Neural Network Action Policy Verification via Predicate Abstraction

Marcel Vinzent, Jörg Hoffmann

{vinzent, hoffmann}@cs.uni-saarland.de

Saarland University

Networks of Automata

- **State variables** $\mathcal{V}$ with a bounded-integer domain, and automaton *location variables* $\mathcal{V}_{loc}$.
- **Linear integer expressions** Exp_{int} over $\mathcal{V}$, $d_1 \cdot v_1 + \cdots + d_r \cdot v_r + c$ with $d_1, \ldots, d_r, c \in \mathbb{Z}$ and $v_1, \ldots, v_r \in \mathcal{V}$.
- **Linear integer constraints** and conjunctions thereof Exp_{bool}, $e_1 \bowtie e_2$ with $e_1, e_2 \in Exp_{int}(\mathcal{V})$ and $\bowtie \in \{\leq, =, \geq\}$.

State space LTS $\Theta = \langle \mathcal{S}, \mathcal{A}, \mathcal{T} \rangle$,

- **states** $\mathcal{S} = \mathcal{S}_{\mathcal{V}_{loc}} \times \mathcal{S}_{\mathcal{V}}$, complete state variable assignments over $\mathcal{V}_{loc}$ and $\mathcal{V}$
- **action** $(g_{loc}, g, u_{loc}, u) \in \mathcal{A}$ composed of:
  - location guard g_{loc} & *location update* u_{loc} (partial variable assignments over $\mathcal{V}_{loc}$),
  - $g \in Exp_{bool}$,
  - *update* $u \subseteq \mathcal{V} \times Exp_{int}$.
- **transition** $((s_{\mathcal{V}_{loc}}, s_{\mathcal{V}}), (g_{loc}, g, u_{loc}, u), (s'_{\mathcal{V}_{loc}}, s'_{\mathcal{V}})) \in \mathcal{T}$ iff
  - $g_{loc} \subseteq s_{\mathcal{V}_{loc}}$,
  - $g(s_{\mathcal{V}})$ evaluates to true,
  - $s'_{\mathcal{V}_{loc}} = s_{\mathcal{V}_{loc}}[u_{loc}]$, and
  - $s'_{\mathcal{V}} = s_{\mathcal{V}}[u(s_{\mathcal{V}})]$.

Neural Network Action Policies

- **Action policy** $\pi : \mathcal{S} \to \mathcal{A}$, implemented by feed-forward neural networks with ReLU activation functions [Nair and Hinton (2010)].
- **Policy restriction** $\Theta^{\pi} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}^{\pi} \rangle$ with $\mathcal{T}^{\pi} = \{(s, a, s') \in \mathcal{T} \mid \pi(s) = a\}$.
- **Policy safety property** $\rho = ((s_{\mathcal{V}_{loc},0}, e_0), (s_{\mathcal{V}_{loc},U}, e_U))$ with partial $s_{\mathcal{V}_{loc},0}, s_{\mathcal{V}_{loc},U}$ over $\mathcal{V}_{loc}$ and $e_0, e_U \in Exp_{bool}$.
- **Start states** $S_0 = \{(s_{\mathcal{V}_{loc}}, s_{\mathcal{V}}) \in \mathcal{S}_{\mathcal{V}_{loc}} \times \mathcal{S}_{\mathcal{V}} \mid s_{\mathcal{V}_{loc},0} \subseteq s_{\mathcal{V}_{loc}} \wedge e_0(s_{\mathcal{V}})\}$.
- **Unsafe states** $S_U = \{(s_{\mathcal{V}_{loc}}, s_{\mathcal{V}}) \in \mathcal{S}_{\mathcal{V}_{loc}} \times \mathcal{S}_{\mathcal{V}} \mid s_{\mathcal{V}_{loc},U} \subseteq s_{\mathcal{V}_{loc}} \wedge e_U(s_{\mathcal{V}})\}$.
- π is **unsafe** with respect to ρ iff there exist $(s_{\mathcal{V}_{loc}}, s_{\mathcal{V}}) \in S_0$, $(t_{\mathcal{V}_{loc}}, t_{\mathcal{V}}) \in S_U$, such that $(t_{\mathcal{V}_{loc}}, t_{\mathcal{V}})$ is reachable from $(s_{\mathcal{V}_{loc}}, s_{\mathcal{V}})$ in Θ^{π}. Otherwise Θ^{π} is **safe** with respect to ρ.

Policy Predicate Abstraction

- Explicit location information & predicates $\mathcal{P} \subseteq Exp_{bool}$,
- **abstraction** of $s_{\mathcal{V}} \in \mathcal{S}_{\mathcal{V}}$: $s_{\mathcal{V}}|_{\mathcal{P}} \in \mathcal{P} \to \mathbb{B}, p \mapsto p(s_{\mathcal{V}})$,
- **concretization** of $s_{\mathcal{P}} \in \mathcal{P} \to \mathbb{B}$: $[s_{\mathcal{P}}] = \{s_{\mathcal{V}} \in \mathcal{S}_{\mathcal{V}} \mid s_{\mathcal{V}}|_{\mathcal{P}} = s_{\mathcal{P}}\}$.
- **Predicate abstraction** $\Theta|_{\mathcal{P}} = \langle \mathcal{S}|_{\mathcal{P}}, \mathcal{A}, \mathcal{T}|_{\mathcal{P}} \rangle$, where $\mathcal{S}|_{\mathcal{P}} = \mathcal{S}_{\mathcal{V}_{loc}} \times (\mathcal{P} \to \mathbb{B})$, and
  $$\mathcal{T}|_{\mathcal{P}} = \{((s_{\mathcal{V}_{loc}}, s_{\mathcal{P}}), a, (s'_{\mathcal{V}_{loc}}, s'_{\mathcal{P}})) \in \mathcal{S}|_{\mathcal{P}} \times \mathcal{A} \times \mathcal{S}|_{\mathcal{P}} \mid \exists s_{\mathcal{V}} \in [s_{\mathcal{P}}], s'_{\mathcal{V}} \in [s'_{\mathcal{P}}] \colon ((s_{\mathcal{V}_{loc}}, s_{\mathcal{V}}), a, (s'_{\mathcal{V}_{loc}}, s'_{\mathcal{V}})) \in \mathcal{T}\}.$$
- **Policy predicate abstraction** $\Theta^{\pi}|_{\mathcal{P}} = \langle \mathcal{S}|_{\mathcal{P}}, \mathcal{A}, \mathcal{T}^{\pi}|_{\mathcal{P}} \rangle$.

Motivation: Safety verification for Θ^{π} via (over-approximating) reachability analysis in $\Theta^{\pi}|_{\mathcal{P}}$.

SMT-Tests to Compute $\Theta^{\pi}|_{\mathcal{P}}$

If $g_{loc} \subseteq s_{\mathcal{V}_{loc}}$ and $s'_{\mathcal{V}_{loc}} = s_{\mathcal{V}_{loc}}[u_{loc}]$ (location constraints), then

$((s_{\mathcal{V}_{loc}}, s_{\mathcal{P}}), a, (s'_{\mathcal{V}_{loc}}, s'_{\mathcal{P}})) \in \mathcal{T}^{\pi}|_{\mathcal{P}}$ iff
$\exists s_{\mathcal{V}} \in [s_{\mathcal{P}}], s'_{\mathcal{V}} \in [s'_{\mathcal{P}}] \colon g(s_{\mathcal{V}}) \wedge s'_{\mathcal{V}} = s_{\mathcal{V}}[u(s_{\mathcal{V}})] \wedge \pi((s_{\mathcal{V}_{loc}}, s_{\mathcal{V}})) = a$

– satisfiability problem over state variable assignments,
– encoded as SMT-test [Barrett *et al.* (1994)]:

An **NN-SAT transition test**, denoted $NNSat(s_{\mathcal{P}}, a, s'_{\mathcal{P}})$, tests the condition $\exists s_{\mathcal{V}} \in [s_{\mathcal{P}}], s'_{\mathcal{V}} \in [s'_{\mathcal{P}}] \colon g(s_{\mathcal{V}}) \wedge s'_{\mathcal{V}} = s_{\mathcal{V}}[u(s_{\mathcal{V}})] \wedge \pi(s_{\mathcal{V}}) = a$.

Problem: NN-SAT tests are expensive $\to$ **over-approximate**.

- SMT transition tests: $SMT(s_{\mathcal{P}}, a, s'_{\mathcal{P}})$ tests $\exists s_{\mathcal{V}} \in [s_{\mathcal{P}}], s'_{\mathcal{V}} \in [s'_{\mathcal{P}}] \colon g(s_{\mathcal{V}}) \wedge s'_{\mathcal{V}} = s_{\mathcal{V}}[u(s_{\mathcal{V}})]$.
- **Applicability tests**: $SMT(s_{\mathcal{P}}, a)$ tests $\exists s_{\mathcal{V}} \in [s_{\mathcal{P}}] \colon g(s_{\mathcal{V}})$, $NNSat(s_{\mathcal{P}}, a)$ tests $\exists s_{\mathcal{V}} \in [s_{\mathcal{P}}] \colon g(s_{\mathcal{V}}) \wedge \pi(s_{\mathcal{V}}) = a$.
- **Continuously-relaxed tests** (relaxing discrete state variables to the continuous domain): $NNSat_{\mathbb{R}}(s_{\mathcal{P}}, a, s'_{\mathcal{P}})$, $NNSat_{\mathbb{R}}(s_{\mathcal{P}}, a)$.

Approach: Compute **fragment** of $\Theta^{\pi}|_{\mathcal{P}}$ **reachable** from $S_0|_{\mathcal{P}}$ in a forward search applying NN-SAT/SMT-tests.
– If $(s'_{\mathcal{V}_{loc}}, s'_{\mathcal{P}}) \notin S_U|_{\mathcal{P}}$ for all reachable $(s'_{\mathcal{V}_{loc}}, s'_{\mathcal{P}})$, then π is safe.

Implementation:
Z3 [de Moura and Bjørner (2008)] for NN-SAT/SMT-tests,
Marabou [Katz *et al.* (2019)] for relaxed NN-SAT tests.

State Expansion

Input: $(s_{\mathcal{V}_{loc}}, s_{\mathcal{P}}) \in \mathcal{S}|_{\mathcal{P}}$, $a \in \mathcal{A}$ with $a = (g_{loc}, g, u_{loc}, u)$

if $\neg g_{loc} \subseteq s_{\mathcal{V}_{loc}}$ **then return**
`// optional applicability tests:`
if $\neg(SMT(s_{\mathcal{P}}, a) \wedge NNSat_{\mathbb{R}}(s_{\mathcal{P}}, a) \wedge NNSat(s_{\mathcal{P}}, a))$ **then return**
$s'_{\mathcal{V}_{loc}} := s_{\mathcal{V}_{loc}}[u_{loc}]$
$s'_{\mathcal{P}} := \{\}$ `// empty truth-value assignment`
$s'_{\mathcal{P}}$ *fixed with respect to* $s_{\mathcal{P}}, g, u$ `// opt`
`enumerate_states(`$s'_{\mathcal{P}}$`)`
Procedure `enumerate_states(`$s'_{\mathcal{P}}$: *predicate state*`)`**:**
  if $dom(s'_{\mathcal{P}}) = \mathcal{P}$ **then**
    `// optional transition tests:`
    if $\neg(SMT(s_{\mathcal{P}}, a, s'_{\mathcal{P}}) \wedge NNSat_{\mathbb{R}}(s_{\mathcal{P}}, a, s'_{\mathcal{P}}))$ **then return**
    if $NNSat(s_{\mathcal{P}}, a, s'_{\mathcal{P}})$ **then** add $((s_{\mathcal{V}_{loc}}, s_{\mathcal{P}}), a, (s'_{\mathcal{V}_{loc}}, s'_{\mathcal{P}}))$ to $\mathcal{T}^{\pi}|_{\mathcal{P}}$
  else
    for some $p \in \mathcal{P} \setminus dom(s'_{\mathcal{P}})$
    let $s'_{\mathcal{P}} := s'_{\mathcal{P}} \uplus \{p \mapsto \text{true}\}$ **in**
      $s'_{\mathcal{P}}$ *fixed with respect to* $\{p \mapsto \text{true}\}$ `// opt`
      `enumerate_states(`$s'_{\mathcal{P}}$`)`
    let $s'_{\mathcal{P}} := s'_{\mathcal{P}} \uplus \{p \mapsto \text{false}\}$ **in**
      $s'_{\mathcal{P}}$ *fixed with respect to* $\{p \mapsto \text{false}\}$ `// opt`
      `enumerate_states(`$s'_{\mathcal{P}}$`)`

Experiments

(on Racetrack modeled in JANI [Budde *et al.* (2017)])