Semantic Flowers for Good-for-Games and Deterministic Automata

Tansholpan Zhanabekova

University of Liverpool

joint work with Daniele Dell'Erba, Sven Schewe, Qiyi Tang

Motivation

- ω-regular languages describe infinite behaviors
- Automata like parity, Rabin, and Streett are used to define them
- Comparing automata's expressive power is complex
- The paper proposes semantic flowers as a simpler framework

Understanding ω-regular languages

- Languages over infinite sequences
- Used to model non-terminating systems
- Accepted by Büchi, Parity, Rabin, Streett, etc.

From Syntactic to Semantic Flowers

- Syntactic flowers: structure in automata (states and transitions)
- Semantic flowers: structure in the language itself

Semantic Flowers

A (semantic) *flower with petals* $c, \ldots, d$ in $\mathcal{L}$ consists of

- a finite word $w_s \in \Sigma^*$, called the stem and
- $d - c + 1$ *petals* $w_c, \ldots, w_d \in \Sigma^+$ with the following properties: for every infinite word $w = w_0', w_1', w_2', \ldots$ such that
  - $w_0' = w_s$ is the stem word, and
  - for all $i > 0$, $w_i' \in \{w_c, \ldots, w_d\}$.

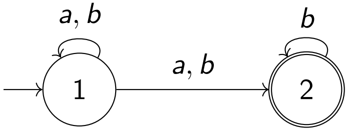
Why Semantic Flowers are useful?

- Effective Complexity Representation
- Synergy of Syntax and Semantics
- Natural Conceptual Framework

Main theorems

- Equivalence of syntactic and semantic flowers for parity automata
- Semantic flowers characterise the expressive limits of:
- Deterministic automata (DPA)
- Good-for-games (GFG) automata
- Rabin, Streett, Muller automata

Finite and Büchi automata



Büchi automata

interpreted over **infinite words** here: over $\Sigma = \{a, b\}$

run: start at some **initial state**

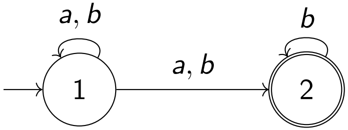
stepwise: read an **input** letter, and traverse the automaton respectively

accepting: is **infinitely often** in a **final state** while processing the complete ω-word

language: words with accepting runs

here: ω-words with **finitely many a's**

Determinisation of Büchi automata



Determinisation of Büchi automata

. . . are **less expressive** than nondeterministic Büchi automata.

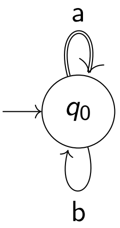
Example Language: All words with finitely many *a*'s

Construct an input word by repeatedly

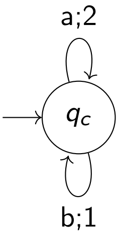
- choosing *b*'s until a final state is reached
- choosing an *a* once.

⇒ determinisation requires **more involved acceptance condition**

Deterministic Büchi Automata



Deterministic Parity Automata



Syntactic flowers

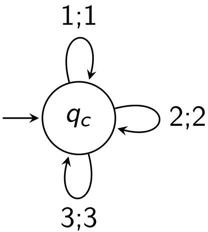
Let $\mathcal{A}$ be a deterministic automaton. A *syntactic flower with petals* $c, \dots, d$ in $\mathcal{A}$ consists of

- a reachable state q_c, called the centre of the flower and
- $d - c + 1$ *petals* $\rho_c, \dots, \rho_d$ with the following properties:
- each petal ρ_i for $c \leq i \leq d$ is a non-trivial run from q_c to itself;

Question

Can you think of $\mathcal{L}$ recognizable by a deterministic parity automata with colours 1,2,3, but not one with colours 0,1,2?

Flowers



What is a Good-for-Games automaton?

- GFG = Good-for-Games
- A nondeterministic automaton with a strategy that resolves choices using only past input
- Behaves deterministically in interaction, despite internal nondeterminism
- The paper uses semantic flowers to simplify reasoning about GFG expressive power

Good-for-Games Automata

Roughly

1. analyse the product Game $\times$ GFGA
2. make decisions on-the-fly
3. you'll get the correct winner & winning strategy
4. essentially the same algorithms as for DPAs
5. same acceptance complexity — pairs, colours

One way to check GFG-ness — letter game

- spoiler: chooses a letter
- verifier: chooses a transition

Spoiler wins iff she can produce a word that should be accepted, but is rejected.

Summary

1. Introduced semantic flowers as a simple and purely semantic way to characterise the complexity of ω-regular languages.
2. Discussed syntactic flowers
3. Explained how semantic flowers extend to Good-for-Games (GFG) automata

Thank you for your attention!