Proofs and model-checking

Gilles Dowek

Joint work with Ying Jiang

I. What is a proof?

A first definition

A derivation for an inductively defined subset of formulae E.g.

$$\frac{}{odd(S(0))}\ \mathrm{a}$$

$$\frac{odd(t)}{odd(S(S(t)))}\ \mathrm{b}$$

$$\cfrac{\cfrac{\cfrac{}{odd(S(0))}\ \mathrm{a}}{odd(S(S(S(0))))}\ \mathrm{b}}{odd(S(S(S(S(S(0))))))}\ \mathrm{b}$$

(Un)decidability

Set of formulae that have a proof: need not be decidable (no decidability of provability): terminating configurations of a Turing machine

Set of pairs $\langle \pi, A \rangle$: decidable (decidability of proof-checking)

Set of formulae that have a proof: is semi-decidable (projection of a decidable set)

A more abstract definition of the notion of proof

$\mathcal{F}$: a set of syntactic objects (natural numbers, formulae, strings, trees, ...)

E: A semi-decidable subset of $\mathcal{F}$

Always the projection of a decidable subset E' of $\mathcal{P} \times \mathcal{F}$

Proofs: elements of $\mathcal{P}$

E.g.: $\mathcal{F}$: configutations of a Turing machine

E: terminating configurations

$\mathcal{P}$: finite traces

Proofs in the decidable case

When E semi-decidable but undecidable

Profit: transform undecidable membership to E into decidable membership to E'

When E is decidable no profit from the point of view of decidability

But may be a profit from the point of view of complexity (communication and memory): from $N\mathcal{C}$ to $\mathcal{C}$

An example: composite numbers

Set of composite (= non prime) numbers decidable

Is 221 a composite number?

Yes perfect answer

Yes, it is 13×17 better

Proof that n composite $\langle p, q \rangle$ such that $p \times q = n$

Communication / memory

One decision algorithm for E: search for a proof

Indeed: checking n is composite is finding $\langle p, q \rangle$

Model checking

Truth (e.g. of a CTL formula) = validity in a finite model: decidable

Finite model replaced by set of configurations of a pushdown system: still decidable

Only advanced model-checking problems (e.g. trace systems) become undecidable

What proofs can be good for in model checking?

- communication / memory

Simpler to check the proof than the truth of the formula

A difference between universal and existential quantifier

$$\frac{P(s_1) \ ... \ P(s_n)}{\forall x \ P(x)}$$

as complex to check the proof than to check the truth of the formula

But ...

$$\frac{P(s_i)}{\exists x\ P(x)}$$

much simpler to check the proof than the truth of the formula (the proof records the element of the model that needs to be rediscovered)

No benefit for AF, but a lot of benefit for EF because the proof records the path that need not be rediscovered

- Use automated theorem proving methods (decision algorithms as automated theorem proving / improve automated theorem proving (Kailiang Ji's talk))
- Communicate with other tools (formal methods)
- Undecidable cases

A very broad project

Only at the beginning (in progress)

Finite state case

Pushdown systems (in progress)

II. Finite models

Inductive modalities

Syntax: $s \Vdash AF(P)$ written $AF_x(P(x))(s)$

$$\frac{(s/x)\phi}{AF_x(\phi)(s)}$$

$$\frac{AF_x(\phi)(s_1) \;\ldots\; AF_x(\phi)(s_n)}{AF_x(\phi)(s)} \; s_1, \ldots, s_n = N(s)$$

$$\frac{}{P(s_1, \ldots, s_n)} \langle s_1, \ldots, s_n \rangle \in P$$

$$\frac{}{\neg P(s_1, \ldots, s_n)} \langle s_1, \ldots, s_n \rangle \notin P$$

$$\frac{\phi_1 \qquad \phi_2}{\phi_1 \wedge \phi_2}$$

$$\frac{\phi_i}{\phi_1 \vee \phi_2}$$

Negation normal form

Recording a path

$$\frac{(s/x)\phi}{EF_x(\phi)(s)}$$

$$\frac{EF_x(\phi)(s')}{EF_x(\phi)(s)}\; s' \in N(s)$$

Decidability (no cut, no contraction, finite search space)

Reasonning and hypothetical reasonning

In general to prove $A \Rightarrow B$, assume A and prove B

Deduction rules do not operate on formulae but on formulae equipped with hypotheses: sequents

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

$$\frac{}{\Gamma \vdash A}\, A \in \Gamma$$

E.g.

$$\dfrac{\overline{P \vdash P}}{\vdash P \Rightarrow P}$$

Here no hypotheses: to prove $A \Rightarrow B$ ($= \neg A \vee B$) two rules

$$\frac{\neg A}{\neg A \vee B} \qquad \frac{B}{\neg A \vee B}$$

To prove $P \Rightarrow P$ prove either P or $\neg P$ (possible thanks to completeness)

Co-inductive modalities

$$\frac{(s/x)\phi \qquad EG_x(\phi)(s')}{EG_x(\phi)(s)}\; s' \in N(s)$$

But ... infinite path: infinite proof

Decidability argument

If there is an infinite path

There is one with twice the same state

Thus there is a regular one

When reach a node that is already in the path, stop with success

Record the path in the hypotheses

Pruning

$$\frac{\vdash (s/x)\phi \qquad \Gamma, EG_x(\phi)(s) \vdash EG_x(\phi)(s')}{\Gamma \vdash EG_x(\phi)(s)}\; s' \in Next(s)$$

$$\frac{}{\Gamma \vdash EG_x(\phi)(s)}\; EG_x(\phi)(s) \in \Gamma$$

III. Pushdown systems

Pruning

Same rules as for the finite case

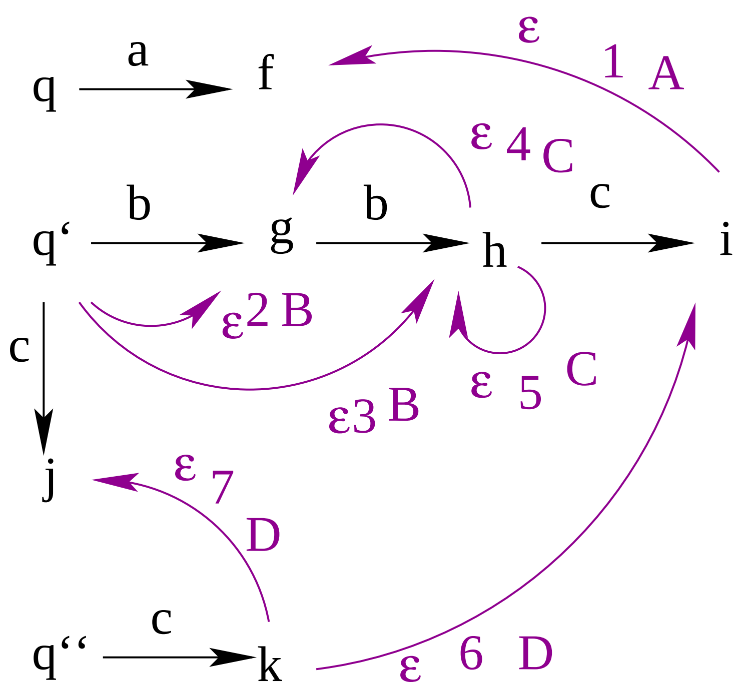
But pruning is more complicated

$$\langle q, a \rangle \longrightarrow \langle q, aa \rangle$$

$$\langle q, az \rangle \longrightarrow \langle q, aaz \rangle \longrightarrow \langle q, aaaz \rangle \longrightarrow \ldots$$

Decidability

(Bouajanni et al.) The accessible configurations are recognized by a (multi)automaton



Two steps: black transitions, then saturation with the purple ones

Step by step saturation

A sequence of transitions in the pushdown system

$$\langle q_0, w_0 \rangle \longrightarrow \langle q_1, w_1 \rangle \longrightarrow \langle q_2, w_2 \rangle \longrightarrow \ldots \longrightarrow \langle q_n, w_n \rangle$$

$$\mathcal{A}_0 \longrightarrow \mathcal{A}_1 \longrightarrow \mathcal{A}_2 \longrightarrow \ldots \longrightarrow \mathcal{A}_n$$

If twice the same automaton: the accessible configuration $\langle q_n, w_n \rangle$ is recognized by the (multi)-automaton with a path that uses twice the same transition

q_n has a “simpler” derivation

Pruning

Sequents of the form $\mathcal{A} \vdash B$

Prune when a transition does not add a new transition to the automaton

Completeness (in progress)

A new kind of sequents

Sequents have to be finitely presented objects (decidability of proof-checking)

But they can contain an infinite number of premises given by a (finite) automaton

Conclusion

Proof-checking and model-checking are not that different

Specific proof-systems for LTL, CTL, ... or not

Benefit in both sides (new automated theorem proving methods, new notion of sequent, ...)