Equivalence Between Model-Checking Flat Counter Systems and Presburger Arithmetic

Amit Kumar Dhar

LIAFA, Paris.
ULB, Belgium.

Casstings Meeting

Joint Work With:

Stéphane Demri,
NYU, New York.
CNRS, France.

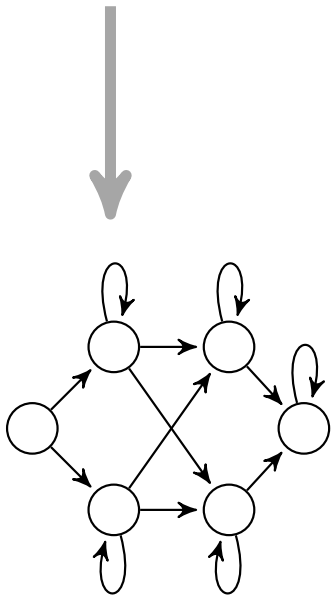
Arnaud Sangnier
LIAFA, Université Paris Diderot, Paris.

Verification 〉 Model Checking

A System **Satisfies** **A Property**

Verification 〉 Model Checking

A System

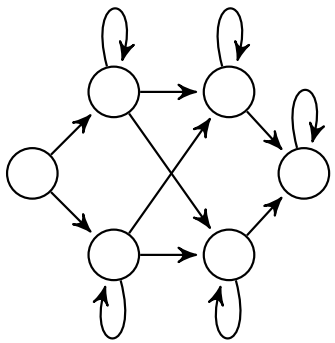


Satisfies

A Property

Verification 〉 Model Checking

A System



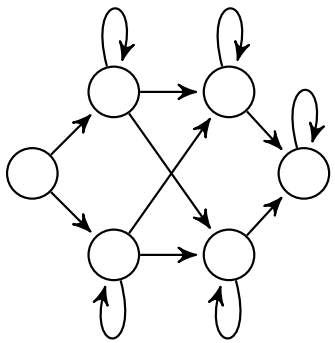
Satisfies

A Property

$\mathcal{A}$

Verification 〉 Model Checking

A System



Satisfies

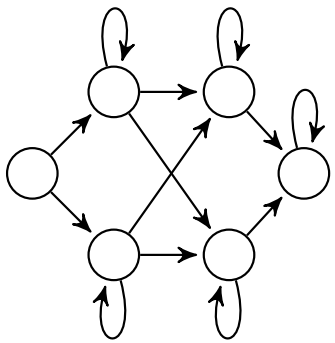
$\models$

A Property

$\mathcal{A}$

Verification 〉 Model Checking

A System | **Satisfies** | **A Property**



$\models$

$\mathcal{A}$

- Model-checking infinite state systems are generally undecidable.
- Decidability can be obtained by restricting expressiveness.

Verification 〉 Contribution

Model-Checking $\{\mathrm{CTL}^\star, \mathrm{CTL}, \mathrm{CTL}_{\mathsf{EF}}\}$

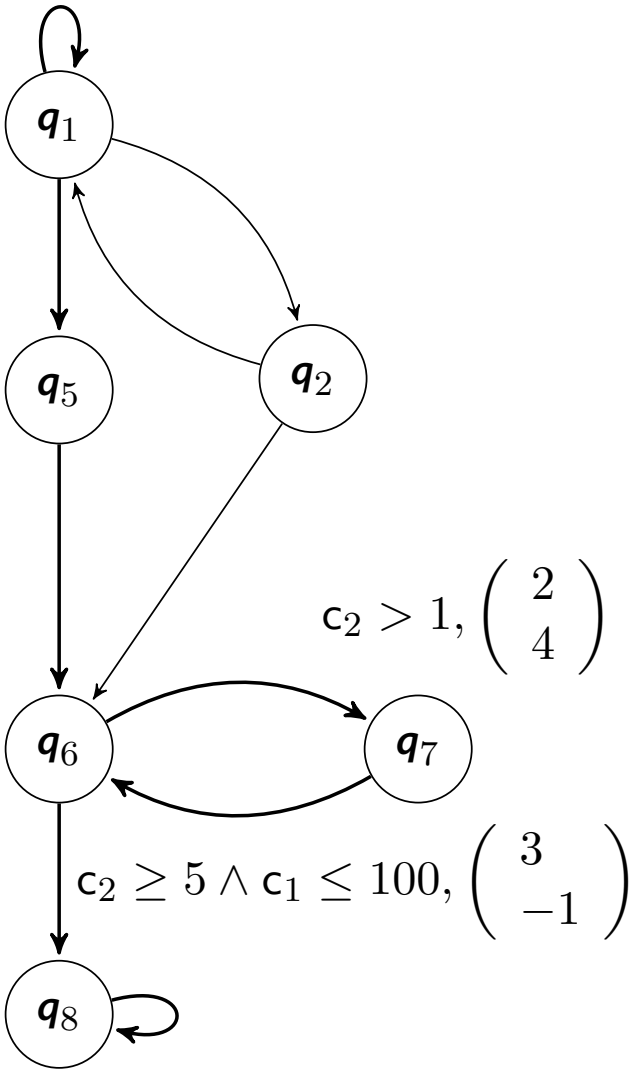
over Flat Counter Systems

is Equivalent to

Satisfiability of Presburger Arithmetic.

$\star$ LogSpace reduction for both sides.

Models 〉 Counter Systems



Counters : $\{c_1, c_2, \ldots, c_n\}$

Updates : $\mathbf{u} \in \mathbb{Z}^n$.

Guards : Boolean Combination of arithmetic constraints

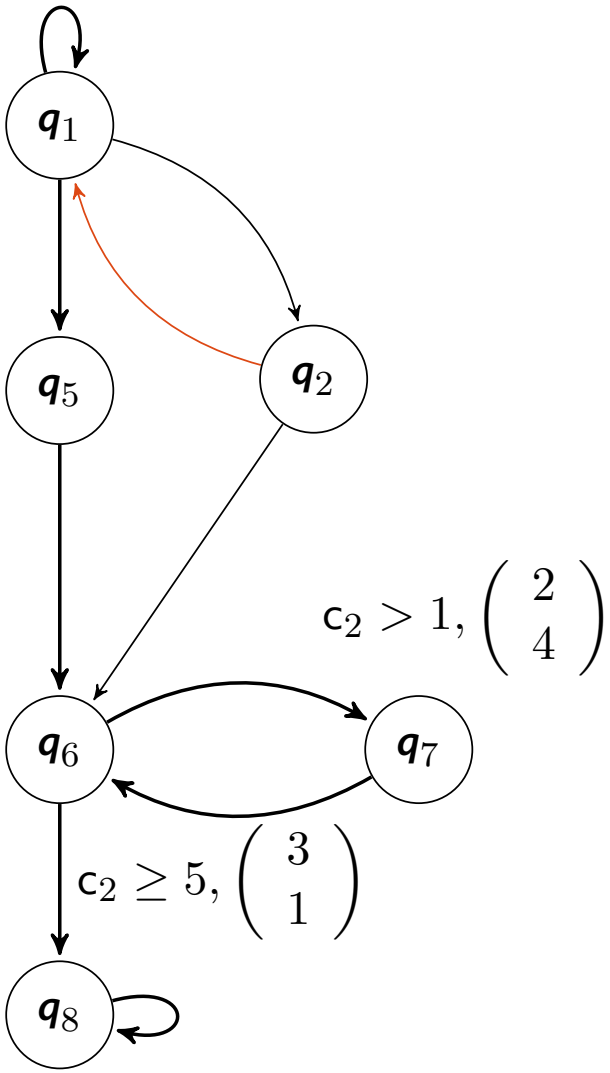
$2.c_1 + 5.c_2 - c_3 \ \{\leq, \geq, <, >\} \ 5.$

Runs of counter systems :

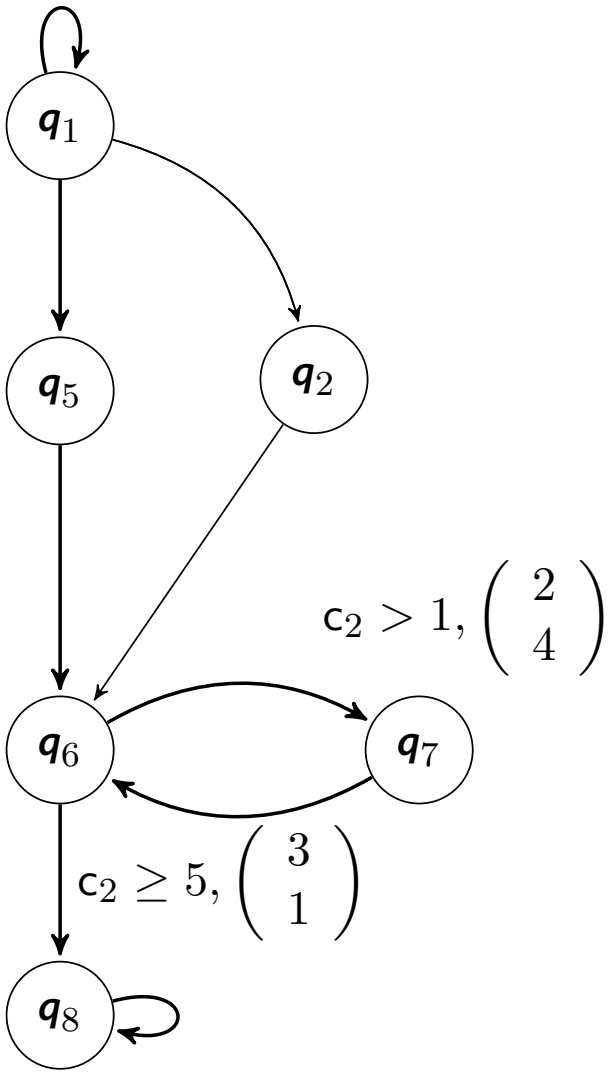
$$\rho = \langle q_1, \mathbf{v}_1 \rangle, \langle q_2, \mathbf{v}_2 \rangle, \ldots, \langle q_j, \mathbf{v}_j \rangle \ldots$$

Models 〉 Flat Counter Systems

No intersecting/nested loops in the structure.



Non-Flat



Flat

Models 〉 Flat Counter Systems

Can still be used to model some systems e.g. Broadcast Protocols

[Finkel, Leroux - FSTTCS'02, Fribourg, Olsén - LOPSTR'96]

Under-approximation of model-checking of counter systems.

[Boigelot - 98, Comon, Jurski - CAV'98, Leroux, Sutre - ATVA'05]

Checking safety property on flat systems with octagonal loop is NP-Complete.

[Bozga, Iosif, Konceny - VMCAI'14]

Computation Tree Logic (CTL)

$$\phi := \boldsymbol{p} \mid g \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{EX}\phi \mid \mathsf{AX}\phi \mid \mathsf{E}[\phi \mathsf{U} \phi] \mid \mathsf{A}[\phi \mathsf{U} \phi].$$

Computation Tree Logic* ($\mathrm{CTL}^\star$)

$$\phi := \boldsymbol{p} \mid g \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi \mid \mathsf{E}\phi.$$

Computation Tree Logic with only EF ($\mathrm{CTL}_{\mathsf{EF}}$)

$$\phi := \boldsymbol{p} \mid g \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{EF}\phi.$$

$\star$ Each contains counter constraints

For a run ρ and position i in the run:

$$
\begin{array}{lcl}
\rho, i \models p & \Leftrightarrow & p \in \mathrm{l}(q), \text{ where } \rho(i) = \langle q, \mathbf{v} \rangle \\
\rho, i \models \mathsf{g} & \Leftrightarrow & \mathbf{v} \models \mathsf{g}, \text{ where } \rho(i) = \langle q, \mathbf{v} \rangle \\
\rho, i \models \mathsf{X}\psi & \Leftrightarrow & \rho, i+1 \models \psi \\
\rho, i \models \psi_1 \mathsf{U} \psi_2 & \Leftrightarrow & \rho, j \models \psi_2 \text{ for some } i \leq j \\
 & & \text{ such that } \rho, k \models \psi_1 \text{ for all } i \leq k < j \\
\rho, i \models \mathsf{E}\phi & \Leftrightarrow & \text{there is a run } \rho' \text{ s.t. } \rho'(0) = \rho(i) \text{ and } \rho', 0 \models \phi
\end{array}
$$

For a counter system S and a configuration c, we write $S, c \models \phi$ iff there exists a run ρ of S such that:

- $\rho, 0 \models \phi$
- $\rho(0) = c$

MC (L, FCS)

INPUT : A flat counter system $\boldsymbol{S}$, a specification $\mathcal{A}$ in logic L, a configuration $\langle \boldsymbol{q}_0, \mathbf{v}_0 \rangle$.

OUTPUT : Does $\boldsymbol{S}, \langle \boldsymbol{q}_0, \mathbf{v}_0 \rangle \models \mathcal{A}$?

Problem 〉 Flat Counter Systems

Model-checking for FOCTL* is decidable.

[Demri et al - JANCL'10]

- Obtained by translation into exponential size Presuburger formula.

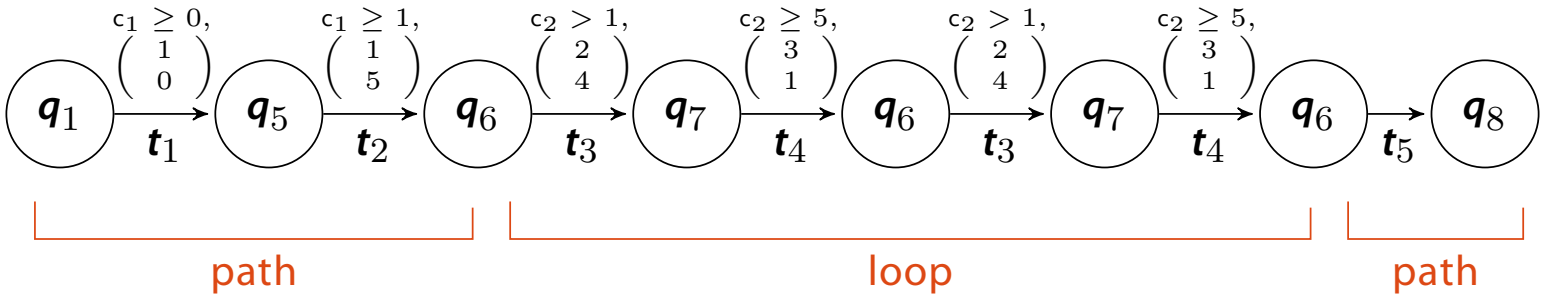
Tight complexity bounds for model-checking linear time properties.

[Demri,D.,Sangnier - IJCAR'12, ICALP'13]

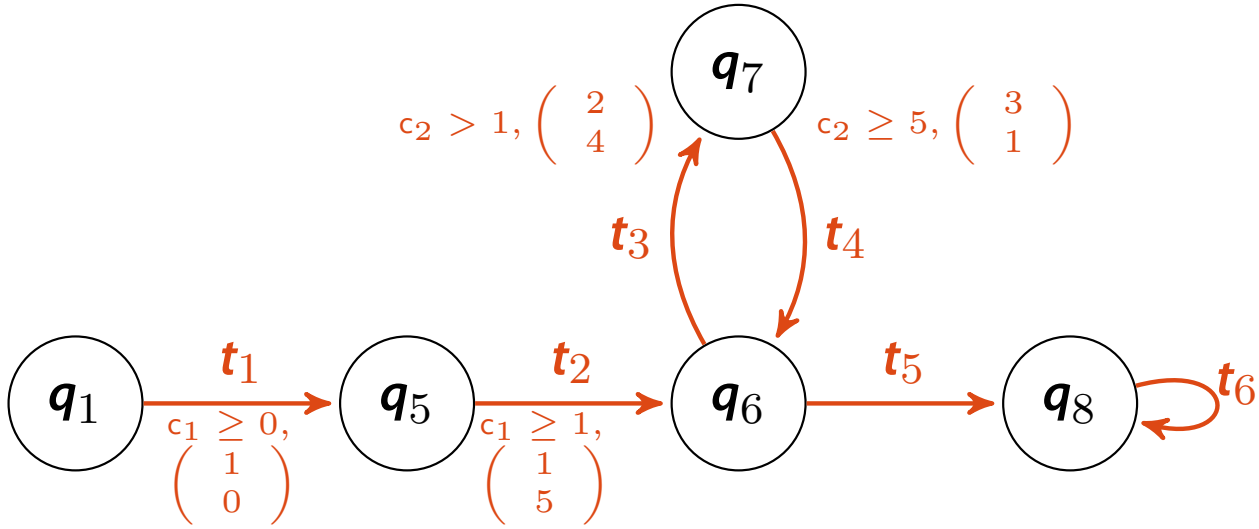
- Past LTL and Büchi Specification - NP-complete.
- FO, Alternating Büchi, Linear μ-calculus - PSPACE-complete.

$\star$ Tight complexity bounds for model-checking branching time properties.

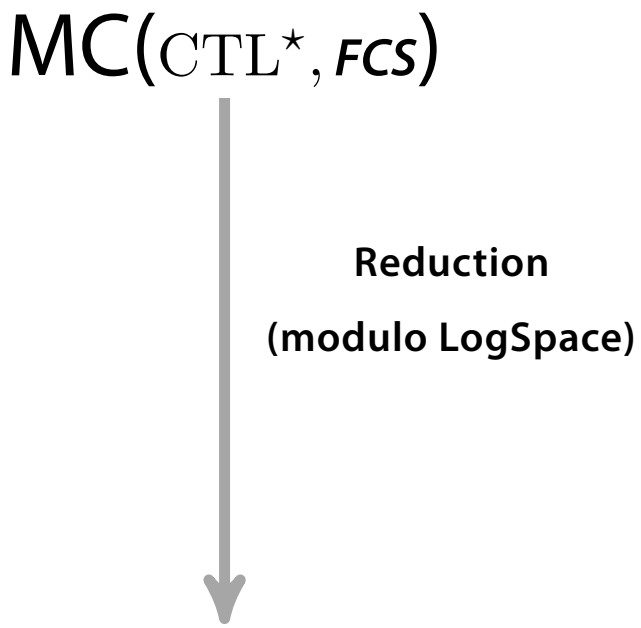
Simpler Models 〉 Path Schemas



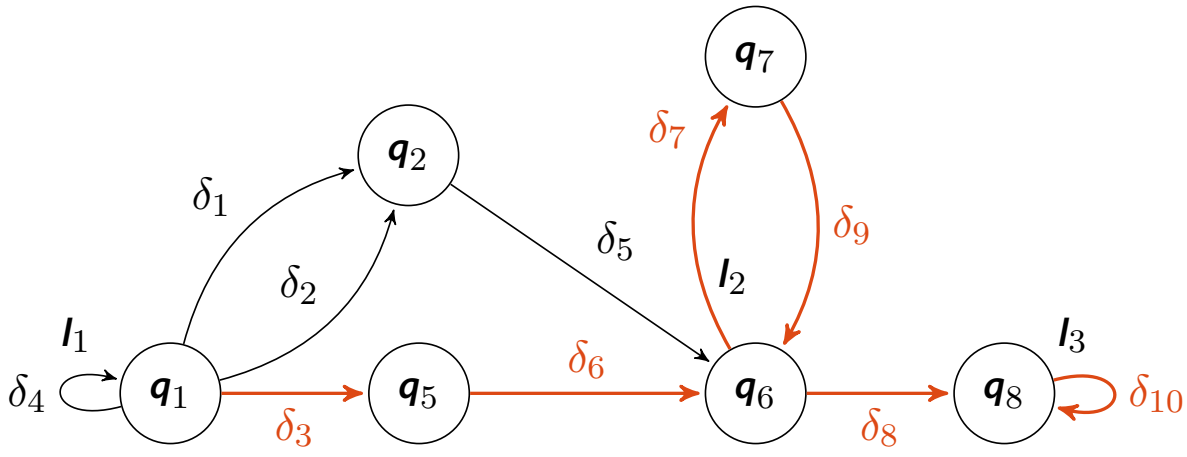
Simpler Models 〉 Path Schemas



- Path Schemas - an alternating sequence of paths and loops - $P = (t_1 t_2)(t_3 t_4)^+ (t_5)(t_6)^\omega$
- A concise way of representing infinite runs = ⟨Path schema, **m**⟩
  - **m** denotes the number of times loops are taken - $\langle P, (2) \rangle$
- At most exponentially many *minimal* path schemas in flat counter systems [Leroux, Sutre - ATVA'05].


MC(CTL*, FCS)
Reduction
(modulo LogSpace)
Satisfiability of Presburger Arithmetic

Encoding Run



$$P = \delta_3 \cdot \delta_6 \cdot (l_2)^{146} \cdot \delta_8 \cdot (l_3)^{\omega}$$

- $\mathbf{v}_p$ - sequence of transition and loop numbers.
  $\mathbf{v}_p = (3, 6, 2, 8, 3, 0, 0, 0, 0, 0, 0, 0, 0)$
- $\mathbf{v}_t$ - Loop or transition ??
  $\mathbf{v}_t = (0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0)$
- $\mathbf{v}_{it}$ - Number of iterations
  $\mathbf{v}_{it} = (1, 1, 146, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$

Encoding Run

$$\delta_3 \cdot \delta_6 \cdot (l_2)^{146} \cdot \delta_8 \cdot (l_3)^{\omega} = \begin{cases} \mathbf{v}_p = (3,6,2,8,3,0,0,0,0,0,0,0,0) \\ \mathbf{v}_t = (0,0,1,0,1,0,0,0,0,0,0,0,0) \\ \mathbf{v}_{it} = (1,1,146,1,0,0,0,0,0,0,0,0,0) \end{cases}$$

1. ϕ_{ps} - Characterizing the properties of path schema
   - $\mathbf{v}_t, \mathbf{v}_p, \mathbf{v}_{it} \models \phi_{PS}$ iff $\mathbf{v}_t, \mathbf{v}_p, \mathbf{v}_{it}$ correspond to the encoding of a path schema
2. ϕ_{run} - Characterizing the runs through path schema
   - $\mathbf{v}_t, \mathbf{v}_p, \mathbf{v}_{it}, c_0 \models \phi_{run}$ iff $\mathbf{v}_t, \mathbf{v}_p, \mathbf{v}_{it}$ correspond to a valid run starting at c_0
3. ϕ_{conf} - Characterizing the runs through path schema
   - $\mathbf{v}_t, \mathbf{v}_p, \mathbf{v}_{it}, c_0, i, c \models \phi_{conf}$ iff $\mathbf{v}_t, \mathbf{v}_p, \mathbf{v}_{it}$ correspond to a valid run starting at c_0 in which i^{th} configuration is c.

Encoding Run

$$\delta_3 \cdot \delta_6 \cdot (l_2)^{146} \cdot \delta_8 \cdot (l_3)^{\omega} = \begin{cases} \mathbf{v}_p = (3,6,2,8,3,0,0,0,0,0,0,0,0) \\ \mathbf{v}_t = (0,0,1,0,1,0,0,0,0,0,0,0,0) \\ \mathbf{v}_{it} = (1,1,146,1,0,0,0,0,0,0,0,0,0) \end{cases}$$

ϕ_{ps} - Characterizing the properties of path schema

- Two loops cannot be adjacent segments of path schema.

$$\bigwedge_{i=1}^{8} ((x_t^i = 1 \wedge x_t^{i+2} = 1) \wedge (x_p^i > 0 \wedge x_p^{i+2} > 0)) \Rightarrow (x_t^{i+1} = 0)$$

Encoding Run

$$\delta_3 \cdot \delta_6 \cdot (l_2)^{146} \cdot \delta_8 \cdot (l_3)^{\omega} = \begin{cases} \mathbf{v}_p = (3,6,2,8,3,0,0,0,0,0,0,0,0,0) \\ \mathbf{v}_t = (0,0,1,0,1,0,0,0,0,0,0,0,0,0) \\ \mathbf{v}_{it} = (1,1,146,1,0,0,0,0,0,0,0,0,0,0) \end{cases}$$

ϕ_{ps} - Characterizing the properties of path schema

- Two loops cannot be adjacent segments of path schema.

$$\bigwedge_{i=1}^{8} ((x_t^i = 1 \wedge x_t^{i+2} = 1) \wedge (x_p^i > 0 \wedge x_p^{i+2} > 0)) \Rightarrow (x_t^{i+1} = 0)$$

- Iterations of transition can only be 1 and for loops it is greater than or equal to 1.

$$\bigwedge_{i=1}^{8} ((x_t^i = 0 \Rightarrow x_{it}^i = 1) \wedge (x_t^i = 1 \Rightarrow x_{it}^i \geq 1))$$

Encoding Run

$$\delta_3 \cdot \delta_6 \cdot (l_2)^{146} \cdot \delta_8 \cdot (l_3)^{\omega} = \begin{cases} \mathbf{v}_p = (3,6,2,8,3,0,0,0,0,0,0,0,0,0) \\ \mathbf{v}_t = (0,0,1,0,1,0,0,0,0,0,0,0,0,0) \\ \mathbf{v}_{it} = (1,1,146,1,0,0,0,0,0,0,0,0,0,0) \end{cases}$$

ϕ_{run} - Characterizing the runs through path schema

- Use witness configurations
  - After δ_3, after $\delta_3 \cdot \delta_6$, after $\delta_3 \cdot \delta_6 \cdot (l_2)^{146}$, etc.
- The number of needed witness configurations is bounded and polynomial
- Check that the next guard is satisfied for each witness configuration
- For each iteration of loops one can compute, the seen configurations and check that the guards are satisfied

Encoding Run

$\phi_{\mathrm{CTL}^\star}$ - Encoding the $\mathrm{CTL}^\star$ formula [Demri et al. - JANCL'10]

$$
\begin{array}{lcl}
Check_{p} & \stackrel{\text{def}}{=} & \exists\, \mathsf{X}\, (Conf(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X}) \wedge \bigvee_{\{j \mid p \in \mathbf{l}(j)\}} \mathsf{x}^1 = j) \\
Check_{\psi(\mathsf{x}_1,\ldots,\mathsf{x}_n)} & \stackrel{\text{def}}{=} & \exists\, \mathsf{X}\, (Conf(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X}) \wedge \psi(\mathsf{X})) \\
Check_{\mathsf{E}\phi} & \stackrel{\text{def}}{=} & \exists\, \mathsf{Z}'_t\, \exists\, \mathsf{Z}'_p\, \exists\, \mathsf{Z}'_{it}\, \exists\, \mathsf{X}\, (Conf(\mathsf{Z}_t, \mathsf{Z}_p, \mathsf{Z}_{it}, \mathsf{X}_0, \mathsf{y}, \mathsf{X}) \wedge \\
 & & Run(\mathsf{Z}'_t, \mathsf{Z}'_p, \mathsf{Z}'_{it}, \mathsf{X}) \wedge \\
 & & \exists\, \mathsf{y}'\, (\mathsf{y}' = 0 \wedge Check_{\phi}(\mathsf{Z}'_t, \mathsf{Z}'_p, \mathsf{Z}'_{it}, \mathsf{X}, \mathsf{y}')))
\end{array}
$$

Property of encoding

$\mathbf{v}_t, \mathbf{v}_p, \mathbf{v}_{it}, \mathbf{c}_0, i \models \phi_\psi$ iff the run characterized by $\mathbf{v}_t, \mathbf{v}_p, \mathbf{v}_{it}$ and starting at $\mathbf{c}_0$ satisfies ψ at position i.

Reduction

$$\phi = \exists \mathsf{x}_{p}^{1} \cdots \mathsf{x}_{p}^{10}, \mathsf{x}_{t}^{1} \cdots \mathsf{x}_{t}^{10}, \mathsf{x}_{it}^{1} \cdots \mathsf{x}_{it}^{10}.(\phi_{ps} \wedge \phi_{run} \wedge \phi_{\mathrm{CTL}^\star})$$

- Polynomial-time reduction compared to exponential time reduction known from [Demri et al. - JANCL'10].
  - No enumeration of path schemas in formula.
  - Encoding runs using a constant number of fixed size integer vectors.
  - Utilizing the power of quantifiers in an essential way.
- Easily extendible for past time operators.

Satisfiability of Presburger Arithmetic

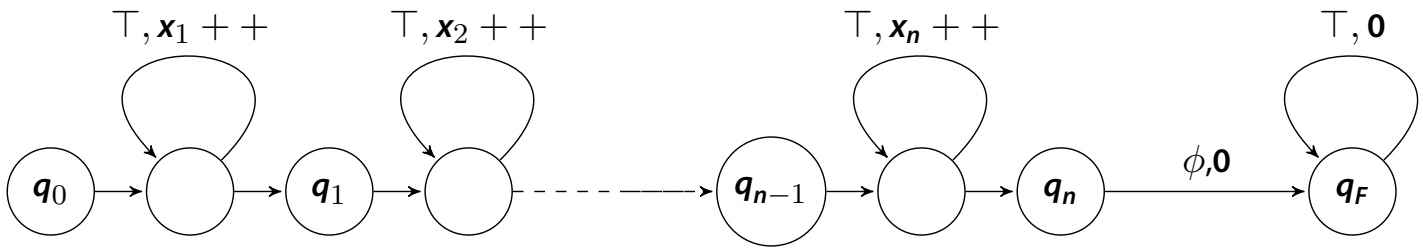
Reduction

(modulo LogSpace)

MC($\mathrm{CTL}_{\mathrm{EF}}$, FCS)

PA Formula : $Q_1 x_1 Q_2 x_2 \cdots Q_n x_n \phi(x_1, x_2, \cdots, x_n)$

$Q_1, Q_2, \cdots Q_n \in \{\exists, \forall\}$



$$\psi_i \stackrel{\text{def}}{=} \begin{cases} \mathsf{EF}(q_i \wedge \psi_{i+1}) & \text{if } Q_i = \exists \\ \mathsf{AG}(q_i \Rightarrow \psi_{i+1}) & \text{if } Q_i = \forall \end{cases}$$

$$\psi \stackrel{\text{def}}{=} \psi_1; \psi_{n+1} \stackrel{\text{def}}{=} \mathsf{EF} q_f$$

Reduction

- Logspace translation
- Number of quantifiers in PA formula = the number of path operators in $\mathrm{CTL}_{\mathrm{EF}}$ formula.
- Quantifier alternation in PA formula = the alternation of path quantifiers in $\mathrm{CTL}_{\mathrm{EF}}$ formula.
- Number of loops in the flat counter system = the number of quantified variables in PA formula.

MC(CTL^*,FCS)

$\updownarrow$

MC(CTL,FCS)

$\updownarrow$

MC($\mathrm{CTL}_{\mathsf{EF}}$,FCS)

$\updownarrow$

Satisfiability of Presburger arithmetic

- Extending the result to more expressive logic : Modal μ-calculus
- Extending with more powerful model : Affine update, difference bound guards.

That's It 〉 Questions?

Thank You

For Your Kind Attention