Cheap CTL Compassion in NuSMV

(Tool Paper)

Daniel Hausmann, Tadeusz Litak, Christoph Rauch and Matthias Zinner

VMCAI 2020 – January 21 2020

Chair for Theoretical Computer Science
Friedrich-Alexander Universität Erlangen-Nürnberg, Germany

Model Checking for Branching Time Logics: Theory

- CTL: express reachability (EF) and safety (AG) properties of paths
  - model checking in $\mathcal{O}(|\mathcal{M}| \cdot |\psi|)$
  - cannot express fairness properties (e.g. EGF)
- CTL* (and FCTL, ECTL$^+$) can express fairness properties, but...
  - model checking is PSPACE-complete (P^{NP}-complete)

Model Checking for Branching Time Logics: Theory

- CTL: express reachability (EF) and safety (AG) properties of paths
  - model checking in $\mathcal{O}(|\mathcal{M}| \cdot |\psi|)$
  - cannot express fairness properties (e.g. EGF)
- CTL* (and FCTL, ECTL$^+$) can express fairness properties, but...
  - model checking is PSPACE-complete (P^{NP}-complete)

- (unconditional) fairness: $\bigwedge_{\psi \in \Psi} \mathsf{GF}\, \psi$
- conditional fairness: $\bigwedge_{(\psi,\phi) \in \Psi} (\mathsf{FG}\, \neg\psi \rightarrow \mathsf{GF}\phi)$
- strong fairness (compassion): $\bigwedge_{(\psi,\phi) \in \Psi} (\mathsf{GF}\, \psi \rightarrow \mathsf{GF}\phi)$

Model Checking for Branching Time Logics: Theory

- CTL: express reachability (EF) and safety (AG) properties of paths
  - model checking in $\mathcal{O}(|\mathcal{M}| \cdot |\psi|)$
  - cannot express fairness properties (e.g. EGF)
- CTL* (and FCTL, ECTL$^+$) can express fairness properties, but...
  - model checking is PSPACE-complete (P^{NP}-complete)

- (unconditional) fairness: $\bigwedge_{\psi \in \Psi} \mathsf{GF}\, \psi$
- conditional fairness: $\bigwedge_{(\psi,\phi) \in \Psi} (\mathsf{FG}\, \neg\psi \rightarrow \mathsf{GF}\phi)$
- strong fairness (compassion): $\bigwedge_{(\psi,\phi) \in \Psi} (\mathsf{GF}\, \psi \rightarrow \mathsf{GF}\phi)$

- Model checking CTL^f [Ghilardi & van Gool, 2016] in $\mathcal{O}(|\mathcal{M}| \cdot |\psi|)$
  - no succinct encoding of strong fairness
- Our contribution: Streett-fair CTL (SFCTL)
  - model checking in $\mathcal{O}(|\mathcal{M}| \cdot |\psi|)$
  - succinct encoding of all above fairness properties

Model Checking for Branching Time Logics: Practice

- NuSMV: symbolic model checker for CTL (and LTL)
  - supports unconditional fairness constraints $\forall(\mathit{fair} \rightarrow \phi)$, $\exists(\mathit{fair} \rightarrow \phi)$, where ϕ CTL-formula, *fair* conjunction of atoms
  - for CTL: no support for conditional fairness, strong fairness or fairness objectives (such as "all paths are fair")

Model Checking for Branching Time Logics: Practice

- NuSMV: symbolic model checker for CTL (and LTL)
  - supports unconditional fairness constraints $\forall(\mathit{fair} \rightarrow \phi)$, $\exists(\mathit{fair} \rightarrow \phi)$, where ϕ CTL-formula, *fair* conjunction of atoms
  - for CTL: no support for conditional fairness, strong fairness or fairness objectives (such as "all paths are fair")
- Our contribution: extension NuSMV$^{\mathsf{sf}}$, model checking for SFCTL
  - adds support for conditional and strong fairness (compassion)
  - adds support for (strong) fairness objectives over formulas

Model Checking for Branching Time Logics: Practice

- NuSMV: symbolic model checker for CTL (and LTL)
  - supports unconditional fairness constraints $\forall(\mathit{fair} \rightarrow \phi)$, $\exists(\mathit{fair} \rightarrow \phi)$, where ϕ CTL-formula, *fair* conjunction of atoms
  - for CTL: no support for conditional fairness, strong fairness or fairness objectives (such as "all paths are fair")
- Our contribution: extension NuSMV$^{\text{sf}}$, model checking for SFCTL
  - adds support for conditional and strong fairness (compassion)
  - adds support for (strong) fairness objectives over formulas

Model checking for SFCTL = solving parity games with 2 priorities

$\rightsquigarrow$ We can benchmark NuSMV$^{\text{sf}}$ against (symbolic) parity game solvers

Streett-Fair CTL (SFCTL)

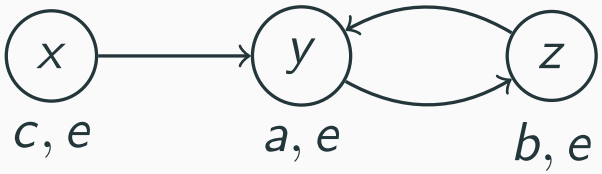
Streett-Fair CTL, Syntax

$$\phi, \psi, \chi := \top \mid p \mid \neg\phi \mid \phi \wedge \psi \mid \mathsf{EX}\,\phi \mid \mathsf{E}\,\phi\,\mathsf{U}\,\psi \mid$$
$$\mathsf{E}\,(\phi_1 \blacktriangleright \psi_1, \ldots, \phi_n \blacktriangleright \psi_n)\,\mathsf{G}\,\chi \qquad p \in \mathsf{At}, n \in \mathbb{N}$$

Streett-Fair CTL, Semantics

- Interpret formulas over transition systems $\mathcal{M} := (S, \longrightarrow, L)$
- Semantics of Boolean operators, EX and EU as usual
- $\mathcal{M}, s \models \mathsf{E}\,(\phi_1 \blacktriangleright \psi_1, \ldots, \phi_n \blacktriangleright \psi_n)\,\mathsf{G}\,\chi$ iff $\exists$ path π starting at s s.t.
  - χ holds globally on π
  - for all $1 \leq i \leq n$:
    if ϕ_i holds infinitely often on π, then ψ_i holds infinitely often on π.

Streett-Fair CTL (SFCTL): Example



$x \models \mathsf{E}\,(a \blacktriangleright b, c \blacktriangleright d)\,\mathsf{G}\,e$

$x \not\models \mathsf{E}\,(e \blacktriangleright c)\,\mathsf{G}\,\top$

$x \models \mathsf{E}((\mathsf{GF}\,a \rightarrow \mathsf{GF}\,b) \wedge (\mathsf{GF}\,c \rightarrow \mathsf{GF}\,d) \wedge \mathsf{G}\,e)$

Embedding SFCTL into the μ-Calculus

Definition

$$\tau(Y) := \bigwedge_{1 \leq i \leq n} (\mathsf{EX}(\mu Z.(\psi_i \wedge Y) \vee (Y \wedge \mathsf{EX}(Z))) \vee (\neg\phi_i \wedge \mathsf{EX}\, Y))$$

Lemma [Emerson & Lei, 1986]

$$\mathsf{E}(\phi_1 \blacktriangleright \psi_1, \ldots, \phi_n \blacktriangleright \psi_n)\, \mathsf{G}\, \chi \equiv \mathsf{E}\, \chi\, \mathsf{U}\, \nu Y.\, (\chi \wedge \tau(Y))$$

Hence SFCTL embeds into the μ-calculus with alternation depth ≤ 2.

Embedding SFCTL into the μ-Calculus

Definition

$$\tau(Y) := \bigwedge_{1 \le i \le n} (\mathsf{EX}(\mu Z.(\psi_i \wedge Y) \vee (Y \wedge \mathsf{EX}(Z))) \vee (\neg\phi_i \wedge \mathsf{EX}\, Y))$$

Lemma [Emerson & Lei, 1986]

$$\mathsf{E}(\phi_1 \blacktriangleright \psi_1, \ldots, \phi_n \blacktriangleright \psi_n)\, \mathsf{G}\, \chi \equiv \mathsf{E}\, \chi\, \mathsf{U}\, \nu Y.\, (\chi \wedge \tau(Y))$$

Hence SFCTL embeds into the μ-calculus with alternation depth ≤ 2.

Corollary

- Model checking for SFCTL_n can be done in time $\mathcal{O}(\mathcal{M} \cdot |\phi|)$.
- Model checking for SFCTL can be done in time $\mathcal{O}(\mathcal{M} \cdot |\phi|^2)$.

Embedding SFCTL into the μ-Calculus

Definition

$$\tau(Y) := \bigwedge_{1 \leq i \leq n} (\mathsf{EX}(\mu Z.(\psi_i \wedge Y) \vee (Y \wedge \mathsf{EX}(Z))) \vee (\neg\phi_i \wedge \mathsf{EX}\, Y))$$

Lemma [Emerson & Lei, 1986]

$$\mathsf{E}(\phi_1 \blacktriangleright \psi_1, \ldots, \phi_n \blacktriangleright \psi_n)\, \mathsf{G}\, \chi \equiv \mathsf{E}\, \chi\, \mathsf{U}\, \nu Y.\, (\chi \wedge \tau(Y))$$

Hence SFCTL embeds into the μ-calculus with alternation depth ≤ 2.

Corollary

- Model checking for SFCTL_n can be done in time $\mathcal{O}(\mathcal{M} \cdot |\phi|)$.
- Model checking for SFCTL can be done in time $\mathcal{O}(\mathcal{M} \cdot |\phi|^2)$.

Extended Streett-Fair CTL (ESFCTL)

Recall: $\tau(Y) = \bigwedge_{1 \leq i \leq n} (\mathsf{EX}(\mu Z.(\psi_i \wedge Y) \vee (Y \wedge \mathsf{EX}(Z))) \vee (\neg \phi_i \wedge \mathsf{EX} Y))$

Extended Streett-Fair CTL (ESFCTL): $\langle \phi_1 \triangleright \psi_1, \ldots, \phi_n \triangleright \psi_n \rangle := \nu Y.\tau(Y)$

"$\forall 1 \leq i \leq n$, if t satisfying ϕ_i is reachable, then u satisfying ψ_i is reachable from t"

Extended Streett-Fair CTL (ESFCTL)

Recall: $\tau(Y) = \bigwedge_{1 \le i \le n} (\mathsf{EX}(\mu Z.(\psi_i \wedge Y) \vee (Y \wedge \mathsf{EX}(Z))) \vee (\neg\phi_i \wedge \mathsf{EX}\, Y))$

Extended Streett-Fair CTL (ESFCTL): $\langle \phi_1 \triangleright \psi_1, \ldots, \phi_n \triangleright \psi_n \rangle := \nu Y.\tau(Y)$

"$\forall 1 \le i \le n$, if t satisfying ϕ_i is reachable, then u satisfying ψ_i is reachable from t"

Fact

$\mathsf{E}(\phi_1 \blacktriangleright \psi_1, \ldots, \phi_n \blacktriangleright \psi_n)\, \mathsf{G}\, \chi \equiv \mathsf{E}\, \chi\, \mathsf{U}\, \langle \phi_1 \triangleright \psi_1, \ldots, \phi_n \triangleright \psi_n, \neg\chi \triangleright \bot \rangle$

ESFCTL also embeds into the μ-calculus with alternation depth ≤ 2.

Extended Streett-Fair CTL (ESFCTL)

Recall: $\tau(Y) = \bigwedge_{1 \leq i \leq n} (\mathsf{EX}(\mu Z.(\psi_i \wedge Y) \vee (Y \wedge \mathsf{EX}(Z))) \vee (\neg \phi_i \wedge \mathsf{EX} Y))$

Extended Streett-Fair CTL (ESFCTL): $\langle \phi_1 \triangleright \psi_1, \ldots, \phi_n \triangleright \psi_n \rangle := \nu Y.\tau(Y)$

"$\forall 1 \leq i \leq n$, if t satisfying ϕ_i is reachable, then u satisfying ψ_i is reachable from t"

Fact

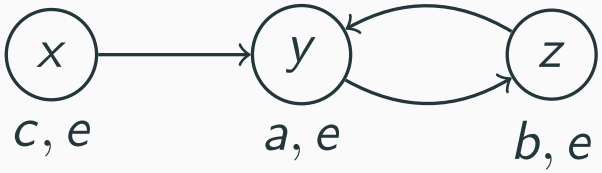
$\mathsf{E}(\phi_1 \blacktriangleright \psi_1, \ldots, \phi_n \blacktriangleright \psi_n)\, \mathsf{G}\, \chi \equiv \mathsf{E}\, \chi\, \mathsf{U}\, \langle \phi_1 \triangleright \psi_1, \ldots, \phi_n \triangleright \psi_n, \neg \chi \triangleright \bot \rangle$

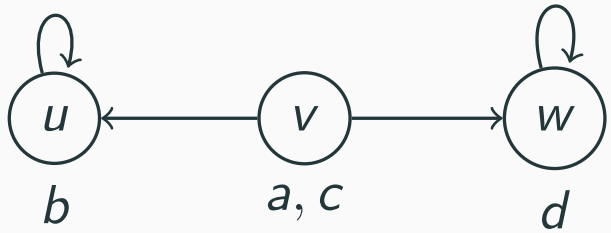
ESFCTL also embeds into the μ-calculus with alternation depth ≤ 2.

Corollary of [Rabinovich & Schnoebelen, 2006]

For all $n \in \mathbb{N}$, (E)SFCTL$_{n+1}$ is more expressive than (E)SFCTL$_n$.

Extended Streett-Fair CTL (ESFCTL): Examples





$x \models \mathsf{E}\,(a \blacktriangleright b, c \blacktriangleright d)\,\mathsf{G}\,e$

$x \not\models \langle a \triangleright b, c \triangleright d \rangle$

$y \models \langle a \triangleright b, c \triangleright d \rangle$

$v \models \langle a \triangleright b, c \triangleright d \rangle$

Does ESFCTL embed into CTL*? Conjecture: No

Implementation as Extension of NuSMV

NuSMV^{sf} adds syntax and BDD-based model checking for ESFCTL, e.g.:

function ExtendedStreettFair$((\phi_1, \psi_1), \ldots, (\phi_n, \psi_n))$
  $y \leftarrow$ BDD for 'true'
  $y' \leftarrow$ BDD for 'false'
  while $y \neq y'$ **do**
    $y' \leftarrow y$
    $y \leftarrow$ BDD for 'true'
    for $i = 1 \ldots n$ **do**
      $z \leftarrow \mathrm{EU}(y', \psi_i \wedge y')$
      $y \leftarrow y \wedge (\mathrm{EX}(z) \vee (\neg\phi_i \wedge \mathrm{EX}(y')))$
  return y

Sources available at `https://git8.cs.fau.de/software/nusmvf`

Benchmarking and Examples

We check various models against SFCTL-formulas (= solve parity games with 2 priorities)

Artifact at `https://doi.org/10.6084/m9.figshare.9977510`

- Compared model checkers / parity game solvers
  - Explicit parity game solver PGSolver[1] (OCaml):
    PGSolver(zlk), PGSolver(mc)
  - Explicit parity game solver Oink[2] (C++):
    Oink(zlk), Oink(pp)
  - BDD-based parity game solvers[3] (Python):
    BDD(zlk), BDD(fpi)
  - BDD-based SFCTL model checker NuSMV^{sf} (C)

[1] `https://github.com/tcsprojects/pgsolver`
[2] `https://github.com/trolando/oink`
[3] [Sanchez & Wesselink & Willemse, 2018]

Benchmarking – Elevator I, FIFO

Check model of FIFO elevator control system with n stages against

$$\mathsf{AG} \bigwedge_{i \leq n} \neg \mathsf{E}(\top \blacktriangleright \mathsf{isPressed(i)})\, \mathsf{G}\, \neg \mathsf{isAt}(i)$$

n	PGSolver(zlk)		Oink(zlk)		NuSMV^{sf}	
	Time	Memory	Time	Memory	Time	Memory
3	0.009	9836	0.002	5080	0.009	17 272
4	0.046	12 656	0.003	5672	0.028	17 852
5	1.252	33 888	0.009	7800	0.112	19 832
6	73.370	168 624	0.070	27 824	0.758	25 480
7	$\dagger_T$	$\dagger_T$	0.771	189 980	2.904	50 792
8	$\dagger_T$	$\dagger_T$	6.812	1 671 940	69.815	67 672
9	$\dagger_M$	$\dagger_M$	$\dagger_M$	$\dagger_M$	$\dagger_T$	$\dagger_T$

Benchmarking – Elevator II, LIFO

Check model of LIFO elevator control system with n stages against

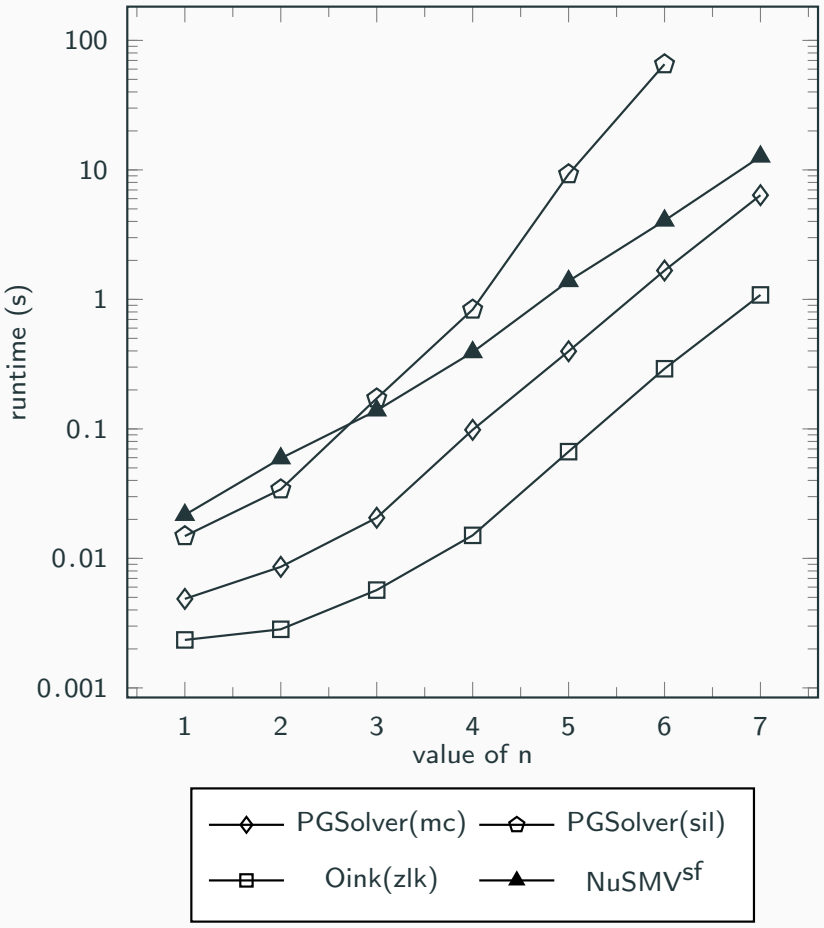
$$\mathsf{AG} \bigwedge_{i \leq n} \neg \mathsf{E}(\top \blacktriangleright \mathsf{isPressed(i)})\, \mathsf{G}\, \neg \mathsf{isAt}(i)$$

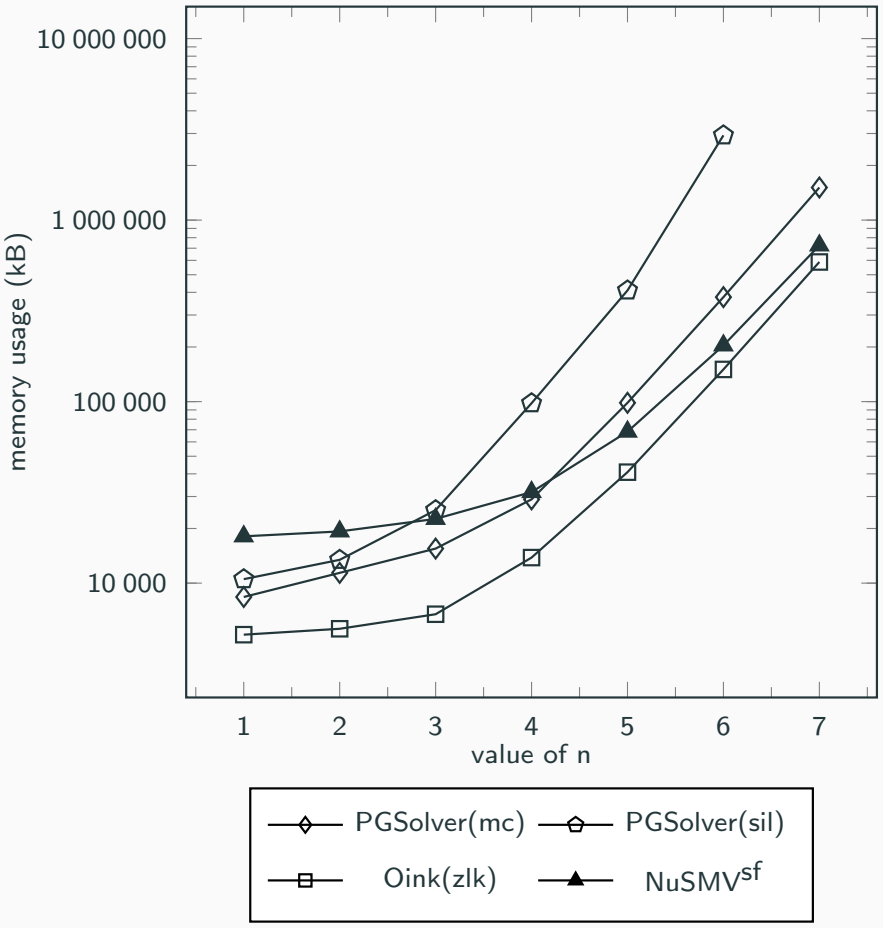
n	PGSolver(zlk)		Oink(zlk)		NuSMV^{sf}	
	Time	Memory	Time	Memory	Time	Memory
3	0.007	9440	0.002	5184	0.012	17 260
4	0.022	12 664	0.004	5676	0.033	17 836
5	0.149	30 652	0.012	8324	0.137	19 748
6	1.440	171 440	0.084	30 916	0.477	26 824
7	14.744	1 434 396	4.005	203 280	2.073	53 876
8	167.345	13 605 024	11.249	1 772 448	71.507	69 800

Benchmarking – Alternating Bit Protocol

Check model of alternating bit protocol against

$$\mathsf{AG} \bigwedge_{d \in D} \neg \mathsf{E}(\top \blacktriangleright \mathsf{EX}(\mathit{transition} = r1(d)))\, \mathsf{G}\, \neg(\mathit{transition} = r1(d)) \qquad |D| = 2^n$$

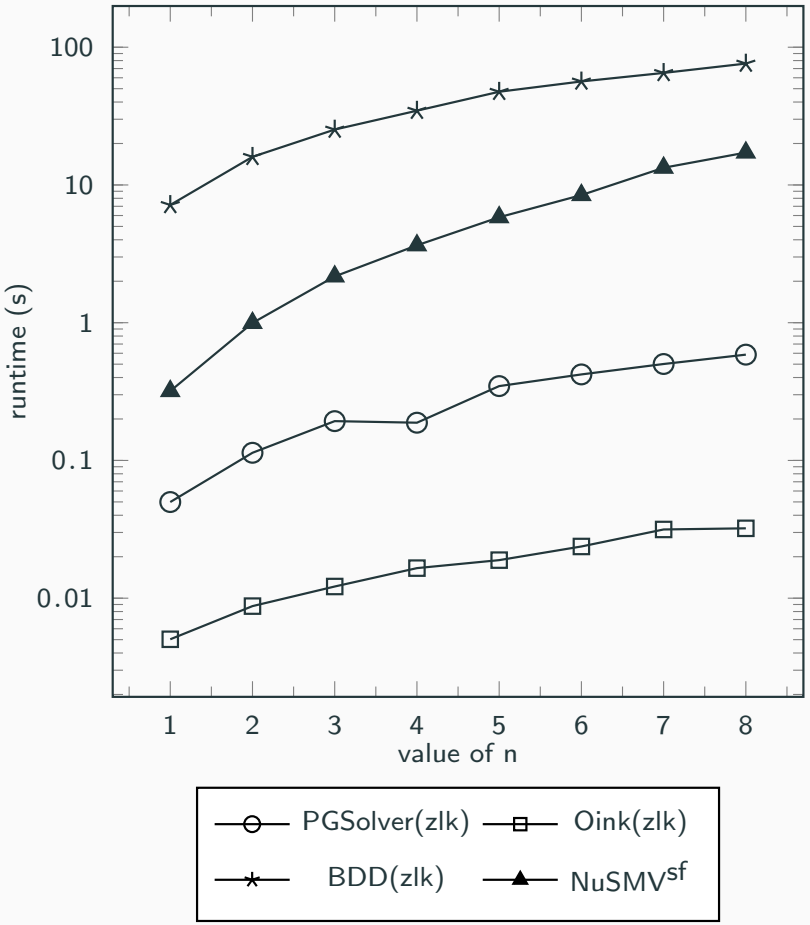


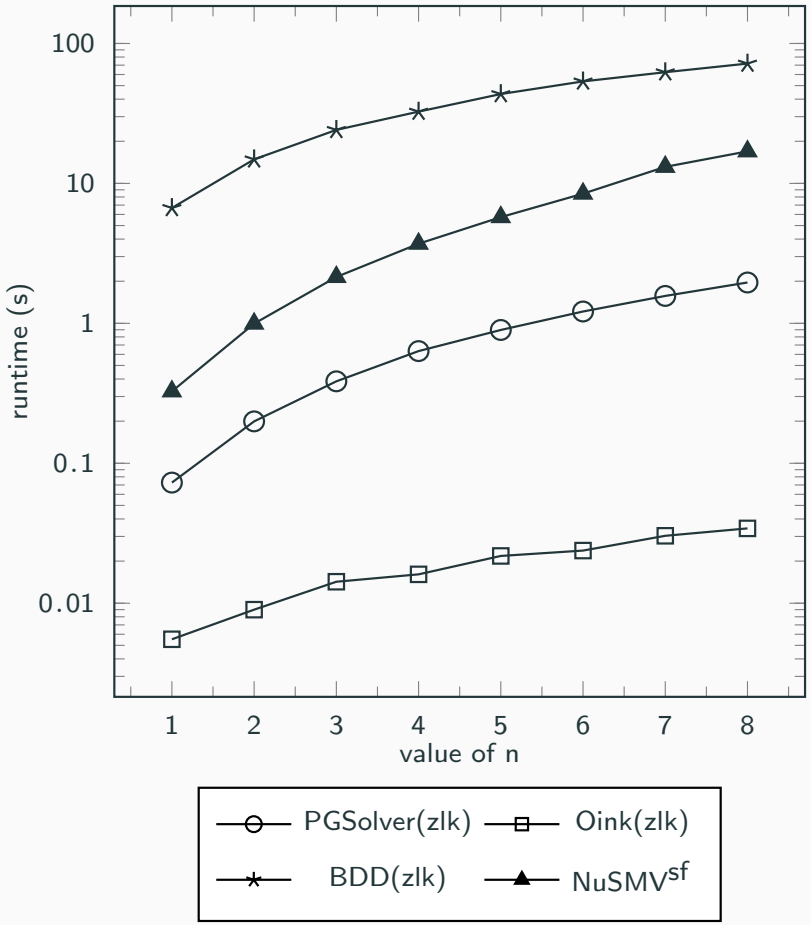


Benchmarking – Random Büchi Automata

Check random (non-)empty Büchi automata for non-emptiness, i.e. against

$$\mathsf{E}(\top \blacktriangleright f)\,\mathsf{G}\,\top$$



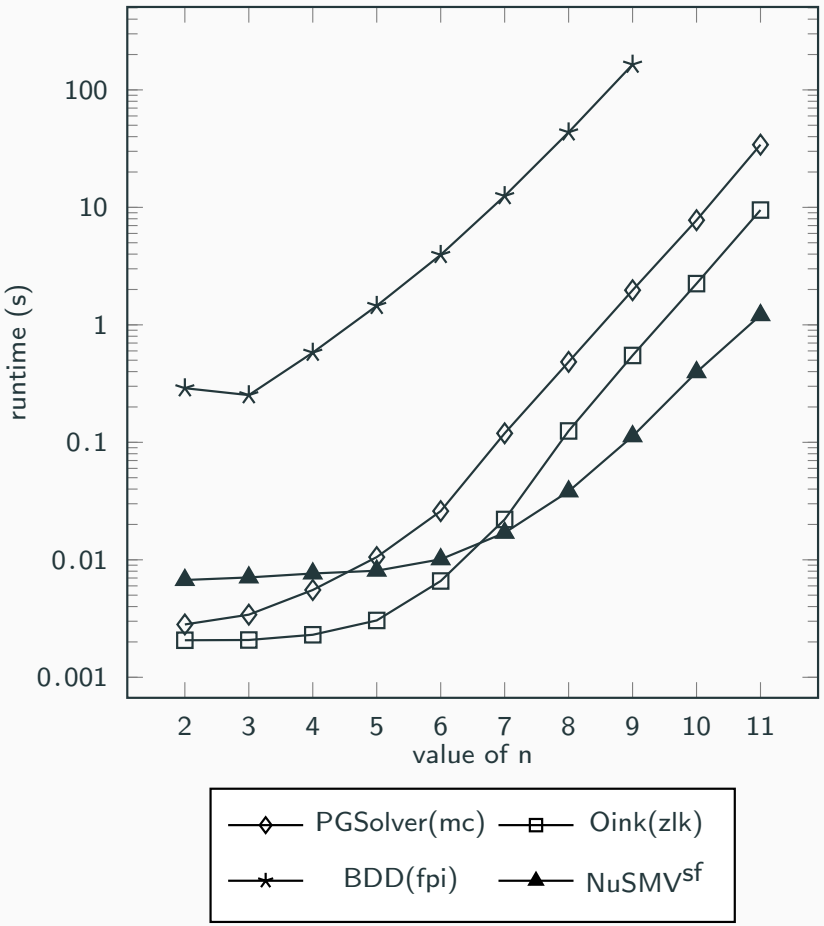


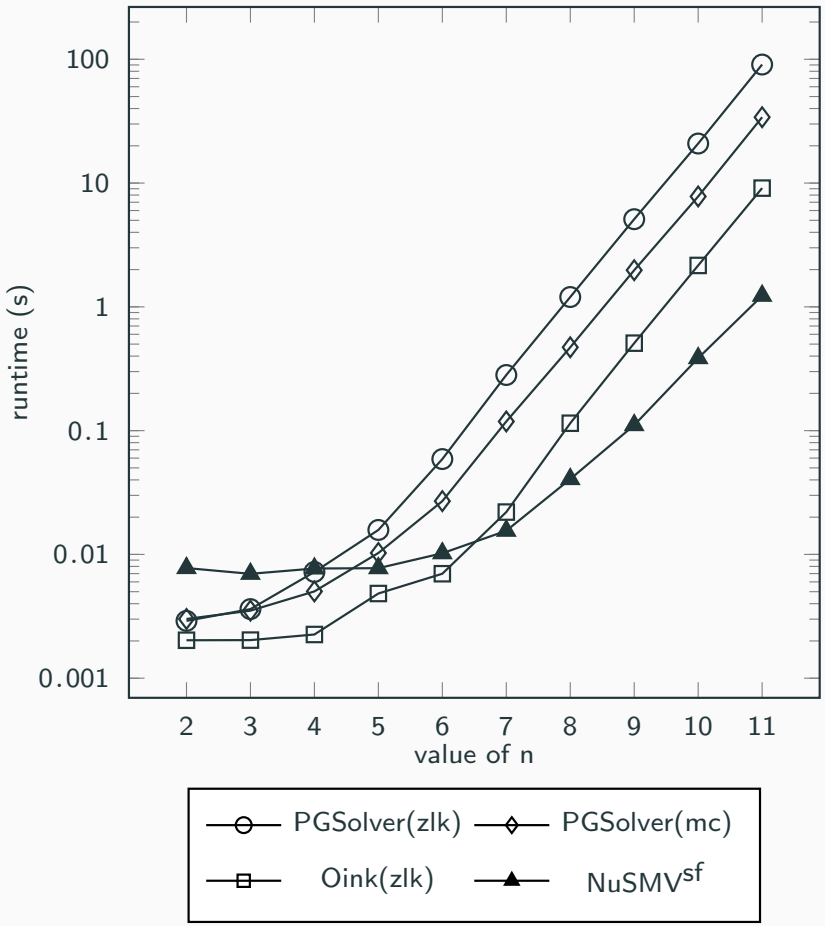
$n \cdot 1000$ states

Benchmarking – Tree-shaped Parity Automata

Check tree-shaped non-empty/totally accepting parity automata against

$$\mathsf{EF} \bigvee_{i \leq 2^n, i \text{ even}} \mathsf{E}(\top \blacktriangleright p_i)\, \mathsf{G}\Big(\bigwedge_{j>i} \neg p_j\Big)$$



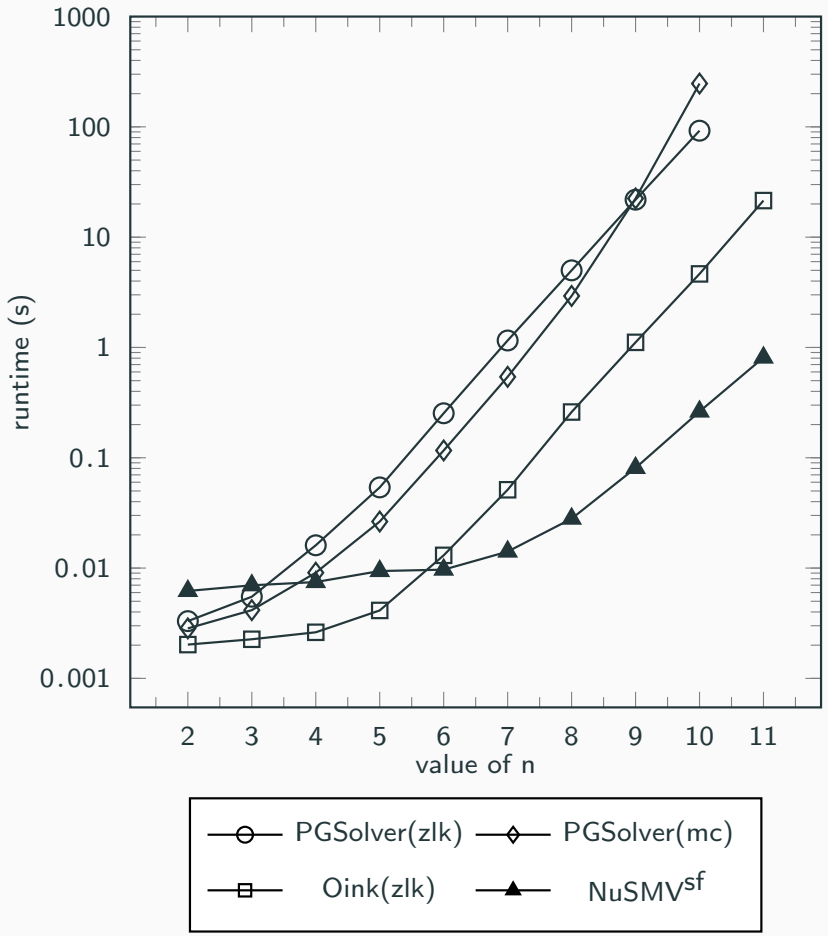


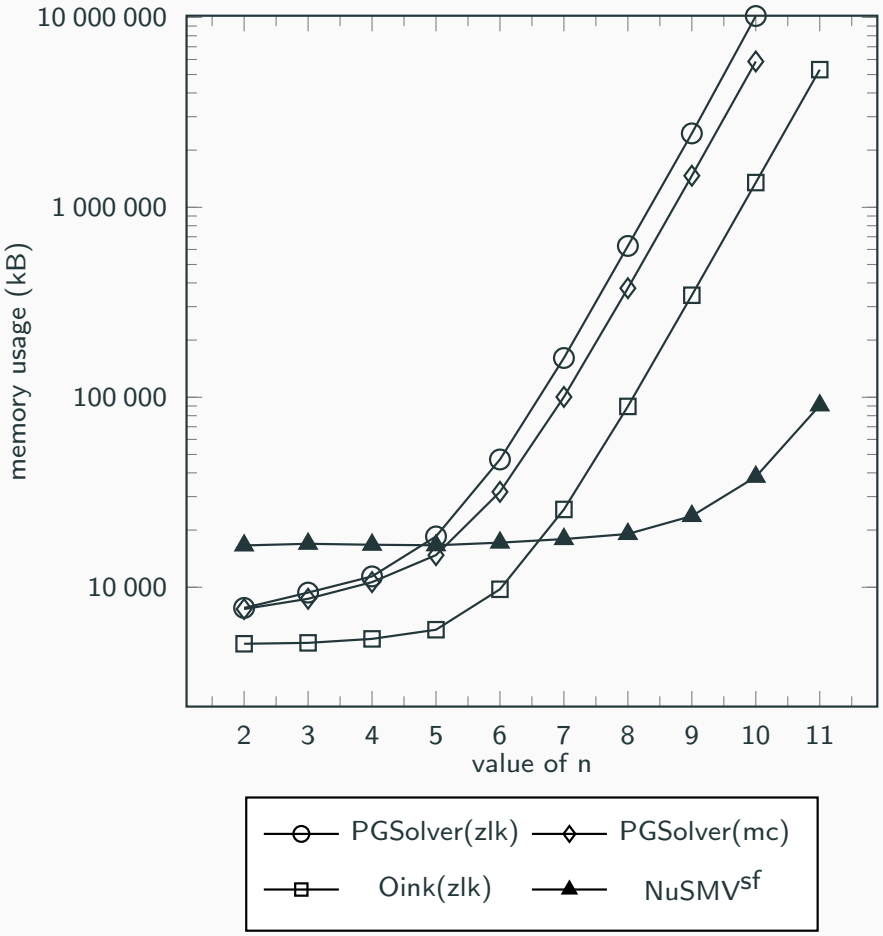
$2^{n+1} - 1$ nodes, leaf number i has priority i, all other nodes priority 0

Benchmarking – Tree-shaped Streett Automata

Check tree-shaped non-empty Streett automata against

$$\mathsf{E}(p_1 \blacktriangleright q_1, \ldots, p_{2^n} \blacktriangleright q_{2^n})\,\mathsf{G}\,\top$$





$2^{n+1} - 1$ nodes, leaf number i belongs to p_i, root to q_j for all j

Conclusion

Results:

- Model checking for Streett-fair CTL (SFCTL) in $\mathcal{O}(|\mathcal{M}| \cdot |\psi|)$ by reduction to parity game solving.
- Added support for SFCTL in symbolic model checker NuSMV.
- Compared performance with symbolic and explicit parity game solvers.

Future work:

- Extend implementation to full μ-calculus (adding full symbolic parity game solver to NuSMV)
- Completeness of a Kozen-style axiomatization of (E)SFCTL