Degrees of Second and Higher Order Polynomials [1]

Donghyun Lim[a]

[a]KAIST klimdhn@kaist.ac.kr

Fourth Workshop on Mathematical Logic and its Applications
22-24 March 2021 Online

[1]This work was supported by the National Research Foundation of Korea (grant 2017R1E1A1A03071032) and by the International Research & Development Program of the Korean Ministry of Science and ICT (grant 2016K1A3A7A03950702).

First-order Polynomials

Syntax of First-order Polynomials

$$p, q ::= 1 \mid x \mid p + q \mid p \cdot q \mid -p$$

- Polynomials defined this way can be transformed into the normal form.

Example: Normal Form

$x \cdot (x + x + 1 + x \cdot (1 + 1 + 1) \cdot x) + x + 1 \Rightarrow 3x^3 + 2x^2 + 2x + 1$

First-order Polynomials

Two Polynomials in Normal Forms

$$c_n x^n + \cdots + c_1 x + c_0 = d_m x^m + \cdots + d_1 x + d_0$$

- normal forms are equal $\Rightarrow$ values are equal at every x
- normal forms are different $\Rightarrow$ values are different at some x
  - Proof.
  - Suppose that p and q have different normal forms.
  - Then $p - q$ cannot be reduced to zero polynomial.
  - By fundamental theorem of algebra, the equation $p(x) - q(x) = 0$ has only finitely many roots.
  - So $p(x) - q(x) \neq 0$ at some x.
- It **needs** proof!
- One consequence of uniqueness of normal form: *degree is well-defined.*
- Our work is something like this, but on second-order polynomials.

Motivations for Second-order Polynomials

- Computational cost is measured in dependence of input size.
- A first-order function problem $\{0,1\}^* \to \{0,1\}^*$
  - Cost : $n \mapsto \max_{w:|w|\le n} \mathrm{cost}(w)$
  - $w \in \{0,1\}^*$
  - Cost : $\mathbb{N} \to \mathbb{N}$
  - (First-order) polynomials characterize important subclasses of feasibly computable functions. [Cobham, 1965]
- A second-order function problem $(\{0,1\}^*)^{\{0,1\}^*} \times \{0,1\}^* \to \{0,1\}^*$
  - Cost : $\ell, n \mapsto \max_{\phi, w:|\phi|\le \ell, |w|\le n} \mathrm{cost}(\phi, w)$
  - $\phi \in (\{0,1\}^*)^{\{0,1\}^*}$, $w \in \{0,1\}^*$, $\ell : \mathbb{N} \to \mathbb{N}$
  - $|\phi| \le \ell$ means $|\phi(x)| \le \ell(|x|)$ for every $x \in \{0,1\}^*$.
  - Cost : $\mathbb{N}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N}$
  - Second-order polynomials characterize important subclasses of feasibly computable functions. [Kapron, 1996]

Overview

Second-order Polynomials

$$P(\ell, n) \in \mathbb{N} \qquad (\ell : \mathbb{N} \to \mathbb{N}, n \in \mathbb{N})$$

Contribution

- We define syntax and semantics of second-order polynomials.
- We prove soundness (syntax $\Rightarrow$ semantics).
- We prove completeness (semantics $\Rightarrow$ syntax).
- We define degree of second-order polynomials.

Ongoing Work

- Generalization to higher-order.
- Applications to complexity theory.

Syntax

Syntax

$$P, Q ::= \mathbf{1} \mid \mathbf{x} \mid P + Q \mid P * Q \mid \mathbf{f}(P)$$

Example

- $\mathbf{1}$
- $\mathbf{1} + \mathbf{x}$
- $\mathbf{f}(\mathbf{1} + \mathbf{x})$
- $\mathbf{f}(\mathbf{1} + \mathbf{x}) + \mathbf{x} * \mathbf{x} * \mathbf{1}$
- $\mathbf{f}(\mathbf{f}(\mathbf{f}(\mathbf{1} + \mathbf{x}) + \mathbf{x} * \mathbf{x} * \mathbf{1})) * \mathbf{f}(\mathbf{1} + \mathbf{x})$

Syntactically different, but must be equivalent

- $\mathbf{1} + \mathbf{x}$ and $\mathbf{x} + \mathbf{1}$
- $\mathbf{x} + (\mathbf{x} + \mathbf{x})$ and $(\mathbf{x} + \mathbf{x}) + \mathbf{x}$
- $\mathbf{1} * \mathbf{x}$ and $\mathbf{x}$

Syntax

Syntactic Equivalence $\sim_{syn}$

The equivalence relation $\sim_{syn}$ generated from

- $(P + Q) + R \sim_{syn} P + (Q + R)$
- $P + Q \sim_{syn} Q + P$
- $(P * Q) * R \sim_{syn} P * (Q * R)$
- $P * Q \sim_{syn} Q * P$
- $P * (Q + R) \sim_{syn} (P * Q) + (P * R)$
- $P * \mathbf{1} \sim_{syn} P$

which is congruent to $+/*/\mathbf{f}$.

Example

- $\mathbf{f}(x * x) * \mathbf{f}(x) + \mathbf{f}(x * x) \sim_{syn} \mathbf{f}(x * x) * (1 + \mathbf{f}(x))$
- $\mathbf{f}(1 + x * 1 + 1 + 1) \sim_{syn} \mathbf{f}(x + 1 + 1 + 1)$

Semantics

Syntax

$$P, Q ::= \mathbf{1} \mid \mathbf{x} \mid P + Q \mid P * Q \mid \mathbf{f}(P)$$

Semantics

- Canonical recursive interpretation as a second-order natural number function $[\![P]\!] : \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N}$
- $[\![\mathbf{1}]\!](\ell, n) := 1$
- $[\![\mathbf{x}]\!](\ell, n) := n$
- $[\![P + Q]\!](\ell, n) := [\![P]\!](\ell, n) + [\![Q]\!](\ell, n)$
- $[\![P * Q]\!](\ell, n) := [\![P]\!](\ell, n) * [\![Q]\!](\ell, n)$
- $[\![\mathbf{f}(P)]\!](\ell, n) := \ell([\![P]\!](\ell, n))$

Soundness

Soundness Theorem

If $P \sim_{syn} Q$, then $[\![P]\!] = [\![Q]\!]$.

- $[\![P]\!] = [\![Q]\!]$ means that $[\![P]\!], [\![Q]\!] : \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N}$ agree on all arguments $\ell : \mathbb{N} \to \mathbb{N}$ and $n \in \mathbb{N}$.

Proof

Induction on the generating rules of $\sim_{syn}$.

- The proof follows directly from definition.

Completeness Theorem

If $[\![P]\!] = [\![Q]\!]$, then $P \sim_{syn} Q$.

Proof Idea

- For example, consider $P = \mathbf{f}(\mathbf{x} + \mathbf{f}(\mathbf{x})) * \mathbf{x} + \mathbf{f}(\mathbf{x} + \mathbf{f}(\mathbf{x})) * \mathbf{f}(\mathbf{1})$.
- Replace $\mathbf{f}(\mathbf{x} + \mathbf{f}(\mathbf{x}))$ by y_1, $\mathbf{f}(\mathbf{1})$ by y_2, $\mathbf{x}$ by x.
- $y_1 * x + y_1 * y_2$
- We transformed P into a first-order multivariate polynomial.
- Do it recursively down below to analyze the structure of P as a graph.
- Suppose $P \nsim_{syn} Q$. Use the following lemma to construct $\ell : \mathbb{N} \to \mathbb{N}$ and $n \in \mathbb{N}$ such that $[\![P]\!](\ell, n) \neq [\![Q]\!](\ell, n)$.

Lemma (well-known elementary fact)

For every distinct multivariate (first-order) polynomials $p, q \in \mathbb{Z}[y_1, \cdots, y_n]$, there exist $a_1, \cdots, a_n \in \mathbb{N}$ such that $p(y_1 := a_1, \cdots, y_n := a_n) \neq q(y_1 := a_1, \cdots, y_n := a_n)$.

Degree of Second-order Polynomials

Syntax

$$P, Q ::= \mathbf{1} \mid \mathbf{x} \mid P + Q \mid P * Q \mid \mathbf{f}(P)$$

Definition of DEG

- $\mathrm{DEG}(\mathbf{1}) := 0$
- $\mathrm{DEG}(\mathbf{x}) := 1$
- $\mathrm{DEG}(P + Q) := \max(\mathrm{DEG}(P), \mathrm{DEG}(Q))$
- $\mathrm{DEG}(P * Q) := \mathrm{DEG}(P) + \mathrm{DEG}(Q)$
- $\mathrm{DEG}(\mathbf{f}(P)) := \mathrm{DEG}(P) \cdot x$

- It coincides with the usual (first-order) polynomial degree.
- For a second-order polynomial P, $\mathrm{DEG}(P)$ is a first-order polynomial.

Example

$\mathrm{DEG}(\mathbf{f}(\mathbf{f}(\mathbf{x})) * \mathbf{f}(\mathbf{x}^5) * \mathbf{x}) = x^2 + 5x + 1$

Degree of Second-order Polynomials

Example

- $\mathrm{DEG}(\mathbf{f}(\mathbf{f}(\mathbf{x})) * \mathbf{f}(\mathbf{x}^5) * \mathbf{x}) = x^2 + 5x + 1$
- $\mathrm{DEG}(\mathbf{f}(\mathbf{x}^{999})) = 999x$
- $\mathrm{DEG}(\mathbf{f}(\mathbf{f}(\mathbf{x})) * \mathbf{f}(\mathbf{x}^5) * \mathbf{x} + \mathbf{f}(\mathbf{x}^{999})) = \max(x^2+5x+1, 999x) = x^2+5x+1$

- max is not pointwise. If it were, the result is not a polynomial.
- Take the one with the larger degree; ties are broken by dictionary order.
- $\max(x^5 + x^4 + 10x^3, x^5 + x^4 + x^3) = x^5 + x^4 + 10x^3$

Example

$\deg(\mathrm{DEG}(\mathbf{f}(\mathbf{f}(\mathbf{x})) * \mathbf{f}(\mathbf{x}^5) * \mathbf{x})) = \deg(x^2 + 5x + 1) = 2$

- The (first-order) degree of the (second-order) degree is the largest nesting depth of **f**.

Degree of Second-order Polynomials

- If $P \sim_{syn} Q$, then $\mathrm{DEG}(P) = \mathrm{DEG}(Q)$
  - Syntactically equivalent polynomials have the same degree.
  - Proof is by straightforward induction.

Completeness Theorem

If $[\![P]\!] = [\![Q]\!]$, then $P \sim_{syn} Q$.

- $[\![P]\!] = [\![Q]\!] \Rightarrow P \sim_{syn} Q \Rightarrow \mathrm{DEG}(P) = \mathrm{DEG}(Q)$
  - By completeness theorem, semantically the same polynomials have the same degree.
  - It would be absurd if the cost of a second-order algorithm $((\{0,1\}^*)^{\{0,1\}^*} \times \{0,1\}^* \to \{0,1\}^*)$ is given by a second-order polynomial $(\mathbb{N}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N})$ which has multiple possible degrees.
  - *Completeness theorem is crucial in well-defining degree!*

Compositions

Elementary Fact

For first-order polynomials $p \neq 0$ and $q \neq 0$,

$$\deg(p \circ q) = \deg(p) \times \deg(q).$$

- We generalize this to second-order polynomials.
- What is the composition of second-order polynomials?

Two (Semantic) Compositions

For $F, G : \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N}$,

- $\lambda \ell . \lambda n . F(\ell, G(\ell, n))$
- $\lambda \ell . F(G(\ell))$ (as maps of type $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$)

- We give syntactic definition of each composition.

$\mathbf{x}$-composition $\circ_{\mathbf{x}}$

Definition

$P \circ_{\mathbf{x}} Q :=$ replace every occurrence of $\mathbf{x}$ in P by Q (at once).

Properties

- $[\![P \circ_{\mathbf{x}} Q]\!](\ell, n) = [\![P]\!](\ell, [\![Q]\!](\ell, n))$
- $\mathrm{DEG}(P \circ_{\mathbf{x}} Q) = \mathrm{DEG}(P) \times \mathrm{DEG}(Q)$

- Proof is by straightforward induction.
- Congruent with respect to $\sim_{syn}$ by soundness and completeness.

Elementary Fact

For first-order polynomials $p \neq 0$ and $q \neq 0$,

$$\deg(p \circ q) = \deg(p) \times \deg(q).$$

$\mathbf{f}$-composition $\circ_{\mathbf{f}}$

Definition

$P \circ_{\mathbf{f}} Q :=$ replace occurrence of subterm $\mathbf{f}(P')$ in P by $Q \circ_{\mathbf{x}} (P' \circ_{\mathbf{f}} Q)$ (recursively from below).

Properties

- $[\![P \circ_{\mathbf{f}} Q]\!](\ell, n) = [\![P]\!]([\![Q]\!](\ell), n)$
- $\mathrm{DEG}(P \circ_{\mathbf{f}} Q) = \mathrm{DEG}(P) \circ \mathrm{DEG}(Q)$

- Proof is by straightforward induction.
- Congruent with respect to $\sim_{syn}$ by soundness and completeness.

Elementary Fact

For first-order polynomials $p \neq 0$ and $q \neq 0$,

$$\deg(p \circ q) = \deg(p) \times \deg(q).$$

Generalization to Higher-order (work in progress)

Definition

A higher-order polynomial is a lambda term of simply typed lambda calculus with base type $\mathbb{N}$ and three constants:

$$\begin{aligned} \mathbf{1} &: \mathbb{N} \\ + &: \mathbb{N} \to \mathbb{N} \to \mathbb{N} \\ * &: \mathbb{N} \to \mathbb{N} \to \mathbb{N} \end{aligned}$$

- A first-order polynomial is a lambda term of type $\mathbb{N} \to \mathbb{N}$.
- A second-order polynomial is a lambda term of type $(\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})$.
- A multivariate first-order polynomial is a lambda term of type $\mathbb{N} \to \mathbb{N} \to \cdots \to \mathbb{N}$.

Application to Complexity Theory (work in progress)

- Cost of computing a string function $f : \{0,1\}^* \to \{0,1\}^*$ is measured by a natural number function $p : \mathbb{N} \to \mathbb{N}$
  - (First-order) polynomials characterize important subclasses of computable functions. [Cobham, 1965]
  - One can further refine these subclasses by considering degrees of polynomials. $(O(n), O(n^2), \cdots)$
- Cost of computing a second-order string function $F : (\{0,1\}^* \to \{0,1\}^*) \times \{0,1\}^* \to \{0,1\}^*$ is measured by a second-order natural number function $P : \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \to \mathbb{N}$.
  - Second-order polynomials characterize important subclasses of computable functions. [Kapron, 1996]
  - *One can further refine these subclasses by considering degrees of second-order polynomials.*
- *Cost of computing a higher-order string function is measured by a higher-order natural number function.*