AGH

AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE

System properties by example

Safety and liveness in first order logic and temporal logic

29 marca 2021

1 General problems

How to define system properties in each formal systems (PL, FOL, TL)? What about defining properties using theories (like integers)? What about using constraint programming?

There are formal definitions of system properties, but I do not know them, they are quite complex. Informally, you just need to find a way of saying: 'bad thing will never happen' – safety, 'something good will eventually occur' – liveness. At least this is how I understand it. In FOL and TL there are good tools for doing such: FOL – quantifiers $\forall$ for all, $\exists$ exists, TL – operators $\Box$ always and $\Diamond$ eventually.

Right now I am assuming user is defining constrains directly in logical system, that is user is using input format similar to TPTP. This requires extensive FOL knowledge and I would say is not a likely business scenario. Business would be in favour of using constraint programing to express properties in tools like MiniZinc or format like SMT-LIB. Unfortunatelly generating random formulas in mentioned languages can easily lead to semantic gap (small change in model, causes huge change in model). Semantic gap would make results of benchamring even more random and unpredictable.

How to define equivalent property in dfferent language and compare performance? Are there converters which preserve equivalence?

More research needed.

How do peple in real world scenarios encode system properties?

It is always some kind of formal system, for example logical systems or formalism provided by standard Minizinc, SMT-LIB, TPTP or input syntax for prover.

2 Representing safety and liveness in FOL and TL

I aim to represent sytem properties in following format, independent from logical system used:

$$safetyClause \land livenessClause \land \ldots \tag{1}$$

where clause is formula with atoms connected with logical or. In FOL safety will be represented with $\forall$, liveness with $\exists$ (in implementation formulas will be encoded in skolem normal form - quantifier free formulas):

$$\forall(atom1 \lor atom2 \lor \ldots) \land \exists(atom3 \lor atom4 \lor \ldots) \land \ldots \tag{2}$$

In TL safety will be represented as operator always $\Box$ and liveless as operator eventually $\Diamond$:

$$\Box(variable1 \lor variable2 \lor \ldots) \land \Diamond(variable3 \lor variable4 \lor \ldots) \land \ldots \tag{3}$$