Model-checking Counting Temporal Logics on Flat Structures

N. Decker[1], P. Habermehl[2], M. Leucker[1], A. Sangnier[2] and D. Thoma[2]

[1] Institute for Software Engineering and Programming Languages,
University of Lübeck, Germany
{decker,leucker,thoma}@isp.uni-luebeck.de

[2] IRIF, Univ. Paris Diderot, France
{haberm,sangnier}@irif.fr

CONCUR 2017, Berlin

September 6, 2017

Counting in Temporal Logic

Relative frequencies

$$(\neg err)\ \mathrm{U}^{80\%}\ safe$$

- Frequency LTL ($\texttt{fLTL}$): $\texttt{LTL} + \mathrm{U}^{\frac{n}{m}}$ [B Bollig, N D., M Leucker '12]
- Frequency CTL ($\texttt{fCTL}$): $\texttt{CTL} + \mathrm{EU}^{\frac{n}{m}} + \mathrm{AU}^{\frac{n}{m}}$
- Frequency CTL* ($\texttt{fCTL}^*$)

Counting in Temporal Logic

Relative frequencies

$$(\neg err)\ \mathrm{U}^{80\%}\ safe$$

- Frequency LTL ($\mathtt{fLTL}$): $\mathtt{LTL}$ + $\mathrm{U}^{\frac{n}{m}}$ [B Bollig, N D., M Leucker '12]
- Frequency CTL ($\mathtt{fCTL}$): $\mathtt{CTL}$ + $\mathrm{EU}^{\frac{n}{m}}$ + $\mathrm{AU}^{\frac{n}{m}}$
- Frequency CTL* ($\mathtt{fCTL}^*$)

Counting in Temporal Logic

Explicit position counting

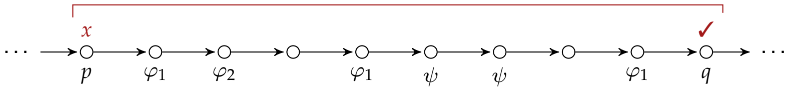
$$\mathrm{G}\left(p \;\rightarrow\; x.\ \mathrm{F}\ q \wedge \#_x(\varphi_1) + \#_x(\varphi_2) \geq 2 \cdot \#_x(\psi)\right)$$

- Count (specific) positions within a scope
- Arithmetic constraints
- Variants: CLTL, CCTL, CCTL*

Counting in Temporal Logic

Explicit position counting

$$\mathrm{G}\Big(p \quad \rightarrow \quad x.\ \mathrm{F} \ \ q \ \wedge \ \#_x(\varphi_1) + \#_x(\varphi_2) \geq 2 \cdot \#_x(\psi) \ \Big)$$

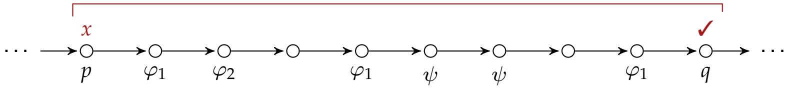


- Count (specific) positions within a scope
- Arithmetic constraints
- Variants: CLTL, CCTL, CCTL*

Counting in Temporal Logic

Explicit position counting

$$\mathrm{G}\Big(p \;\rightarrow\; x.\ \mathrm{F}\ q \wedge \#_x(\varphi_1) + \#_x(\varphi_2) \geq 2 \cdot \#_x(\psi)\Big)$$



- Count (specific) positions within a scope
- Arithmetic constraints
- Variants: CLTL, CCTL, CCTL*

Counting in Temporal Logic

Similar formalisms

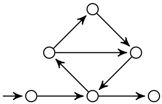
- Counting/Presburger LTL [A Bouajjani, R Echahed, P Habermehl '95]
- Counting LTL/CTL [F Laroussinie, A Meyer, E Petonnet '10 '12]
- Regular Availability Expressions [J Hönicke, R Meyer, E-R Olderog '10]

Model Checking

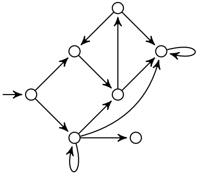
- Frequency LTL: *undecidable* [B Bollig, N D., M Leucker '12]
- Counting LTL: *undecidable* [A Bouajjani, R Echahed, P Habermehl '95] [F Laroussinie, A Meyer, E Petonnet '10]
- Counting CTL: *undecidable* [F Laroussinie, A Meyer, E Petonnet '12]
- Regular Availability Expressions: *non-elementary*
  with intersection: *undecidable* [P Abdulla, MF Atig, R Meyer, MS Salehi '15]

Flat Systems

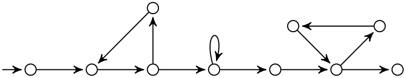
(not flat)



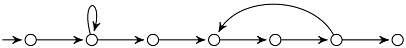
(not flat)



(flat)



(flat)



(flat)

(Existential) Model-checking Flat Systems

Flat Kripke Structures

- LTL: NP-complete

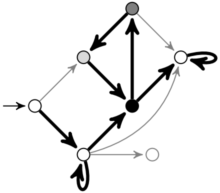
[Kuhtz, Finkbeiner '11]

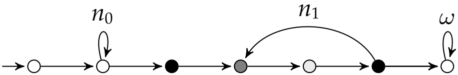
Flat Counter Systems

- LTL+Past: NP-complete
- FO, L_μ: PSPACE-complete
- CTL^*: SAT(PA)-equivalent

[Demri, Dhar, Sangnier '13,'14,'15]

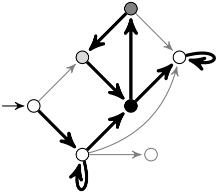
Path Schemas

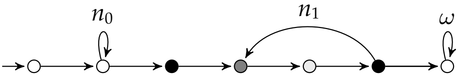




- Concise representation of specific runs and sets of runs
- Flat systems coverable by finite union
- Small witnesses for LTL formulae [Kuhtz, Finkbeiner '11]
- Encoding (with counters) in *quantifier-free* PA [Demri, Dhar, Sangnier '14]

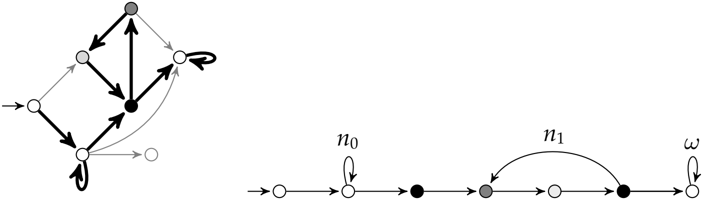
Path Schemas





- Concise representation of specific runs and sets of runs
- Flat systems coverable by finite union
- Small witnesses for LTL formulae [Kuhtz, Finkbeiner '11]
- Encoding (with counters) in *quantifier-free* PA [Demri, Dhar, Sangnier '14]

Path Schemas



- Concise representation of specific runs and sets of runs
- Flat systems coverable by finite union
- Small witnesses for LTL formulae [Kuhtz, Finkbeiner '11]
- Encoding (with counters) in *quantifier-free* PA [Demri, Dhar, Sangnier '14]

Model-checking Frequency LTL

Model-checking problem

Input: Flat Kripke structure $\mathcal{K}$, $\texttt{fLTL}$ formula Φ.

Question: $\exists \rho \in \mathcal{K} : \rho \models \Phi$

- Symbolic models (*augmented path schemas*)
- Syntactic criterion (*consistency*)
- Small model property (exp. bound in $|\Phi| + |\mathcal{K}|$)

$\Rightarrow$ Guess&Check algorithm: decidable in NEXP

Model-checking Frequency CTL/CTL*

fCTL* (on flat Kripke structures)

- CTL labelling algorithm
- Handle $\mathrm{E}\,\varphi$ and $\mathrm{A}\,\varphi \equiv \neg\,\mathrm{E}\,\neg\varphi$ by (determin.) fLTL-subprocedure (EXPSPACE)

fCTL (on Kripke structures)

- CTL labelling algorithm
- For $\mathrm{EU}^{\frac{n}{m}}$ and $\mathrm{AU}^{\frac{n}{m}}$ construct 1-$\mathbb{Z}$-counter system
- Decide (repeated) reachability in polynomial time

Model-checking Frequency CTL/CTL*

$\texttt{fCTL}^*$ (on flat Kripke structures)

- $\texttt{CTL}$ labelling algorithm
- Handle $\mathrm{E}\,\varphi$ and $\mathrm{A}\,\varphi \equiv \neg\,\mathrm{E}\,\neg\varphi$ by (determin.) $\texttt{fLTL}$-subprocedure (EXPSPACE)

$\texttt{fCTL}$ (on Kripke structures)

- $\texttt{CTL}$ labelling algorithm
- For $\mathrm{EU}^{\frac{n}{m}}$ and $\mathrm{AU}^{\frac{n}{m}}$ construct 1-$\mathbb{Z}$-counter system
- Decide (repeated) reachability in polynomial time

Model-checking Counting Temporal Logics

fLTL (FKS):	NEXP	(NP-hard)
fCTL* (FKS):	EXPSPACE	(NP-hard)
fCTL (KS):	P-*complete*	
CLTL, CCTL, CCTL* (FKS):	*decidable*	

Model-checking Counting Temporal Logics

fLTL (FKS):	NEXP	(NP-hard)
fCTL* (FKS):	EXPSPACE	(NP-hard)
fCTL (KS):	P-*complete*	
CLTL, CCTL, CCTL* (FKS):	*decidable*	

Counting in FO

Härtig's Equicardinality Quantifier

$$I_{x,y}(\varphi(x), \psi(y)) \;:\Leftrightarrow\; Card(\{v \mid \varphi(v)\}) = Card(\{v \mid \psi(v)\})$$ [K Härtig '62]

Presburger Arithmetic with Härtig Quantifier (PH)

$$\exists n. \quad \exists y. n = 2y \quad \wedge \quad \exists^{=n} x. P(x)$$

- Count number of solutions
- SAT(PH) is *decidable* (non-elementary) [H Apelt '66] [W Pugh '94] [N Schweikhardt '05]

Counting in FO

Härtig's Equicardinality Quantifier

$$I_{x,y}(\varphi(x), \psi(y)) \;:\Leftrightarrow\; Card(\{v \mid \varphi(v)\}) = Card(\{v \mid \psi(v)\})$$ [K Härtig '62]

Presburger Arithmetic with Härtig Quantifier (PH)

$$\exists n. \quad \exists y. n = 2y \quad \wedge \quad \exists^{=n} x. P(x)$$

- Count number of solutions
- SAT(PH) is *decidable* (non-elementary) [H Apelt '66] [W Pugh '94] [N Schweikhardt '05]

Model-checking CCTL*

MC(CCTL*, FKS) ≤ SAT(PH)

- Encode runs as fixed-size vectors based on path schemas
- Express CCTL* semantics in PH

SAT(PH) $\leq_{\text{Exp}}$ MC(CLTL, FKS)

- Encode variable values in distances between scopes
- Counting positions between scopes, translate arithmetic constraints

Model-checking $\mathtt{CCTL}^*$

$\mathsf{MC}(\mathtt{CCTL}^*, \mathtt{FKS}) \leq \mathsf{SAT}(\mathtt{PH})$

- Encode runs as fixed-size vectors based on path schemas
- Express $\mathtt{CCTL}^*$ semantics in $\mathtt{PH}$

$\mathsf{SAT}(\mathtt{PH}) \leq_{\text{Exp}} \mathsf{MC}(\mathtt{CLTL}, \mathtt{FKS})$

- Encode variable values in distances between scopes
- Counting positions between scopes, translate arithmetic constraints

Model-checking Counting Temporal Logics

fLTL (FKS):	NEXP	(NP-hard)
fCTL^* (FKS):	EXPSPACE	(NP-hard)
fCTL (KS):	P-*complete*	
CLTL, CCTL, CCTL^* (FKS):	*decidable*	

Model-checking Counting Temporal Logics

fLTL (FKS):	NEXP	(NP-hard)
fCTL* (FKS):	EXPSPACE	(NP-hard)
fCTL (KS):	P-*complete*	
CLTL, CCTL, CCTL* (FKS):	SAT(PH)-*equivalent*	(EXP reduction)

Outlook

Theory

- Generalise
  - fLTL procedure to CLTL (with restrictions)
  - from flat Kripke structures to flat counter systems
  - fCTL procedure to CCTL (with restrictions)
- Precise complexity for fLTL

  (NP-c. for fixed nesting depth! NP in general?)

Verification

- Under-approximation technique for non-flat structures
- Identify suitable application domains

References I

- Parosh Aziz Abdulla, Mohamed Faouzi Atig, Roland Meyer, and Mehdi Seyed Salehi. What's decidable about availability languages? In *FSTTCS*, volume 45 of *LIPIcs*, pages 192–205. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.
- H. Apelt. Axiomatische Untersuchungen über einige mit der Presburgerschen Arithmetik verwandten Systeme. *Z. Math. Logik Grundlagen Math.*, 12:131–168, 1966.
- Benedikt Bollig, Normann Decker, and Martin Leucker. Frequency linear-time temporal logic. In *TASE*, pages 85–92. IEEE Computer Society, 2012.
- Ahmed Bouajjani, Rachid Echahed, and Peter Habermehl. On the verification problem of nonregular properties for nonregular processes. In *LICS*, pages 123–133. IEEE Computer Society, 1995.
- Stéphane Demri, Amit Kumar Dhar, and Arnaud Sangnier. On the complexity of verifying regular properties on flat counter systems,. In *ICALP (2)*, volume 7966 of *Lecture Notes in Computer Science*, pages 162–173. Springer, 2013.
- Stéphane Demri, Amit Kumar Dhar, and Arnaud Sangnier. Equivalence between model-checking flat counter systems and presburger arithmetic. In *RP*, volume 8762 of *Lecture Notes in Computer Science*, pages 85–97. Springer, 2014.

References II

- Stéphane Demri, Amit Kumar Dhar, and Arnaud Sangnier. Taming past LTL and flat counter systems. *Inf. Comput.*, 242:306–339, 2015.
- Jochen Hoenicke, Roland Meyer, and Ernst-Rüdiger Olderog. Kleene, rabin, and scott are available. In *CONCUR*, volume 6269 of *Lecture Notes in Computer Science*, pages 462–477. Springer, 2010.
- Klaus Härtig. Über einen Quantifikator mit zwei Wirkungsbereichen. In L. Kalmar, editor, *Colloquium on the foundations of mathematics, mathematical machines and their applications*, pages 31–36. Akademiai Kiado, Budapest, 1962.
- Lars Kuhtz and Bernd Finkbeiner. Weak kripke structures and LTL. In *CONCUR*, volume 6901 of *Lecture Notes in Computer Science*, pages 419–433. Springer, 2011.
- François Laroussinie, Antoine Meyer, and Eudes Petonnet. Counting LTL. In *TIME*, pages 51–58. IEEE Computer Society, 2010.
- François Laroussinie, Antoine Meyer, and Eudes Petonnet. Counting CTL. *Logical Methods in Computer Science*, 9(1), 2012.
- William Pugh. Counting solutions to presburger formulas: How and why. In *PLDI*, pages 121–134. ACM, 1994.

References III

Nicole Schweikardt. Arithmetic, first-order logic, and counting quantifiers. *ACM Trans. Comput. Log.*, 6(3):634–671, 2005.