Language Theory and Infinite Graphs

Colin Stirling
School of Informatics
University of Edinburgh

Main aim of course

- Prove one result:

  language equivalence is decidable for deterministic context-free languages

- Was a notorious open problem in language theory

- Its solution paves the way for more difficult open problems from the 1960s: last lecture

- New applications: algorithmics of game semantics (equivalence of open programs in idealised algol, Ong et als)

Pushdown automata

A pushdown automaton, PDA, consists of

- A finite set of states P
- A finite set of stack symbols S
- A finite alphabet A
- A finite set of basic transitions T

Basic transition $pX \xrightarrow{a} q\alpha$ where p, $q \in$ P, $X \in$ S, $a \in$ A $\cup \{\epsilon\}$ and $\alpha \in$ S*

Configurations

- A configuration $p\beta$, $p \in \mathsf{P}$ and $\beta \in \mathsf{S}^*$

- Transitions of a configuration

  If $pX \xrightarrow{a} q\alpha \in \mathsf{T}$ then $pX\delta \xrightarrow{a} q\alpha\delta$

Disjointness assumption

If $pX \xrightarrow{\epsilon} q\beta \in \mathsf{T}$ and $pX \xrightarrow{a} r\lambda \in \mathsf{T}$ then $a = \epsilon$

Configurations are unstable, only have ϵ-transitions, or stable, have no ϵ-transitions

Transition graphs $\mathsf{G}(p\beta)$

Generated by deriving all possible transitions from $p\beta$ and every configuration reachable from it

Example

$\mathsf{P} = \{p\}$, $\mathsf{S} = \{X\}$, $\mathsf{A} = \{a\}$, $\mathsf{T} = \{pX \xrightarrow{a} pXX\}$

$\mathsf{G}(pX)$ is:

$$pX \xrightarrow{a} pXX \xrightarrow{a} pXXX \xrightarrow{a} pXXXX \xrightarrow{a} \ldots$$

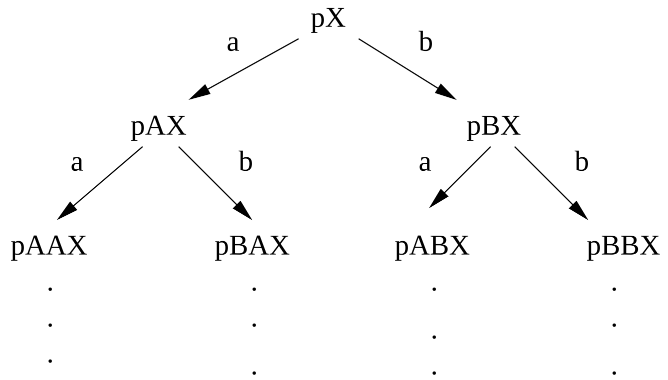
Transition $pXXX \xrightarrow{a} pXXXX$ follows from $pX \xrightarrow{a} pXX \in \mathsf{T}$

$$pX\,XX \xrightarrow{a} pXX\,XX$$

Example

P $= \{p\}$, S $= \{X, A, B\}$ and A $= \{a, b\}$, T is

$$pX \xrightarrow{a} pAX \qquad pA \xrightarrow{a} pAA \qquad pB \xrightarrow{a} pAB$$
$$pX \xrightarrow{b} pBX \qquad pA \xrightarrow{b} pBA \qquad pB \xrightarrow{b} pBB$$



Example

P $= \{p, q, r\}$, S $= \{X\}$, A $= \{a, b\}$ and T is

$$pX \xrightarrow{a} pXX \qquad pX \xrightarrow{b} r\epsilon \qquad rX \xrightarrow{\epsilon} r\epsilon$$
$$pX \xrightarrow{b} q\epsilon \qquad qX \xrightarrow{b} q\epsilon$$

G(pX) is

$$\begin{array}{ccccccc}
q\epsilon & \xleftarrow{b} & qX & \xleftarrow{b} & qXX & \xleftarrow{b} & \dots \\
\uparrow b & & \uparrow b & & \uparrow b & & \\
pX & \xrightarrow{a} & pXX & \xrightarrow{a} & pXXX & \xrightarrow{a} & \dots \\
\downarrow b & & \downarrow b & & \downarrow b & & \\
r\epsilon & \xleftarrow{\epsilon} & rX & \xleftarrow{\epsilon} & rXX & \xleftarrow{\epsilon} & \dots
\end{array}$$

Normal form (up to graph isomorphism)

- If $pX \xrightarrow{a} q\alpha \in \mathsf{T}$ then the length of α, $|\alpha|$, is at most 2.

Achieving normal form

- by introducing extra stack symbols
- assume maximum length of any α in $pX \xrightarrow{a} q\alpha \in \mathsf{T}$ is n
- new stack symbols $[\beta]$, $1 < |\beta| \leq n$ are introduced

Example

$$pX \xrightarrow{a} pX_1X_2X_3X_4 \qquad pX_1 \xrightarrow{a} p\epsilon \qquad pX_4 \xrightarrow{a} pX_1$$
$$pX_2 \xrightarrow{a} pX_1X_3X_4 \qquad pX_3 \xrightarrow{a} pX_1X_1X_1X_4$$

This set is transformed as follows when starting from pX

- $pX \xrightarrow{a} pX_1[X_2X_3X_4]$ and $p[X_2X_3X_4] \xrightarrow{a} p[X_1X_3X_4][X_3X_4]$
- $p[X_1X_3X_4] \xrightarrow{a} p[X_3X_4]$ and $p[X_3X_4] \xrightarrow{a} p[X_1X_1X_1X_4]X_4$
- $p[X_1X_1X_1X_4] \xrightarrow{a} p[X_1X_1X_4]$ and $p[X_1X_1X_4] \xrightarrow{a} p[X_1X_4]$
- $p[X_1X_4] \xrightarrow{a} pX_4$ and transitions for pX_1 and pX_4 are unchanged

Bounded in/out degree of G($p\alpha$)

- Out degree of terminal configuration $p\alpha$ is 0
- Out degree of $pX\alpha$ is bounded by the number of basic transitions in T of form $pX \xrightarrow{a} q\beta$
- In degree of $p\alpha$ is bounded by the number of basic transitions in T of form $qY \xrightarrow{a} p\alpha'$, where α' is a prefix of α

Word transitions

- extend transitions to words $w \in \mathsf{A}^*$
- $p\alpha \xrightarrow{w} q\beta$ if that there is a w-path from $p\alpha$ to $q\beta$ in $\mathsf{G}(p\alpha)$
- so, $p\alpha \xrightarrow{\epsilon} p\alpha$ for any $p\alpha$
- role of ϵ-transitions in T thereby differs from a-transitions in T when $a \in \mathsf{A}$, because ϵ is the empty word

Example

$$
\begin{array}{ccccccc}
q\epsilon & \xleftarrow{b} & qX & \xleftarrow{b} & qXX & \xleftarrow{b} & \ldots \\
\uparrow b & & \uparrow b & & \uparrow b & & \\
pX & \xrightarrow{a} & pXX & \xrightarrow{a} & pXXX & \xrightarrow{a} & \ldots \\
\downarrow b & & \downarrow b & & \downarrow b & & \\
r\epsilon & \xleftarrow{\epsilon} & rX & \xleftarrow{\epsilon} & rXX & \xleftarrow{\epsilon} & \ldots
\end{array}
$$

$pXX \xrightarrow{ab} r\epsilon$ because $pXX \xrightarrow{a} pXXX \xrightarrow{b} rXX \xrightarrow{\epsilon} rX \xrightarrow{\epsilon} r\epsilon$

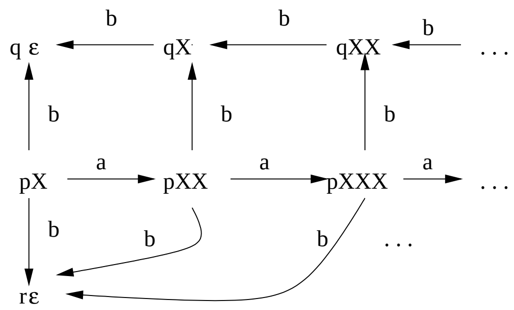
Collapsed transition graphs $\mathsf{G}^c(p\alpha)$

- consider word transitions $\xrightarrow{a}$, $a \in \mathsf{A}$, between stable configurations
- for $p\alpha$, $q\beta$ stable, $p\alpha \xrightarrow{a} q\beta$ if $p\alpha \xrightarrow{a} r\gamma(\xrightarrow{\epsilon})^* q\beta$
- this is the collapsed graph, without ϵ-transitions
- if $p\alpha$ is stable then $\mathsf{G}^c(p\alpha)$ is its collapsed graph
- $\mathsf{G}(p\alpha) = \mathsf{G}^{\mathsf{c}}(p\alpha)$ if the PDA does not have ϵ-transitions

Example

$$
\begin{array}{ccccccc}
q\epsilon & \stackrel{b}{\longleftarrow} & qX & \stackrel{b}{\longleftarrow} & qXX & \stackrel{b}{\longleftarrow} & \ldots \\
\uparrow b & & \uparrow b & & \uparrow b & & \\
pX & \stackrel{a}{\longrightarrow} & pXX & \stackrel{a}{\longrightarrow} & pXXX & \stackrel{a}{\longrightarrow} & \ldots \\
\downarrow b & & \downarrow b & & \downarrow b & & \\
r\epsilon & \stackrel{\epsilon}{\longleftarrow} & rX & \stackrel{\epsilon}{\longleftarrow} & rXX & \stackrel{\epsilon}{\longleftarrow} & \ldots
\end{array}
$$

So $\mathsf{G}^c(pX)$ is

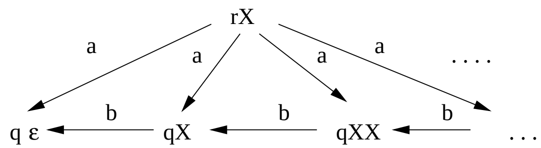


Infinite out degree

$rX \xrightarrow{a} pX \quad pX \xrightarrow{\epsilon} pXX \quad pX \xrightarrow{\epsilon} q\epsilon \quad qX \xrightarrow{b} q\epsilon$

$$\begin{array}{ccccccccc}
rX & \xrightarrow{a} & pX & \xrightarrow{\epsilon} & pXX & \xrightarrow{\epsilon} & pXXX & \xrightarrow{\epsilon} & \ldots \\
 & & \downarrow \epsilon & & \downarrow \epsilon & & \downarrow \epsilon & & \\
 & & q\epsilon & \xleftarrow{b} & qX & \xleftarrow{b} & qXX & \xleftarrow{b} & \ldots
\end{array}$$

$\mathsf{G}^c(rX)$ is



Infinite out degree only if there is **nondeterminism** in basic ϵ-transitions

Exercise

If $\mathsf{G}(pX)$ is

$$pX \xrightarrow{a} qX \xrightarrow{\epsilon} qXX \xrightarrow{\epsilon} qXXX \xrightarrow{\epsilon} \ldots$$

then what is $\mathsf{G}^{\mathsf{c}}(pX)$?

Subclasses of pushdown automata

- real-time: there are no ϵ-transitions
- single-state: P is a singleton set
- ϵ-deterministic: if $pX \xrightarrow{\epsilon} q\beta$, $pX \xrightarrow{\epsilon} r\gamma \in$ T then $q = r$ and $\beta = \gamma$
- A-deterministic: if $a \in$ A and $pX \xrightarrow{a} q\beta$, $pX \xrightarrow{a} r\gamma \in$ T, then $q = r$ and $\beta = \gamma$
- deterministic: if it is both ϵ-deterministic and A-deterministic
- normed (or without redundancy): for any configuration $q\beta$ of a graph G$(p\alpha)$ there is a word $u \in$ A* and a state r such that $q\beta \xrightarrow{u} r\epsilon$

Normal form for ϵ-determinisitic PDA

- if $pX \xrightarrow{\epsilon} q\beta \in$ T then $\beta = \epsilon$ ϵ-transitions only pop the stack

- Equivalence is isomorphism of collapsed graphs

To prove this, there are three cases to consider for each offending transition

$$pX \xrightarrow{\epsilon} p_1 X_1 \alpha_1$$

Proof of normal form

1. If $pX \xrightarrow{\epsilon} p_1X_1\alpha_1 \xrightarrow{\epsilon} p_2X_2\alpha_2 \xrightarrow{\epsilon} \ldots$ then there are no configurations $pX\beta$ in collapsed graph. Therefore, $pX \xrightarrow{\epsilon} p_1X_1\alpha_1$ and any transition $qY \xrightarrow{a} pX\alpha$, $a \in \mathsf{A} \cup \{\epsilon\}$, are removed from T.

2. If $pX \xrightarrow{\epsilon} p_1X_1\alpha_1 \xrightarrow{\epsilon} \ldots \xrightarrow{\epsilon} p_nX_n\alpha_n$ and $p_nX\alpha_n$ is stable then we remove $pX \xrightarrow{\epsilon} p_1X_1\alpha_1$ from T and for each transition $p_nX_n \xrightarrow{a} q\beta \in$ T add the transition $pX \xrightarrow{a} q\beta\alpha_n$ to T.

3. If $pX \xrightarrow{\epsilon} p_1X_1\alpha_1 \xrightarrow{\epsilon} \ldots \xrightarrow{\epsilon} p_n\epsilon$ then we remove $pX \xrightarrow{\epsilon} p_1X_1\alpha_1$ from T and add the new transition $pX \xrightarrow{\epsilon} p_n\epsilon$ to T.

Decision questions I

- Model checking: $\mathsf{G}^c(q\beta)$ is a model with respect to logical formulae

- Example, monadic second-order logic: first-order logic (with equality and atomic predicates E_a for $a \in \mathsf{A}$ with interpretation $\xrightarrow{a}$) + quantification over sets of vertices.

  Given $\mathsf{G}^c(q\beta)$ and $\Phi(x)$ with one free variable, is $\Phi(x)$ true at $q\beta$?

- Long history here: Büchi 1962 (example slide 5), Rabin 1968 (example slide 6), Muller and Schupp 1985 (real-time PDA) and Caucal 1996 (all PDA = prefix-recognisable graphs)

- Standard temporal logics, LTL, CTL, CTL* and modal μ-calculus, are sublogics

Decision questions II

- Equivalence checking: given $p\alpha$ and $q\beta$ are they equivalent ?

- Classically, formal language theory views the language generable from a configuration as paramount.

- $L(p\alpha) = \{w \in \mathsf{A}^* \, : \, p\alpha \xrightarrow{w} q\epsilon \text{ for some } q \in \mathsf{P}\}$ Empty stack acceptance

- Languages accepted by PDA are the context-free languages

Example

$$
\begin{array}{ccccccc}
q\epsilon & \xleftarrow{b} & qX & \xleftarrow{b} & qXX & \xleftarrow{b} & \ldots \\
\uparrow b & & \uparrow b & & \uparrow b & & \\
pX & \xrightarrow{a} & pXX & \xrightarrow{a} & pXXX & \xrightarrow{a} & \ldots \\
\downarrow b & & \downarrow b & & \downarrow b & & \\
r\epsilon & \xleftarrow{\epsilon} & rX & \xleftarrow{\epsilon} & rXX & \xleftarrow{\epsilon} & \ldots
\end{array}
$$

$L(pX)$ is $\{a^n b^{n+1} : n \geq 0\} \cup \{a^n b : n \geq 0\}$.

Language equivalence

- In 1960s: $L(p\alpha) = L(q\beta)$ undecidable for single-state real-time PDA
- Active research question of 1960/70s: is $L(p\alpha) = L(q\beta)$ decidable for deterministic PDA? The DPDA equivalence problem.
  - Decidable for real-time single-state DPDA , Korenjak and Hopcroft in 1966. (Exponential algorithm. Improved to polynomial time.)
  - Decidable for single stack . . .
  - Decidable for finite-turn . . .
  - ⋮
  - Decidable for real-time DPDA, Oyamaguchi, Honda and Inagaki in 1980. (No complexity bound: procedure is two semi-decision procedures.)

Positive solution of DPDA problem

- Sénizergues (full proof, 166 pages, 2001). Configuration notation not rich enough. Deeper algebraic structure needed. Sénizergues's proof is primarily algebraic. Simplified proof (Stirling 2001) using bisimulation equivalence and graphs. Redone algebraically (Sénizergues 2002).

- Solution is two semi-decision procedures: one for inequivalence, find a smallest distinguishing word, one for equivalence, a nondeterministic proof procedure. No complexity upper bound.

- Simpler deterministic decision procedure with a primitive recursive complexity upper bound (Stirling 2002) using bisimulation equivalence and graphs $\mathsf{G}^c(p\alpha)$. Topic of these lectures.

Aside: final state vs empty stack

- A more classical definition of a PDA includes final states $\mathsf{F} \subseteq \mathsf{P}$
- $L(p\alpha)$ is $\{w \in \mathsf{A}^* \; : \; p\alpha \xrightarrow{w} q\beta, \; q \in \mathsf{F}\}$ Final state acceptance
- For PDA, languages recognised under final state acceptance coincide with those recognised under empty stack acceptance
- Not true for DPDA. Languages recognised under final state acceptance are deterministic context-free languages. Languages accepted under empty stack acceptance is a proper subset. The DPDA equivalence problem is really about deterministic context-free languages.

Empty stack acceptance is OK

For any deterministic context-free language L, there is a DPDA configuration that accepts $L\$$ where $\$ \notin$ A is an end of word marker

$$L\$ = \{w\$ \,:\, w \in L\}$$

Proof

- Add new bottom of stack symbol B to a DPDA with final state acceptance: $p\alpha \xrightarrow{u} q\beta$ iff $p\alpha B \xrightarrow{u} q\beta B$ in amended DPDA. Construct DPDA with empty stack acceptance from amended DPDA.

- If for final q and X, $qX \xrightarrow{\epsilon} r\epsilon \in$ T and r not final, then add new final r' and replace transition with $qX \xrightarrow{\epsilon} r'\epsilon$ and add transitions for r' same as for r.

- For each final q and X, if qX is stable then add new transition $qX \xrightarrow{\$} e\epsilon$ where e erases all stack elements (including B), so $eY \xrightarrow{\varepsilon} e\varepsilon$ for all Y.

- So, $p\alpha B \xrightarrow{u} q\beta B$ and q is final in amended DPDA with final state acceptance iff $p\alpha B \xrightarrow{u\$} e\epsilon$ in DPDA that accepts with empty stack acceptance.

Bisimulation equivalence

- Based on existence of a binary relation between vertices of a transition graph which is preserved by transitions.
- R is a bisimulation on vertices whenever $(E, F) \in R$, for all $a \in$ A,
  - if $E \xrightarrow{a} E'$ then there is an F'. $F \xrightarrow{a} F'$ and $(E', F') \in R$
  - if $F \xrightarrow{a} F'$ then there is an E'. $E \xrightarrow{a} E'$ and $(E', F') \in R$
- $E \sim F$ if there is a bisimulation containing (E, F)
- $\sim$ is bisimulation equivalence

Example

$$
\begin{array}{ccccccccccc}
p_1X & \xrightarrow{b} & p_1\epsilon & & q_1'\epsilon & \xleftarrow{b} & q_1X & \xrightarrow{b} & q_1\epsilon & & & r_1\epsilon \\
\uparrow a & & & & & & \uparrow a & & & & & \uparrow b \\
pX & & & & & & qX & & & rX & \xrightarrow{a} & r_1X \\
\downarrow a & & & & & & \downarrow a & & & & & \downarrow c \\
p_2X & \xrightarrow{c} & p_2\epsilon & & q_2'\epsilon & \xleftarrow{c} & q_2X & \xrightarrow{c} & q_2\epsilon & & & r_2\epsilon
\end{array}
$$

$pX \sim qX$. The bisimulation relation R which witnesses this is

$$\{(pX, qX), (p_1X, q_1X), (p_2X, q_2X), (p_1\epsilon, q_1'\epsilon), (p_1\epsilon, q_1\epsilon), (p_2\epsilon, q_2'\epsilon), (p_2\epsilon, q_2\epsilon)\}$$

The proof that R is a bisimulation relation is straightforward

Example

$$
\begin{array}{ccccccccccc}
p_1X & \xrightarrow{b} & p_1\epsilon & & q_1'\epsilon & \xleftarrow{b} & q_1X & \xrightarrow{b} & q_1\epsilon & & & r_1\epsilon \\
\uparrow a & & & & & & \uparrow a & & & & & \uparrow b \\
pX & & & & & & qX & & & rX & \xrightarrow{a} & r_1X \\
\downarrow a & & & & & & \downarrow a & & & & & \downarrow c \\
p_2X & \xrightarrow{c} & p_2\epsilon & & q_2'\epsilon & \xleftarrow{c} & q_2X & \xrightarrow{c} & q_2\epsilon & & & r_2\epsilon
\end{array}
$$

$pX \not\sim rX$ although $L(pX) = L(rX)$.

The problem is how to match the transition $rX \xrightarrow{a} r_1X$.

Example

$$
\begin{array}{ccccccccc}
pX & \xrightarrow{a} & pXX & \xrightarrow{a} & pXXX & \xrightarrow{a} & pXXXX & \xrightarrow{a} & \dots \\
qY & \xrightarrow{a} & q_1Y & \xrightarrow{a} & qYY & \xrightarrow{a} & q_1YY & \xrightarrow{a} & \dots
\end{array}
$$

The bisimulation R which witnesses $pX \sim qY$ is infinite

$$\{(pX^{2n}, q_1Y^n) \,:\, n > 0\} \cup \{(pX^{2n+1}, qY^n) \,:\, n > 0\}$$

For instance $(pX^{24}, q_1Y^{12}) \in R$. There is a single transition $pX^{24} \xrightarrow{a} pX^{25}$ and $q_1Y^{12} \xrightarrow{a} qY^{13}$, and $(pX^{25}, qY^{13}) \in R$.

Bisimulation approximants

The family $\{\sim_n : n \geq 0\}$ is defined inductively

$$E \sim_0 F \text{ for all vertices } E, F$$
$$E \sim_{n+1} F \text{ iff for all } a \in \mathsf{A}$$
$$\text{if } E \xrightarrow{a} E' \text{ then there is an } F'.\ F \xrightarrow{a} F' \text{ and } E' \sim_n F', \text{ and}$$
$$\text{if } F \xrightarrow{a} F' \text{ then there is an } E'.\ E \xrightarrow{a} E' \text{ and } E' \sim_n F'$$

Fact If $p\alpha \sim q\beta$ then for all $n \geq 0$. $p\alpha \sim_n q\beta$

Fact If the transition graphs containing E and F have finite out degree and for all $n \geq 0$. $E \sim_n F$ then $E \sim F$

Decision question $p\alpha \sim q\beta$?

- Decidable for normed single-state real-time PDA, irredundant context-free grammars. (Baeten, Bergstra and Klop 1987). Extended to all single-state real-time PDA, all context-free grammars. (Christensen, Hüttel and Stirling 1992).

- Decidability extended to normed real-time PDA, real-time PDA and collapsed graphs of ϵ-deterministic PDA. (Stirling 1996, Sènizergues 2002)

- Undecidable for arbitrary collapsed graphs of PDA (a consequence of Srba 2002).

Language and bisimulation equivalence

- Fact For any collapsed graph of a PDA, if $q\beta \sim r\delta$ then $L(q\beta) = L(r\delta)$

- Converse only holds if the PDA is deterministic and normed. Proof: verify then that the following relation on collapsed graph is a bisimulation.

$$\{(q\beta, r\delta) \,:\, L(q\beta) = L(r\delta)\}$$