Comparing SMT and Canonical Set Representation for Analyzing Network Policies

Shai Guendelman

What will be in this lecture?

- Seminar, not a lecture!
- NP-Guard – static analysis tool for Kubernetes
- Comparing SMT and Canonical Representation

Background

Kubernetes

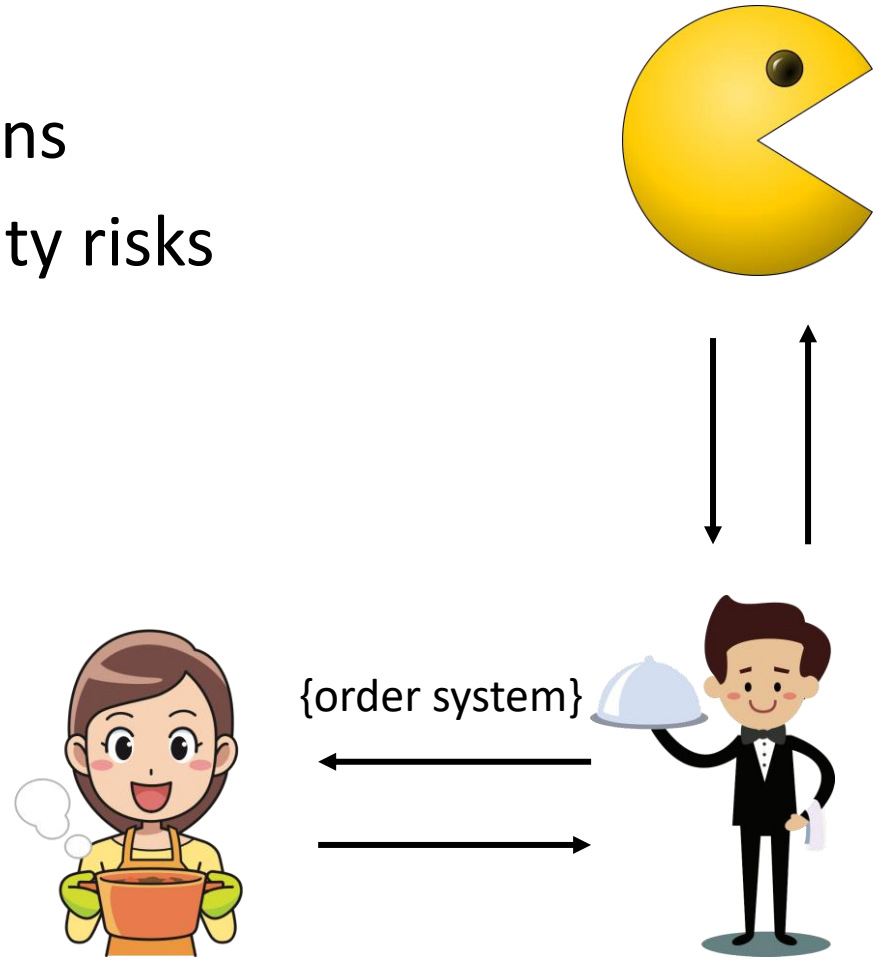
- Micro-services vs. Monolith
- Kubernetes supports micro-services developers
- Used in many projects. For example – Spotify, booking.com, CERN

Kubernetes – Restaurant Metaphor

- Different logical components
- Simplicity
- Useful automations
- For example –
  - Auto-scaling
  - Self-healing

Network Policies

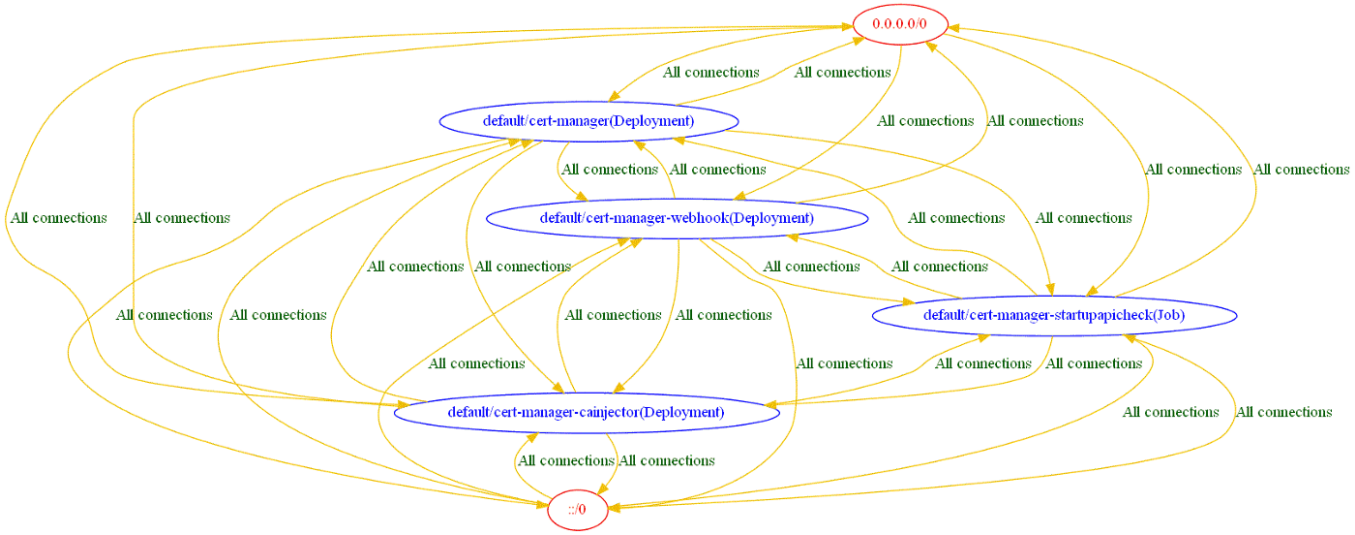
- Limit internal and external communications
- Open communication channels are security risks
- Communications graph
  - Nodes: pods & outside entities
  - Edges: allowed communications
  - Edge labels: conditions



Network Policies – Example

Network Policies – Difficulties

- Lack of awareness
  - Connectivity map
  - CI/CD integration
- Easy to do mistakes
  - Change Impact Analysis

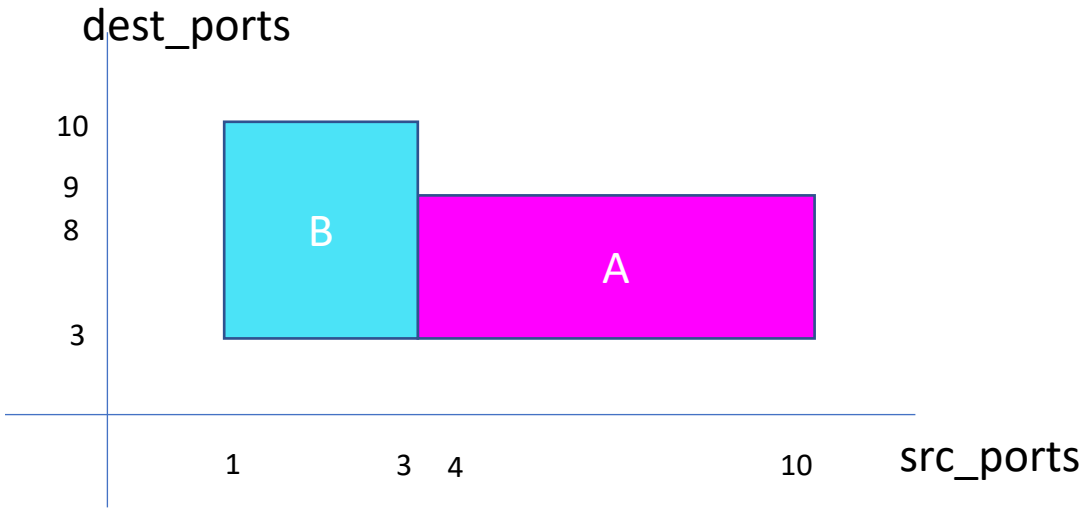



```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: paymentservice-netpol-with-typo
  namespace: default
spec:
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: checkoutservice
  - ports:  # the typo is here
    - port: 50051
      protocol: TCP
  podSelector:
    matchLabels:
      app: paymentservice
```

```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: paymentservice-netpol
  namespace: default
spec:
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: checkoutservice
    ports:
    - port: 50051
      protocol: TCP
  podSelector:
    matchLabels:
      app: paymentservice
```

Hyper-Cube Set

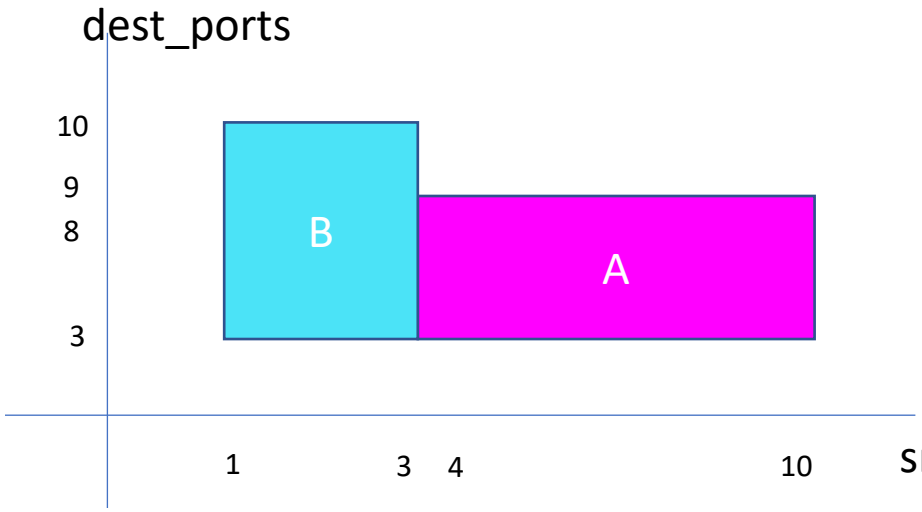
- Connectivity can be modelled as a set of hyper-cubes
- Dimensions are either **Strings** or **Integers**
- Analysis reduces to set operations
  - Change impact analysis ➔ set difference

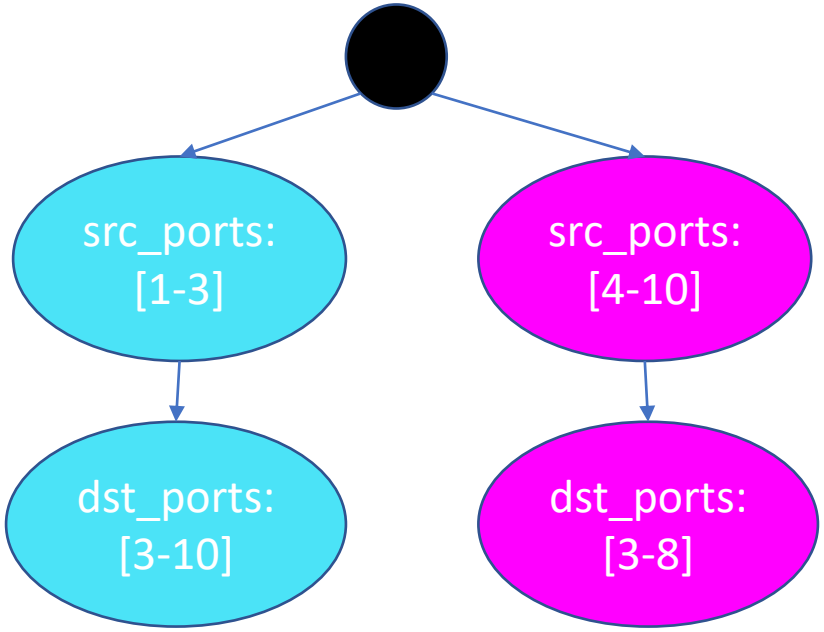


Questions?

Canonical Representation

- Efficient equality check
- Supports set operations – union, intersection, subtraction
- Integers –
  - Interval Sets: {[1-10], [100-200], [432-543]}
- Strings –
  - Minimal DFAs
- Hyper-Cubes –
  - Tree





SMT (**S**atisfiability **M**odulo **T**heory) Solver

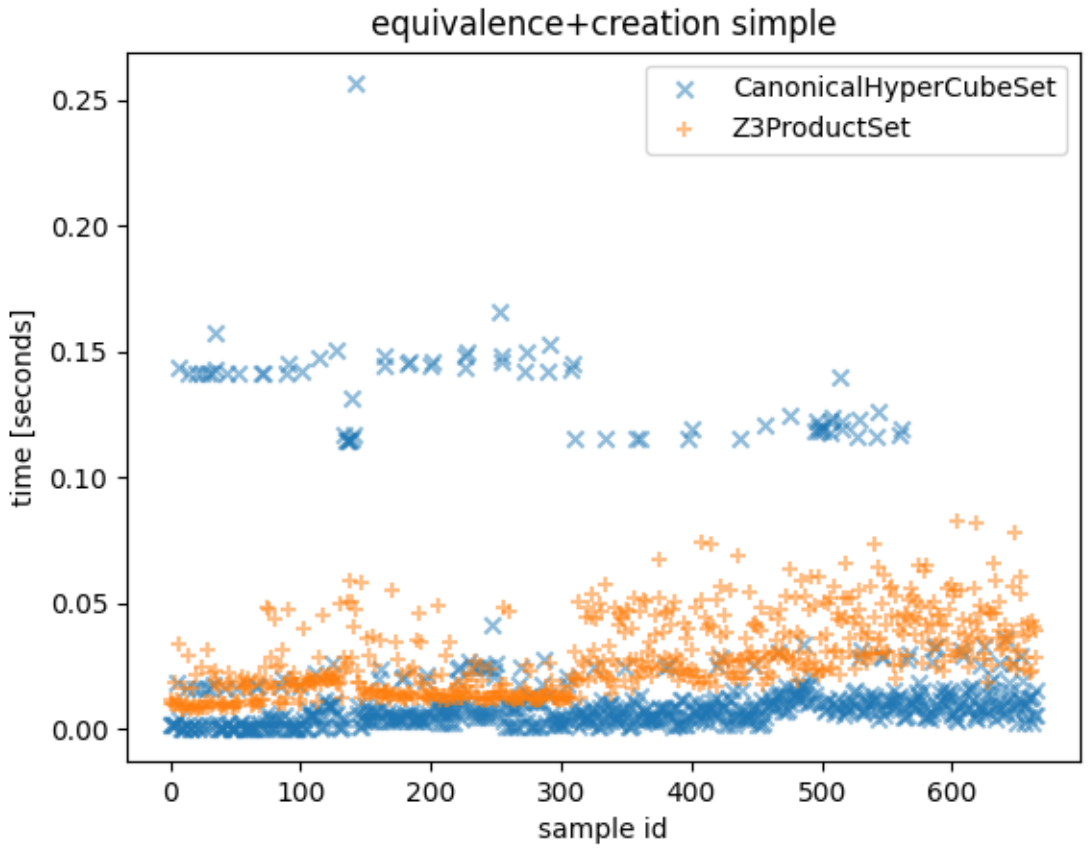
- Is formula ψ satisfiable?
- Integers:
  - $0 \leq x \wedge x \leq 10 \mapsto \text{sat}\ (x = 0)$
  - $7 \leq x \wedge x \leq 6 \mapsto \text{unsat}$
- Strings:
  - $\text{PrefixOf}(\text{"www"}, s) \wedge \text{SuffixOf}(\text{".}com\text{"}, s) \mapsto \text{sat}\ (s = \text{"}www.com\text{"})$
  - $\text{PrefixOf}(\text{"}abc\text{"}, s) \wedge \text{PrefixOf}(\text{"}xyz\text{"}, s) \mapsto \text{unsat}$

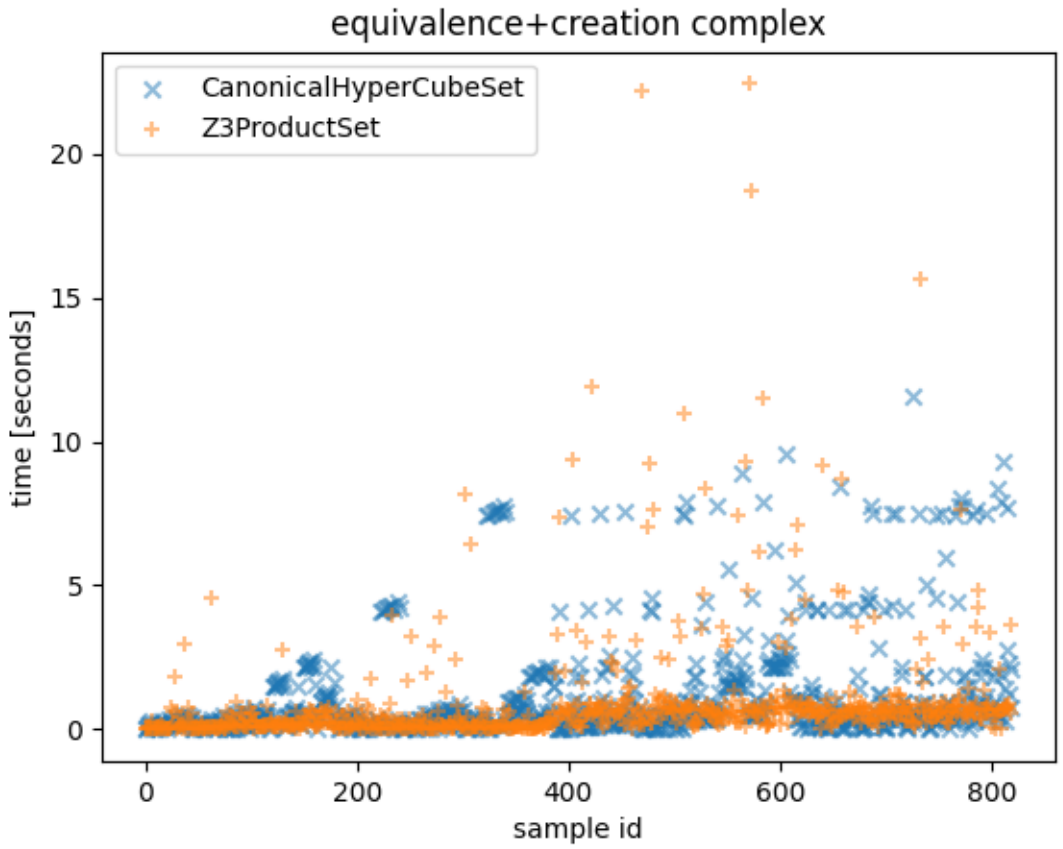
SMT (**S**atisfiability **M**odulo **T**heory) Solver

- Single hyper-cube: $C_1 = (port \leq 20 \wedge port \geq 10) \wedge (\text{PrefixOf}(\text{"}us/\text{"}, host))$
- Hyper-cube set: $\text{A} = C_1 \vee C_2 \vee \cdots \vee C_n$
- Can only check sat / unsat
- $A == B \mapsto (A \wedge \neg B) \vee (\neg A \wedge B)$ is unsat
- No canonical representation

Results

Comparison on Realistic Examples

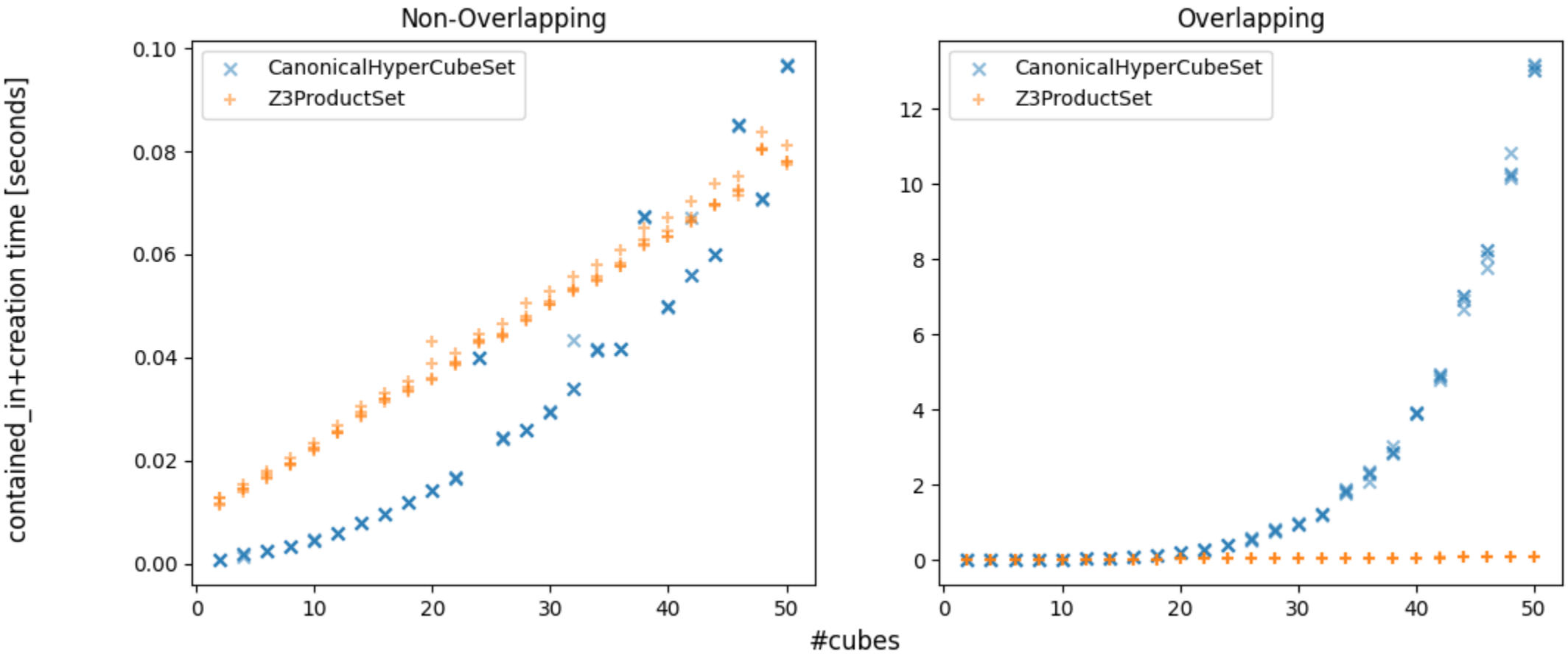




Comparison on Realistic Examples

operation	mode	#samples	#z3 wins	z3 wins (%)	total time		max time		max advantage	
					z3	canonical	z3	canonical	z3	canonical
contained_in	simple	1332	316	23.724	31.74	35.873	0.085	0.545	0.504	0.073
contained_in	complex	1640	1063	64.817	1610.665	2029.715	40.11	11.658	11.376	32.217
emptiness	simple	37	3	8.108	0.744	0.409	0.05	0.141	0.129	0.047
emptiness	complex	41	23	56.098	16.974	24.29	7.847	7.445	3.974	0.505
equivalence	simple	666	114	17.117	18.793	14.721	0.083	0.256	0.208	0.079
equivalence	complex	820	517	63.049	705.715	971.611	22.419	11.562	11.328	22.149

Overlapping / Non-Overlapping Cubes



Summary

- Canonical Representation Advantages:
  - Z3 unable to deal with general regular expressions
  - Can be used for connectivity map
  - Better at small common examples
- Z3 advantages:
  - Simple if we do simple things
  - Scales better
  - May be optimized?

Thanks!

Questions?