

# Inference of Regular Expressions for Text Extraction from Examples



A. Bartoli, A. De Lorenzo, E. Medvet, F. Tarlao  
University of Trieste, Italy



# Regular Expressions Inference From Examples

- Regular expressions:
  - Used **routinely** in **many** different domains
  - Since a **long time**
- We developed a **GP-based** method for regular expression inference
- **IEEE Transactions on Knowledge and Data Engineering**
- **IEEE Intelligent Systems**

# Why human-competitive? (H)

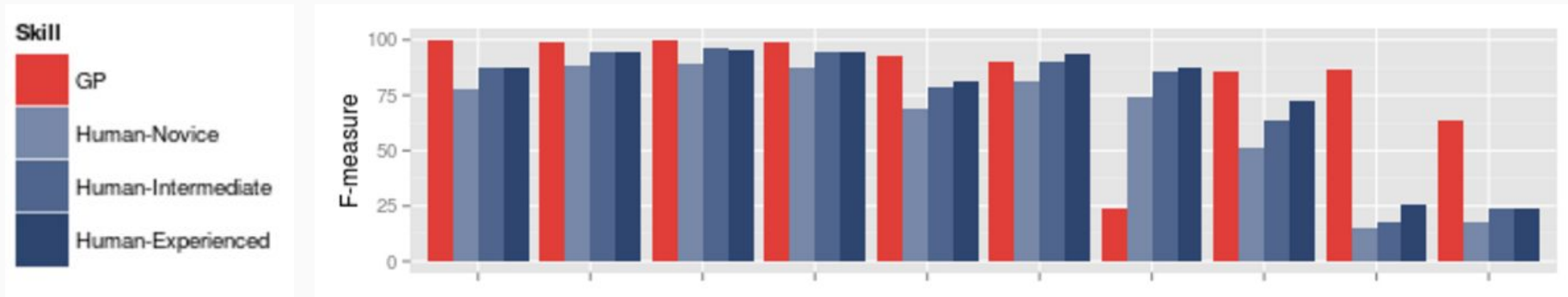
The result holds its own or wins a **regulated competition involving human contestants** (in the form of either live human players or human-written computer programs)

- Web challenge: 10 regex-writing tasks specified by examples
- **1700 (one thousand seven hundreds)** participants (!!!) in a few days



The screenshot shows a Reddit post interface. At the top left is the Reddit logo (a white alien head) and the word "reddit" in a bold, lowercase font. To the right of the logo is the word "PROGRAMMING" in all caps. Below this, there are three tabs: "commenti" (highlighted in orange), "pertinente", and "altre discussioni (1)". The main text of the post is "How good are you in writing regex? Challenge yourself!" in a blue font, followed by "(play.inginf.units.it)" in a smaller grey font. Below the title, there is an upvote arrow with the number "32" and the text "inviato 4 mesi fa da mark-allei". At the bottom, there is a downvote arrow with the text "34 commenti" and "condividi".

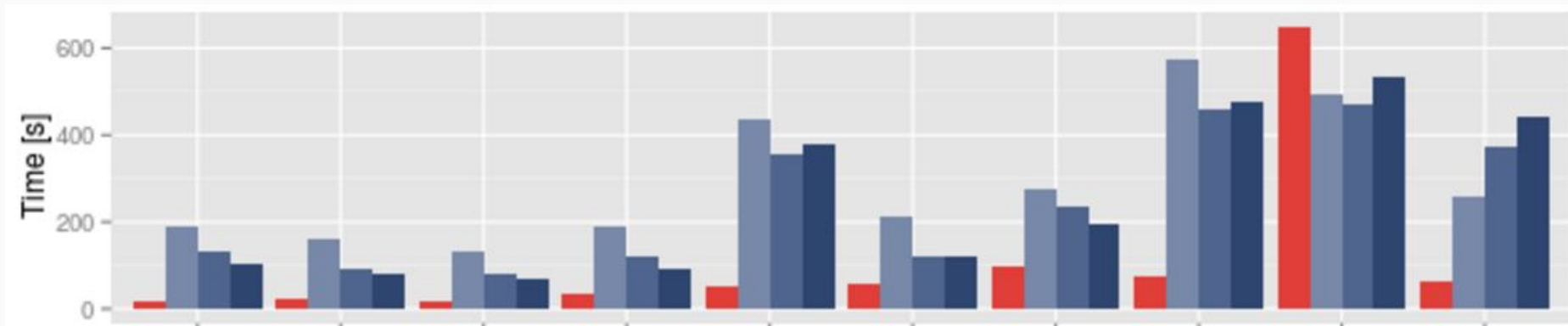
# Why human-competitive? (H): Quality of constructed solution



- **Quality** of constructed regex (F-measure):  
(almost always) **better than** the average of each user category

# Why human-competitive? (H): Time for constructing a solution

- **Time** for constructing the regular expression:  
(almost always) **faster than** the average of each user category



# Why human-competitive? (B)

*The result is equal to or better than a result that was accepted as **a new scientific result** at the time when it was published in a **peer-reviewed scientific journal***

- We improve significantly over 3 baseline methods
  - IEEE TPAMI (2005)
  - IEEE Computer (2014)
  - ACM PLDI (2014)
- Full details in our **IEEE-TKDE** paper

# Why human-competitive? (D)

The result is publishable in its own right as a new scientific result **independent of the fact that the result was mechanically created**

- **IEEE-TKDE**: "*the most popular flagship journal in the broad, data related areas, including data science, big data, data engineering, data mining, databases and systems, information retrieval and many others*"
- Concerned only with **quality** and **novelty** of the results
- The **nature** of the methods used for achieving those results is **irrelevant**

# Why human-competitive? (E)

The result is equal to or better than the most recent human-created solution to a **long-standing problem** for which there has been a succession of increasingly better human-created solutions

## Regular Language Induction with Genetic Programming

BERTRAND DANIEL DUNAY  
FREDERICK E. PETRY  
BILL P. BUCKLES

## Learning Regular Languages Using Genetic Programming

Børge Svingen  
Department of Computer and Information Science  
Norwegian University of Science and Technology  
N-7034 Trondheim  
Norway  
bsvingen@idi.ntnu.no

## Learning Regular Languages from Simple Positive Examples

FRANÇOIS DENIS  
Equipe Grappa, LIFL, Université de Lille 1, 59655 Villeneuve d'Ascq Cedex, France

denis@lifl.fr

## Algorithms for learning regular expressions from positive data<sup>☆</sup>

Henning Fernau

Universität Trier, FB 4, Abt. Informatik, Germany

## Learning Regular Expressions from Representative Examples and Membership Queries

Elim Kimber

## Regular Expression Learning for Information Extraction

Yunyao Li, Rajasekar Krishnamurthy, Sriram Raghavan, Shivakumar Vaithyanathan

## Results of the Abbadingo One DFA Learning Competition and

a New Evidence Driven State Merging Algorithm

Kevin J. Lang      Barak A. Pearlmutter      Rodney A. Price\*

## Inducing Grammars from Sparse Data Sets: A Survey of Algorithms and Results

Orlando Cicchello  
Stefan C. Kremer

OCICHEL@UOGUELPH.CA  
SKREMER@UOGUELPH.CA

Josh Bongard  
Hod Lipson

## Active Coevolutionary Learning of Deterministic Finite Automata

JOSH.BONGARD@CORNELL.EDU  
HOD.LIPSON@CORNELL.EDU



# Why human-competitive? (E)

*The result is equal to or better than the most recent human-created solution to a **long-standing problem** for which there has been a succession of increasingly better human-created solutions*

- **Many** proposals for automatic inference of regular expressions (from 1993 onwards)
- Ours improves over them significantly
- Only the most recent ones could address **non-trivial** text extraction tasks
- None could (meaningfully) use **humans** as a baseline

# Why human-competitive? (G)

The result solves *a problem of indisputable difficulty* in its field



stackoverflow

## Tags

regex × 147834

Regular expressions provide a declarative language to match patterns within strings. They are commonly used for string validation,

81 asked today, 505 this week

- Stackoverflow: Most popular programming forum
- “**regex**”: 26-th most popular tag in a set of more than 44,000 tags
- More than 144,000 questions with this tag

# Why the best entry? (1)

## Nature of the problem

- Construction of regular expressions:
  - **Practically relevant** problem in a **variety** of application domains
  - Requires a considerable amount of **skill, expertise** and **creativity**
- **Automatic** construction of regular expressions:
  - **Long-standing** scientific problem  
(many proposals since 1992)

# Why the best entry? (2)

## Quality of our solution

- **First** method capable of addressing **practical** tasks of **realistic complexity**
- Human-competitiveness: **more than 1700 human users on 10 tasks**
  - Better than/similar to **skilled** users (accuracy and construction time)
- Top-tier journal in which nature of the method is irrelevant
  - Better than 3 journal-published baselines

# Why the best entry? (3)

## Last but not least

- **Public prototype** (<http://regex.inginf.units.it>)
- **Full source code** (<http://github.com/MaLeLabTs/RegexGenerator>)