Synthesis of Symbolic Controllers: A Parallelized and Sparsity-Aware Approach

Mahmoud Khaled, Eric S. Kim, Murat Arcak, and Majid Zamani

Professorship of Hybrid Control Systems

ETAPS 2019 : Poster session

April 8-11, 2019

Introduction to abstraction-based controller synthesis for dynamical systems

Phase 1: construction of a finite abstraction:

- Dynamical systems: physical systems modeled by differential/difference equations.

$$\Sigma : x^+ = f(x,u), \quad x \in X \subseteq R^n, \text{ and } u \in U \subseteq R^m$$

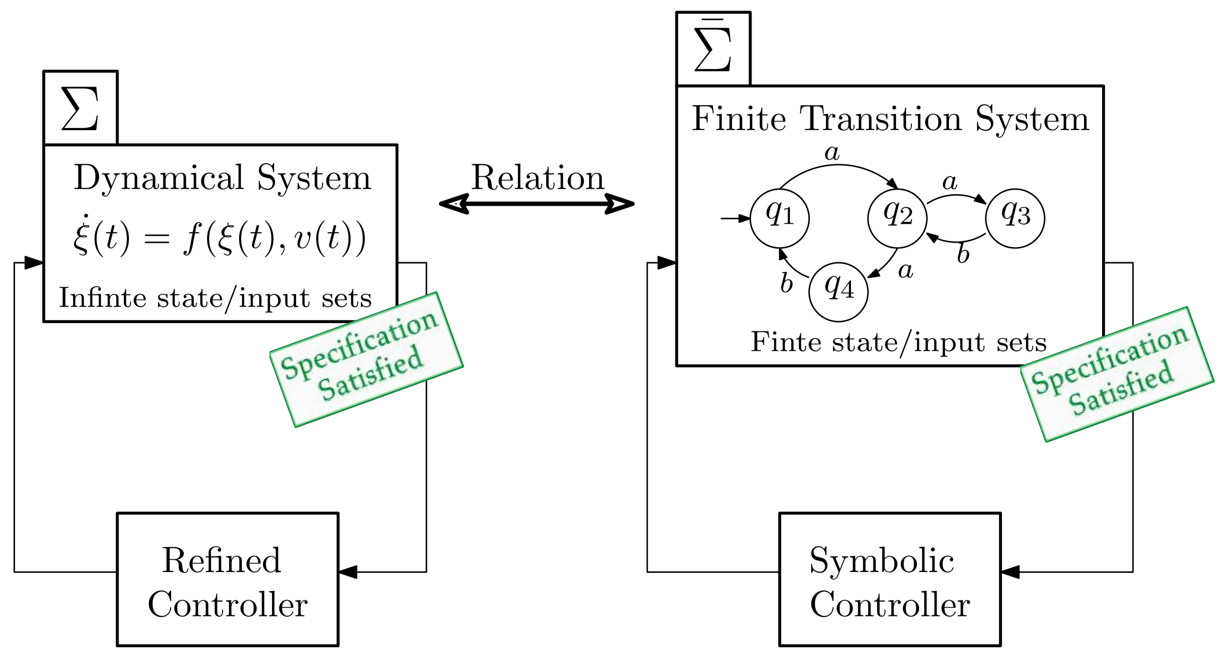
- Abstraction: finite systems to mimic original plants up to a predefined accuracy.

$$\bar{\Sigma} = (\bar{X}, \bar{U}, T), \quad \bar{X} := [X]_{\eta_x}, \bar{U} := [U]_{\eta_u}, \text{ and } T \subseteq \bar{X} \times \bar{U} \times \bar{X}$$

Phase 2: controller synthesis and refinement:

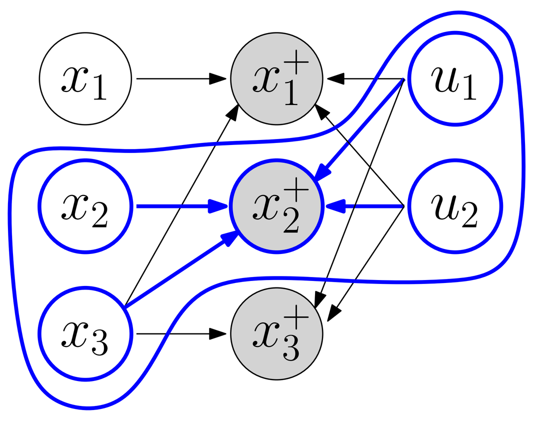
- Algorithmic synthesis: fixed-point iterations on subsets $Z_i \subseteq \bar{X} \times \bar{U}$ using update map $G(Z)$.
- Handles complex specifications: LTL specification ψ is mapped to a set $Z_\psi \subseteq \bar{X} \times \bar{U}$

Problem: Exponential time/space bottlenecks w.r.t. $m+n$.



Literature: sparsity-aware construction of $\bar{\Sigma}$

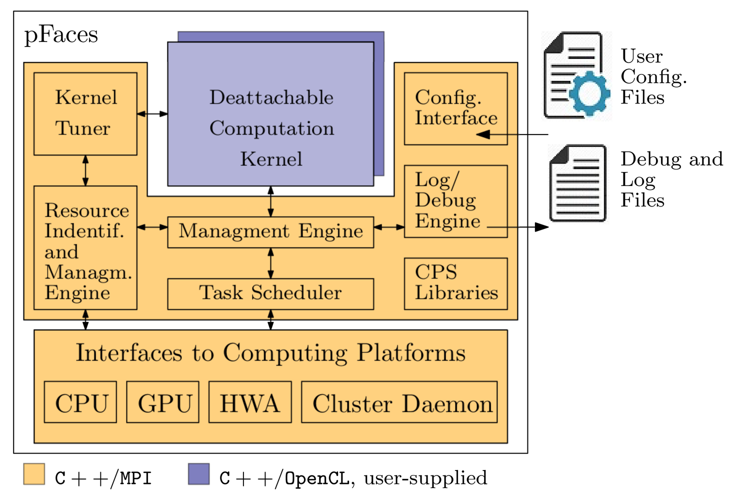
- Sparsity: density of the dependency graph.
- No need to iterate non-affecting states/inputs.
- Example: x_2^+ depends only on states x_2 and x_3. Then, no need to iterate x_1 for computing x_2^+.
- Sparsity is only utilized for constructing $\bar{\Sigma}$.
- Inefficient storage of abstraction.



F. Gruber, E. Kim, and M. Arcak. Sparsity-aware finite abstraction. CDC 2017.

Literature: parallel implementation

- pFaces: an acceleration ecosystem.
- Introduced parallel kernel:
  - constructing $\bar{\Sigma}$.
  - Fixed-point controller synthesis
- Supports Cloud/local processing elements (PEs).
- Many wasted compute-threads.



M. Khaled and M. Zamani. pFaces: An acceleration ecosystem for symbolic control. HSCC 2019.

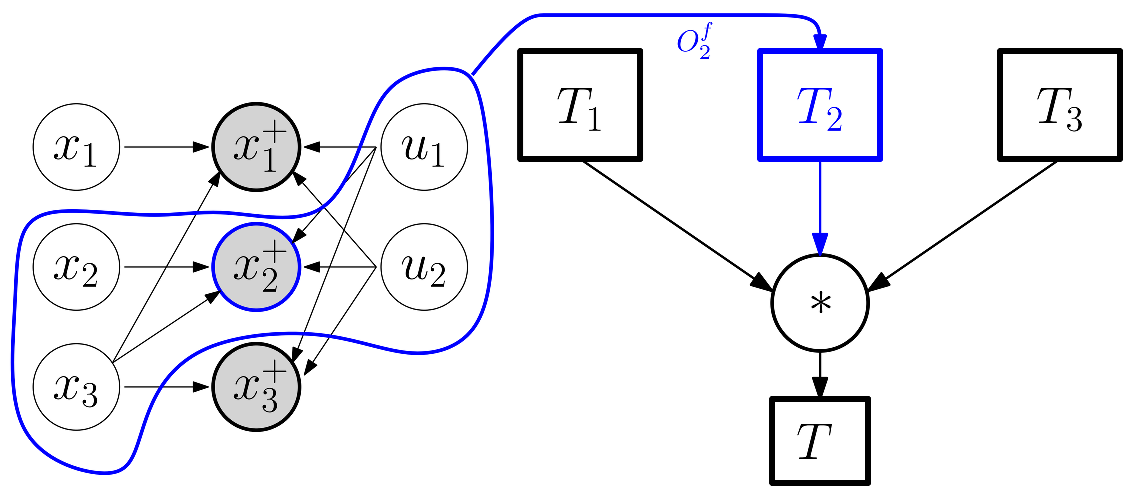
Contribution 1: Combining parallel and sparsity-aware approaches to construct $\bar{\Sigma}$

Traditional algorithm:

$T \leftarrow \emptyset$;
for all $\bar{x} \in \bar{X}$ **do**
  for all $\bar{u} \in \bar{U}$ **do**
    for all $\bar{x}' \in O^f(\bar{x}, \bar{u})$ **do**
      $T \leftarrow T \cup \{(\bar{x}, \bar{u}, \bar{x}')\}$;
    end
  end
end

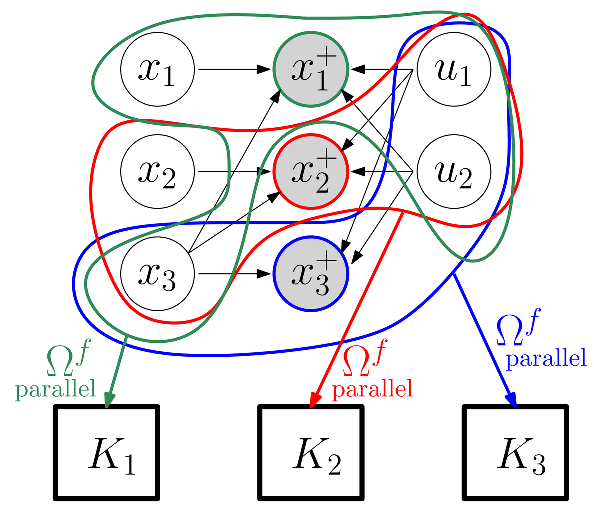
- Exhaustive iterations on (x, u).
- O^f explodes in bigger $m + n$.

Sparsity-aware approach:



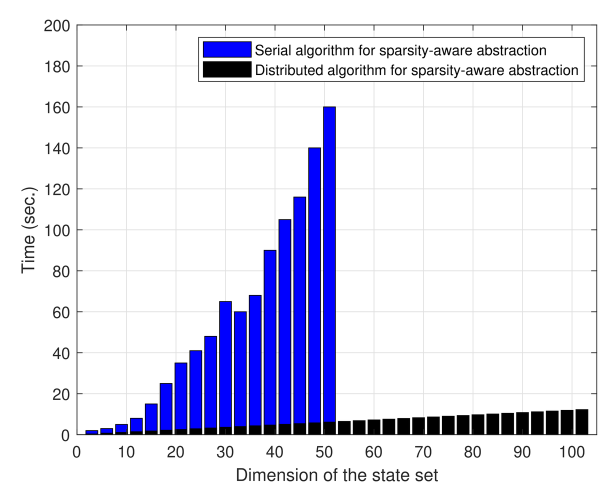
- O^f: Over-approximation of reachable sets.
- Bottleneck in combining T_i for T.

Parallel sparsity-aware:



- Ω^f: Corners of reachable sets.
- K_i: Locally-stored storage.

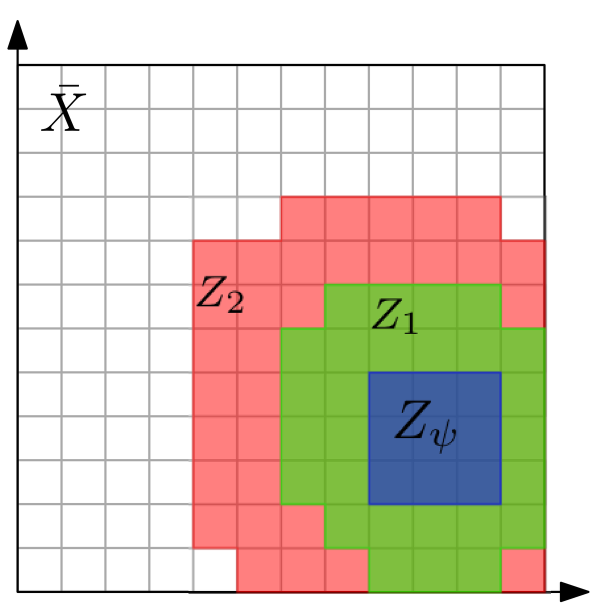
Comparison:



- Traffic network (3D = intersection).
- Noticeable speedup !

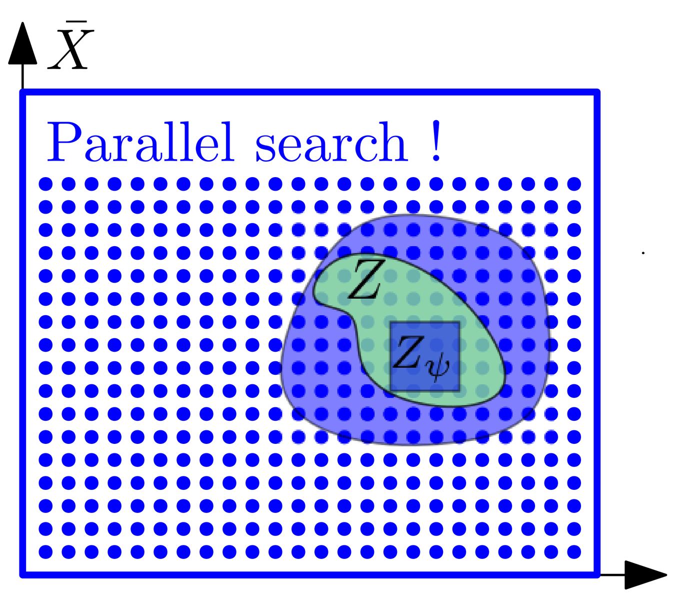
Contribution 2: A novel parallel approach that utilizes sparsity in the synthesis of controllers

Traditional approach:



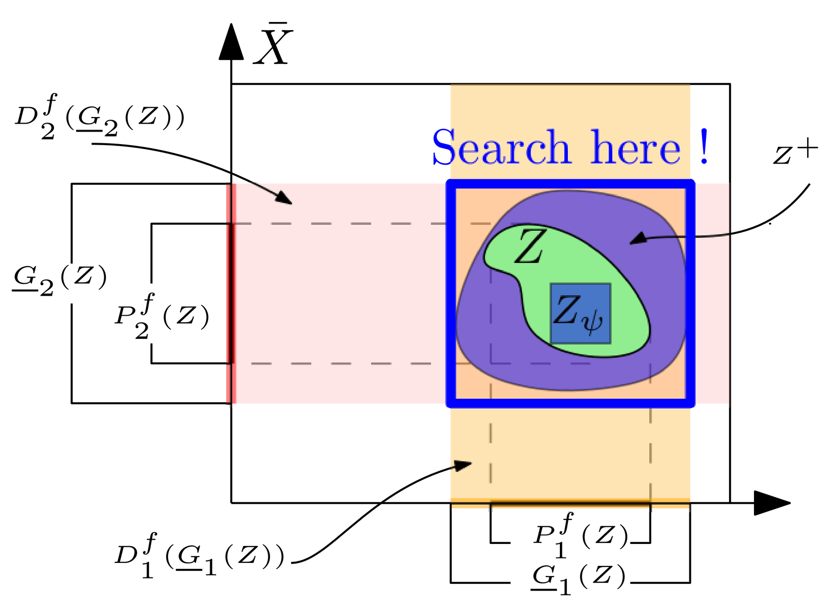
- New Z: Iterative/symb. search.
- Settles when $Z_{k-1} = Z_k$!

Parallel approach:



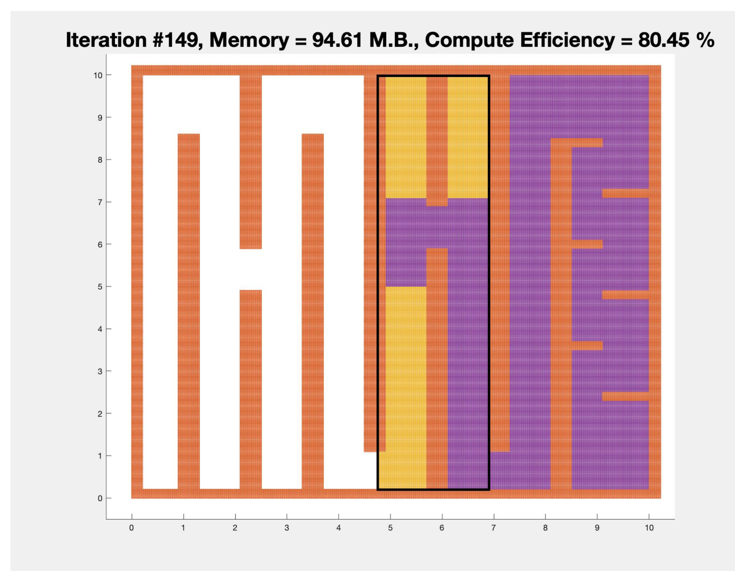
- Massively parallel.
- Wasted computations.

Parallel sparsity-aware:



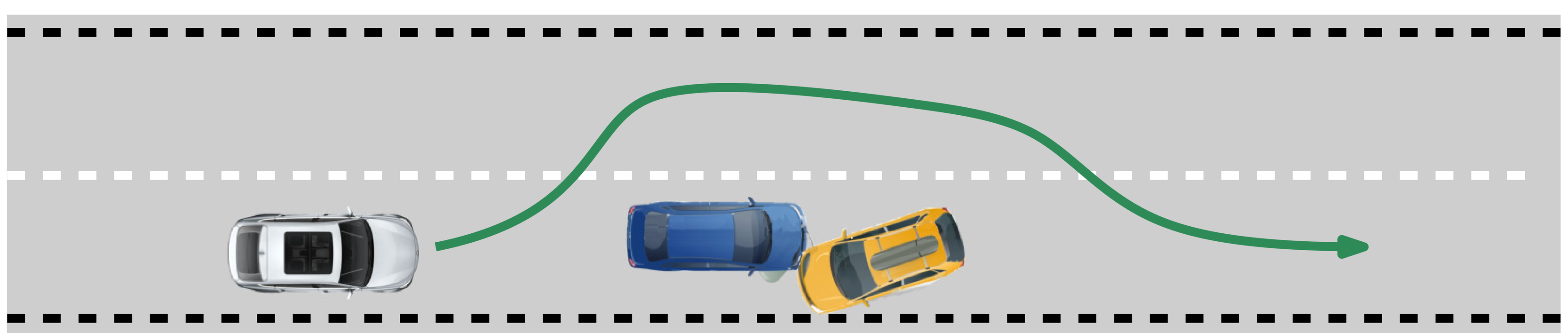
- P_i^f: sparsity-aware projection map.
- D_i^f: sparsity-aware recovery map.

Example:



- Robot maze: reach/avoid.
- Avg. compute efficiency: 80%.

Case study: A 7D BMW 320i car avoids autonomously blocks on highway



Used HW configuration	PEs	EX_1 pFaces	EX_1 this	EX_2 pFaces	EX_2 this
Local machine: Intel Xeon E5-1620	8	–	–	24 H	8.7 H
AWS `p3.16xlarge`: Intel Xeon E5-2686	64	2.1 H	0.5 H	8.1 H	3.2 H
AWS `c5.18xlarge`: Intel Xeon P 8000	72	1.9 H	0.4 H	–	–