My DIVINE contribution

IV115 2018

Tadeáš Kučera

ParaDiSe
Parallel & Distributed
Systems Laboratory

Masaryk University

Brno, Czech Republic

February 26, 2018

Diploma thesis

Contents of my thesis

- C/C++ API for monitoring ω regular properties of verified programs
- May consider implementation of own translation of LTL
  + easier usage – bigger automata
- Implement it into DIVINE

Previuous work

PROGRESS

- LTL parser
- decided for our own embedded translation LTL to Buchi
- LTL to TGBA by D. Giannakopoulou and F. Lerda (see [5])
- implemented it and used SPOT to test it
- ltlc.cpp then generates c++ API of the TGBA
- standard (stupid) degeneralizer of TGBA

Where is it now?

- /divine/divine/ltl
- /divine/divine/ui/ltlc.cpp
- /divine/runtime/dios/lib/degeneralizer.hpp
- /divine/runtime/libc/include/sys/monitor.h

More about TGBA?

Definition (TGBA)

TGBA is a 5-touple (S, A, T, q_0, F), where

- S is a finite set of states and $q_0 \in S$ is initial state
- A is a finite alphabet (set of used atomical propositions),
- $T \subseteq S \times A \times S$ is a set of all transitions,
- $F \subseteq 2^T$ is a set of sets of accepting transitions (colors).

Definition (TGBA accepting condition)

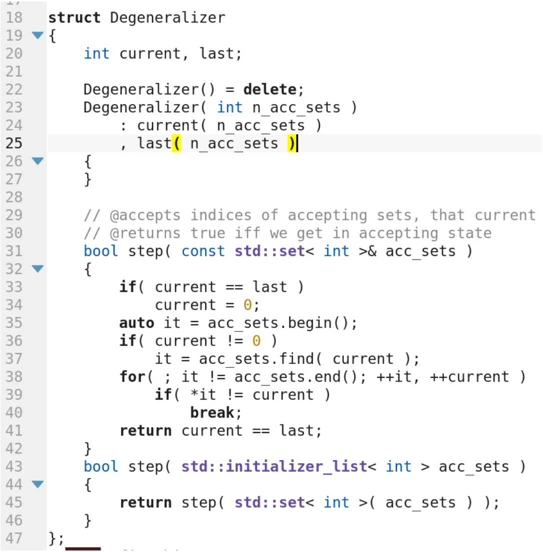
An infinite word $w \in A^*$ is accepted by the TGBA iff there exists an execution θ of the automaton on w that for every $C \in F$ contains at least one element from C infinitely many times.

Statistics

Used SPOT randltl and ltlcross to test our LTL -> TGBA on 400 random formulas (manual in [4])

Current work

- Testing the C++ API of TGBA and its Degeneralizer on some of our examples of synchronous systems
- Reading [3].
- Looking for some smarter Degeneralizer

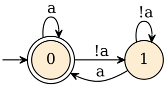


```
struct Degeneralizer
{
    int current, last;

    Degeneralizer() = delete;
    Degeneralizer( int n_acc_sets )
        : current( n_acc_sets )
        , last( n_acc_sets )
    {
    }

    // @accepts indices of accepting sets, that current
    // @returns true iff we get in accepting state
    bool step( const std::set< int >& acc_sets )
    {
        if( current == last )
            current = 0;
        auto it = acc_sets.begin();
        if( current != 0 )
            it = acc_sets.find( current );
        for( ; it != acc_sets.end(); ++it, ++current )
            if( *it != current )
                break;
        return current == last;
    }
    bool step( std::initializer_list< int > acc_sets )
    {
        return step( std::set< int >( acc_sets ) );
    }
};
```

Future work

Smarter Degeneralizer - why should we try?

- There is simple conversion from state to transition based acceptance with NO SPACE INCREASE.
- Not the other way:



- DIVINE uses TBA -> transition acceptance is fully enough.

Smarter Degeneralizer - what all could that bring to us?

- Smaller product with our TGBA
- Smaller product with SPOTs TGBA - possible even smaller than their state based BA

Lets start with [1] and [2]!

Souheib Baarir and Alexandre Duret-Lutz.
Mechanizing the minimization of deterministic generalized büchi automata.
In Erika Ábrahám and Catuscia Palamidessi, editors, *Formal Techniques for Distributed Objects, Components, and Systems*, pages 266–283, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

Tomáš Babiak, Thomas Badie, Alexandre Duret-Lutz, Mojmír Křetínský, and Jan Strejček.
Compositional approach to suspension and other improvements to ltl translation.
In Ezio Bartocci and C. R. Ramakrishnan, editors, *Model Checking Software*, pages 81–98, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

Vincent Bloemen, Alexandre Duret-Lutz, and Jaco van de Pol.
Explicit state model checking with generalized büchi and rabin automata.
In *Proceedings of the 24th ACM SIGSOFT International SPIN Symposium on Model Checking of Software*, SPIN 2017, pages 50–59, New York, NY, USA, 2017. ACM.

Alexandre Duret-Lutz, Alexandre Lewkowicz, Amaury Fauchille, Thibaud Michaud, Étienne Renault, and Laurent Xu.
Spot 2.0 — a framework for ltl and ω-automata manipulation.
In Cyrille Artho, Axel Legay, and Doron Peled, editors, *Automated Technology for Verification and Analysis*, pages 122–129, Cham, 2016. Springer International Publishing.

Dimitra Giannakopoulou and Flavio Lerda.
From states to transitions: Improving translation of ltl formulae to büchi automata.
In Doron A. Peled and Moshe Y. Vardi, editors, *Formal Techniques for Networked and Distributed Sytems — FORTE 2002*, pages 308–326, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.