \begin{array}{|c|c|}
\hline
l & Z-\text{value} \\
\hline
50 & 1.645 \\
60 & 1.282 \\
70 & 1.043 \\
80 & 0.675 \\
90 & 0.433 \\
99 & 0.12 \\
100 & 0.043 \\
\hline
\end{array} \]
"On-line" monitoring of probability

Let’s consider a simple frequentist approach to estimate probabilities.

Our monitors

- have a parameter: confidence level $0 \leq \ell \leq 100$
- keep estimating probabilities:

\[ \hat{p}_{ij} = \frac{\# \text{branch } ij}{\# \text{choice } i} \]

\[ \hat{p}_i, \text{guess} = \frac{\# \text{times } c \text{ selected } \text{guess}}{\# \text{times } c \text{ had } \text{relevant}} \]

\[ \hat{p}_i, \text{incorrect} = \frac{\# \text{times } c \text{ selected } \text{incorrect}}{\# \text{times } c \text{ present}} \]

- use $E_{ij} = Z(\ell) \sqrt{\frac{\hat{p}_{ij} (1-\hat{p}_{ij})}{\# \text{ choice } i}}$

\[ \text{to flag warnings when } \hat{p}_{ij} \notin [p_{ij} - E_{ij}, p_{ij} + E_{ij}] \]

Flagging may be needed.
Wrapping up

- Applications
  - Support to take decisions (e.g. flagging possible fraud detections)
  - Checking AI's learning
  - Flagging faulty components

- Future work
  - Try different estimators or different warning policies
  - Non-uniform confidence levels
  - "Global" probabilistic behaviour (PdF ("a big purchase shortly after small ones") is very small)
  - Can we formulate monitors' correctness?
  - What is monitorable?
  - When are constraints on probabilities inconsistent?
Thanks!