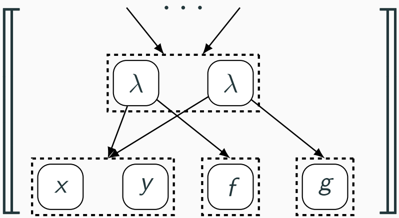
Categorical E-Graphs for λ-calculi



Aleksei Tiurin, Dan Ghica, Nick Hu
University of Birmingham

July, 2025

Equality saturation $\simeq$ E-graphs

Equality saturation

First introduced as a decision procedure for the quantifier-free fragment of theory of uninterpreted functional symbols with equality

Equality saturation

First introduced as a decision procedure for the quantifier-free fragment of theory of uninterpreted functional symbols with equality

Program optimisation technique that obviates the need to worry about phase ordering

Equality saturation

First introduced as a decision procedure for the quantifier-free fragment of theory of uninterpreted functional symbols with equality

Program optimisation technique that obviates the need to worry about phase ordering

Equalities are applied non-destructively until the representation is "saturated" with respect to these equalities

Equality saturation

First introduced as a decision procedure for the quantifier-free fragment of theory of uninterpreted functional symbols with equality

Program optimisation technique that obviates the need to worry about phase ordering

Equalities are applied non-destructively until the representation is "saturated" with respect to these equalities

Once saturated, the best candidate can be extracted from the set

Equality saturation

First introduced as a decision procedure for the quantifier-free fragment of theory of uninterpreted functional symbols with equality

Program optimisation technique that obviates the need to worry about phase ordering

Equalities are applied non-destructively until the representation is “saturated” with respect to these equalities

Once saturated, the best candidate can be extracted from the set

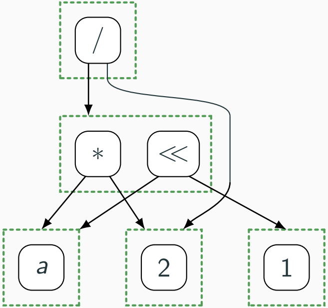
The best candidate is domain-specific

- Best program in compiler optimisation
- Smallest test-case in fuzzy testing
- Smallest digital circuit
- ...

E-graphs [Willsey et al. 2021]

E-graph is a data structure that facilitates equality saturation

Example



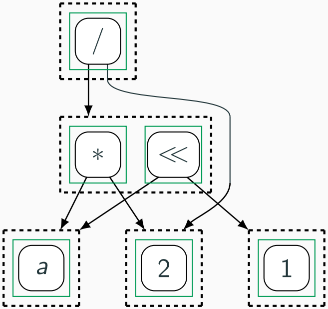
E-classes

- Set of equivalent terms represented by enodes rooted at this particular e-class

E-graphs [Willsey et al. 2021]

E-graph is a data structure that facilitates equality saturation

Example



E-nodes 

- A representative of an e-class once the root node is fixed

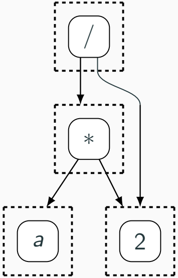
E-graphs [Willsey et al. 2021]

Can also be understood as a device computing the quotient of a free term-algebra over an algebraic signature

Can also be understood as a device computing the quotient of a free term-algebra over an algebraic signature

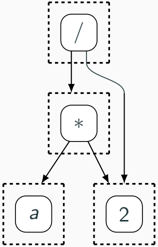
- $\Sigma = \{\mathbb{N} \setminus \{0\}, * : 2 \to 1, / : 2 \to 1, \ll : 2 \to 1, a : 0 \to 1\}$
- $\mathcal{E} = \{x * 2 = x \ll 1, (x * y)/z = x * (y/z), x/x = 1, x * 1 = x\}$
- a, $a * 2$, $a \ll 1$, $\ldots$

Example $((a * 2)/2)$

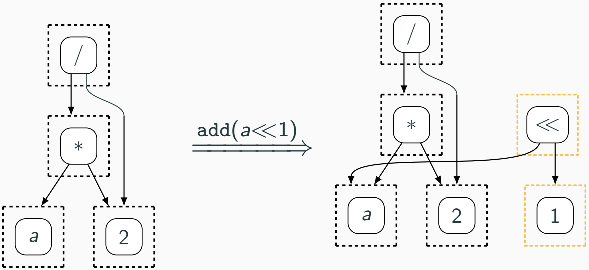


E-graphs [Willsey et al. 2021]

Example $(x * 2 \to x \ll 1)$

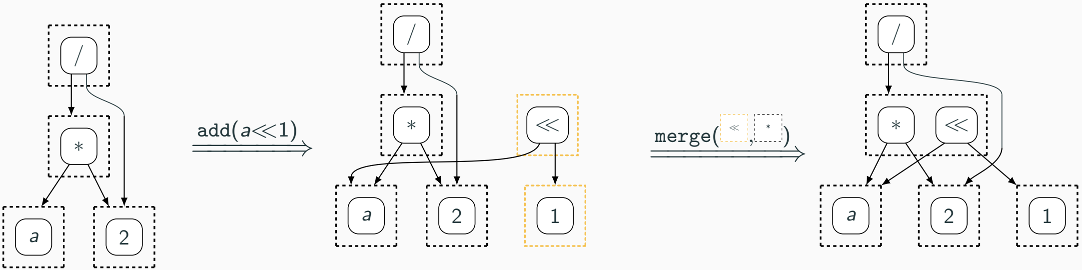


Example $(x * 2 \to x \ll 1)$



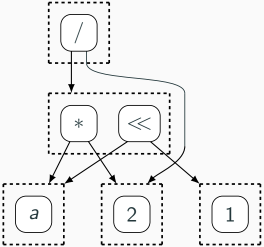
E-graphs [Willsey et al. 2021]

Example $(x * 2 \to x \ll 1)$



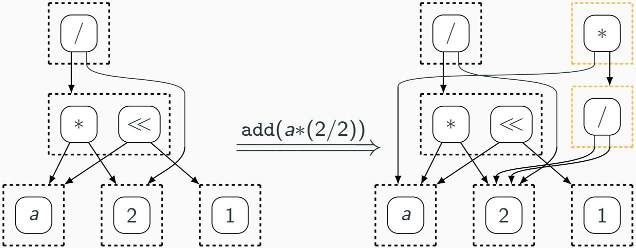
E-graphs [Willsey et al. 2021]

Example $((x * y)/z \to x * (y/z))$



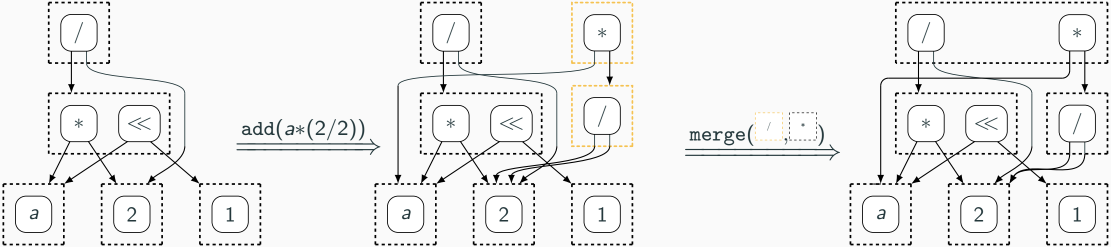
E-graphs [Willsey et al. 2021]

Example $((x * y)/z \to x * (y/z))$



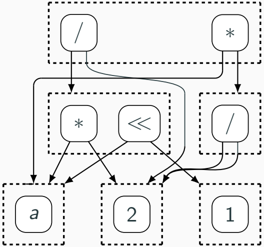
E-graphs [Willsey et al. 2021]

Example $((x * y)/z \to x * (y/z))$



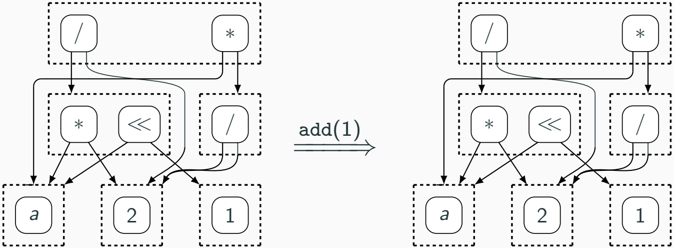
E-graphs [Willsey et al. 2021]

Example ($(x/x) \to 1$)



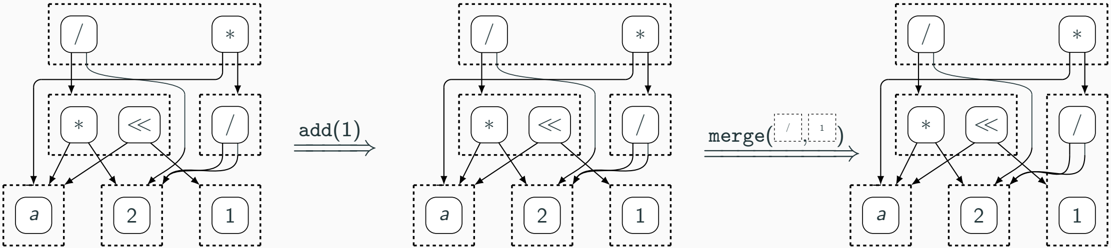
E-graphs [Willsey et al. 2021]

Example $((x/x) \to 1)$



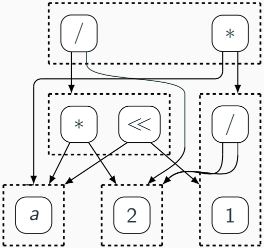
E-graphs [Willsey et al. 2021]

Example $((x/x) \to 1)$



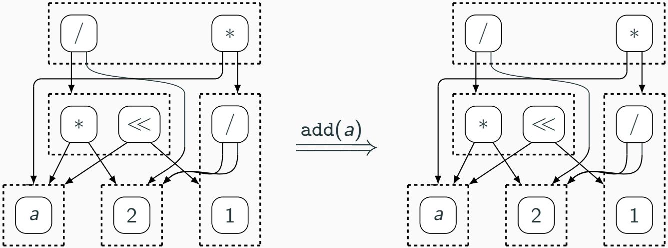
E-graphs [Willsey et al. 2021]

Example ($x * 1 \to x$)



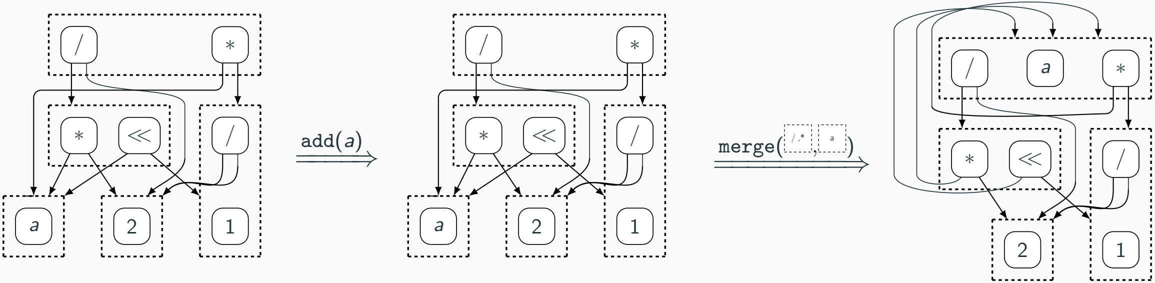
E-graphs [Willsey et al. 2021]

Example ($x * 1 \to x$)



E-graphs [Willsey et al. 2021]

Example $(x * 1 \to x)$



Challenges

- Do not scale well in the case of (symmetric) monoidal theories
- No first-class notion of variables, binding and substitution

Symmetric monoidal theories

Presented by a monoidal signature $\Sigma = (\Sigma_0, \Sigma_1)$ and a set of equations $\mathcal{E}$ between Σ-terms

Example

$\Sigma = \{\mu : 1 \to 2, \eta : 1 \to 0\}$

$\mathcal{E} = \{\mu; \mathsf{sym} \equiv \mu, \mu; (\eta \otimes \mathrm{id}) \equiv \mathrm{id}, \mu; (\mathrm{id} \otimes \mu) \equiv \mu; (\mu \otimes \mathrm{id})\}$

Two kinds of composition: $\fatsemi$ and $\otimes$

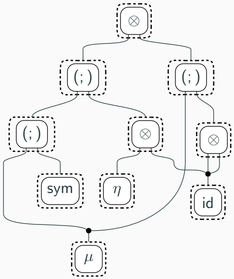
Additional generators id_n, $\mathsf{sym}_{m,n}$

Must coherently interact with each other: $f \fatsemi \mathsf{id}_n = f$, $(f \otimes g) \fatsemi (h \otimes k) = (f \fatsemi h) \otimes (g \fatsemi k)$, ...

Symmetric monoidal theories

Can be still presented algebraically by lifting $\fatsemi$ and $\otimes$ to the level of functional symbols

Example $(((\mu \fatsemi \mathsf{sym}) \otimes (\eta \otimes \mathsf{id})) \otimes (\mu \fatsemi (\mathsf{id} \otimes \mathsf{id})))$



- Saturating the e-graph yields...

Rule		(→)	(←)
Left identity	$(\mathrm{id}_n \mathbin{;} f = f)$	—	899
Right identity	$(f \mathbin{;} \mathrm{id}_n = f)$	4	308
Composition assoc.	$((f \mathbin{;} g) \mathbin{;} h = f \mathbin{;} (g \mathbin{;} h))$	1,325	580
Bifunctoriality	$((f_1 \otimes f_2) \mathbin{;} (g_1 \otimes g_2) = (f_1 \mathbin{;} g_1) \otimes (f_2 \mathbin{;} g_2))$	157	92
Tensor assoc.	$((f \otimes g) \otimes h = f \otimes (g \otimes h))$	1,695	2,174
Tensor unit	$(\mathrm{id}_0 \otimes f = f,\ f \otimes \mathrm{id}_0 = f)$	—	2,005
Sym. naturality	$(\mathrm{sym} \mathbin{;} (f \otimes g) = (g \otimes f) \mathbin{;} \mathrm{sym})$	10	—
Sym. involution	$(\mathrm{sym} \mathbin{;} \mathrm{sym} = \mathrm{id})$	—	1
Commutativity	$(\mu \mathbin{;} \mathrm{sym} = \mu)$	1	—
Counit	$(\mu; (\eta \otimes \mathrm{id}) = \mathrm{id})$	1	—
Total		3,193	6,059

Traditional e-graphs model variables promoting them to functional symbols with no inputs. No notion of binding and substitution

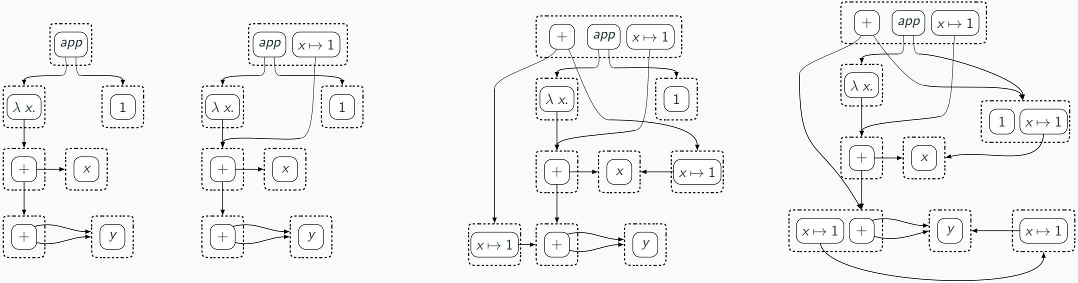
Example ([Gonzalez, Giridharan, and Kurashige 2023])

$(\lambda\, x.(y+y)+x)\ 1$

$(\lambda\, x.(y+y)+x)\ 1 = [x \mapsto 1](y+y)+x$

$[x \mapsto 1](y+y)+x = [x \mapsto 1](y+y)+[x \mapsto 1]x$

$[x \mapsto 1]x = 1$
$[x \mapsto 1](y+y) = [x \mapsto 1]y + [x \mapsto 1]y$
$[x \mapsto 1]y = y$



Categorical E-graphs

What we do

Both challenges can be addressed by studying e-graphs from a categorical perspective

We view e-graphs as morphisms in a particular category induced by the signature

Categorical approach has clear notions of variables and binding

Helps to justify certain congruences stipulated by e-graphs

A concrete representation of e-graphs can be obtained by considering string diagrams for the corresponding category

Categorical semantics [Tiurin et al. 2025]

To capture the notion of equivalence between (sub)terms we freely enrich a given signature-generated category over a category of semilattices

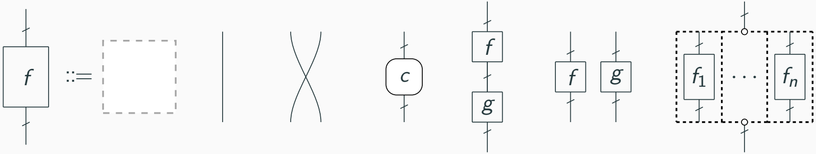
Semilattices are sets with an idempotent commutative associative binary operation $+$

This means we have the following properties in the category

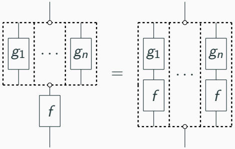
- $f + f = f$
- $f + g = g + f$
- $(f + g) \fatsemi h = (f \fatsemi h) + (g \fatsemi h)$
- $f \fatsemi (g + h) = (f \fatsemi g) + (f \fatsemi h)$
- $f \otimes (g + h) = (f \otimes g) + (f \otimes h)$
- $(f + g) \otimes h = (f \otimes h) + (g \otimes h)$

Or, string diagrammatically

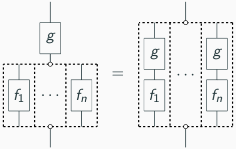
$$f \quad ::= \quad \mathrm{id}_I \qquad \mathrm{id}_1 \quad \mathrm{sym}_{1,1} \quad c \in \Sigma \quad f;g \qquad f \otimes g \qquad f_1 + \ldots + f_n$$



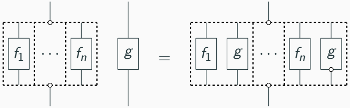
Or, string diagrammatically



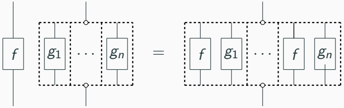
$f ⨟ (g_1 \oplus \ldots \oplus g_n) = f ⨟ g_1 \oplus \ldots \oplus f ⨟ g_n$



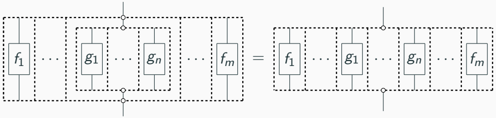
$(f_1 \oplus \ldots \oplus f_n) ⨟ g = f_1 ⨟ g \oplus \ldots \oplus f_n ⨟ g$



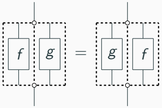
$(f_1 \oplus \ldots \oplus f_n) \otimes g = (f_1 \otimes g) \oplus \ldots \oplus (f_n \otimes g)$



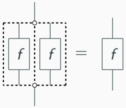
$f \otimes (g_1 \oplus \ldots \oplus g_n) = f_1 \otimes g \oplus \ldots f_n \otimes \oplus g$



$\oplus$ is an n-ary associative operation

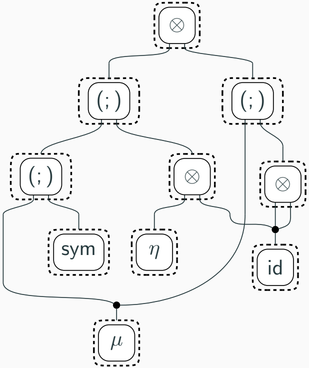


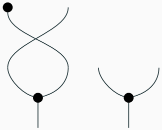
$f \oplus g = g \oplus f$



$f \oplus f = f$

Example



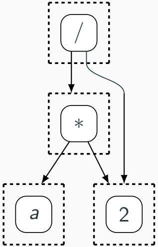


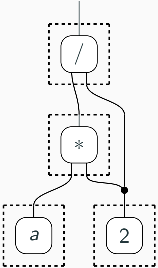
A concrete representation can literally absorb all the non-domain-specific equations (and some domain-specific ones) [Bonchi et al. 2016]

Extend this representation to support string diagrams with dashed boxes [Tiurin et al. 2025]

Traditional e-graphs are recovered as string diagrams for cartesian semilattice-enriched categories

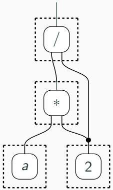
Example $((a * 2)/2)$



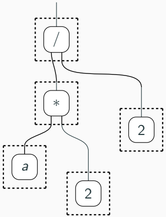


Example

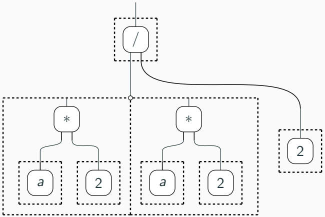
(1)



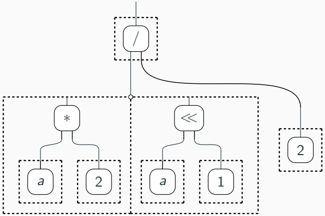
(2)



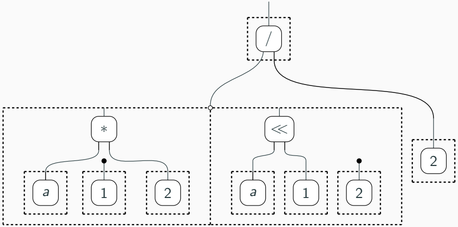
(4)



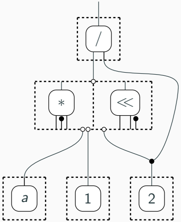
(4)



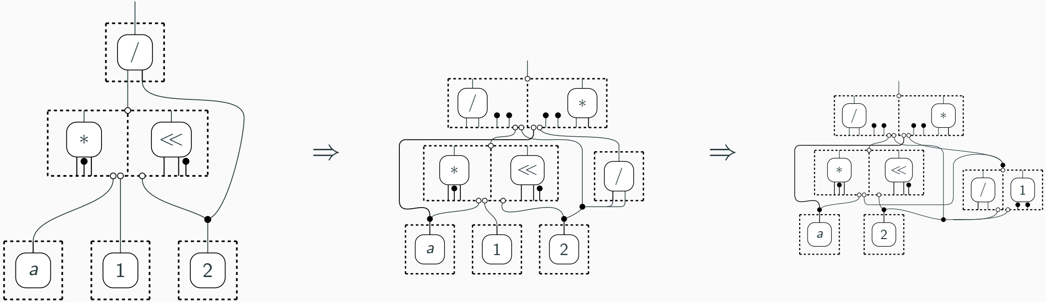
(5)



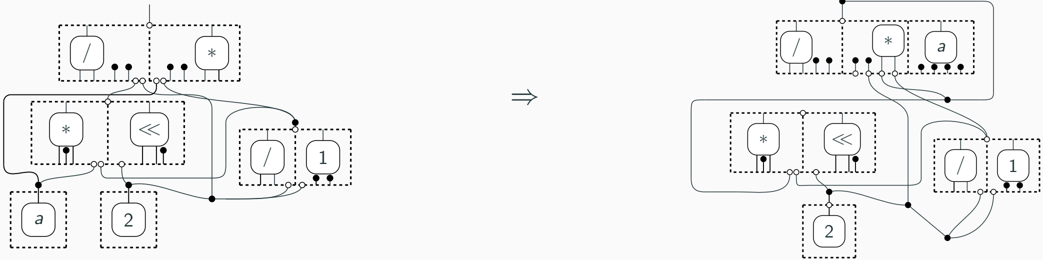
(6)



Example



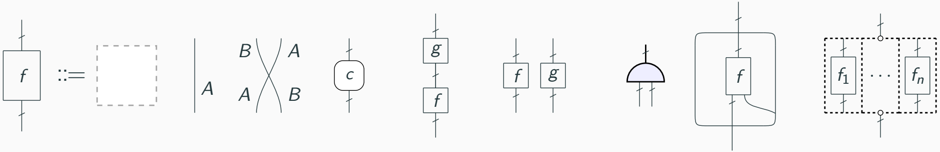
Example



This work

Semilattice-enriched closed symmetric monoidal categories

$$f \quad ::= \quad \mathrm{id}_I \quad \mathrm{id}_A \quad \mathrm{sym}_{A,B} \quad c \in \Sigma \quad f \mathbin{⨟} g \quad f \otimes g \quad \mathrm{ev} \quad \Lambda(f) \quad f_1 + \ldots + f_n$$



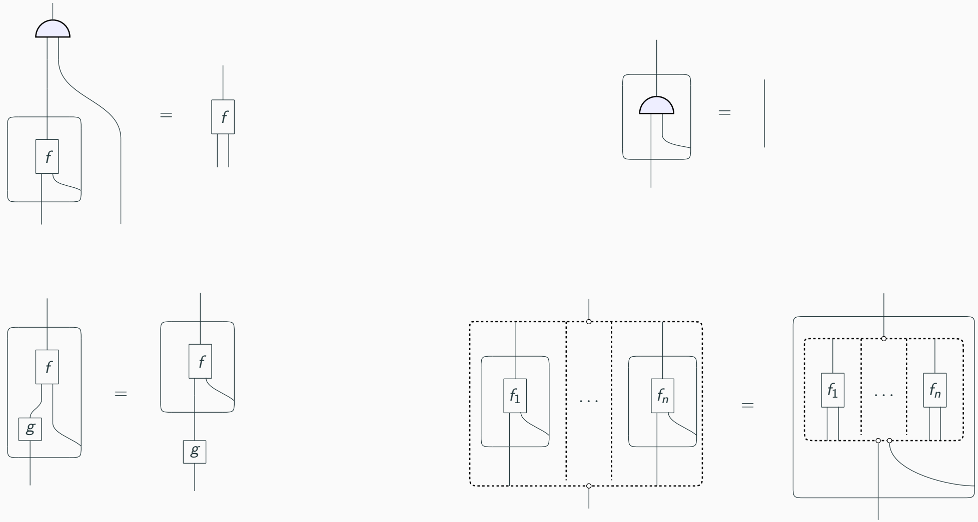
$$f = \Lambda_{A,B,C}(f) \otimes \mathrm{id}_B \mathbin{⨟} \mathbf{ev}_{B,C}$$

$$\mathrm{id}_{B \multimap C} = \Lambda_{B \multimap C,B,C}(\mathbf{ev}_{B,C})$$

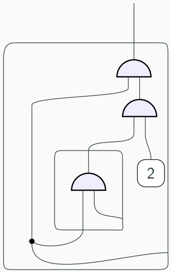
$$\Lambda_{Z,B,C}(g \otimes \mathrm{id}_B \mathbin{⨟} f) = g \mathbin{⨟} \Lambda_{A,B,C}(f)$$

$$\Lambda_{A,B,C}(f + g) = \Lambda_{A,B,C}(f) + \Lambda_{A,B,C}(g)$$

Or, string diagrammatically



$\lambda f.f\,((\lambda x.f\,x)\;2)$



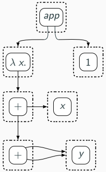
Invariant under α-equivalence

Invariant under non-interacting substitutions

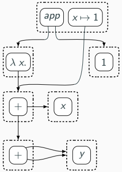
Substitution is composition

β-reduction is diagram rewiring

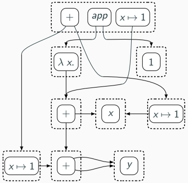
$(\lambda\, x.(y+y)+x)\ 1$



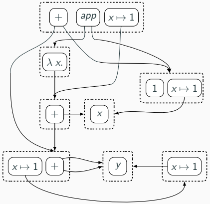
$((y+y)+x)\,[x/1]$

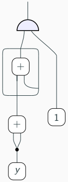


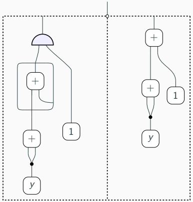
$(y+y)[x/1]+x[x/1]$

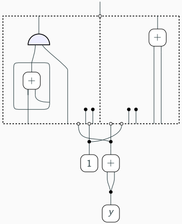


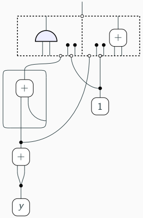
$x[x/1] = 1$
$y[x/1] + y[x/1]$
$y[x/1] = y$



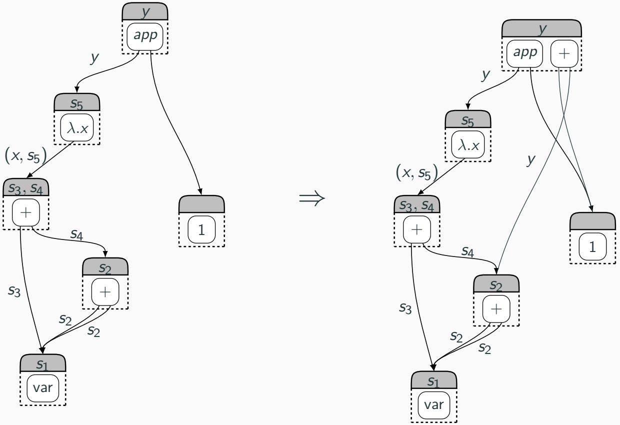








Slotted e-graphs [Schneider, Koehler, and Steuwer 2025]



Enrichment helps to justify certain equalities

$$\begin{aligned}
\Lambda_{A,B,C}(f+g) &= \eta_A; (B \multimap (f+g)) \\
&= \eta_A; (B \multimap f + B \multimap g) \\
&= \eta_A; (B \multimap f) + \eta_A; (B \multimap g) \\
&= \Lambda_{A,B,C}(f) + \Lambda_{A,B,C}(g).
\end{aligned}$$

That appeared in slotted e-graphs

$$\frac{E \vdash tc \cong tc'}{E \vdash \text{bind } \$x\ tc \cong \text{bind } \$x\ tc'}\ \textsc{cong. (bind)},$$

Lambda calculus with explicit substitution aka let [Accattoli et al. 2014]

$$t \coloneqq x \mid tt \mid \lambda x.t \mid t[x/t]$$

$$\begin{aligned}
(\lambda x.t)Lu &\to_{d\beta} t[x/u]L, && \\
C[\![x]\!][x/u] &\to_{ls} C[\![u]\!][x/u], && \\
t[x/u] &\to_{gc} t, && x \notin \mathsf{fv}(t),
\end{aligned}$$

$$\begin{aligned}
t[x/u][y/v] &\sim t[y/v][x/u], && x \notin \mathsf{fv}(v) \wedge y \notin \mathsf{fv}(u), \\
(\lambda y.t)[x/u] &\sim \lambda y.(t[x/u]), && y \notin \mathsf{fv}(u), \qquad (*) \\
(tv)[x/u] &\sim t[x/u]v, && x \notin \mathsf{fv}(v),
\end{aligned}$$

Set of equations	$d\beta + ls + gc$	$d\beta + ls + gc$ $+$ `lambda_subst`	$d\beta + ls + gc$ $+$ `lambda_subst` $+$ `subst_comm`	$d\beta + ls + gc$ $+$ `lambda_subst` $+$ `subst_comm` $+$ `subst_app`
# of e-nodes	88	185	244	846

Rewrite rule	# of applications ($d\beta$ + ls + gc)	# of applications ($d\beta$ + ls + gc + `lambda_subst`)	# of applications ($d\beta$ + ls + gc + `lambda_subst` + `subst_comm`)	# of applications ($d\beta$ + ls + gc + `lambda_subst` + `subst_comm` + `subst_app`)	Absorbed?
`beta`	14	57	49	80	No
`subst_no_var`	35	113	118	197	No
`ls_rule_*`	48	141	140	302	Yes
`subst_lambda`	—	76	109	336	No
`subst_comm`	—	—	104	321	Yes
`subst_app`	—	—	—	263	Yes

Conclusion

We extend the categorical semantics of e-graphs to support binding, moving from semilattice-enriched symmetric monoidal to semillattice-enriched symmetric monoidal closed categories

A single string diagram encodes both binding (closed structure) and equivalence of subterms (semilattice enrichment)

Closed e-hypergraphs give these diagrams a combinatorial representation, with DPOI rewriting shown sound and complete w.r.t. term rewriting modulo SMC equations

Binding comes for free: no explicit-substitution nodes or de Bruijn shifting — β-reduction is just re-wiring, and α-equivalence built in

These *categorical e-graphs* have clear advantages over slotted e-graphs in some scenarios and built from a semantics-first approach

References

Accattoli, Beniamino et al. (2014). **"A Nonstandard Standardization Theorem".** In: *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. Popl '14. San Diego, California, USA: Association for Computing Machinery, pp. 659–670. ISBN: 978-1-4503-2544-8. DOI: 10.1145/2535838.2535886. URL: https://doi.org/10.1145/2535838.2535886.

Bonchi, Filippo et al. (2016). **"Rewriting modulo symmetric monoidal structure".** In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS '16. New York, NY, USA: Association for Computing Machinery, pp. 710–719. ISBN: 9781450343916. DOI: 10.1145/2933575.2935316. URL: https://doi.org/10.1145/2933575.2935316.

References ii

- Gonzalez, Emmanuel Anaya, Aditya Giridharan, and Cole Kurashige (2023). ***Optimizing Beta-Reduction in E-Graphs (EGRAPHS 2023 - E-Graph Research, Applications, Practices, and Human-factors Symposium) - PLDI 2023.*** URL: https://pldi23.sigplan.org/details/egraphs-2023-papers/12/Optimizing-Beta-Reduction-in-E-Graphs (visited on 07/17/2023).
- Schneider, Rudi, Thomas Koehler, and Michel Steuwer (2025). ***Slotted E-Graphs — First-Class Support for (Bound) Variables in E-Graphs.*** Accepted to PLDI 2025. URL: https://michel.steuwer.info/files/publications/2025/PLDI-2025.pdf.
- Tiurin, Aleksei et al. (2025). **"Equivalence Hypergraphs: DPO Rewriting for Monoidal E-Graphs".** In: *2025 40th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pp. 209–222. DOI: 10.1109/LICS65433.2025.00023.

Willsey, Max et al. (Jan. 2021). **“Egg: Fast and Extensible Equality Saturation”.** In: *Proc. ACM Program. Lang.* 5.POPL. DOI: 10.1145/3434304. URL: https://doi.org/10.1145/3434304.