Cut-elimination and the decidability of reachability in alternating pushdown systems

Gilles Dowek and Ying Jiang

Locali
Decidability

To prove the decidability of a problem

- **reduce** it to provability in some decidable logic
- **reduce** it to reachability in some transition system (where reachability is decidable)
The set of even numbers is decidable

\[
\begin{align*}
\text{even}(0) \\
\text{even}(x) & \quad \Rightarrow \quad \text{odd}(S(x)) \\
\text{odd}(x) & \quad \Rightarrow \quad \text{even}(S(x))
\end{align*}
\]

\(n\) is even iff \(\text{even}(n)\) is provable

Provability in this logic is decidable
The set of even numbers is decidable

\[ \text{even}, 0 \rightarrow f \]
\[ \text{even}, Sw \rightarrow \text{odd}, w \]
\[ \text{odd}, Sw \rightarrow \text{even}, w \]

\( n \) is even iff even, \( n \in \text{Pre}^*(f) \)

Reachability in this transition system is decidable
At a first glance look alike: objects (propositions / states) and (deduction / transition) rules that permit to go step by step from one object to another

But, after a second look, details quite different: cut-elimination, ...

/ finite state automaton, ...
This talk

Deeper connections between logics and transition systems

A “new” method to prove cut-elimination: saturation

Application: “new” proof of decidability of reachability in alternating pushdown systems (LTL, CTL, $\mu$-calculus over pushdown systems)
I. A path to decidability: Cut-elimination
Cuts, cut-elimination and the sub-formula property

\[
\frac{\pi \quad \pi'}{A \quad B} \quad \text{intro}
\]

\[
\frac{A \land B}{A} \quad \text{elim}
\]

In a cut-free proof: all propositions are sub-formulae of the conclusion.

The notion of sub-formula varies with logics (\(\text{sub}(A)\) may the full set of propositions in some logics (e.g. SOL, HOL, ...))
Decidability

If

cut-elimination + $\text{sub}(A)$ finite

then

decidability

Search space finite

E.g. Kleene’s proof of decidability of propositional logic (classical and constructive)
II. Alternating pushdown systems
State, word, configuration

A language $\mathcal{L}$ in predicate logic

a finite number of monadic predicate symbols states

a finite number of monadic function symbols stack symbols

a constant $\varepsilon$ the empty word
State, word, configuration

A language $\mathcal{L}$ in predicate logic

a finite number of monadic predicate symbols states

a finite number of monadic function symbols stack symbols

a constant $\varepsilon$ the empty word

Closed term: $\gamma_1(\gamma_2\ldots(\gamma_n(\varepsilon)))$ word ($w = \gamma_1\gamma_2\ldots\gamma_n$)

Open term: $\gamma_1(\gamma_2\ldots(\gamma_n(x)))$ ($wx$ for $w = \gamma_1\gamma_2\ldots\gamma_n$)

Closed atomic proposition: $P(w)$ configuration

Open atomic proposition: $P(wx)$
Alternating pushdown system

A finite set of deduction rules, transition rules

\[
\frac{P_1(v_1 x) \ldots P_n(v_n x)}{Q(wx)}
\]

\[
\frac{}{Q(\varepsilon)}
\]

\[
\langle Q, w \rangle \longrightarrow \{\langle P_1, v_1 \rangle, \ldots, \langle P_n, v_n \rangle\}
\]

\[
\langle Q, \varepsilon \rangle \longrightarrow \emptyset
\]
Proof as usual

A provable from $B_1, \ldots, B_p$ ($A \in Pre^*(\{B_1, \ldots, B_p\})$) if exists a tree (proof) s.t.

(1) root labeled by $A$

(2) leaves labeled by $B_1, \ldots, B_p$

(3) for each internal node $N$, a transition rule

\[
P_1(v_1x) \ldots P_n(v_nx) \quad \begin{array}{c} \overline{Q(wx)} \end{array}\]

and a word $u$ s.t. $N$ labeled with $Q(wu)$ and its children labeled with $P_1(v_1u), \ldots, P_n(v_nu)$
An example

\[
\frac{Q(x)}{P(ax)} \quad i_1 \quad \frac{T(x)}{P(bx)} \quad i_2 \quad \frac{T(x)}{R(ax)} \quad i_3 \quad \frac{R(bx)}{} \quad i_4
\]

\[
\frac{P(x)}{Q(x)} \quad \frac{R(x)}{Q(x)} \quad n_1 \quad \frac{T(x)}{} \quad n_2
\]

\[
\frac{P(ax)}{S(x)} \quad e_1
\]
\[
\begin{align*}
\frac{T(\varepsilon)}{P(b)} \quad & \quad \frac{Q(b)}{P(ab)} \quad \frac{T(b)}{R(ab)} \\
\quad & \quad \frac{R(b)}{P(ab)} \quad \quad \quad \frac{Q(ab)}{P(aab)} \quad \quad \quad \frac{R(b)}{S(ab)}
\end{align*}
\]
When at most one premise in each rule

Lists rather than trees

\[
\begin{array}{c}
even(0) \\
\hline
odd(S(0)) \\
\hline
even(S(S(0))) \\
\hline
odd(S(S(S(0)))) \\
\hline
even(S(S(S(S(0)))))
\end{array}
\]

\[
even(S(S(S(S(0))))) \rightarrow odd(S(S(S(0)))) \rightarrow \]
\[even(S(S(0))) \rightarrow odd(S(0)) \rightarrow even(0) \rightarrow f
\]
In general: transitions on sets: \textbf{Alternating transition systems}

\[
\begin{align*}
\overline{T} & \quad \overline{T} \quad \overline{T} \\
& \quad T \lor (T \lor T) \\
\{T \lor (T \lor T)\} & \rightarrow \{T, T \lor T\} \rightarrow \{T \lor T\} \rightarrow \\
\{T, T\} & \rightarrow \{T\} \rightarrow \emptyset
\end{align*}
\]

Proofs as lists rather than trees
Alternating?

Inference systems

Unary inference systems

Alternating transition systems

Transition systems

Alternating: a huge step in the direction of proof-theory
III. Decidability of reachability
Small-step alternating pushdown systems

\[ \frac{P_1(x) \ldots P_n(x)}{Q(\gamma x)} \]  \text{ introduction rule}

\[ \frac{P_1(\gamma x) P_2(x) \ldots P_n(x)}{Q(x)} \]  \text{ elimination rule}

\[ \frac{P_1(x) \ldots P_n(x)}{Q(x)} \]  \text{ neutral rule}

Any alternating pushdown system can be transformed into a small-step one
Alternating multi-automaton

Introduction rules only

\[
\frac{P_1(x) \ldots P_n(x)}{Q(\gamma x)}
\]

also written

\[
\langle Q, \gamma \rangle \rightarrow \{\langle P_1, \varepsilon \rangle, \ldots, \langle P_n, \varepsilon \rangle\}
\]

\[
Q \rightarrow^\gamma \{P_1, \ldots, P_n\}
\]

\(P(w)\) is provable: \(w\) recognized in \(P\)

Particular case of alternating pushdown systems
Provability in an alternating multi-automaton

Obviously decidable: bottom-up proof search terminates

In contrast, with arbitrary rules

\[
\frac{P(ax)}{P(x)}
\]

bottom-up proof-search yields \(P(a), P(aa), P(aaa), P(aaaa), \ldots\)

Transforming (small step) alternating pushdown systems into alternating multi-automata: Cut-elimination
Another form intro + neutral
Not every small-step alternating pushdown system cut-elimination property

Every small step alternating pushdown systems: an extension with derivable rules, that has cut-elimination property

Similar to Knuth-Bendix method
Saturation

\[
\frac{P_1(x) \ldots P_m(x)}{Q_1(\gamma x)} \text{ intro} \quad \frac{Q_1(\gamma x) Q_2(x) \ldots Q_n(x)}{R(x)} \text{ elim}
\]

add

\[
\frac{P_1(x) \ldots P_m(x) Q_2(x) \ldots Q_n(x)}{R(x)} \text{ neutral}
\]

+ another form intro + neutral $\rightarrow$ intro

Termination: only a finite number of possible rules
Cut-elimination

\[
\begin{align*}
\frac{\pi_1}{P_1(w)} \quad \frac{\pi_m}{P_m(w)} \\
\frac{\pi_1}{Q_1(\gamma w)} \quad \frac{\pi_m}{Q_m(w)} \\
\frac{\rho_2}{Q_2(w)} \quad \frac{\rho_n}{Q_n(w)} \\
\frac{\rho_2}{R(w)} \quad \frac{\rho_n}{R(w)}
\end{align*}
\]

replaced by

\[
\begin{align*}
\frac{\pi_1}{P_1(w)} \quad \frac{\pi_m}{P_m(w)} \\
\frac{\pi_1}{Q_1(\gamma w)} \quad \frac{\pi_m}{Q_m(w)} \\
\frac{\rho_2}{Q_2(w)} \quad \frac{\rho_n}{Q_n(w)} \\
\frac{\rho_2}{R(w)} \quad \frac{\rho_n}{R(w)}
\end{align*}
\]

Termination: a proof \(\rightarrow\) a cut-free proof
Example

\[
\begin{align*}
\frac{T(\varepsilon)}{P(b)} & \quad \frac{Q(b)}{P(ab)} \quad \frac{T(b)}{R(ab)} \\
\frac{Q(ab)}{P(aab)} & \quad \frac{R(b)}{P(ab)} \\
\frac{Q(ab)}{P(aab)} & \quad \frac{Q(b)}{P(ab)} \quad \frac{T(b)}{R(ab)}
\end{align*}
\]
Transformed into

\[
\frac{T(\varepsilon)}{Q(b)} \frac{T(b)}{S(ab)}
\]
A cut-free proof contains introduction rules only

The proof has the form

\[
\begin{array}{c}
\pi_1 & \pi_n \\
\hline
A_1 & \ldots & A_n \\
\hline
B
\end{array}
\]

Induction hypothesis: \( \pi_1, \ldots, \pi_n \) contain introduction rules only

Cut-free: last rule neither elimination, nor neutral: introduction

Iterated: last rule property (no hypotheses)

All the other rules can be dropped (alternating multi-automaton)
Decidability

Alternating pushdown system

→ small-step alternating pushdown system

→ saturated small-step alternating pushdown system

→ alternating multi-automaton