Improving Genetic Programming with Behavioral Consistency Measure

Krzysztof Krawiec[1] and Armando Solar-Lezama[2]

[1]Institute of Computing Science, Poznan University of Technology, Poznań, Poland

[2]Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA

Key insights

- Program error (fitness) is the objective quality measure of a program, but not necessarily the best search driver (meant as a means to *guide* the search)
- Better search drivers can be constructed by analyzing the internal behavior of programs

Main result

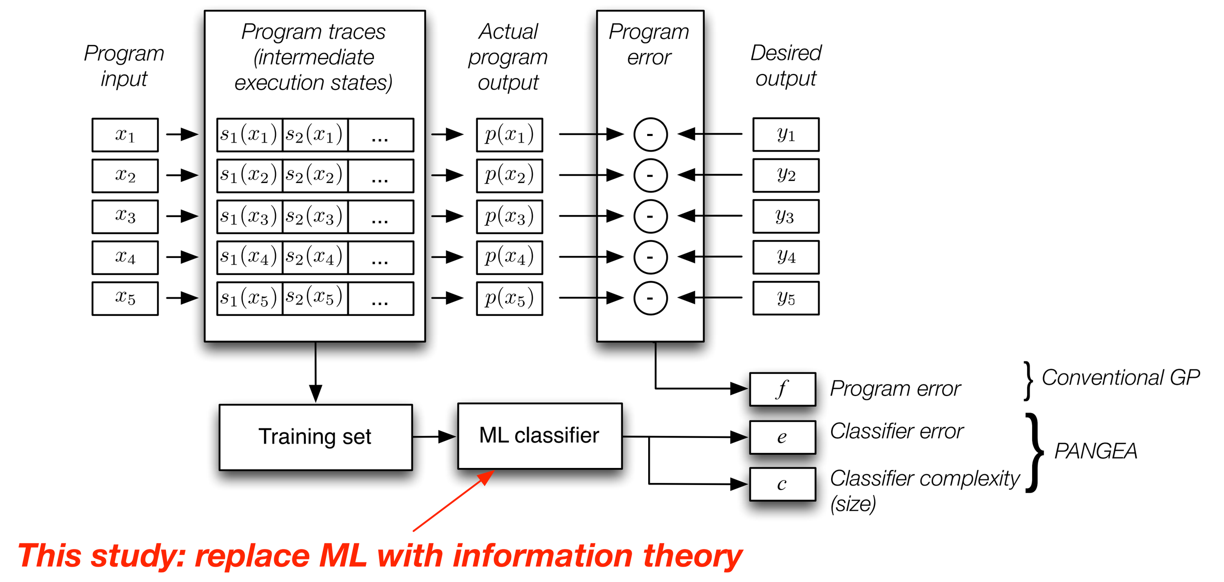
- A search driver based on information content of program behavior that improves performance of genetic programming

Hypotheses

- Even a poorly performing program may feature hidden relationships between its internal behavior and desired output
- Such relationships can be captured using measures based on information theory
- Such measures can be better search drivers than conventional fitness

Behavioral evaluation

Semantic GP taken to the next level: evaluate the internal behavior of programs, not only the outputs.



(PANGEA, Krawiec, Swan, O'Reilly, GECCO'13, GECCO'14, EuroGP'14)

Idea: Use information-based measures to examine program behavior

Given:

- Y - random variable associated with the desired program output
- S_k - random variable associated with the kth intermediate execution state

Define:

- $H(Y|S_k)$ - amount of information that Y adds to S_k
- $H(S_k|Y)$ - amount of information that S_k adds to Y

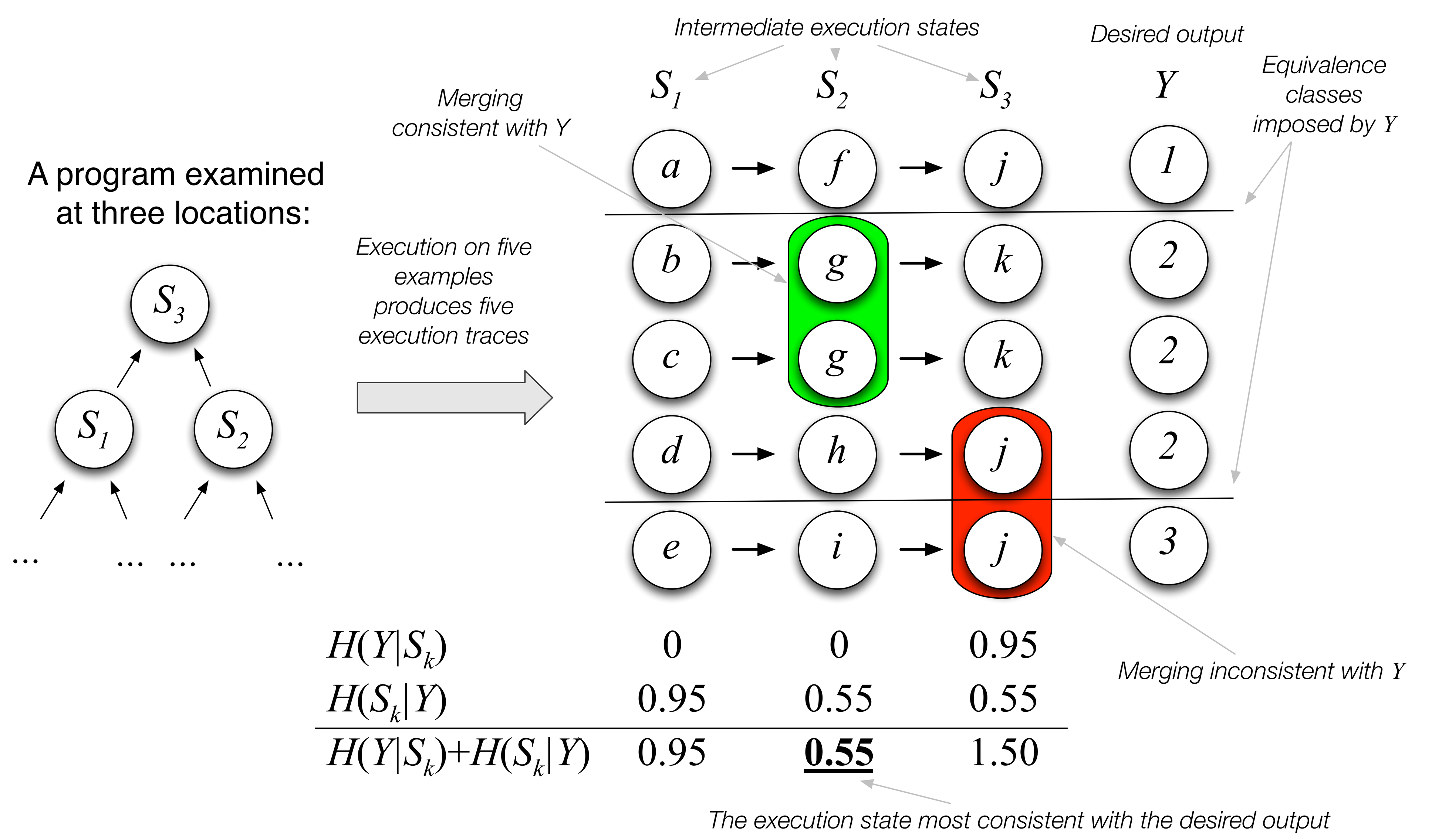
Note:

- $H(Y|S_k) > 0$ => S_k alone cannot predict Y
- $H(S_k|Y) > 0$ => S_k partitions the examples within Y's equivalence classes

We want to penalise both, hence the **proposed measure**:

$$I(p) = \min_k H(Y|S_k(p)) + H(S_k(p)|Y)$$

Example: three intermediate execution states, five examples



Experiment

Configurations:

- FxI - scalar aggregation of fitness F and I
- FI - multiobjective approach (NSGA-II)
- F - standard GP, F only (baseline)

Three domains and 35 benchmarks:

Domain	Instruction set	Problem	v	m	k
Boolean	and, nand, or, nor	Cmp6, Maj6, Mux6, Par6	6	64	2^{64}
		Cmp8, Maj8, Par8	8	256	2^{256}
		Mux11	11	2048	2^{2048}
Categorical	$a_l(x,y)$	D-a1, D-a2, D-a3, D-a4, D-a5	3	27	3^{27}
	$a_l(x,y)$	M-a1, M-a2, M-a3, M-a4, M-a5	3	15	3^{15}
Regression	$+, -, *, \%$, sin, cos, log, exp, $-x$	Keij1, Keij4, Nguy3..7, Sext	1	20	–
		Keij5, Keij11..14, Nguy9..10, Nguy12	2		
		Keij15	3		

Results

Budget = number of evaluations:

	FI	FxI	F	Friedman p
All problems	**1.60**	2.21	2.19	< 0.01
Categorical	**1.20**	2.05	2.75	< 0.01
Regression	**1.82**	2.00	2.18	< 0.10
Boolean	**1.62**	2.50	1.88	< 0.01

Budget = time:

	FI	FxI	F	Friedman p
All problems	**1.77**	2.17	2.06	0.08

Conclusions

- Behavioral evaluation:
  - leads to significant performance improvements
  - promotes some behaviors without explicitly specifying them
  - has a moderate computational overhead
- A step towards more information-rich evaluation and 'better-informed' program synthesis

Reasons (?) for moderate performance on Boolean problems:

- no negation in instruction set (leads to hard-to-escape-from local optima),
- relatively large number of inputs,
- low discriminative power of entropy on binary variables.

Acknowledgments

K. Krawiec acknowledges support from the Polish-U.S. Fulbright Commission and NCN grant no. DEC- 2011/01/B/ST6/07318, and A. Solar-Lezama from grant no. NSF-CCF-1161775.

Related

K. Krawiec, J. Swan. *Pattern-guided genetic programming*, GECCO'13.

K. Krawiec, U.-M. O'Reilly, *Behavioral Programming: A Broader and More Detailed Take on Semantic GP*, GECCO'14.

K. Krawiec, U.-M. O'Reilly, *Behavioral Search Drivers for Genetic Programing*, EuroGP'14.

PPSN 2014

September 13-17, 2014, Ljubljana, Slovenia