On the Complexity of Path Checking in Temporal Logics

Daniel Bundala
Joël Ouaknine

Department of Computer Science
University of Oxford

Given LTL formula $(\mathrm{X}p) \wedge ((\mathrm{G}q)\ \mathrm{U}\ r))$

Given LTL formula $(\mathrm{X}p) \wedge ((\mathrm{G}q)\,\mathrm{U}\,r))$

and finite trace

p :	0	1	0	1	1
q :	0	1	1	1	1
r :	0	0	0	1	0

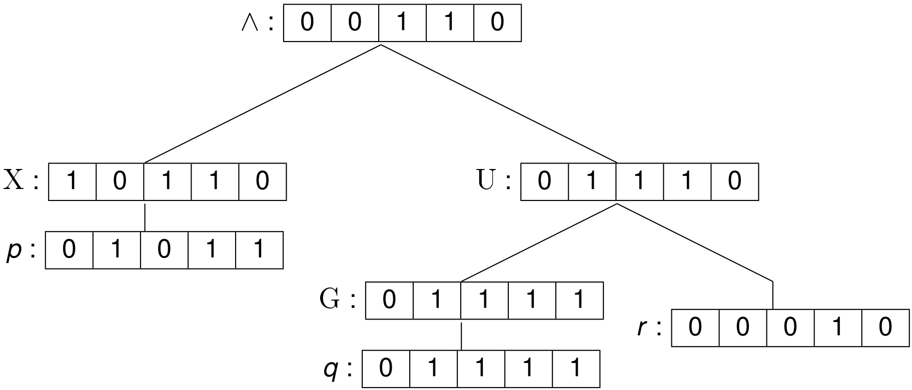
Given LTL formula $(\mathrm{X}p) \wedge ((\mathrm{G}q)\,\mathrm{U}\,r))$

and finite trace

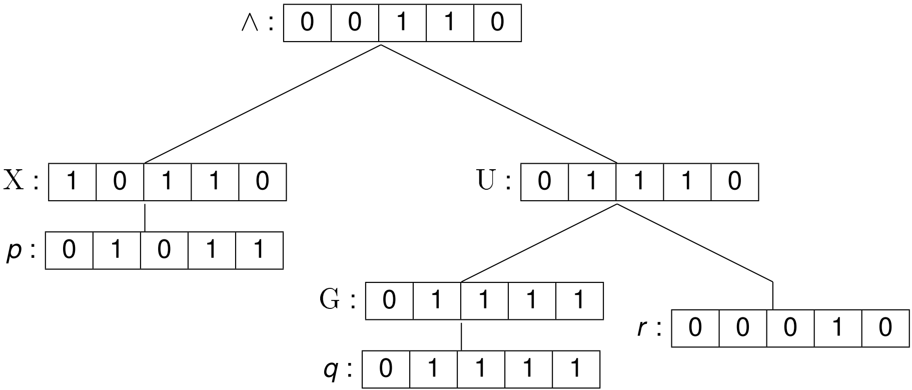
p :	0	1	0	1	1
q :	0	1	1	1	1
r :	0	0	0	1	0

Does it satisfy the formula? What is the complexity of checking?

$$(\mathrm{X}p) \wedge ((\mathrm{G}q)\,\mathrm{U}\,r))$$

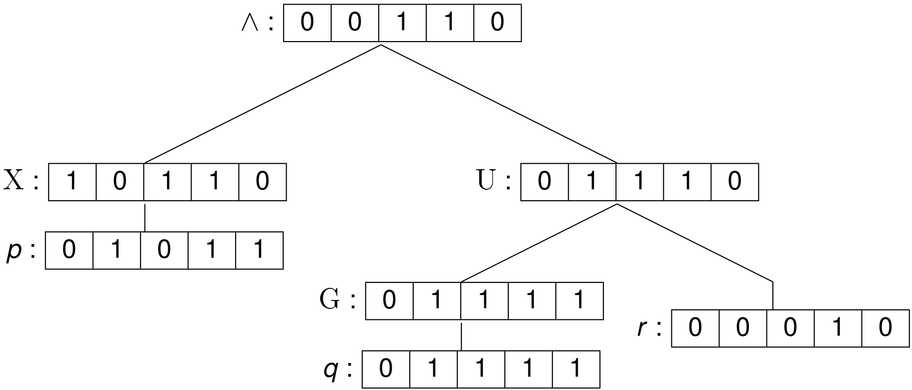


$$(\mathrm{X}p) \wedge ((\mathrm{G}q)\,\mathrm{U}\,r))$$



Upper bound: polynomial time: $O(|\mathrm{formula}||\mathrm{trace}|)$.

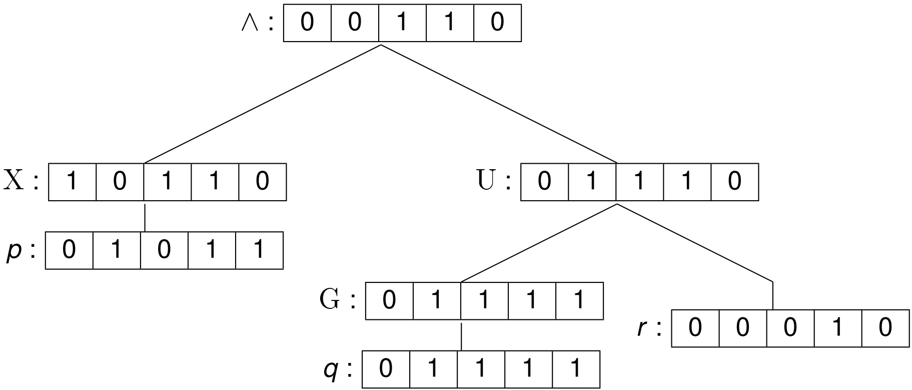
$$(\mathrm{X}p) \wedge ((\mathrm{G}q)\ \mathrm{U}\ r))$$



Upper bound: polynomial time: $O(|\mathrm{formula}||\mathrm{trace}|)$.

Best lower bound: NC^1-hard - propositional formula evalution.

$$(\mathrm{X}p) \wedge ((\mathrm{G}q)\ \mathrm{U}\ r))$$

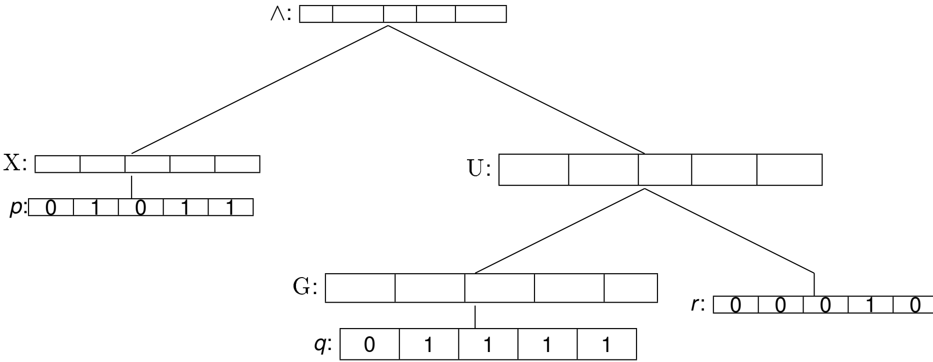


Upper bound: polynomial time: $O(|\mathrm{formula}||\mathrm{trace}|)$.

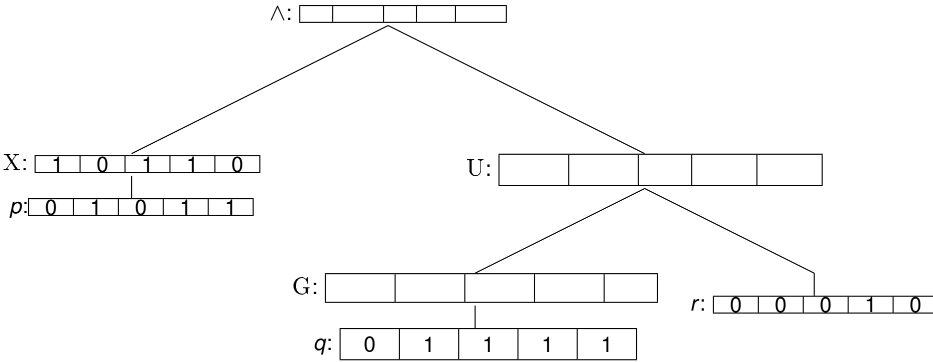
Best lower bound: NC^1-hard - propositional formula evalution.

Can we do better than P?

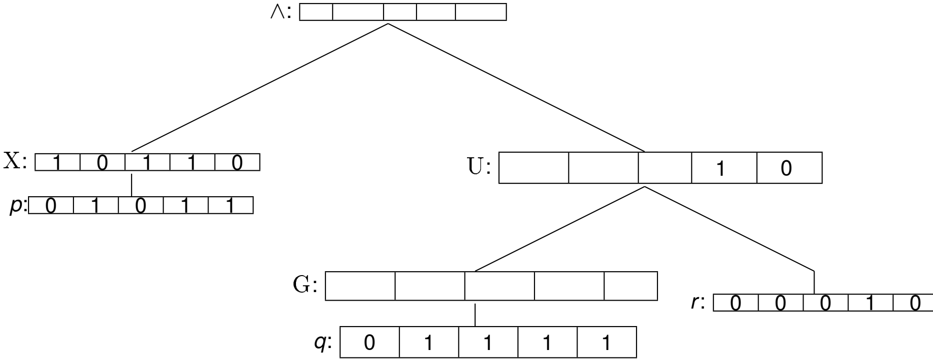
Tree Contraction Algorithm, Kuhtz and Finkbeiner, ICALP'09



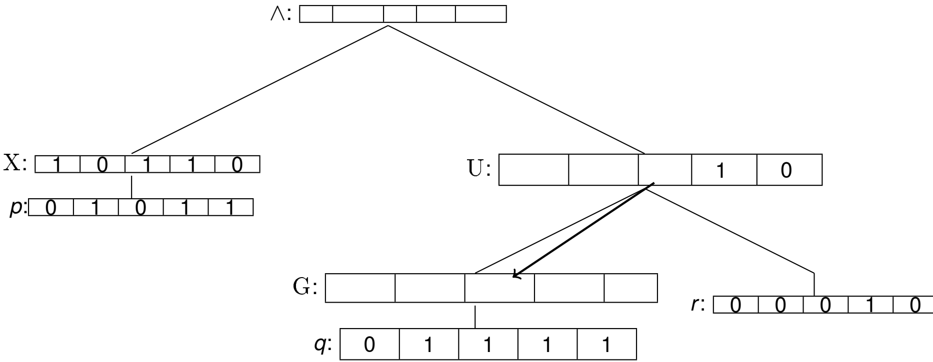
Tree Contraction Algorithm, Kuhtz and Finkbeiner, ICALP'09



Tree Contraction Algorithm, Kuhtz and Finkbeiner, ICALP'09

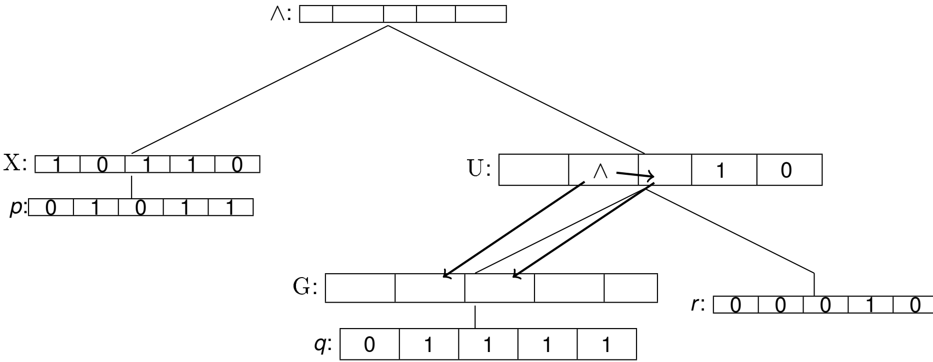


Tree Contraction Algorithm, Kuhtz and Finkbeiner, ICALP'09



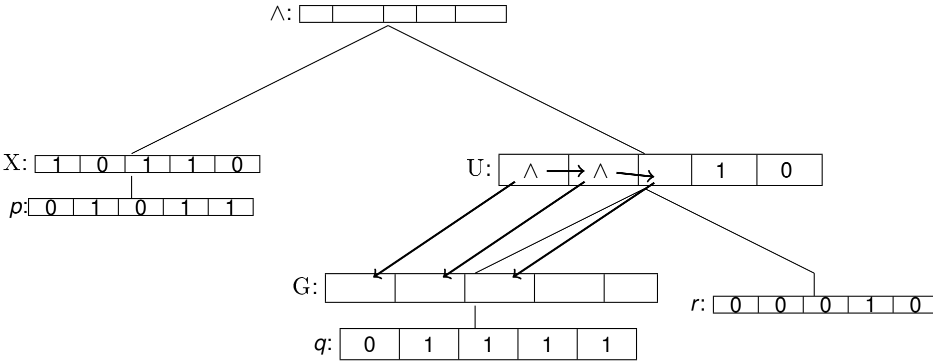
$$\varphi \, \mathrm{U} \, \psi = \psi \vee (\varphi \wedge \mathrm{X}(\varphi \, \mathrm{U} \, \psi))$$

Tree Contraction Algorithm, Kuhtz and Finkbeiner, ICALP'09



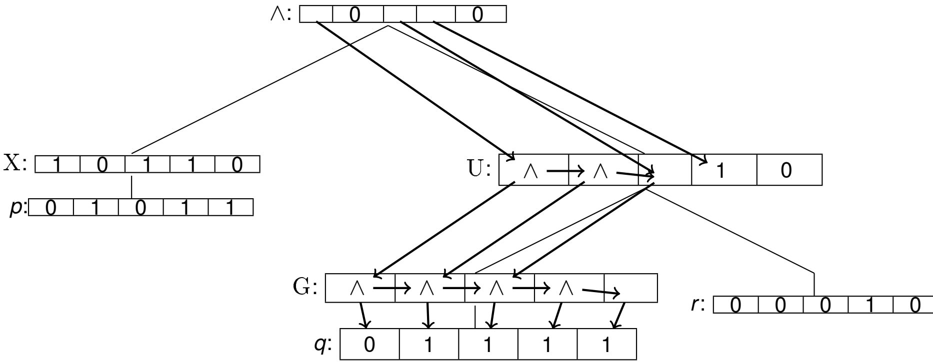
$$\varphi \,\mathrm{U}\, \psi = \psi \vee (\varphi \wedge \mathrm{X}(\varphi \,\mathrm{U}\, \psi))$$

Tree Contraction Algorithm, Kuhtz and Finkbeiner, ICALP'09



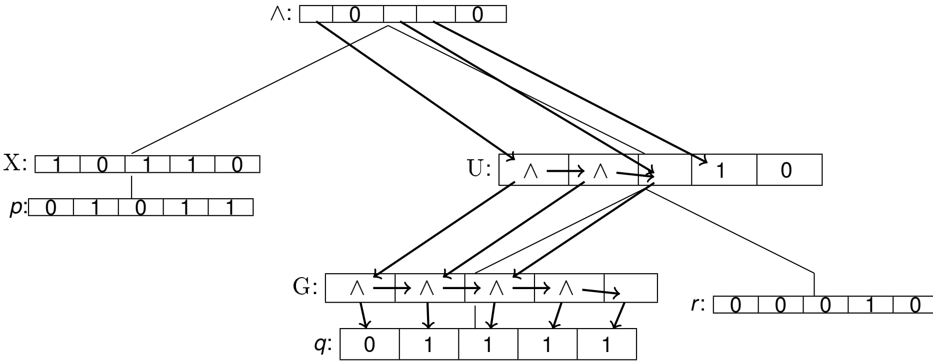
$$\varphi \,\mathrm{U}\, \psi = \psi \vee (\varphi \wedge \mathrm{X}(\varphi \,\mathrm{U}\, \psi))$$

Tree Contraction Algorithm, Kuhtz and Finkbeiner, ICALP'09



$$\varphi \,\mathrm{U}\, \psi = \psi \vee (\varphi \wedge \mathrm{X}(\varphi \,\mathrm{U}\, \psi))$$

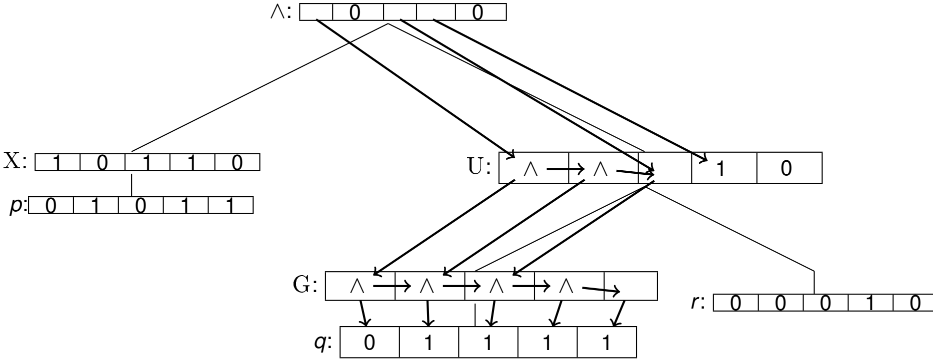
Tree Contraction Algorithm, Kuhtz and Finkbeiner, ICALP'09



$\varphi \, \mathrm{U} \, \psi = \psi \vee (\varphi \wedge \mathrm{X}(\varphi \, \mathrm{U} \, \psi))$

Build circuits for independent leaves in parallel.

Tree Contraction Algorithm, Kuhtz and Finkbeiner, ICALP'09

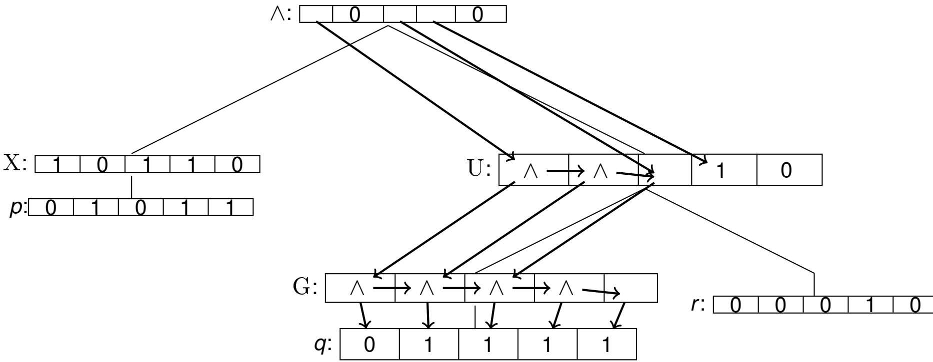


$\varphi \,\mathrm{U}\, \psi = \psi \vee (\varphi \wedge \mathrm{X}(\varphi \,\mathrm{U}\, \psi))$

Build circuits for independent leaves in parallel.

Circuits:

Tree Contraction Algorithm, Kuhtz and Finkbeiner, ICALP'09

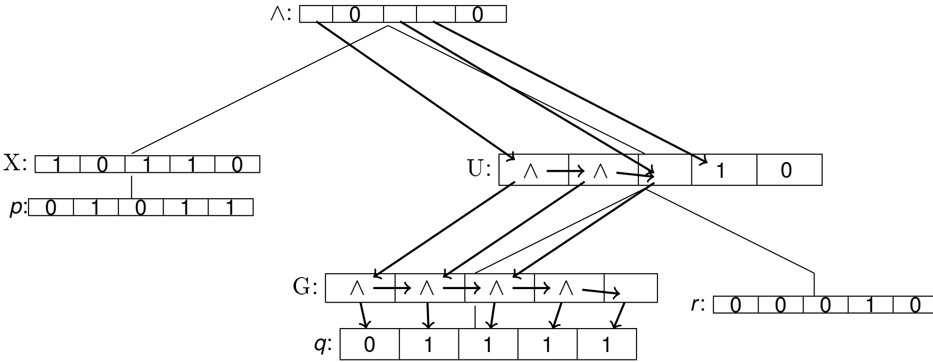


$\varphi \, \mathrm{U} \, \psi = \psi \vee (\varphi \wedge \mathrm{X}(\varphi \, \mathrm{U} \, \psi))$

Build circuits for independent leaves in parallel.

Circuits: planar, layered, stratified, monotone

Tree Contraction Algorithm, Kuhtz and Finkbeiner, ICALP'09

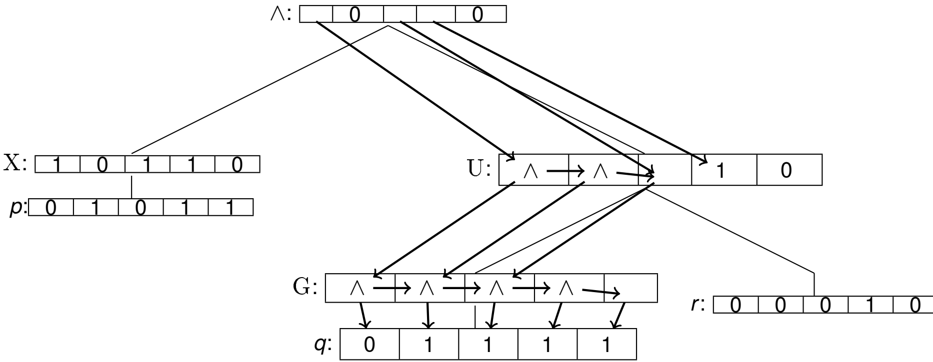


$\varphi \,\mathrm{U}\, \psi = \psi \vee (\varphi \wedge \mathrm{X}(\varphi \,\mathrm{U}\, \psi))$

Build circuits for independent leaves in parallel.

Circuits: planar, layered, stratified, monotone (logDCFL $\subseteq$ AC^1)

Tree Contraction Algorithm, Kuhtz and Finkbeiner, ICALP'09



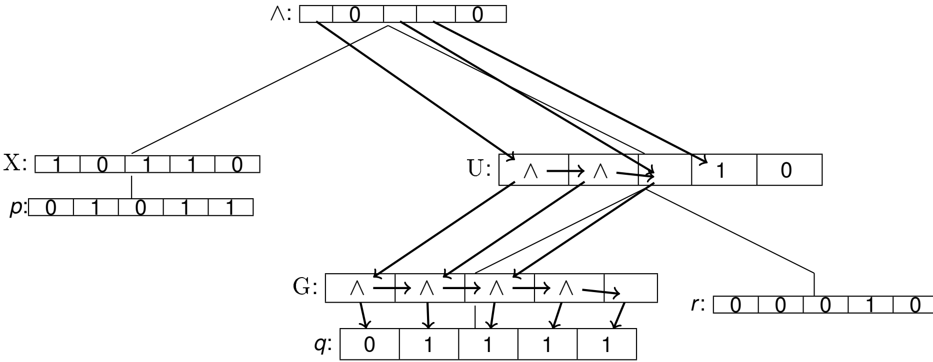
$\varphi \,\mathrm{U}\, \psi = \psi \vee (\varphi \wedge \mathrm{X}(\varphi \,\mathrm{U}\, \psi))$

Build circuits for independent leaves in parallel.

Circuits: planar, layered, stratified, monotone (logDCFL $\subseteq$ AC^1)

log $|\mathrm{formula}|$ parallel stages are sufficient.

Tree Contraction Algorithm, Kuhtz and Finkbeiner, ICALP'09



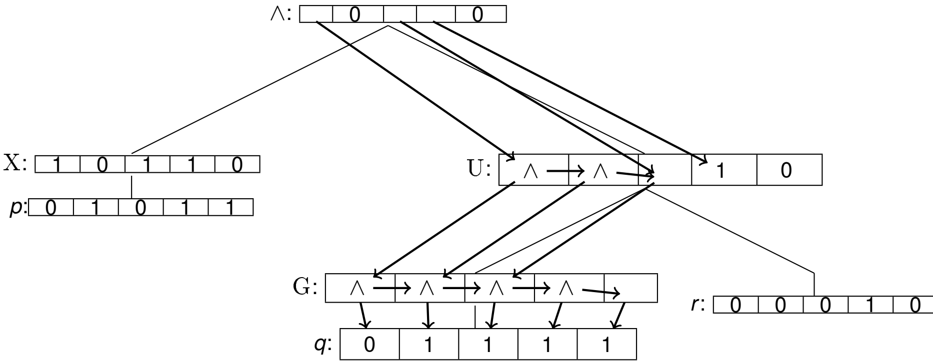
$\varphi \, \mathrm{U} \, \psi = \psi \vee (\varphi \wedge \mathrm{X}(\varphi \, \mathrm{U} \, \psi))$

Build circuits for independent leaves in parallel.

Circuits: planar, layered, stratified, monotone (logDCFL $\subseteq$ AC^1)

log $|\mathrm{formula}|$ parallel stages are sufficient. Gives AC^1[logDCFL] algorithm

Tree Contraction Algorithm, Kuhtz and Finkbeiner, ICALP'09



$\varphi \,\mathrm{U}\, \psi = \psi \vee (\varphi \wedge \mathrm{X}(\varphi \,\mathrm{U}\, \psi))$

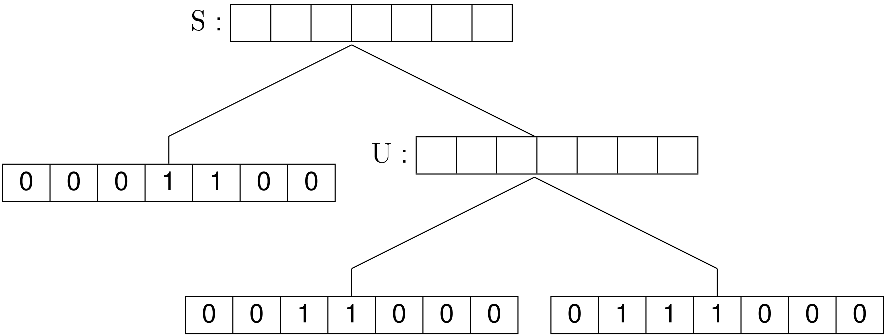
Build circuits for independent leaves in parallel.

Circuits: planar, layered, stratified, monotone (logDCFL $\subseteq$ AC^1)

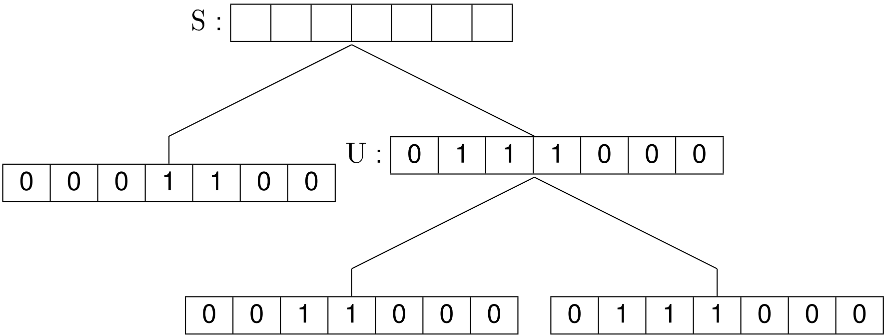
log $|\mathrm{formula}|$ parallel stages are sufficient. Gives AC^1[logDCFL] algorithm

AC^1[logDCFL] $\subseteq$ AC^2 $\Longrightarrow$ efficient parallel algorithm for LTL path checking. LTL path checking unlikely P-hard

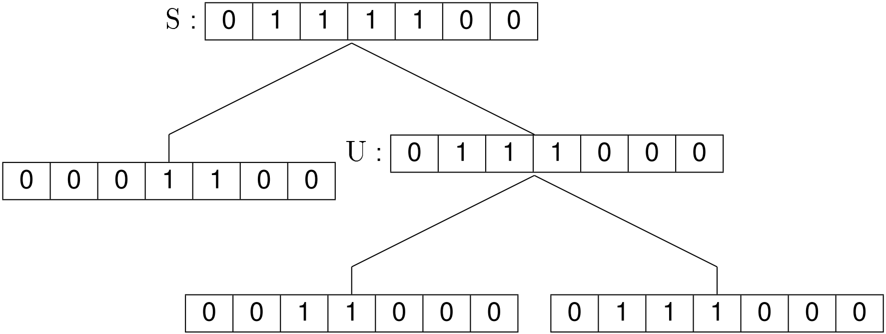
Circuit Evaluation



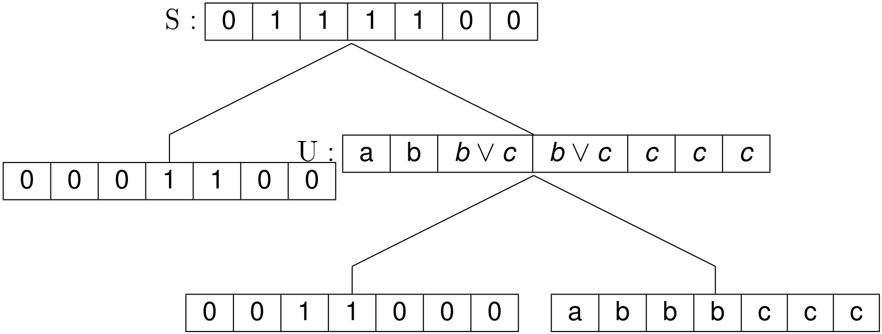
Circuit Evaluation



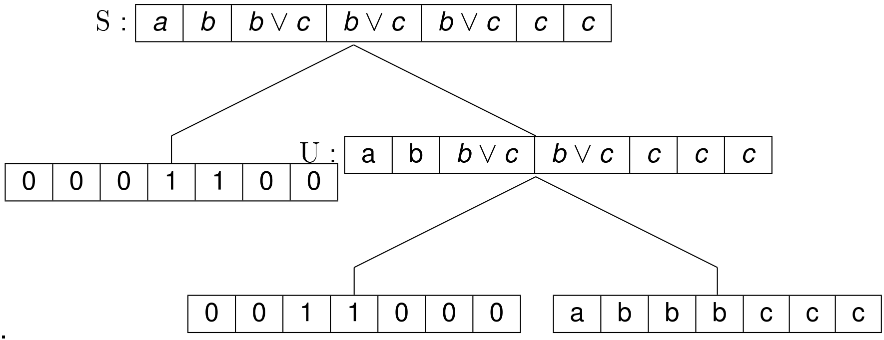
Circuit Evaluation



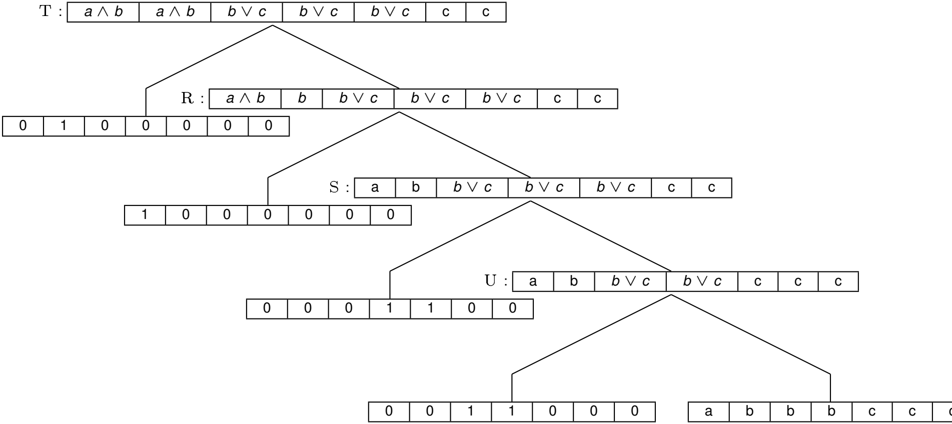
Circuit Evaluation



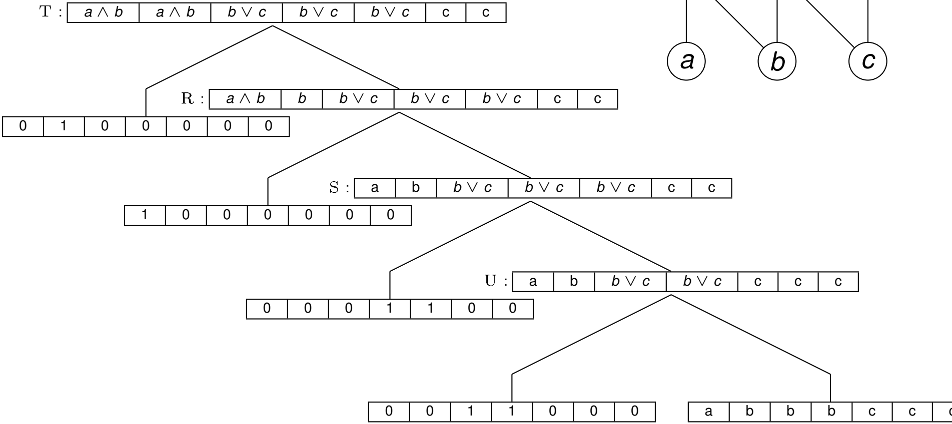
Circuit Evaluation

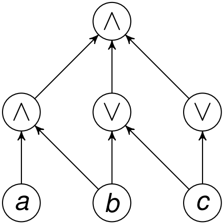


Circuit Evaluation Continued



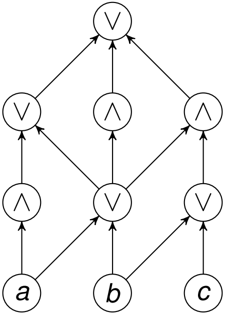
Circuit Evaluation Continued





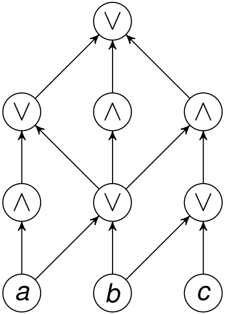
Evaluated a single layer of the circuit

Circuit Evaluation Continued



Inductively, circuit evaluation reduces to LTL path checking

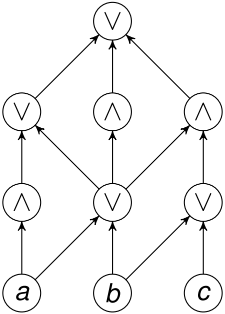
Circuit Evaluation Continued



Inductively, circuit evaluation reduces to LTL path checking

Matching upper bounds ($AC^1[\log DCFL]$)

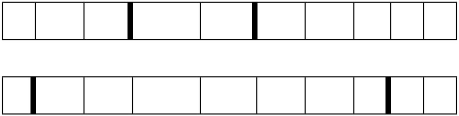
Circuit Evaluation Continued



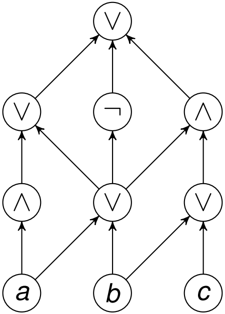
Inductively, circuit evaluation reduces to LTL path checking

Matching upper bounds (AC^1[logDCFL])

Nonmonotone circuits: PTIME-Complete



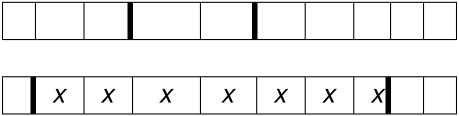
Circuit Evaluation Continued



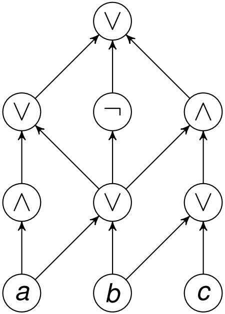
Inductively, circuit evaluation reduces to LTL path checking

Matching upper bounds (AC^1[logDCFL])

Nonmonotone circuits: PTIME-Complete



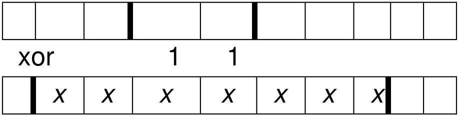
Circuit Evaluation Continued



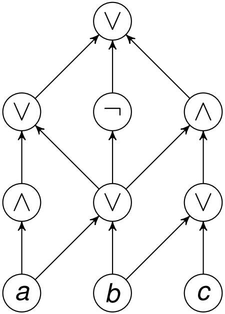
Inductively, circuit evaluation reduces to LTL path checking

Matching upper bounds (AC^1[logDCFL])

Nonmonotone circuits: PTIME-Complete



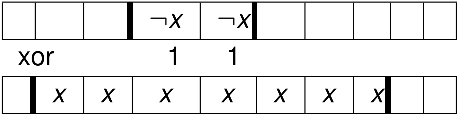
Circuit Evaluation Continued



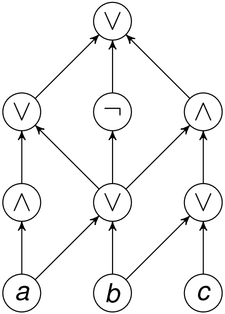
Inductively, circuit evaluation reduces to LTL path checking

Matching upper bounds (AC^1[logDCFL])

Nonmonotone circuits: PTIME-Complete



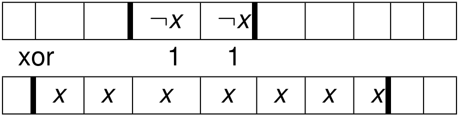
Circuit Evaluation Continued



Inductively, circuit evaluation reduces to LTL path checking

Matching upper bounds (AC^1[logDCFL])

Nonmonotone circuits: PTIME-Complete



$\Longrightarrow$ LTL+XOR path checking is PTIME-complete

$$\mathrm{UTL} \subseteq \mathrm{LTL} \subseteq \mathrm{MTL}$$

$\mathrm{UTL} \subseteq \mathrm{LTL} \subseteq \mathrm{MTL}$

MTL

Formula $p \,\mathrm{U}_{[1,3]}\, q$:

U :	?	?	?	?	?	?	?	?

p :	*	*	*	*	*	*	*	*
q :	0	0	0	0	1	0	1	1
t :	1	2	3	3.5	3.8	4	4.5	5

MTL

Formula $p \, \mathrm{U}_{[1,3]} \, q$:

U :	?	?	?	?	?	?	?	?

p :	*	*	*	*	*	*	*	*
q :	0	0	0	0	1	0	1	1
t :	1	2	3	3.5	3.8	4	4.5	5

MTL

Formula $p \, \mathrm{U}_{[1,3]} \, q$:

U :	?	?	?	?	?	?	?	?

p :	*	*	*	*	*	*	*	*
q :	0	0	0	0	1	0	1	1
t :	1	2	3	3.5	3.8	4	4.5	5

MTL

Formula $p \, \mathrm{U}_{[1,3]} \, q$:

U :	?	?	?	?	?	?	?	?

p :	*	*	*	*	*	*	*	*
q :	0	0	0	0	1	0	1	1
t :	1	2	3	3.5	3.8	4	4.5	5

MTL

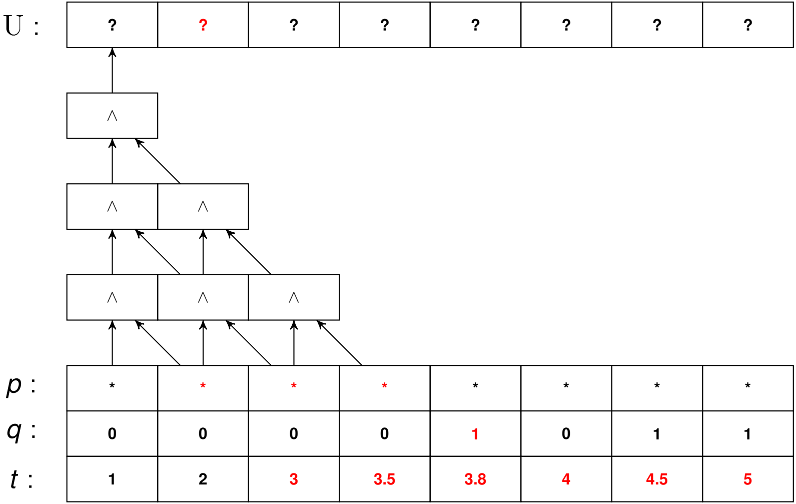
Formula $p \, \mathrm{U}_{[1,3]} \, q$:

U :	?	?	?	?	?	?	?	?

p :	*	*	*	*	*	*	*	*
q :	0	0	0	0	1	0	1	1
t :	1	2	3	3.5	3.8	4	4.5	5

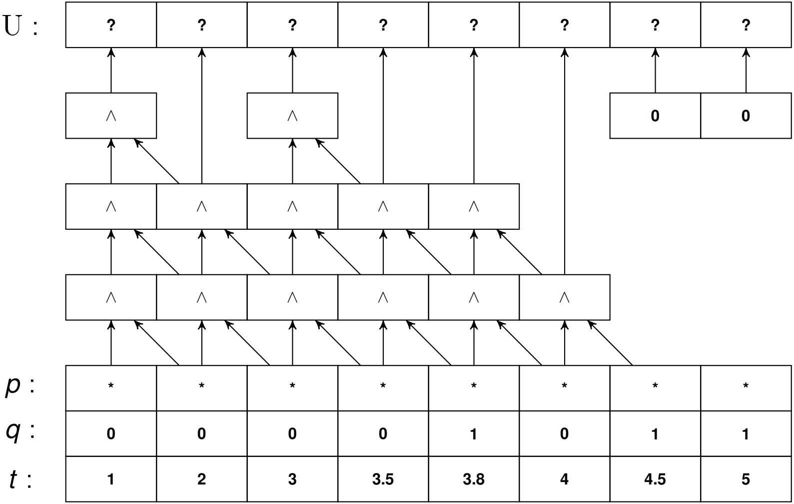
MTL

Formula $p \, \mathrm{U}_{[1,3]} \, q$:



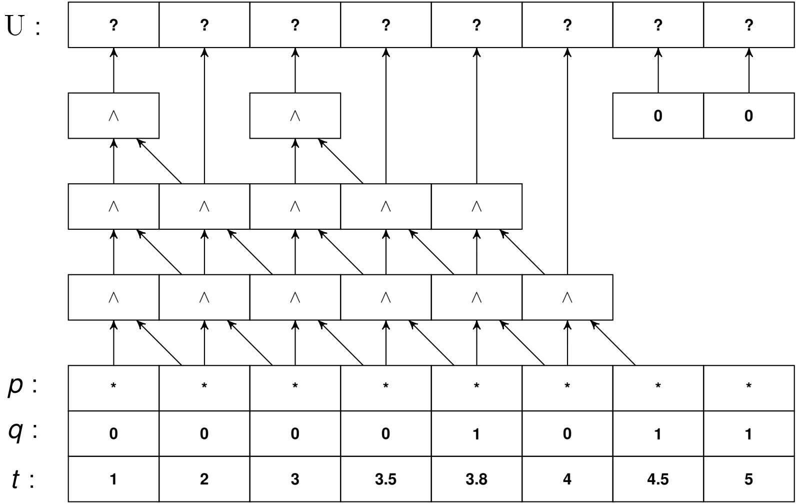
MTL

Formula $p \,\mathrm{U}_{[1,3]}\, q$:



MTL

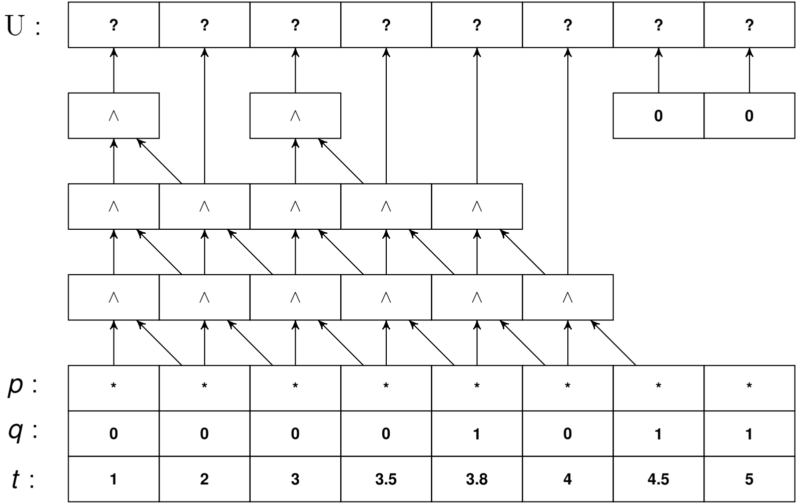
Formula $p\,\mathrm{U}_{[1,3]}\,q$:



Similar circuits for other operators

MTL

Formula $p \, \mathrm{U}_{[1,3]} \, q$:



Similar circuits for other operators

$\Longrightarrow$ MTL path checking in AC^1[logDCFL].

$\mathsf{UTL} \subseteq \mathsf{LTL} \subseteq \mathsf{MTL}$

$\mathsf{UTL} \subseteq \mathsf{LTL} \subseteq \mathsf{MTL}$

UTL - $\mathrm{G}, \mathrm{F}, \wedge, \vee, \mathrm{X}$

G:	0	0	0	0	0
p:	*	*	*	*	0

UTL - $\mathrm{G}, \mathrm{F}, \wedge, \vee, \mathrm{X}$

G:	0	0	0	0	1
p:	*	*	*	0	1

UTL - $\mathrm{G}, \mathrm{F}, \wedge, \vee, \mathrm{X}$

G:	0	0	0	1	1
p:	*	*	0	1	1

UTL - $\mathrm{G}, \mathrm{F}, \wedge, \vee, \mathrm{X}$

G:	0	0	1	1	1
p:	*	0	1	1	1

UTL - $\mathrm{G}, \mathrm{F}, \wedge, \vee, \mathrm{X}$

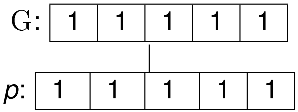
G:	0	1	1	1	1
p:	0	1	1	1	1

UTL - $\mathrm{G}, \mathrm{F}, \wedge, \vee, \mathrm{X}$

G:	1	1	1	1	1
p:	1	1	1	1	1

Polynomially many possibilities for $\mathrm{G}p \implies$ can be stored explicitly in polynomial space

UTL - $\mathrm{G}, \mathrm{F}, \wedge, \vee, \mathrm{X}$



Polynomially many possibilities for $\mathrm{G}p \implies$ can be stored explicitly in polynomial space

Leads to AC^1 algorithm for UTL

UTL - $\mathrm{G}, \mathrm{F}, \wedge, \vee, \mathrm{X}$

G:	1	1	1	1	1

p:	1	1	1	1	1

Polynomially many possibilities for $\mathrm{G}p \implies$ can be stored explicitly in polynomial space

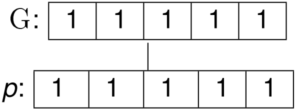
Leads to AC^1 algorithm for UTL

$$NC^1 \subseteq AC^1 \subseteq AC^1[\text{logDCFL}] \subseteq AC^2 \subseteq \cdots \subseteq \mathit{PTIME}$$

UTL LTL MTL LTL+Xor

Can we close the gaps?

UTL - $\mathrm{G}, \mathrm{F}, \wedge, \vee, \mathrm{X}$



Polynomially many possibilities for $\mathrm{G}p \implies$ can be stored explicitly in polynomial space

Leads to AC^1 algorithm for UTL

$$\mathsf{NC}^1 \subseteq \mathsf{AC}^1 \subseteq \mathsf{AC}^1[\mathsf{logDCFL}] \subseteq \mathsf{AC}^2 \subseteq \cdots \subseteq \mathit{PTIME}$$

UTL LTL MTL LTL+Xor

Can we close the gaps?

Questions?