# Model Checking a Path

**N. Markey**[*°], Ph. Schnoebelen[*]

{markey,phs}@lsv.ens-cachan.fr

[*] LSV, ENS de Cachan & CNRS UMR 8643

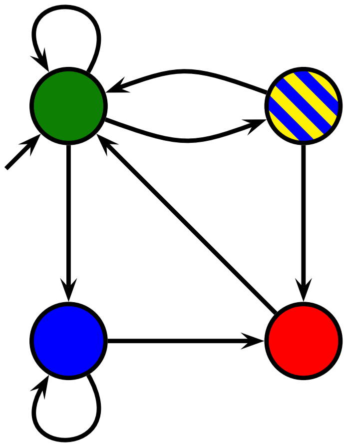
[°] LIFO, Univ. Orléans & CNRS FRE 2490

# LTL Model Checking

Verify that a specification holds along all paths of a model

# LTL Model Checking

Verify that a specification holds along all paths of a model



$$LTL \ni \varphi ::= \neg\varphi \mid \varphi \vee \varphi \mid X\,\varphi \mid \varphi\, U\, \varphi \mid \quad \mid \ldots$$

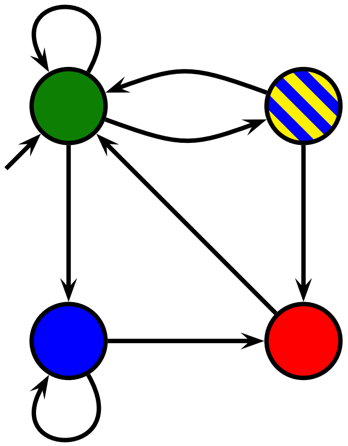
Abbreviations:

$$F\varphi = \top U \varphi$$

$$G\varphi = \neg F \neg \varphi$$

# LTL Model Checking

Verify that a specification holds along all paths of a model



$LTL \ni \varphi ::= \neg\varphi \mid \varphi \vee \varphi \mid X\,\varphi \mid \varphi\, U\, \varphi \mid ● \mid \ldots$

Abbreviations:

$F\varphi = \top U \varphi$

$G\varphi = \neg F \neg \varphi$

Example:

$G(● \vee (● \, U \, ●))$

# LTL Path Model Checking

Verify that a specification holds along one given path

# LTL Path Model Checking

Verify that a specification holds along one given path

Several interesting occurences:

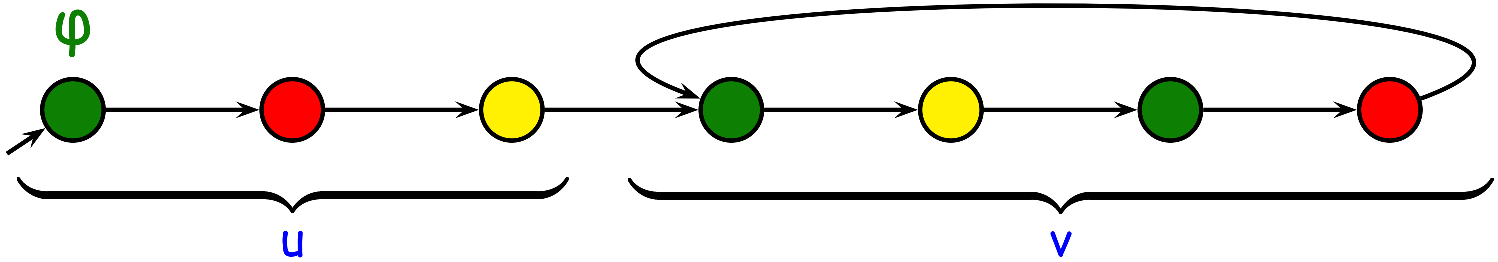
- Testing;
- Model checking probabilistic systems;
- On-the-fly and runtime model checking;
- Analysis of counter-examples;
- Analysis of log files;
- ...

# The PMC problem

**PMC:**

Input: two finite words $u$ and $v$, and a formula $\varphi$.

Output: yes iff $u \cdot v^\omega \models \varphi$.

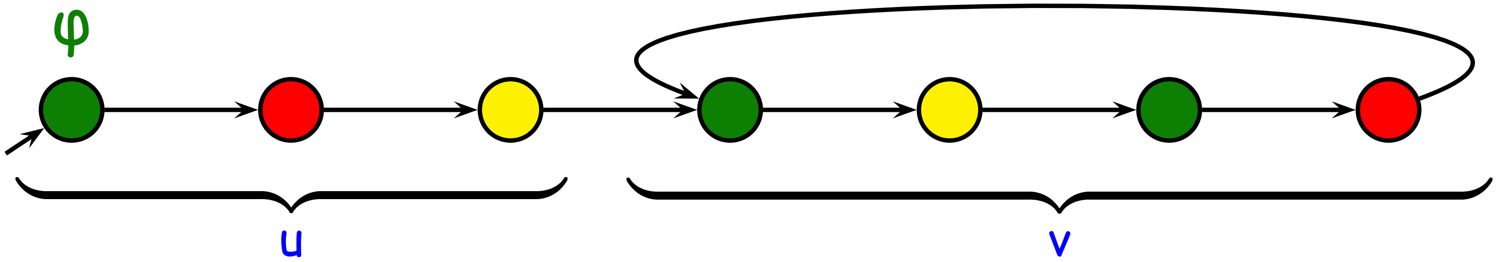


# The PMC problem

PMC:

Input: two finite words u and v, and a formula $\varphi$.

Output: yes iff $u \cdot v^{\omega} \models \varphi$.




Note: v can be empty, i.e. we deal with finite paths as well as with infinite paths.

# Our results

- PMC is easy: much easier than standard MC
    - can efficiently handle more expressive logics,
    - can efficiently handle compressed paths.
- Ultimately periodic paths are not more difficult than finite paths,
- Polynomial time algorithms by using dynamic programming techniques.

# Outline of the talk

- LTL Path Model Checking
- PMC for richer logics:
  - LTL + Past, LTL + Past + Now
  - LTL + Chop
  - first order logic
- PMC for compressed paths

# Outline of the talk

- LTL Path Model Checking
- PMC for richer logics:
  - LTL + Past, LTL + Past + Now
  - LTL + Chop
  - first order logic
- PMC for compressed paths

# LTL PMC

- For a single path, LTL = CTL.

  $\rightsquigarrow$ algorithm in time $O(|\text{model}| \times |\text{formula}|)$

# LTL PMC

- For a single path, LTL = CTL.

  $\leadsto$algorithm in time $O(|\text{model}| \times |\text{formula}|)$

- lower bound?
  - PTIME-hardness for CTL standard model checking
  - the best we have is $NC^1$-hardness for LTL PMC

# LTL PMC

- For a single path, LTL = CTL.

  $\leadsto$algorithm in time $O(|\text{model}| \times |\text{formula}|)$

- lower bound?
  - PTIME-hardness for CTL standard model checking
  - the best we have is $NC^1$-hardness for LTL PMC

PTIME

LOG(CFL)

NLOGSPACE

LOGSPACE

$NC^1$ = ALOGTIME

$AC^0$

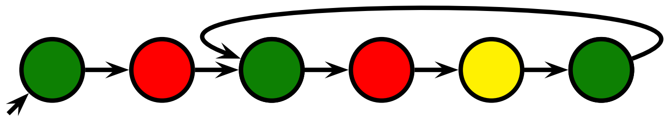
Large (?) gap between upper and lower bound.

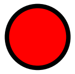
# LTL PMC

PMC can be reduced to finite PMC.

# LTL PMC

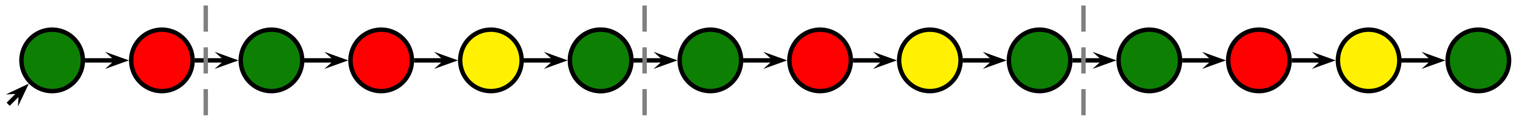
PMC can be reduced to finite PMC.

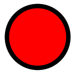


G F 

# LTL PMC

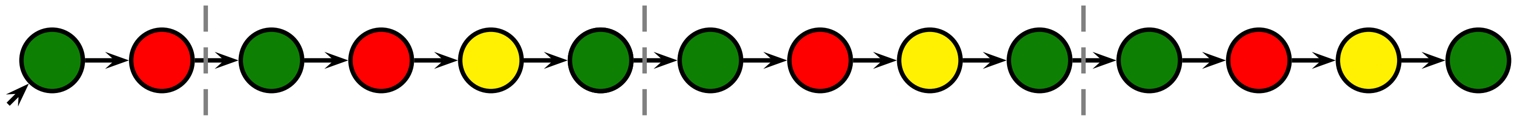
PMC can be reduced to finite PMC.



G F 

# LTL PMC

PMC can be reduced to finite PMC.

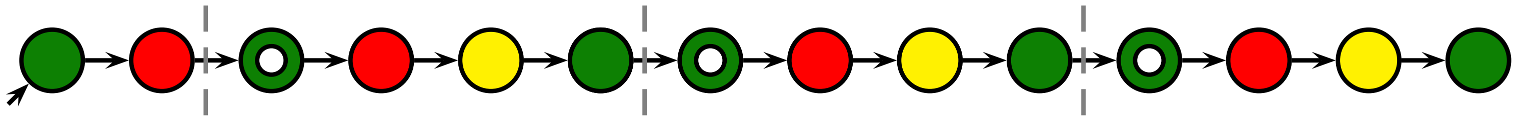


G F ●

We have to modify the formula (for example $\mathbf{G}\ \mathbf{X}\ \top$)

# LTL PMC

PMC can be reduced to finite PMC.



G F

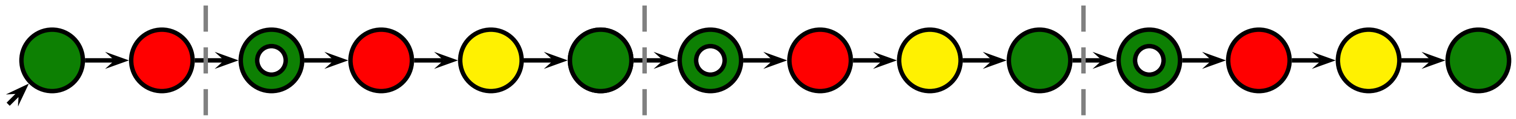
We have to modify the formula (for example G X $\top$)

$$\overline{\varphi\ U\ \psi} = \overline{\varphi}\ U\ (\overline{\psi} \wedge \gamma_{h(\varphi\ U\ \psi)})$$

with $\gamma_p = F(\circ \wedge X\gamma_{p-1})$ and $\gamma_1 = \top$

# LTL PMC

PMC can be reduced to finite PMC.



$G\,F\,\bullet$

We have to modify the formula (for example $G\,X\,\top$)

$$\overline{\varphi\; U\; \psi} = \overline{\varphi}\; U\; (\overline{\psi} \wedge \gamma_{h(\varphi\, U\, \psi)})$$

with $\gamma_p = F(\circ \wedge X\gamma_{p-1})$ and $\gamma_1 = \top$

For example:

$$\overline{G\,F\,\bullet} = G\,(F\,(\bullet \wedge \gamma_1) \vee \neg\gamma_2)$$

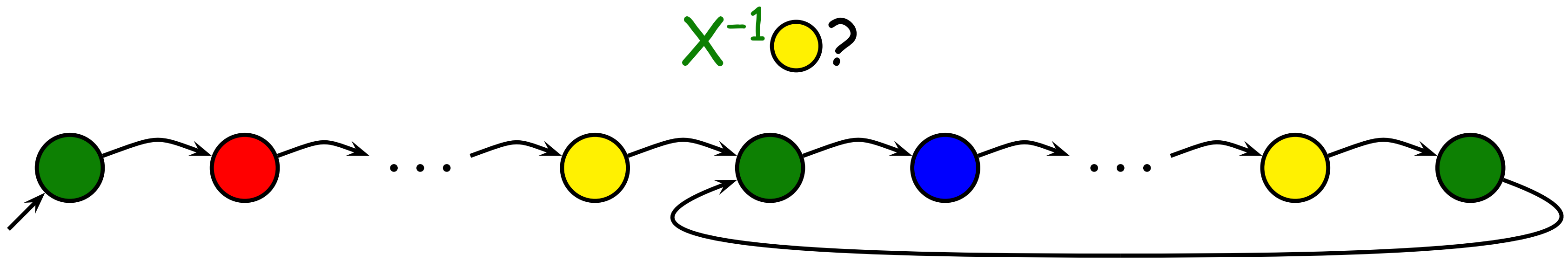
# Outline of the talk

- LTL Path Model Checking
- PMC for richer logics:
  - LTL + Past, LTL + Past + Now
  - LTL + Chop
  - first order logic
- PMC for compressed paths

# Complexity of PMC

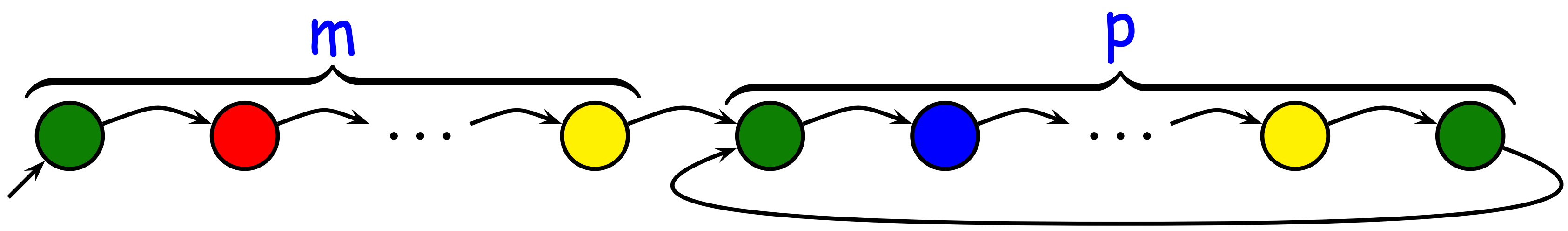
| Logic | MC | Path MC |
| --- | --- | --- |
| LTL | PSPACE-complete | PTIME-easy |
| LTL+Past | PSPACE-complete | |
| LTL+Past+Now | EXPSPACE-complete | |
| LTL+Chop | nonelementary | |
| FOMLO | nonelementary | |

# PMC for LTL+Past




We cannot apply a CTL-like algorithm since past is not "deterministic".

# PMC for LTL+Past

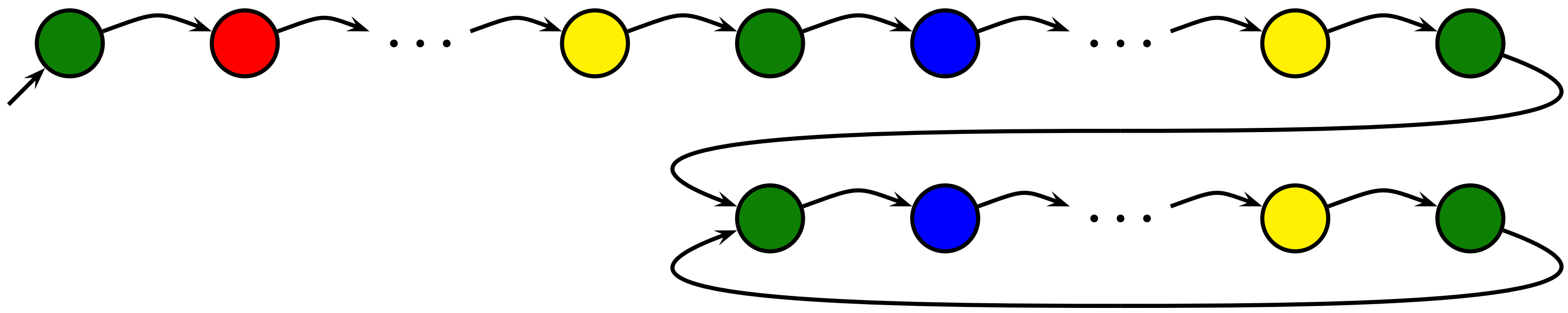



We cannot apply a CTL-like algorithm since past is not "deterministic".

Lemma: For any loop $(u, v)$ with $|u| = m$ and $|v| = p$, for any LTL+Past formula $\varphi$ with at most $h$ nested past-time modalities, and for any $k \geq m + p \cdot h$, we have

$$u \cdot v^\omega, k \models \varphi \quad \Leftrightarrow \quad u \cdot v^\omega, k + p \models \varphi.$$

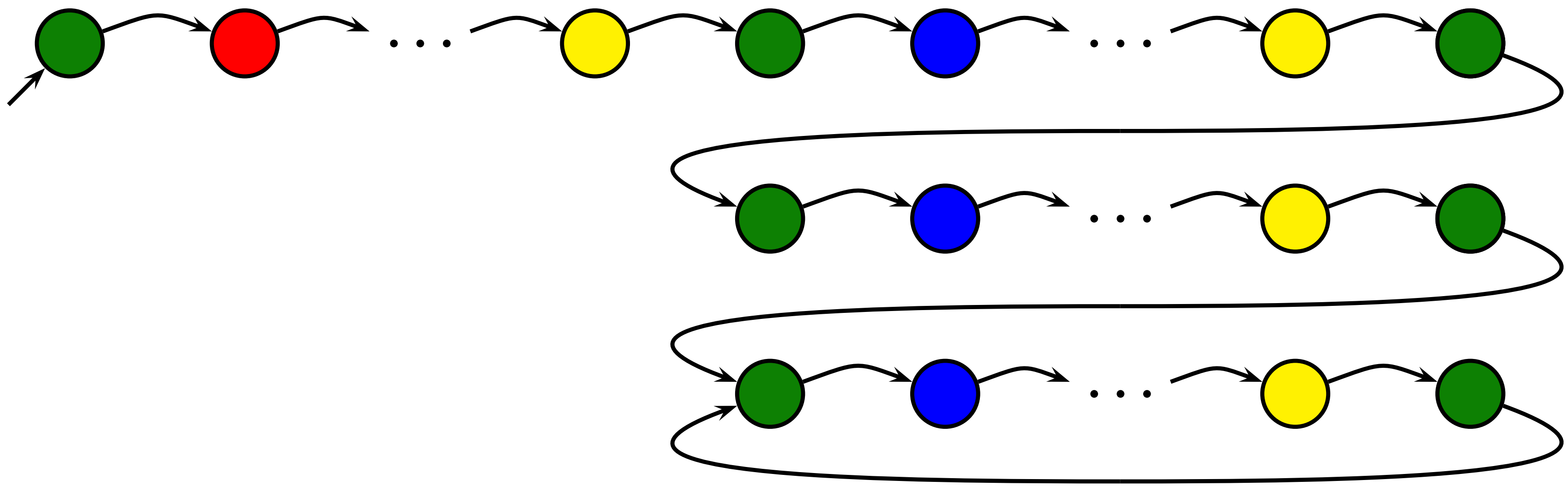
# PMC for LTL+Past



Lemma: For any loop $(u, v)$ with $|u| = m$ and $|v| = p$, for any LTL+Past formula $\varphi$ with at most $h$ nested past-time modalities, and for any $k \geq m + p \cdot h$, we have

$$u \cdot v^{\omega}, k \models \varphi \quad \Leftrightarrow \quad u \cdot v^{\omega}, k + p \models \varphi.$$

# PMC for LTL+Past



Lemma: For any loop $(u, v)$ with $|u| = m$ and $|v| = p$, for any LTL+Past formula $\varphi$ with at most $h$ nested past-time modalities, and for any $k \geq m + p \cdot h$, we have

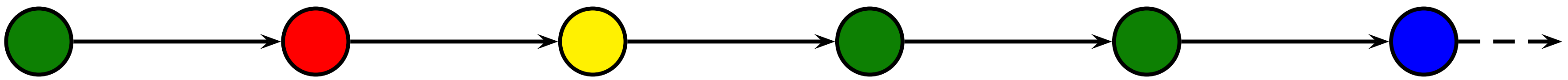
$$u \cdot v^\omega, k \models \varphi \quad \Leftrightarrow \quad u \cdot v^\omega, k + p \models \varphi.$$

# Summary

| Logic | MC | Path MC |
|---|---|---|
| LTL | PSPACE-complete | PTIME-easy |
| LTL+Past | PSPACE-complete | PTIME-easy |
| LTL+Past+Now | EXPSPACE-complete | |
| LTL+Chop | nonelementary | |
| FOMLO | nonelementary | |

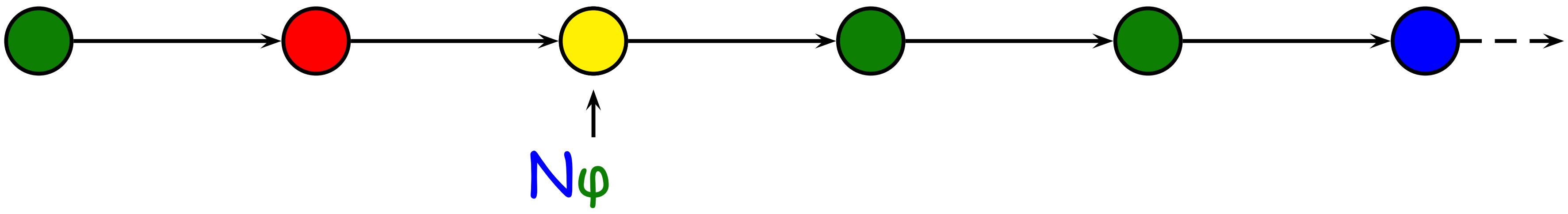
# PMC for LTL+Past+Now

Semantics of Now ([LS95]):



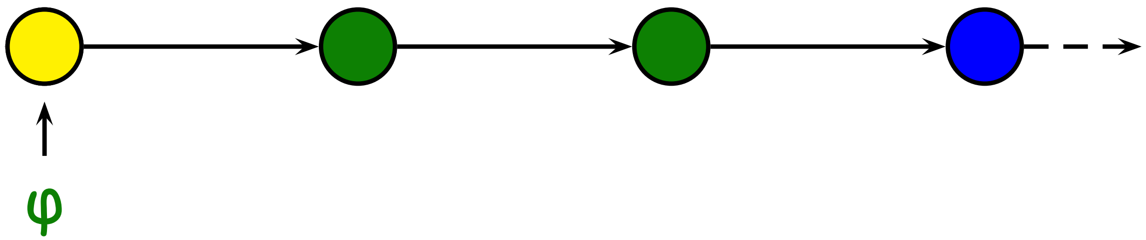
# PMC for LTL+Past+Now

Semantics of Now ([LS95]):



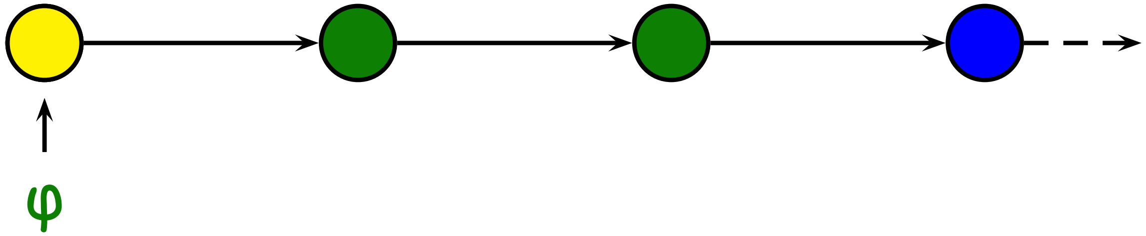
# PMC for LTL+Past+Now

Semantics of Now ([LS95]):



# PMC for LTL+Past+Now

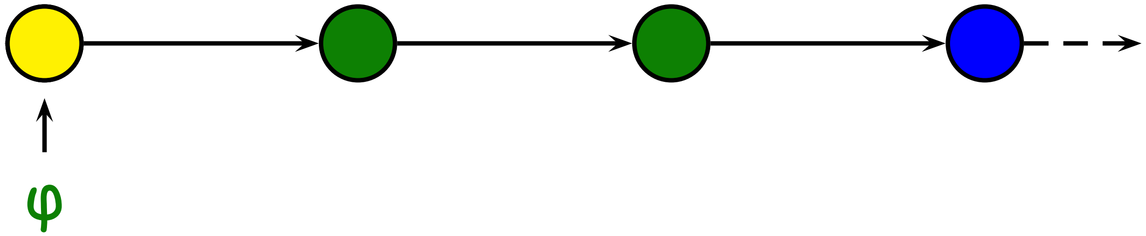
Semantics of Now ([LS95]):




Model checking LTL+Past+Now is EXPSPACE-complete ([LMS02]).

# PMC for LTL+Past+Now

Semantics of Now ([LS95]):




Model checking LTL+Past+Now is EXPSPACE-complete ([LMS02]).

Theorem:
Path Model Checking LTL+Past+Now is PTIME-complete.

# PMC for LTL+Past+Now

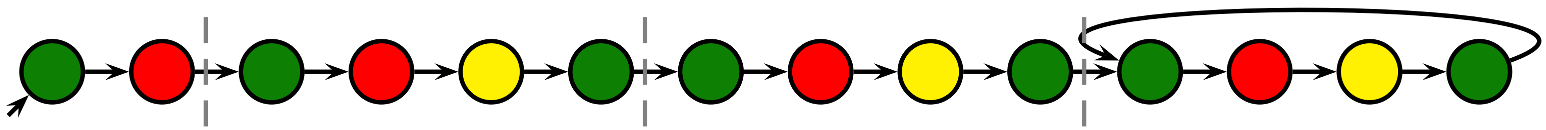
Lemma: For any loop $(u, v)$ with $|u| = m$ and $|v| = p$, for any LTL+Past+Now formula $\varphi$ with at most $h$ nested past-time modalities, and for any $k \geq m + p \cdot h$, we have

$$u \cdot v^{\omega}, k \models \varphi \qquad \Leftrightarrow \qquad u \cdot v^{\omega}, k + p \models \varphi.$$

# PMC for LTL+Past+Now

Lemma: For any loop $(u, v)$ with $|u| = m$ and $|v| = p$, for any LTL+Past+Now formula $\varphi$ with at most $h$ nested past-time modalities, and for any $k \geq m + p \cdot h$, we have

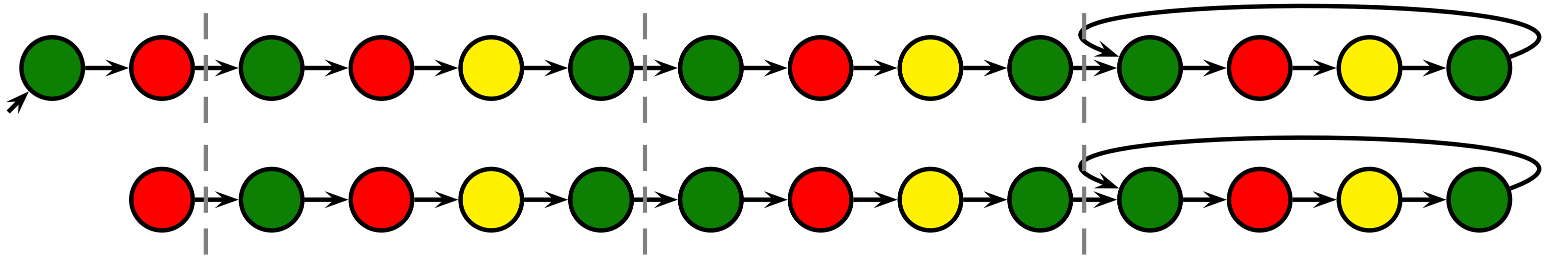
$$u \cdot v^{\omega}, k \models \varphi \qquad \Leftrightarrow \qquad u \cdot v^{\omega}, k + p \models \varphi.$$



# PMC for LTL+Past+Now

Lemma: For any loop $(u, v)$ with $|u| = m$ and $|v| = p$, for any LTL+Past+Now formula $\varphi$ with at most $h$ nested past-time modalities, and for any $k \geq m + p \cdot h$, we have

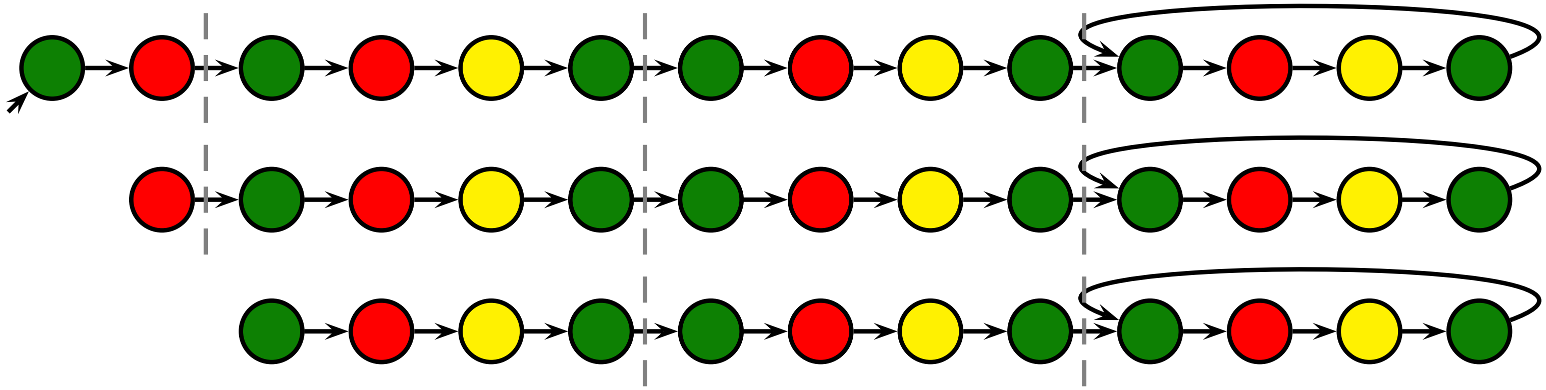
$$u \cdot v^{\omega}, k \models \varphi \qquad \Leftrightarrow \qquad u \cdot v^{\omega}, k + p \models \varphi.$$



# PMC for LTL+Past+Now

Lemma: For any loop $(u, v)$ with $|u| = m$ and $|v| = p$, for any LTL+Past+Now formula $\varphi$ with at most $h$ nested past-time modalities, and for any $k \geq m + p \cdot h$, we have

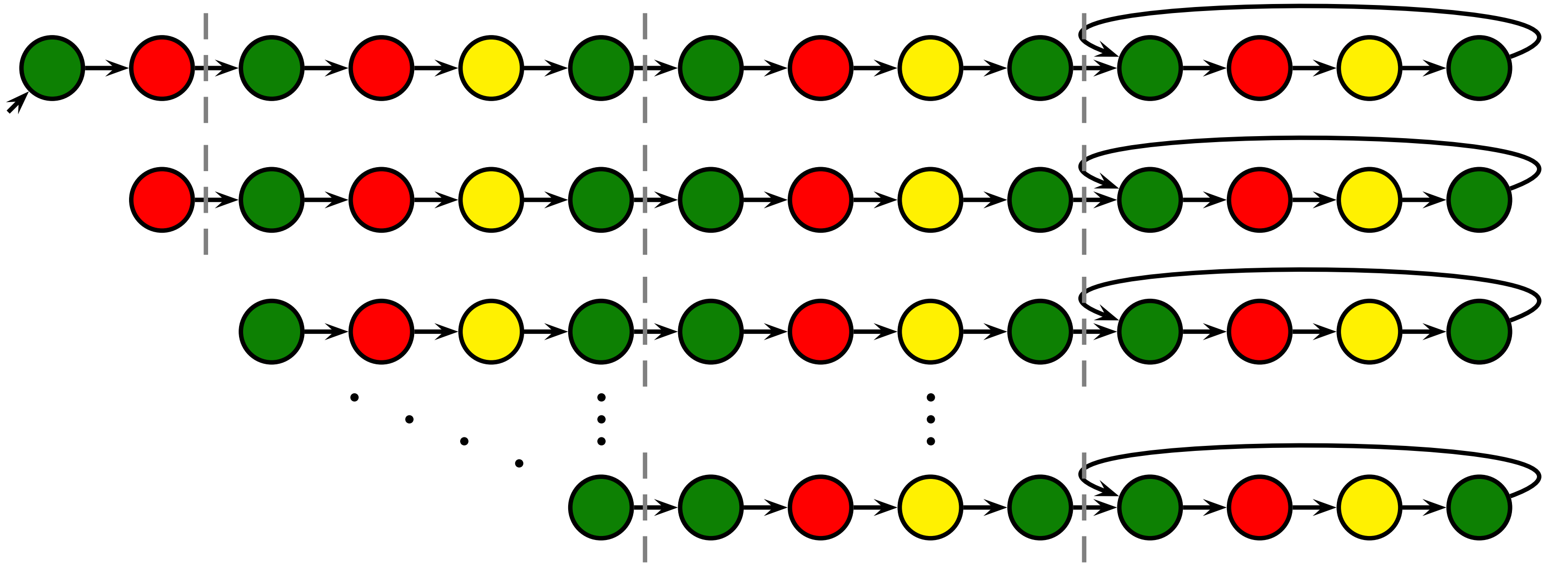
$$u \cdot v^{\omega}, k \models \varphi \quad \Leftrightarrow \quad u \cdot v^{\omega}, k + p \models \varphi.$$



# PMC for LTL+Past+Now

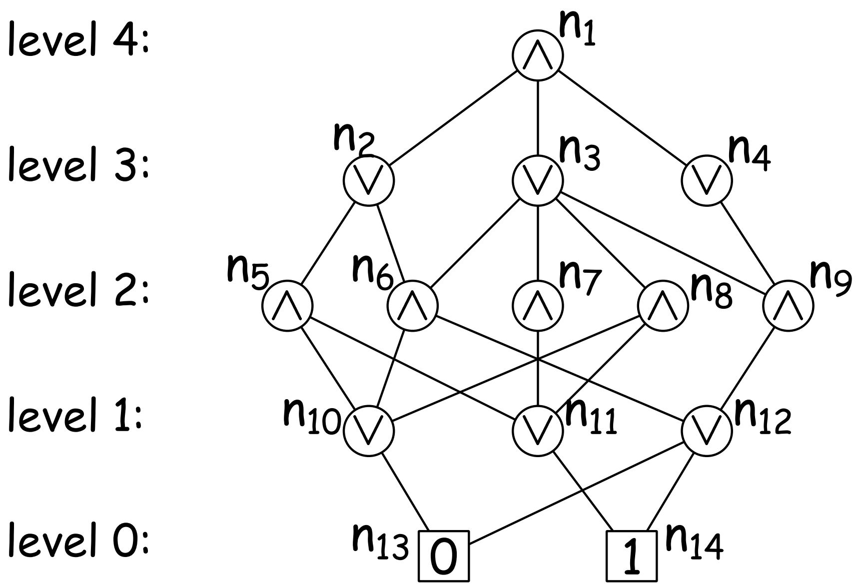
Lemma: For any loop $(u, v)$ with $|u| = m$ and $|v| = p$, for any LTL+Past+Now formula $\varphi$ with at most $h$ nested past-time modalities, and for any $k \geq m + p \cdot h$, we have

$$u \cdot v^{\omega}, k \models \varphi \quad \Leftrightarrow \quad u \cdot v^{\omega}, k + p \models \varphi.$$



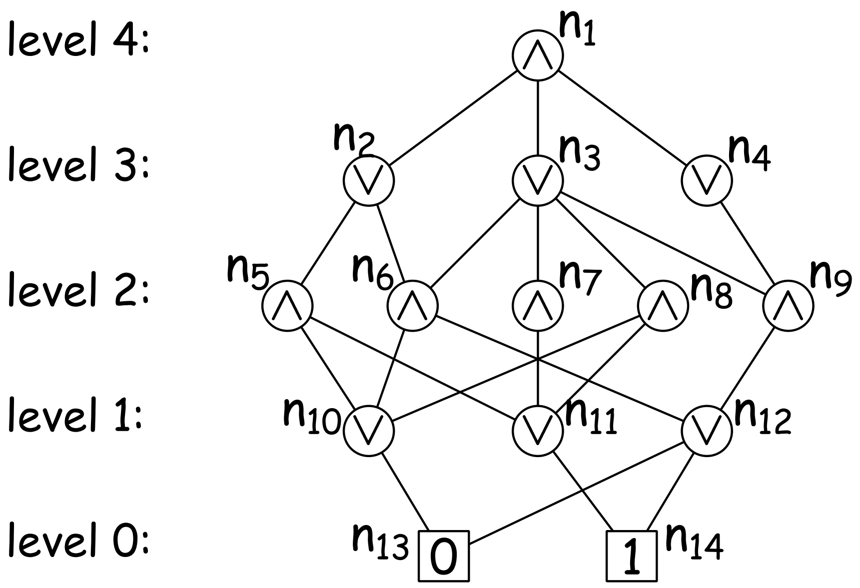
# PMC for LTL+Past+Now

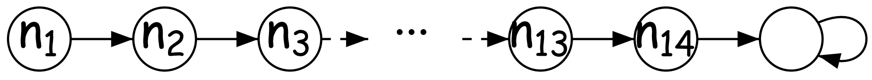
Reduction of Circuit-Value:



# PMC for LTL+Past+Now

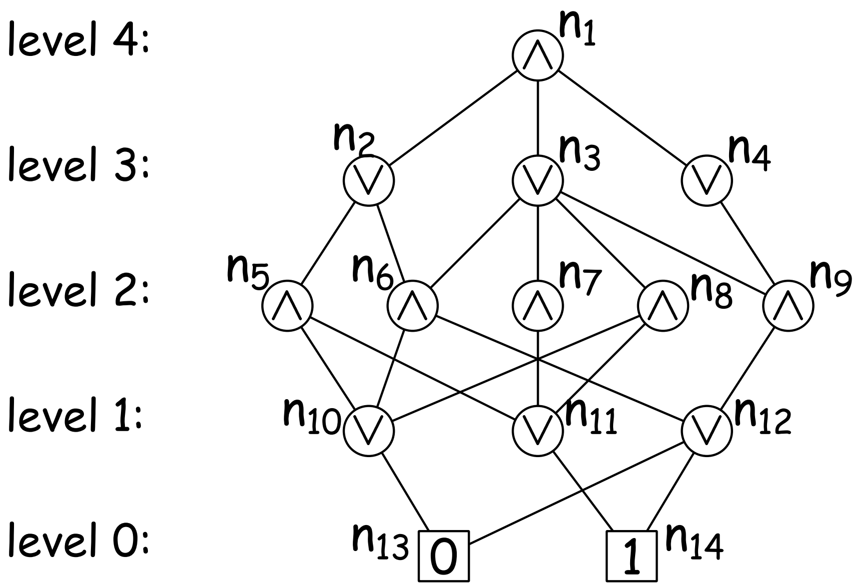
Reduction of Circuit-Value:

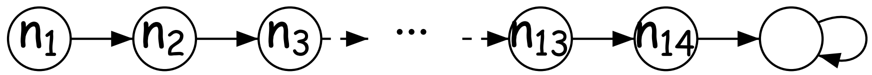


# PMC for LTL+Past+Now

Reduction of Circuit-Value:

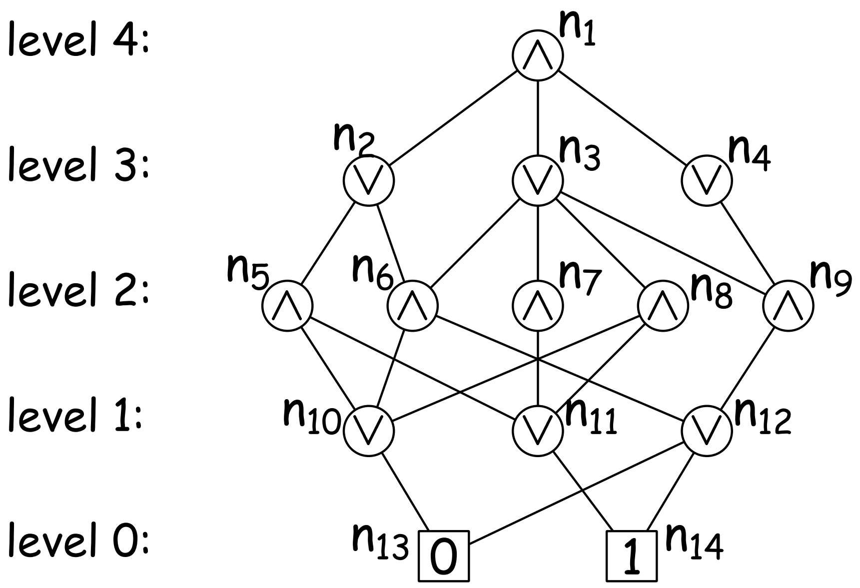


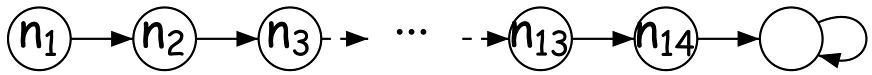




$\varphi_0 = n_{14}$

# PMC for LTL+Past+Now

Reduction of Circuit-Value:



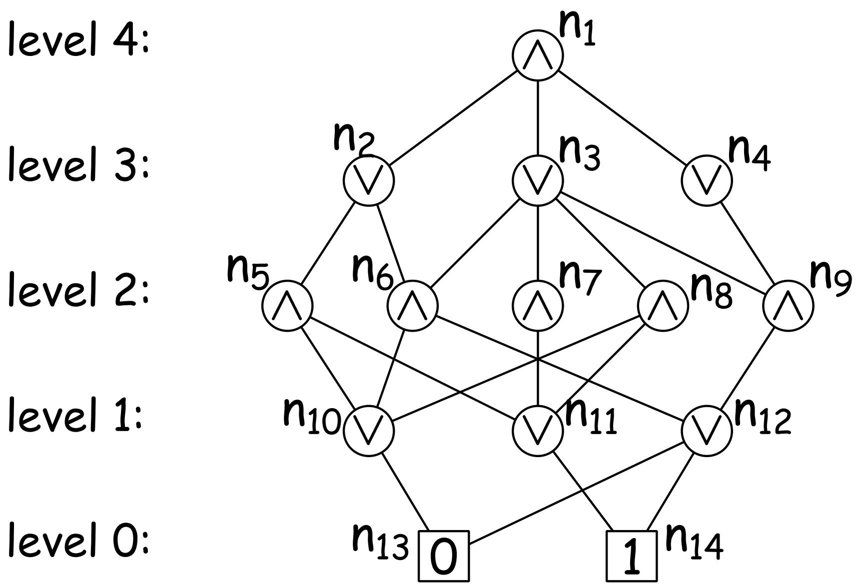



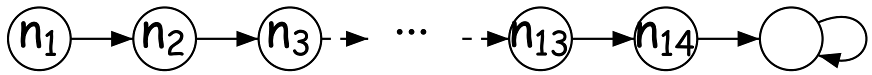

$$\varphi_0 = n_{14}$$

$$\varphi_{2k+1} = \mathrm{NF}(\varphi_{next} \wedge \varphi_{2k})$$

# PMC for LTL+Past+Now

Reduction of Circuit-Value:



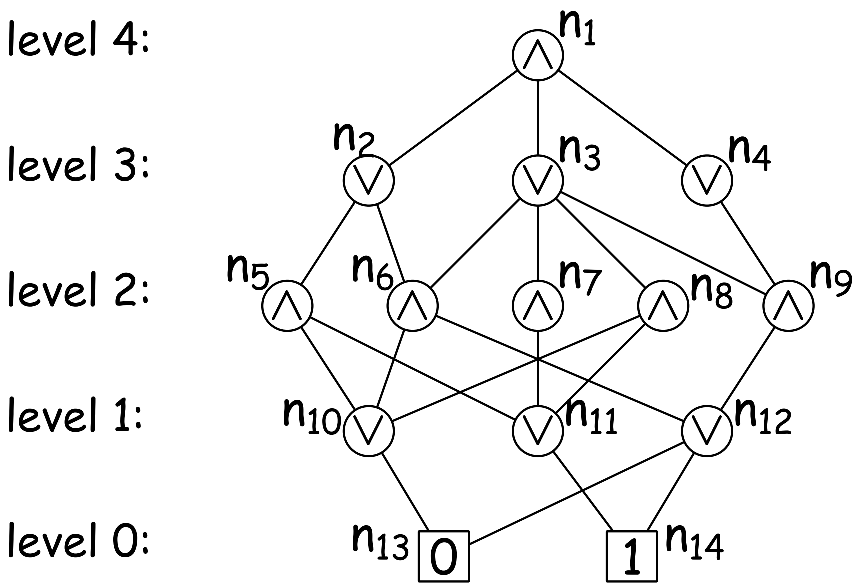

$$\varphi_0 = n_{14}$$

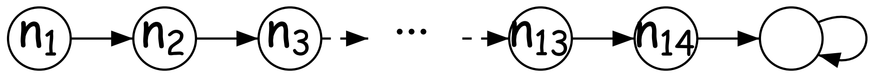
$$\varphi_{2k+1} = \mathrm{NF}(\varphi_{next} \wedge \varphi_{2k})$$

$$\varphi_{2k+2} = \mathrm{NG}(\varphi_{next} \Rightarrow \varphi_{2k+1})$$

# PMC for LTL+Past+Now

Reduction of Circuit-Value:




$$\varphi_0 = n_{14}$$

$$\varphi_{2k+1} = \mathrm{NF}(\varphi_{next} \wedge \varphi_{2k})$$

$$\varphi_{2k+2} = \mathrm{NG}(\varphi_{next} \Rightarrow \varphi_{2k+1})$$

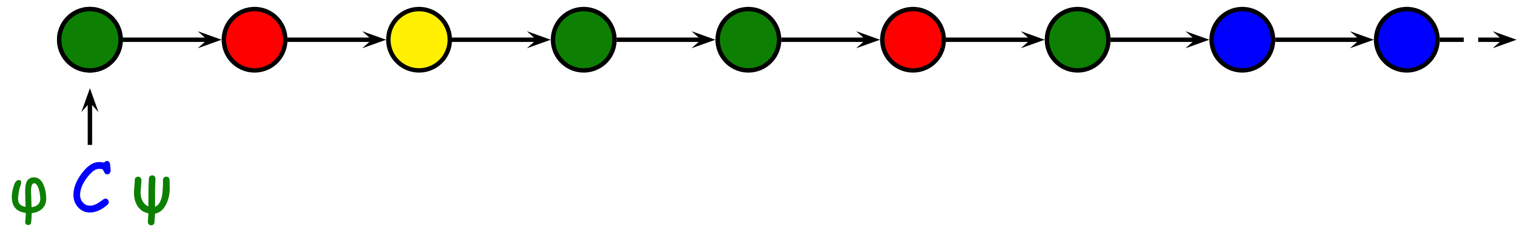
where $\varphi_{next} = \bigvee_{(n,n')\in E} (n' \wedge F^{-1}G^{-1}n)$

# Summary

| Logic | MC | Path MC |
|---|---|---|
| LTL | PSPACE-complete | PTIME-easy |
| LTL+Past | PSPACE-complete | PTIME-easy |
| LTL+Past+Now | EXPSPACE-complete | PTIME-complete |
| LTL+Chop | nonelementary | |
| FOMLO | nonelementary | |

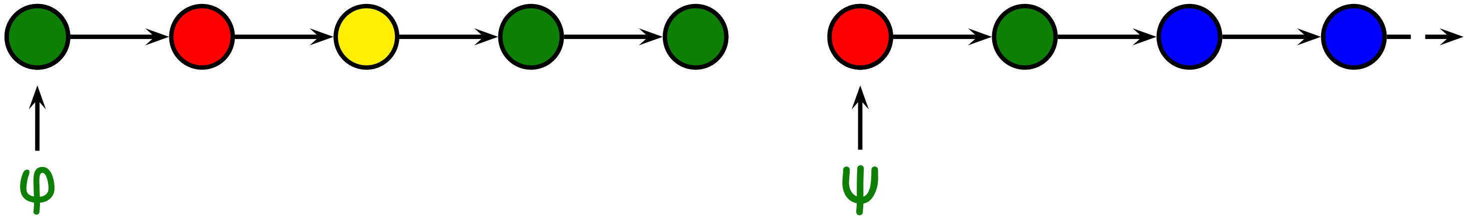
# PMC for LTL+Chop

Semantics of Chop:



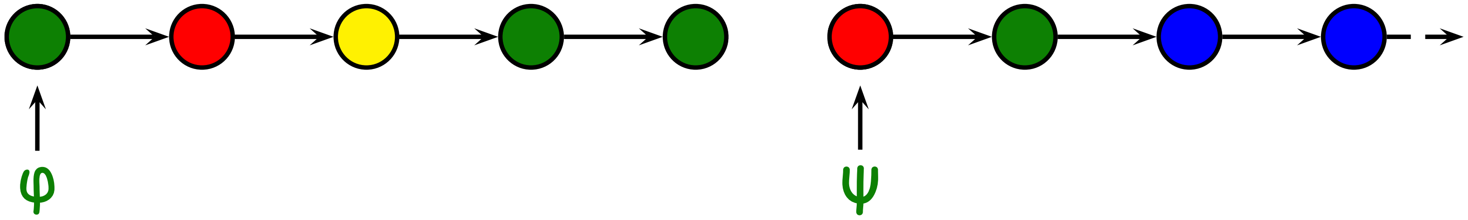
# PMC for LTL+Chop

Semantics of Chop:



# PMC for LTL+Chop

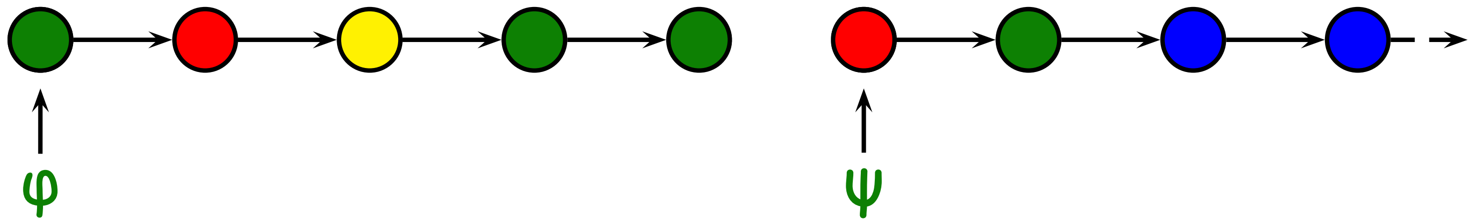
Semantics of Chop:




Model checking LTL+Chop is non-elementary ([RP86]).

# PMC for LTL+Chop

Semantics of Chop:




Model checking LTL+Chop is non-elementary ([RP86]).

Theorem:
Path Model Checking LTL+Chop is PTIME-complete.

# Summary

| Logic | MC | Path MC |
|---|---|---|
| LTL | PSPACE-complete | PTIME-easy |
| LTL+Past | PSPACE-complete | PTIME-easy |
| LTL+Past+Now | EXPSPACE-complete | PTIME-complete |
| LTL+Chop | nonelementary | PTIME-complete |
| FOMLO | nonelementary | |

# PMC for First Order Logic

Model checking FOMLO properties is nonelementary ([Sto74])

Theorem:
Path Model Checking FOMLO is PSPACE-complete.

# PMC for First Order Logic

Model checking FOMLO properties is nonelementary ([Sto74])

Theorem:
Path Model Checking FOMLO is PSPACE-complete.

- PSPACE-hardness: reduction from QBF.
- PSPACE-easiness:
  - a finite (exponential) prefix of the path is sufficient (but, as for LTL, we have to modify the formula)
  - store values of variables in binary.

# Summary

| Logic | MC | Path MC |
|---|---|---|
| LTL | PSPACE-complete | PTIME-easy |
| LTL+Past | PSPACE-complete | PTIME-easy |
| LTL+Past+Now | EXPSPACE-complete | PTIME-complete |
| LTL+Chop | nonelementary | PTIME-complete |
| FOMLO | nonelementary | PSPACE-complete |

# Outline of the talk

- LTL Path Model Checking
- PMC for richer logics:
  - LTL + Past, LTL + Past + Now
  - LTL + Chop
  - first order logic
- PMC for compressed paths

# Compressed Paths

Useful in analysis of log files, DNA strings, ...

Exponential compression:

$$[(a \cdot b^{22})^5 \cdot c \cdot a^{32}]^{96} \cdot a \cdot [(b^5 \cdot c)^{34} \cdot a]^{19}$$

Possible compression methods:

- Lempel-Ziv (LZ):
  - reuse subwords of what has already been compressed;
  - implemented in **gzip**,
- Straight-Line Programs (SLP): special context-free grammars;
- ...

# Straight Line Programs

example of a SLP:

$$P = \begin{cases} N_1 \rightarrow N_2 \cdot N_3 \\ N_2 \rightarrow N_3 \cdot N_4 \\ N_3 \rightarrow N_4 \cdot N_5 \\ N_4 \rightarrow N_5 \cdot N_6 \end{cases} \quad \begin{array}{l} N_5 \rightarrow a \cdot N_6 \\ N_6 \rightarrow N_7 \cdot b \\ N_7 \rightarrow c \cdot b \end{array}$$

This SLP encodes the following word:

$$w(P) = acbbcbbacbbacbbcbbacbbcbbacbb$$

# SLP and Finite State Automata

Theorem:
Given a SLP P and a finite state automaton $\mathcal{A}$, deciding whether w(P) is accepted by $\mathcal{A}$ is PTIME-complete.

- easiness [PR99]: fill a boolean table T[r,s,i] s.t. T[r,s,i] is true iff there is a path from state r to state s labeled with $w(N_i)$.

$$N_i \to N_j \cdot N_k \;\Rightarrow\; T[r,s,i] = \bigvee_u (T[r,u,j] \wedge T[u,s,k])$$

- hardness: encode NAND-CIRCUIT VALUE, which is PTIME-complete.

# Compressed Path Model Checking

| | PMC for LTL | Accept. by FSA |
|---|---|---|
| uncompressed | PTIME-easy | NL-complete |
| compressed | PSPACE-complete | PTIME-complete |

PSPACE-hardness is “due to” the formula:

compressed PMC can be achieved in time $O(|P| \times 2^{O(|\varphi|)})$.

# Conclusion

- "new" problem: Path Model Checking;
- efficient algorithms for rich logics:
  - PTIME-complete for LTL+Past+Now and LTL+Chop;
  - PSPACE-complete for First Order Logic,
- Efficient LTL PMC on compressed paths.