On a Decidable Fragment of $d\mathcal{L}$
or, The Next 700 (Un)decidable Fragments of $d\mathcal{L}$

David M Kahn    Siva Somayyajula

Carnegie Mellon University

December 11, 2018
Motivation

If you or a loved one has been frustrated trying to formally verify systems,
Motivation

If you or a loved one has been frustrated trying to formally verify systems, you may be entitled to righteous indignation.
Motivation

Why is formal verification so frustrating?

- complicated and tedious proofs
- lots of work for no product change
- people only care it looks like it works
Motivation

Why is formal verification so frustrating?

- complicated and tedious proofs
- lots of work for no user-facing change
- people only care it looks like it works

*Cyberphysical systems are life-critical!*
Results

- Found and implemented decidable fragments of $d\mathcal{L}$ to ease verifying cyberphysical systems
- Found undecidable/inter-decidable fragments of $d\mathcal{L}$ to ease future decidability research
## (Un)decidability Results

### Arithmetical Approaches

<table>
<thead>
<tr>
<th></th>
<th>Integer Arithmetic</th>
<th>$d\mathcal{L}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive $\exists$</td>
<td>MRDP’s Diophantine</td>
<td>Post Correspondence</td>
</tr>
<tr>
<td>positive $\forall$</td>
<td>polynomial ID testing</td>
<td>extended Platzer-Tan</td>
</tr>
<tr>
<td>bounded</td>
<td>finitary checking</td>
<td></td>
</tr>
<tr>
<td>single variable</td>
<td>trivial</td>
<td>Post Correspondence</td>
</tr>
<tr>
<td>purely $+$</td>
<td>Presburger</td>
<td>Post Correspondence</td>
</tr>
<tr>
<td>purely $\times$</td>
<td>Skolem</td>
<td>Post Correspondence</td>
</tr>
</tbody>
</table>
### Structural Approaches

<table>
<thead>
<tr>
<th>Without $\cup$</th>
<th>$d\mathcal{L}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without ;</td>
<td>MRDP’s Diophantine</td>
</tr>
<tr>
<td>Without $*$</td>
<td>(exponential) polynomial star-free</td>
</tr>
<tr>
<td>Only :=</td>
<td>Post Correspondence</td>
</tr>
<tr>
<td>Only $?(\neg)$</td>
<td>reduction to $\text{FOL}_R$</td>
</tr>
<tr>
<td>Only $x' = f(x) &amp; Q$</td>
<td>piecewise constant derivative reachability</td>
</tr>
<tr>
<td>Simultaneously $[\alpha]P \land \langle\alpha\rangle P$</td>
<td>when $[\alpha]P$ is</td>
</tr>
</tbody>
</table>
Polynomial Star-Free Fragment

How can this be used for theorem proving?

- Work with simple ODEs
- Human identifies loop invariant
- That’s it! Everything else is free.
Polynomial Star-Free Fragment

- Idea: sound translation to FOL$_R$

\[ x := e \rightarrow P(x) \leftrightarrow P(e) \]

\[ \alpha; \beta \rightarrow P \leftrightarrow \alpha[\beta]P \]

\[ x' = f(x) \rightarrow P(x) \leftrightarrow \forall t \geq 0 P(x(t)) \text{ where } x'(t) = f(x(t)) \]

Remove iteration (star/asterate)

\[ \alpha^* = ? \]

Loop invariants?

Encode integer arithmetic: undecidable

Restrict to polynomial solutions of ODEs
Polynomial Star-Free Fragment

- Idea: sound translation to $\text{FOL}^\mathbb{R}$
  - $[x := e]P(x) \iff P(e)$
  - $[\alpha; \beta]P \iff [\alpha][\beta]P$

Remove iteration (star/asterate)

$\alpha^* = \text{true} \cup \alpha^*$

Loop invariants?

Encode integer arithmetic: undecidable

Restrict to polynomial solutions of ODEs
Polynomial Star-Free Fragment

- Idea: sound translation to $\text{FOL}_R$
  - $[x := e]P(x) \iff P(e)$
  - $[\alpha; \beta]P \iff [\alpha][\beta]P$
  - $[x' = f(x)]P(x) \iff \forall t \geq 0 \ P(x(t)) \text{ where } x'(t) = f(x(t))$
Polynomial Star-Free Fragment

- Idea: sound translation to $\text{FOL}_\mathbb{R}$
  - $[x := e]P(x) \leftrightarrow P(e)$
  - $[\alpha; \beta]P \leftrightarrow [\alpha][\beta]P$
  - $[x' = f(x)]P(x) \leftrightarrow \forall t \geq 0 P(x(t))$ where $x'(t) = f(x(t))$
- Remove iteration (star/asterate)
Polynomial Star-Free Fragment

- Idea: sound translation to $\text{FOL}_R$
  - $[x := e]P(x) \Leftrightarrow P(e)$
  - $[\alpha; \beta]P \Leftrightarrow [\alpha][\beta]P$
  - $[x' = f(x)]P(x) \Leftrightarrow \forall t \geq 0 P(x(t))$ where $x'(t) = f(x(t))$
- Remove iteration (star/asterate)
  - $\alpha^* = \text{true} \cup \alpha; \alpha^*$
Polynomial Star-Free Fragment

- Idea: sound translation to $\text{FOL}_{\mathbb{R}}$
  - $[x := e]P(x) \leftrightarrow P(e)$
  - $[\alpha; \beta]P \leftrightarrow [\alpha][\beta]P$
  - $[x' = f(x)]P(x) \leftrightarrow \forall t \geq 0 \ P(x(t))$ where $x'(t) = f(x(t))$

- Remove iteration (star/asterate)
  - $\alpha^* = ?true \cup \alpha; \alpha^*$
  - Loop invariants?
Polynomial Star-Free Fragment

- **Idea:** sound translation to $\text{FOL}_\mathbb{R}$
  - $[x := e]P(x) \leftrightarrow P(e)$
  - $[\alpha; \beta]P \leftrightarrow [\alpha][\beta]P$
  - $[x' = f(x)]P(x) \leftrightarrow \forall t \geq 0 P(x(t))$ where $x'(t) = f(x(t))$

- **Remove iteration (star/asterate)**
  - $\alpha^* = ?\text{true} \cup \alpha; \alpha^*$
  - Loop invariants?
  - Encode integer arithmetic: undecidable
Polynomial Star-Free Fragment

- Idea: sound translation to $\text{FOL}^\mathbb{R}$
  - $[x := e]P(x) \iff P(e)$
  - $[\alpha; \beta]P \iff [\alpha][\beta]P$
  - $[x' = f(x)]P(x) \iff \forall t \geq 0 \ P(x(t))$ where $x'(t) = f(x(t))$

- Remove iteration (star/asterate)
  - $\alpha^* = ?\text{true} \cup \alpha; \alpha^*$
  - Loop invariants?
    - Encode integer arithmetic: undecidable

- Restrict to polynomial solutions of ODEs
Theorem (DAG condition)

Given $S \equiv x'_1 = e_1, \ldots, x'_n = e_n$, let $G$ be a digraph s.t.

edge from $x'_i = e_i$ to $x'_j = e_j \iff x_i$ occurs in $e_j$

Then, $S$ has a polynomial solution $\iff G$ is acyclic.
Theorem (DAG condition)

Given $S \equiv x'_1 = e_1, \ldots, x'_n = e_n$, let $G$ be a digraph s.t.

$$\text{edge from } x'_i = e_i \text{ to } x'_j = e_j \iff x_i \text{ occurs in } e_j$$

Then, $S$ has a polynomial solution $\iff G$ is acyclic.

Proof sketch.

Back-sub in the topological order of $G$. 

David M Kahn, Siva Somayyajula (CMU)  On a Decidable Fragment of $dL$  December 11, 2018  11 / 15
Polynomial Star-Free: Implementation

- ~ 500 lines in OCaml
Polynomial Star-Free: Implementation

- ~ 500 lines in OCaml
- Shallow embedding of dŁ using weak higher-order abstract syntax
Polynomial Star-Free: Implementation

- ~ 500 lines in OCaml
- Shallow embedding of $d\mathcal{L}$ using weak higher-order abstract syntax
- Polynomial manipulation and ODE solver
Polynomial Star-Free: Implementation

- ~ 500 lines in OCaml
- Shallow embedding of $d\mathcal{L}$ using weak higher-order abstract syntax
- Polynomial manipulation and ODE solver
- Z3 for quantifier elimination
Polynomial Star-Free: Demo

Verifying $x \geq 0 \land v \geq 0 \land a \geq 0 \rightarrow [x' = v, v' = a] x \geq 0$

```plaintext
utop # check
((x >= !0. && v_ >= !0. && a >= !0.) =>
 (!! (["x" ^= v_; "v" ^= a] & tt) !! (x >= !0.)))));
```

```plaintext
Common.Valid
"unsat
((declare-fun _x0!0 () Real)
(proof
(let ((?x254 (* a _x0!0 _x0!0)))
(let ((?x257 (* 251 ?x257)))
(let ((?x287 (>= ?x260 0.0)))
```

David M Kahn, Siva Somayyajula (CMU)
Conclusion and Future Work

- Survey of restrictions for (un)decidability
Conclusion and Future Work

- Survey of restrictions for (un)decidability
- Decision procedures for theorem proving
DECIDABILITY

It’s Free VERIFICATION

imgflip.com