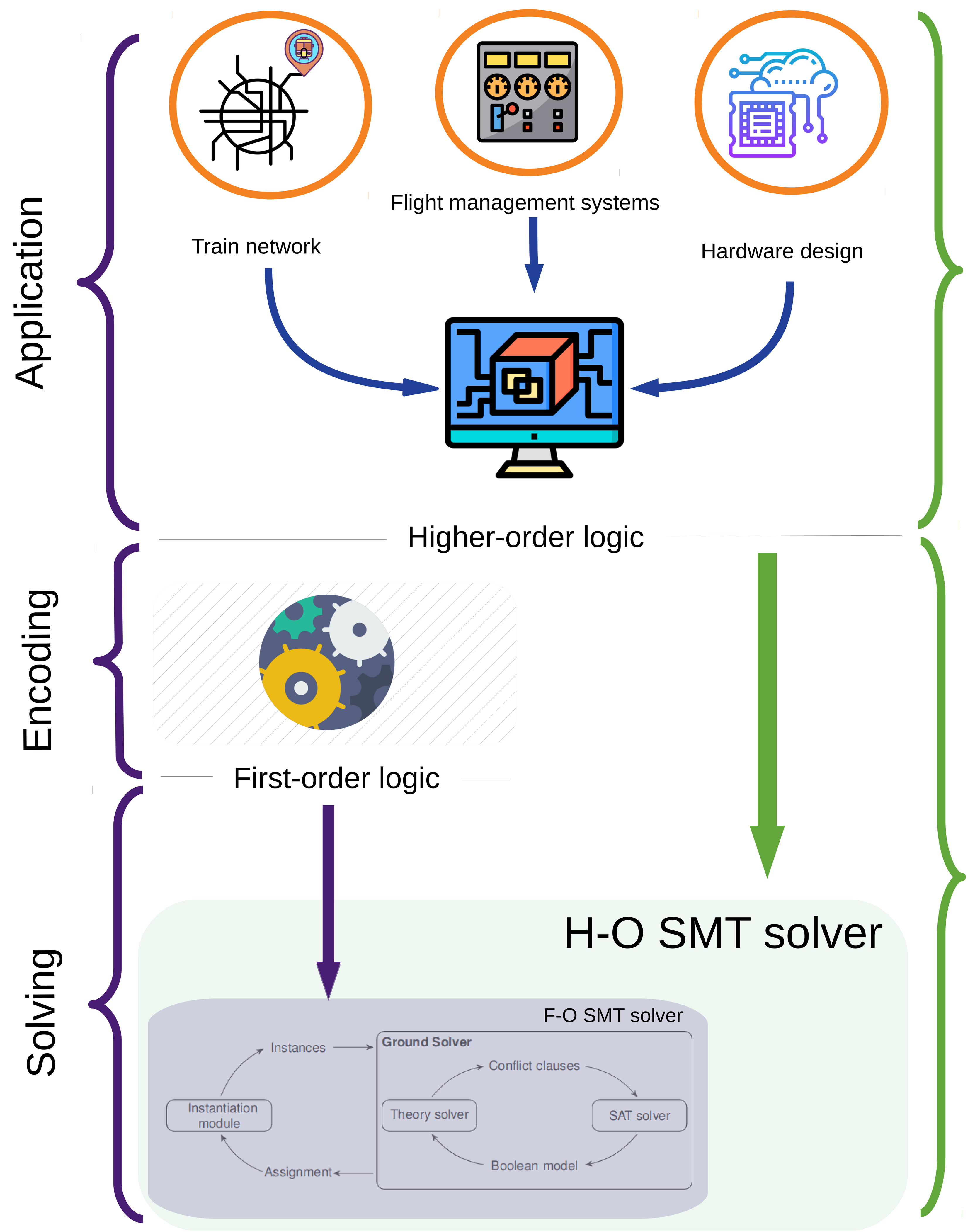
Extending SMT Solvers to Higher-Order Logic

Haniel Barbosa, Andrew Reynolds, **Daniel El Ouraoui**, Cesare Tinelli, Clark Barrett
The University of Iowa, Iowa City, USA
University of Lorraine, CNRS, Inria, and LORIA, Nancy, France
Stanford University, USA



Applicative encoding

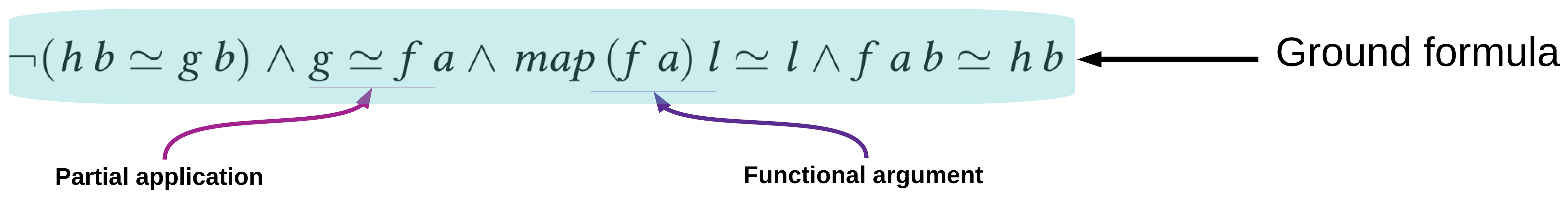
- The notation **f(a,b)** denotes the application of the function symbol **f** of arity 2 to the arguments **a** and **b** (F-O syntax)
- The notation **f a b** should be read as **(f a) b** denoting that **(f a)** is a function applied to **b** and similarly with **f** to **a** (H-O syntax)
- The operator @ simulate the notation **f a b** at the F-O level

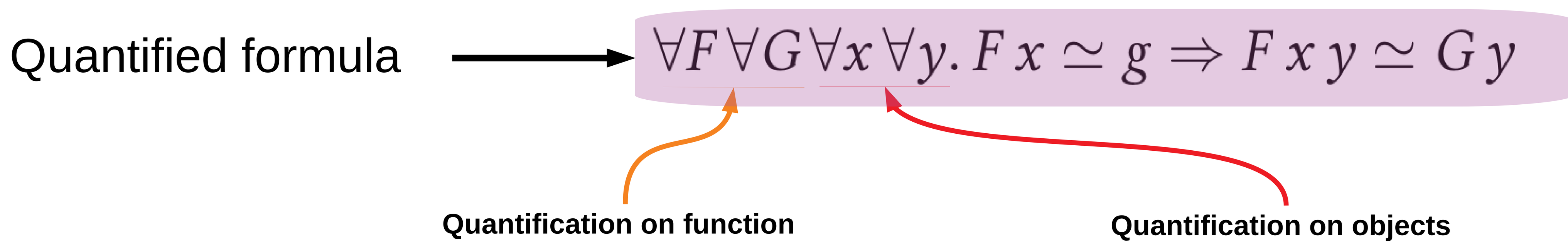
$$f\,a\,b \simeq b \wedge f\,a\,(f\,a\,b) \simeq g\,b$$

Translation

$$@(@(f,a),b) \simeq b \wedge @(@(f,a),\,@(@(f,a),b)) \simeq @(g,b)$$

Higher-order Logic





Ground solver

$$\frac{t \in \mathbf{T}^{\mathrm{curr}}(E)}{t \simeq t \in E_{\mathrm{curr}}}\ \textsc{Refl}_{\textsc{curr}} \qquad \frac{t \simeq u}{u \simeq t \in E_{\mathrm{curr}}}\ \textsc{Sym} \qquad \frac{s \simeq t,\ t \simeq u}{s \simeq u \in E_{\mathrm{curr}}}\ \textsc{Trans}$$

$$\frac{t \simeq u \quad t' \simeq u' \quad (t,\,t'),(u,\,u') \in \mathbf{T}^{\mathrm{curr}}(E)}{(t,\,t') \simeq (u,\,u') \in E_{\mathrm{curr}}}\ \textsc{Cong}_{\textsc{curr}} \qquad \frac{t \simeq u,\ t \not\simeq u}{\bot}\ \textsc{Conflict}$$

$$\frac{t \simeq u \qquad \texttt{uncurry}(t) \text{ or } \texttt{uncurry}(u) \text{ is shared with theory } \mathcal{T}}{\texttt{uncurry}(t) \simeq \texttt{uncurry}(u) \in E}\ \textsc{Theory-prop}$$

$$\frac{t,\,u \in \mathbf{T}^{\mathrm{curr}}(E) \qquad t,\,u : \bar{\tau}_n \to \tau,\ n > 0}{\forall F, G : \bar{\tau}_n \to \tau.\ F \not\simeq G \Rightarrow F(\mathsf{sk}_1, \ldots, \mathsf{sk}_n) \not\simeq G(\mathsf{sk}_1, \ldots, \mathsf{sk}_n) \in Q}\ \textsc{Ext-Ax}$$

where $\mathsf{sk}_1, \ldots, \mathsf{sk}_n$ are fresh symbols of respective sorts $\tau_1, \ldots, \tau_n$.

E-matching

$$e(x,t,S) = \{\sigma \cup \{x \mapsto t\} \mid \sigma \in S,\ s \notin dom(\sigma))\} \cup \{\sigma \mid \sigma \in S,\ E \models x\sigma \simeq t\}$$

$$e(t',t,S) = \begin{cases} S \text{ if } E \models t' \simeq t \\ \varnothing \text{ otherwise} \end{cases}$$

$$e(\mathsf{a}(\overline{s_n}),t,S) = \bigcup_{\substack{\mathsf{a}(\overline{t_n}) \in \mathbf{T}^{\mathrm{uncurry}}(E_{\mathrm{curr}}) \\ E \models \mathsf{a}(\overline{t_n}) \simeq t}} e(s_n,t_n,\ldots,e(s_1,t_1,S)\ldots)$$

$$e(F(\overline{s_n}),t,S) = \bigcup_{\substack{\mathtt{Ty}(F) = \mathtt{Ty}(\mathsf{a}) \\ \mathsf{a}(\overline{t_n}) \in \mathbf{T}^{\mathrm{uncurry}}(E_{\mathrm{curr}}) \\ E \models \mathsf{a}(\overline{t_n}) \simeq t}} e(s_n,t_n,\ldots,e(s_1,t_1,S)\ldots) \cup \{F \mapsto \mathsf{a}\}$$

Features comparison

Features	First-order	Λ-free Higher-order logic	Higher-order logic
Functions	✓	✓	✓
Predicates	✓	✓	✓
Functional arguments	✗	✓	✓
Quantification on objects	✓	✓	✓
Quantification on functions	✗	✓	✓
Partial applications	✗	✓	✓
Anonymous functions	✗	✗	✓

Results

- @vt is veriT using the applicative encoding
- vt is veriT using the native higher-order approach

- THO is λ-free and without first-class Boolean
- oTH0 is only λ-free
- λoTH0 is full higher-order
- JD* are Sledghammer problems with respectively 32 and 512 axioms and whether λ-abstractions are removed via λ-lifting (lift) or via SK-style combinators (combs).
- λoSH is Sledghammer problems with 1024 axioms and full higher-order

Solver	Total	TH0	oTH0	λoTH0	$\mathrm{JD}^{32}_{\mathrm{lift}}$	$\mathrm{JD}^{32}_{\mathrm{combs}}$	$\mathrm{JD}^{512}_{\mathrm{lift}}$	$\mathrm{JD}^{512}_{\mathrm{combs}}$	$\lambda o\mathrm{SH}^{1024}$
#	9032	530	743	1915	1253	1253	1253	1253	832
@cvc	4318	384	344	940	457	459	655	667	**412**
@cvc-sax	4348	390	373	937	456	457	655	**668**	**412**
cvc	4232	389	342	865	463	447	**667**	654	405
cvc-sax	4275	389	376	883	458	443	**667**	654	405
@vt	2556	370	332		404	396	525	529	
vt	2671	369	346		426	424	550	556	
Ehoh	2631	394			489	481	637	630	
Leo-III	**4410**	**402**	452	1178	**491**	**482**	609	565	231
Satallax	3961	392	**457**	**1215**	394	390	407	404	302

Number of proved theorems per benchmark set. Best results are inbold.

Credits images :
Icon « wrong » made by xnimrodx from www.flaticon.com
Icon « check » made by smashicons from www.flaticon.com
Icon « control panel » by itim2101 from www.flaticon.com
Icon « microchip » made by eucalyp from www.flaticon.com
Icon « subway map » made by freepik from www.flaticon.com
Icon « setting » made by smashicons from www.flaticon.com
Icon «train» made by icongeek26 from www.flaticon.com

- Haniel Barbosa, Andrew Reynolds, Daniel El Ouraoui, Cesare Tinelli, and Clark Barrett. Extending smt solvers to higher-order logic. In Pascal Fontaine, editor, Automated Deduction – CADE 27, pages 35–54, Cham, 2019. Springer International Publishing.
- Haniel Barbosa, Jasmin Christian Blanchette, Simon Cruanes, Daniel El Ouraoui, and Pascal Fontaine. Language and proofs for higher-order SMT (work in progress). volume 262 of EPTCS, pages 15–22, 2017.
- Haniel Barbosa, Andrew Reynolds, Pascal Fontaine, Daniel El Ouraoui, and Cesare Tinelli. Higher-order smt solving (work in progress). In 16th International Workshop on Satisfiability Modulo Theories 2018.
- Haniel Barbosa, Andrew Reynolds, Daniel El Ouraoui, Cesare Tinelli, and Clark Barrett. Extending SMT solvers to higher-order logic. Technical report, The University of Iowa, May 2019.