Group Meeting - 4

Member: Yong Li, Depeng Liu, Weizhi Feng, Xie Li, Shizhen Yu, Zongxin Liu

June 16, 2021

MemSafety: Progress and Problem Encountered

Previous problem: add separation logic assertion to front-end.

- Memory configuration using SL formulae.
- Assertions using SL formulae.

Give the syntax of the program considered and the semantic (or symbolic execution rule).

$$
\begin{aligned}
e &::= n \mid x \mid x + n \\
B &::= e = e \mid e \neq e \mid e \leq e \\
S &::= x := e \mid x := \texttt{load}(e) \mid \texttt{store}(e, e) \mid x := \texttt{malloc}(e) \mid \texttt{free}(x) \\
C &::= B \mid C; C \mid \texttt{if}(B)\texttt{then}\{C\}\texttt{else}\{C\}
\end{aligned}
$$

A Simple Case

Example

```
int main(){
    // emp
    int *i = malloc(2*sizeof(int));
    // blk(i, i+8)
    int *j = malloc(sizeof(int));
    // blk(i, i+8) * blk(j, j+4)
    *i = 0;
    // i ↦ 0 * blk(i+4, i+8) * blk(j, j+4)
    *j = 1;
    // i ↦ 0 * blk(i+4, i+8) * j ↦ 1
    free(i);
    // j ↦ 1
    free(j);
    // emp

}
```

Symbolic Execution Rules

Symbolic Heap: $Q \equiv \Pi \mid \Sigma$

$$\Pi \mid \Sigma \Rightarrow_{\{x:=e\}} \Pi[x'/x] \wedge x = e[x'/x] \mid \Sigma[x'/x]$$

$$\Pi \mid \Sigma * e \mapsto E \Rightarrow_{\{x:=[e]\}} \Pi[x'/x] \wedge x = E[x'/x] \mid \Sigma[x'/x]$$

$$\Pi \mid \Sigma \Rightarrow_{\{x:=\texttt{malloc}(e)\}} \Pi[x'/x] \mid \Sigma[x'/x] * x \mapsto e[x'/x] * \texttt{blk}(x+1, x+1+e[x'/x])$$

$$\Pi \mid \Sigma * e_0 \mapsto e' \Rightarrow_{\{[e_0]:=e_1\}} \Pi \mid \Sigma * e_0 \mapsto e_1$$

$$\Pi \mid \Sigma_0 * \texttt{blk}(e, f) * \Sigma_1 \Rightarrow_{\{[e_0]:=e_1\} \wedge e \leq e_0 < f} \Pi \mid \Sigma_0 * \texttt{blk}(e, e_0) * e_0 \mapsto e_1 * \texttt{blk}(e_0 + 1, f) * \Sigma_1$$

$$\Pi \mid \Sigma_0 * x \mapsto e * \Sigma_1 * \Sigma_2 \Rightarrow_{\{\texttt{free}(x)\} \wedge \texttt{linked}(\Sigma_1) \wedge \texttt{tail}(\Sigma_1)} \Pi = x + e + 1 \mid \Sigma_0 * \Sigma_2$$

Problems

- How to distinguish the fresh variables?
- For the free rule, where to put the condition.
- The formula will explode if using the rules.
-

MemSafety: TODOs

- Implement the above transformation into the frontend SMACK and try to generate SL formula in boogie.
- Modify the Boogie parser in Boogie: require some work to clarify the syntax.

Weizhi Feng: Outline

- Bounded Büchi automata.
- Reading.
- Plan.

BBA: Last Week

- Proved the intersection of bounded languages is not closed.
- Plan: thinking about union and complementation;

BBA: This Week

- Consider the construction of the union of bounded Büchi automata.
  - Still thinking, not sure..
  - $d = lcm(d_1, d_2)$;
  - Extend the loop starting and ending from the accepting states.
  - We should ensure the accepting word won't change.

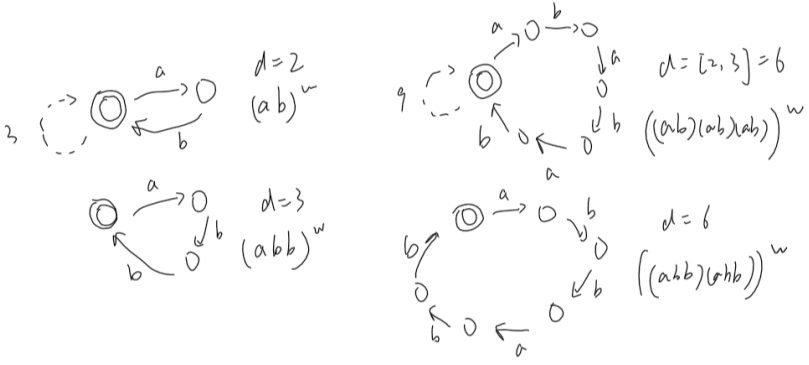

Figure 1: union

BBA: This Week

- Consider the subsets of bounded Büchi language;
  - The relationship of the property (e.g. safety) described by LTL and bounded Büchi language.

Reading

- Reading some paper about automata.
  - LTL2BA;
  - Unambiguous Büchi automata complementation.

Plan

- Consider the union and complementation;
- Consider the relationship of LTL and bounded Büchi language.

Pufferfish/Differential Privacy

Currently:

- Pufferfish: Submit the paper to FM;
- Differential privacy: Write the epmc plugin;

Plan:

- Differential privacy: Complete the plugin in a week; Then write descriptions.
- Pufferfish: Can extend to a journal paper with existing materials.
- NN attack: Evaluate privacy preservation/accuracy effect for reconstructed neural networks with PAC learning?

FDFA model checking

Currently:

- Assume the model language is an ω-regular language; Then the FDFA learning algorithm can be applied to learn the model.
- With the FDFA model and the FDFA converted by LTL property, model checking can be done by FDFA inclusion checking.

Plan:

- If applicable, do experiments with ROLL and our algorithm; Compare with?
- Find a class of LTL that is more suitable for FDFA than Büchi?