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

Gilles Dowek

Joint work with Ying Jiang
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

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

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

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

$\langle \text{even}, 0 \rangle \xrightarrow{f} \langle \text{even}, \text{Sw} \rangle \xrightarrow{\langle \text{odd}, w \rangle} \langle \text{odd}, \text{Sw} \rangle \xrightarrow{\langle \text{even}, w \rangle}$

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

Reachability in this transition system is decidable
Logics and transition systems

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, μ-calculus over pushdown systems)
I. A path to decidability: Cut-elimination
Why are proofs more complex than theorems?

\[ \frac{A \quad B}{A \land B} \]
Why are proofs more complex than theorems?

\[
\begin{array}{c}
A \\
A \land B \\
A \land B \\
A
\end{array}
\]
Why are proofs more complex than theorems?

\[
\begin{align*}
\frac{A \quad B}{A \land B} & \quad \text{intro} \\
\frac{A \land B}{A} & \quad \text{elim}
\end{align*}
\]
A cut: a proof of the form

\[
\begin{array}{c}
\pi \\
\frac{A}{A} \\
\frac{A \land B}{A} \quad \text{intro}
\end{array}
\]

\[
\begin{array}{c}
\quad \\
\frac{A}{A} \quad \text{elim}
\end{array}
\]
Cut-elimination

Replace this proof by

\[
\pi \\
A
\]

Eliminating a cut may create new cuts

Termination depends on the system

When termination: each provable proposition has a cut-free proof (cut-elimination property)
Cut-elimination and the sub-formula property

\[
\pi \\
\hline
A \\
\hline
A \land B \\
\hline
A
\]

intro
elim

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

The notion of sub-formula varies with logics
Decidability

If

\begin{align*}
& \text{cut-elimination } + \ \text{sub}(A) \ \text{finite} \\
\end{align*}

then

\begin{align*}
& \text{decidability} \\
\end{align*}

Search space finite

E.g. Kleene's proof of decidability of propositional logic (classical and constructive)
II. Pushdown systems and alternating pushdown systems
Finite transition systems

$P$ reachable from $R$

$R \in \text{Pre}^*(P)$

$\text{Pre}^*(P) = \{P, Q, R\}$

Decidable
Pushdown systems: configurations

A pair \( \langle P, v \rangle \)

\( P \): a state taken in a finite set

\( v \): a word (stack): finite sequence in a finite alphabet
Pushdown systems: rules

\[ \langle P, \gamma x \rangle \rightarrow \langle Q, vx \rangle \]

Reachability still decidable (Bouajjani, Esparza, Maler)

\( Pre^* (\langle R, w \rangle) \) infinite but regular (recognized by a finite automaton)
Alternating pushdown systems

\[ \langle P, \gamma x \rangle \longrightarrow \{ \langle Q_1, v_1 x \rangle, \ldots, \langle Q_n, v_n x \rangle \} \]
How is reachability defined?

In finite transition systems, pushdown systems, etc.

\[ c \in Pre^*(c) \]

if \( c \rightarrow c' \) and \( c' \in Pre^*(c'') \) then \( c \in Pre^*(c'') \)

In **alternating** finite transition systems, pushdown systems, etc.

\[ c \in Pre^*(c) \]

if \( c \rightarrow \{ c_1, \ldots, c_n \} \) and \( c_1 \in Pre^*(c''), \ldots, c_n \in Pre^*(c'') \), then \( c \in Pre^*(c'') \)
Decidability

Reachability still decidable (Bouajjani, Esparza, Maler)

$Pre^*(\emptyset)$ infinite but recognized by a finite alternating automaton
III. A logical approach to 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 \textit{states}

a finite number of monadic function symbols \textit{stack symbols}

a constant $\varepsilon$ \textit{the empty word}

Closed term: $\gamma_1(\gamma_2 \cdots (\gamma_n(\varepsilon)))$ \textit{word} ($w = \gamma_1 \gamma_2 \cdots \gamma_n$)

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

Closed atomic proposition: $P(w)$ \textit{configuration}

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

A finite set of deduction rules, transition rules

\[
P_1(v_1x) \ldots P_n(v_nx) \Rightarrow Q(wx)
\]

\[
Q(\varepsilon) \Rightarrow \emptyset
\]

\[
\langle Q, wx \rangle \rightarrow \{ \langle P_1, v_1x \rangle, \ldots, \langle P_n, v_nx \rangle \}
\]

\[
\langle Q, \varepsilon \rangle \rightarrow \emptyset
\]

Beware: \( w \) arbitrary (may be \( \varepsilon \))
Proof

A provable \((A \in Pre^*(\emptyset))\)
if exists a tree (proof) s.t.
(1) root labeled by \(A\)
(2) for each node \(N\), a transition rule
\[
\begin{array}{c}
P_1(v_1x) \ldots P_n(v_nx) \\
\hline \\
Q(wx)
\end{array}
\]
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

\[
\begin{align*}
\frac{Q(x)}{P(ax)} & \quad i_1 \\
\frac{T(x)}{P(bx)} & \quad i_2 \\
\frac{T(x)}{R(ax)} & \quad i_3 \\
\overline{R(bx)} & \quad i_4
\end{align*}
\]

\[
\begin{align*}
\frac{P(x)R(x)}{Q(x)} & \quad n_1 \\
\overline{T(x)} & \quad n_2
\end{align*}
\]

\[
\begin{align*}
\frac{P(ax)}{S'(x)} & \quad e_1
\end{align*}
\]
When at most one premise in each rule

Lists rather than trees

\[
\begin{align*}
\text{even}(0) \\
\sqrt{odd(S(0))} \\
\sqrt{even(S(S(0)))} \\
\sqrt{odd(S(S(S(0))))} \\
\sqrt{even(S(S(S(S(0)))))}
\end{align*}
\]

\[
\begin{align*}
even(S(S(S(S(0)))) & \rightarrow odd(S(S(S(0)))) \\
even(S(S(0))) & \rightarrow odd(S(0)) \\
even(0) & \rightarrow f
\end{align*}
\]
In general: transitions on sets: Alternating transition systems

\[
\begin{array}{c}
\overline{T} \\
\overline{T} \\
\overline{T}
\end{array}
\]

\[
\frac{T \land T \land T}{T \land (T \land T)}
\]

\[
\{T \land (T \land T)\} \rightarrow \{T, T \land T\} \rightarrow \{T \land T\} \rightarrow \\
\{T, T\} \rightarrow \{T\} \rightarrow \emptyset
\]
Alternating?

Inference systems

Unary inference systems

Alternating transition systems

Transition systems

Alternating: a huge step in the direction of proof-theory
Where does the word “alternating” come from?

To prove $Q$ we have to prove either $(P_1$ and $P_2)$ or $(P_3$ and $P_4)$
Introduction rule, elimination rule, neutral rule

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

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

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

\(\emptyset\) empty in elimination and neutral rules
Any alternating pushdown system can be transformed into a small-step one, e.g.

\[
\begin{align*}
P(aax) & \quad Q(x) \\
P(ax) & \quad P'(x) \\
P'(ax) & \quad P'(x) \\
& \quad Q(x)
\end{align*}
\]
Alternating multi-automaton

Introduction rules only

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

also written

\[
\langle Q, \gamma x \rangle \longrightarrow \{\langle P_1, x \rangle, \ldots, \langle P_n, x \rangle\}
\]

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

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

Particular case of alternating pushdown systems
IV. Decidability of reachability
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
May give a name to each rule and write the proof with these names only (labeled transition systems, Curry-de Bruijn-Howard)
Rule names

In

\[
\frac{R_1(\varepsilon)}{Q_1(b)} \quad \frac{R_2(\varepsilon)}{Q_2(b)} \quad \varepsilon \quad \varepsilon \\
\frac{b}{b} \quad \frac{b}{b} \quad a \\
P(ab)
\]

\(a(b(\varepsilon), b(\varepsilon))\)

All branches are labeled with the same list: \(ab\)

\(ab\) both the proof and the argument of \(P\)
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

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

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

add

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

+ another form intro + neutral \(\rightarrow\) intro

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

\[
\begin{array}{c}
\frac{\pi_1}{P_1(w)} \ldots \frac{\pi_m}{P_m(w)} \quad \frac{\pi_m}{Q_1(\gamma w)} \quad \frac{\rho_2}{Q_2(w)} \ldots \frac{\rho_n}{Q_n(w)} \\
\text{intro} \quad \text{elim} \\
\frac{}{R(w)} \\
\end{array}
\]

replaced by

\[
\begin{array}{c}
\frac{\pi_1}{P_1(w)} \ldots \frac{\pi_m}{P_m(w)} \quad \frac{\rho_2}{Q_2(w)} \ldots \frac{\rho_n}{Q_n(w)} \\
\text{neutral} \\
\frac{}{R(w)} \\
\end{array}
\]

Termination: a proof $\rightarrow$ a cut-free proof
Example
Transformed into

\[
\frac{T(\varepsilon)}{Q(b)} \frac{i_{10}}{i_{10}} \frac{T(b)}{i_{10}} \frac{i_{9}}{i_{8}} \frac{S(ab)}{i_{8}}
\]
A cut-free proof contains introduction rules only

The proof has the form

\[
\frac{\pi_1 \quad \pi_n}{A_1 \quad \ldots \quad A_n} \quad B
\]

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

$\rightarrow$ small-step alternating pushdown system

$\rightarrow$ saturated small-step alternating pushdown system

$\rightarrow$ alternating multi-automaton