Taming Strategy Logic: Non-Recurrent Fragments

Fabio Mogavero
Università degli Studi di Napoli Federico II

(joint work with Massimo Benerecetti & Adriano Peron)

Time'22
November 9, 2022

From Temporal To Strategic Reasoning

Classic Temporal Logics (LTL, CTL, CTL*, PDL, ...)

MC

SYNTESIS

SAT

Monolitic Nondeterministic (Closed) Systems

From Temporal To Strategic Reasoning

MC

SYNTESIS

Classic Temporal Logics (LTL, CTL, CTL*, PDL, ...) — Monolitic Nondeterministic (Closed) Systems

SAT

MC

SYNTESIS

Alternating Temporal Logics (ATL, ATL*, GL, ...) — Modular Interactive (Open) Systems

SAT

Non-Cooperative Bi-Coalition Properties

From Temporal To Strategic Reasoning

Classic Temporal Logics (LTL, CTL, CTL*, PDL, ...) — MC / SYNTESIS / SAT — Monolitic Nondeterministic (Closed) Systems

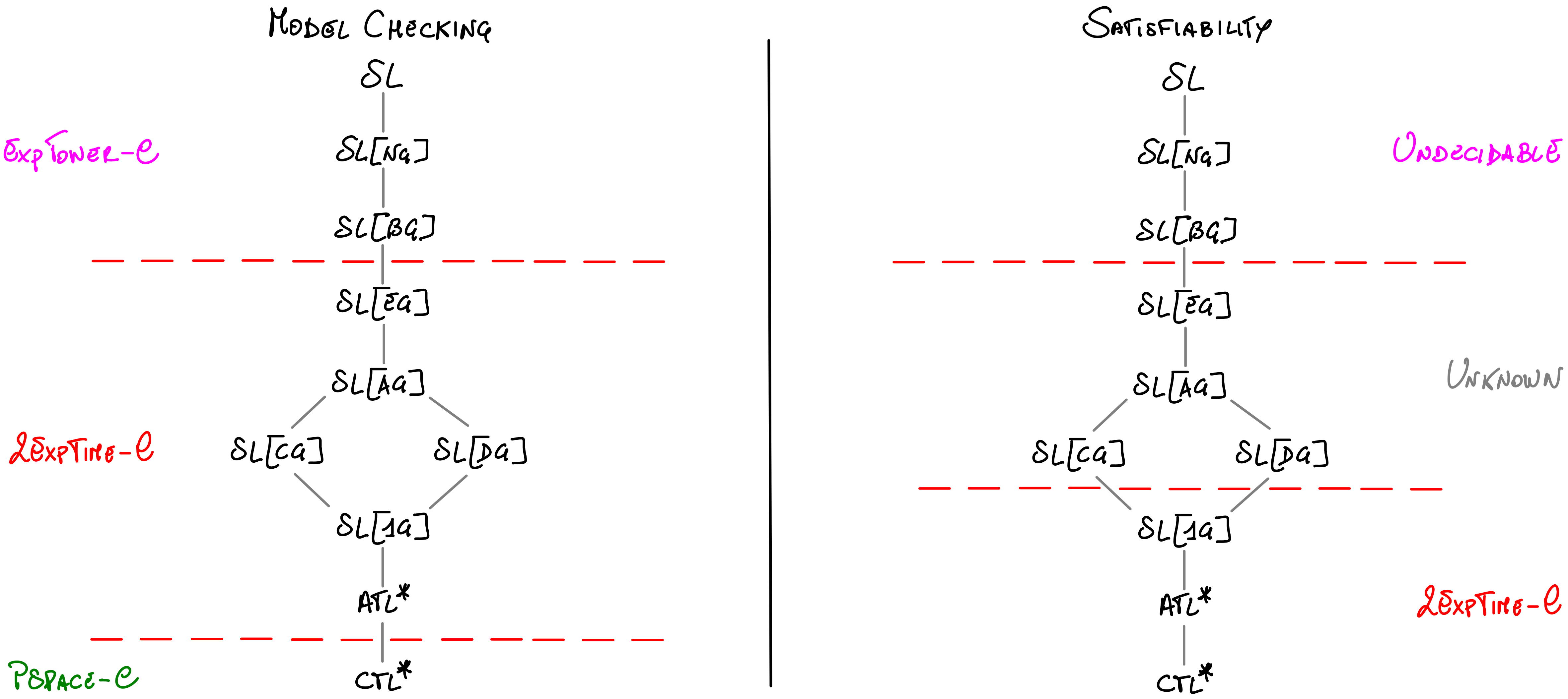
Alternating Temporal Logics (ATL, ATL*, GL, ...) — MC / SYNTESIS / SAT — Modular Interactive (Open) Systems

Non-Cooperative Bi-Coalition Properties

Cooperative/Non-Cooperative Multi-Coalition Properties

Strategy Logics (SL, ...) — MC / SYNTESIS / SAT — Multi-Agent Systems

What is Known About Strategy Logic



Our Contribution

We study new fragments of **Strategy Logic** with Sat easier than MC

+ Highlight a **deep connection** between fragments of **FOL** and SL

+ Analyse **new classes of tree models** called Bounded-Fork Trees

+ Introduce two **new classes of automata** on Words & Trees (Bounded-Fork TA & Prefix-Deterministic WA)

Non Recurrent One-Goal Strategy Logic

One Goal Strategy Logic

* $SL[1G]$: $\begin{cases} \varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \gamma b.\psi \\ \psi ::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid X\psi \mid \psi U \psi \mid \psi R \psi \end{cases}$

One Goal Strategy Logic

Atomic Propositions

* SL[1G]:

$$\begin{cases} \varphi ::= p \mid \neg\varphi \mid \varphi\wedge\varphi \mid \varphi\vee\varphi \mid \wp\flat.\psi \\ \psi ::= \varphi \mid \underbrace{\neg\psi \mid \psi\wedge\psi \mid \psi\vee\psi}_{\text{Boolean Connectives}} \mid \underbrace{X\psi \mid \psi U\psi \mid \psi R\psi}_{\text{Temporal Operators}} \end{cases}$$

Sentences

One Goal Strategy Logic

Quantifier Prefix (e.g. $\forall x \exists y$)

Binding Prefix (e.g. $(\alpha, x)(\beta, y)$)

Strategies are first-class citizen

Atomic Propositions

$$* \; SL[1G]: \begin{cases} \varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp\, b.\psi \\ \psi ::= \varphi \mid \underbrace{\neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi}_{\text{Boolean Connectives}} \mid \underbrace{X\psi \mid \psi U \psi \mid \psi R \psi}_{\text{Temporal Operators}} \end{cases}$$

Sentences

One Goal Strategy Logic

Atomic Propositions

Quantifier Prefix (e.g. $\forall x \exists y$)

Binding Prefix (e.g. $(\alpha,x)(\beta,y)$)

Strategies are first-class citizen

* SL[1G]:

$$\varphi ::= p \mid \neg\varphi \mid \varphi\wedge\varphi \mid \varphi\vee\varphi \mid \wp\flat.\psi$$

$$\psi ::= \varphi \mid \underbrace{\neg\psi \mid \psi\wedge\psi \mid \psi\vee\psi}_{\text{Boolean Connectives}} \mid \underbrace{X\psi \mid \psi U\psi \mid \psi R\psi}_{\text{Temporal Operators}}$$

Sentences

* SL[1G] generalises ATL* to strategy quantifiers with arbitrary alternation

One Goal Strategy Logic

Atomic propositions

Quantifier prefix (e.g. $\forall x \exists y$)

Binding prefix (e.g. $(\alpha, x)(\beta, y)$)

Strategies are first-class citizen

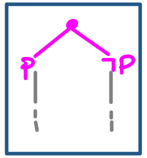
* SL[1G]: $\begin{cases} \varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp b.\psi \\ \psi ::= \varphi \mid \underbrace{\neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi}_{\text{Boolean connectives}} \mid \underbrace{X\psi \mid \psi U \psi \mid \psi R \psi}_{\text{Temporal operators}} \end{cases}$

Sentences

* SL[1G] generalises ATL* to strategy quantifiers with arbitrary alternation

* Examples of simple properties:

  - $\varphi \triangleq \exists x \exists y (\alpha, x)(\alpha, \beta). Xp \wedge \exists x \exists y (\alpha, x)(\alpha, \beta). X\neg p \equiv EXp \wedge EX\neg p$



One Goal Strategy Logic

Atomic Propositions

Quantifier Prefix (e.g. $\forall x \exists y$)

Binding Prefix (e.g. $(\alpha, x)(\beta, y)$)

Strategies are first-class citizen

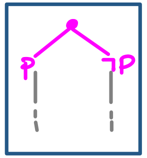
* SL[1G]:
$$\begin{cases} \varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp \flat . \psi \\ \psi ::= \varphi \mid \underbrace{\neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi}_{\text{Boolean Connectives}} \mid \underbrace{X\psi \mid \psi U \psi \mid \psi R \psi}_{\text{Temporal Operators}} \end{cases}$$

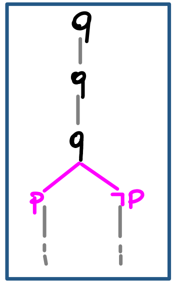
Sentences

* SL[1G] generalises ATL* to strategy quantifiers with arbitrary alternation

* Examples of simple properties:

    - $\varphi \triangleq \exists x \exists y (\alpha, x)(\alpha, \beta). Xp \wedge \exists x \exists y (\alpha, x)(\alpha, \beta). X\neg p \equiv EXp \wedge EX\neg p$
    - $\exists x \exists y (\alpha, x)(\alpha, \beta). q U \varphi \equiv E q U \varphi$





One Goal Strategy Logic

* SL[1G]:

Atomic Propositions

Quantifier Prefix (e.g. $\forall x \exists y$)

Binding Prefix (e.g. $(\alpha,x)(\beta,y)$)

Strategies are first-class citizen

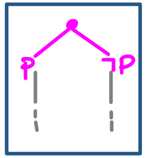
$$\varphi ::= p \mid \neg\varphi \mid \varphi\wedge\varphi \mid \varphi\vee\varphi \mid \wp b.\psi$$

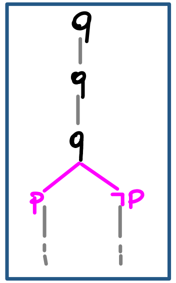
$$\psi ::= \varphi \mid \underbrace{\neg\psi \mid \psi\wedge\psi \mid \psi\vee\psi}_{\text{Boolean Connectives}} \mid \underbrace{X\psi \mid \psi U\psi \mid \psi R\psi}_{\text{Temporal Operators}}$$

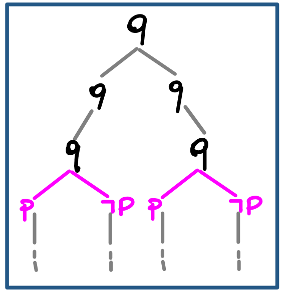
Sentences

* SL[1G] generalises ATL* to strategy quantifiers with arbitrary alternation

* Examples of simple properties:







- $\varphi \triangleq \exists x\exists y(\alpha,x)(\alpha,\beta).Xp \wedge \exists x\exists y(\alpha,x)(\alpha,\beta).X\neg p \equiv EXp \wedge EX\neg p$
- $\exists x\exists y(\alpha,x)(\alpha,\beta).q\,U\,\varphi \equiv E\,q\,U\,\varphi$
- $\forall x\forall y(\alpha,x)(\alpha,\beta).\varphi\,R\,q \equiv A\,\varphi\,R\,q$

One Goal Strategy Logic

Atomic Propositions

Quantifier Prefix (e.g. $\forall x \exists y$)

Binding Prefix (e.g. $(\alpha,x)(\beta,y)$)

Strategies are first-class citizen

* SL[1G]:

$$\begin{cases} \varphi ::= p \mid \neg\varphi \mid \varphi\wedge\varphi \mid \varphi\vee\varphi \mid \wp\flat.\psi \\ \psi ::= \varphi \mid \underbrace{\neg\psi \mid \psi\wedge\psi \mid \psi\vee\psi}_{\text{Boolean Connectives}} \mid \underbrace{X\psi \mid \psi U\psi \mid \psi R\psi}_{\text{Temporal Operators}} \end{cases}$$

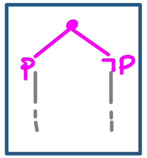
Sentences

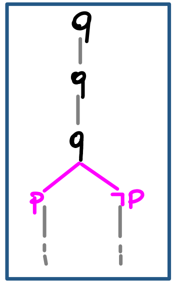
* SL[1G] generalises ATL* to strategy quantifiers with arbitrary alternation

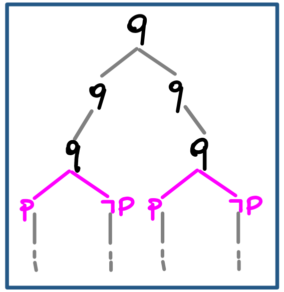
* Examples of simple properties:

Bounded fork models

- $\varphi \triangleq \exists x\exists y(\alpha,x)(\alpha,\beta).Xp \wedge \exists x\exists y(\alpha,x)(\alpha,\beta).X\neg p \equiv EXp \wedge EX\neg p$
- $\exists x\exists y(\alpha,x)(\alpha,\beta).q U \varphi \equiv E\, qU\varphi$
- $\forall x\forall y(\alpha,x)(\alpha,\beta).\varphi R q \equiv A\varphi R q$







All paths have a finite number of bifurcations

One Goal Strategy Logic

Quantifier Prefix (e.g. $\forall x \exists y$)

Binding Prefix (e.g. $(\alpha, x)(\beta, y)$)

Strategies are first-class citizen

Atomic Propositions

* SL[1G]:

$$\begin{cases} \varphi ::= p \mid \neg\varphi \mid \varphi\wedge\varphi \mid \varphi\vee\varphi \mid \wp\flat.\psi \\ \psi ::= \varphi \mid \underbrace{\neg\psi \mid \psi\wedge\psi \mid \psi\vee\psi}_{\text{Boolean Connectives}} \mid \underbrace{X\psi \mid \psi U\psi \mid \psi R\psi}_{\text{Temporal Operators}} \end{cases}$$

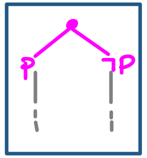
Sentences

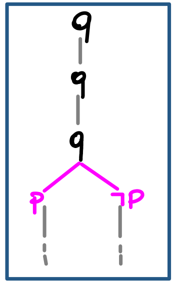
* SL[1G] generalises ATL* to strategy quantifiers with arbitrary alternation

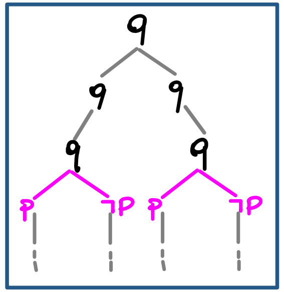
* Examples of simple properties:

Bounded Fork Models

- $\varphi \triangleq \exists x \exists y (\alpha, x)(\beta, y).Xp \wedge \exists x \exists y (\alpha, x)(\beta, y).X\neg p \equiv EXp \wedge EX\neg p$
- $\exists x \exists y (\alpha, x)(\beta, y).qU\varphi \equiv EqU\varphi$
- $\forall x \forall y (\alpha, x)(\beta, y).\varphi R q \equiv A\varphi R q$

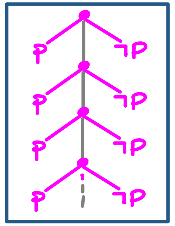






All paths have a finite number of bifurcations

- $\exists x \exists y (\alpha, x)(\beta, y).G\varphi \equiv EG\varphi$



One Goal Strategy Logic

$\ast$ SL[1G]:

$$\begin{cases} \varphi ::= p \mid \neg\varphi \mid \varphi\wedge\varphi \mid \varphi\vee\varphi \mid \wp\, \flat.\psi \\ \psi ::= \varphi \mid \neg\psi \mid \psi\wedge\psi \mid \psi\vee\psi \mid X\psi \mid \psi U\psi \mid \psi R\psi \end{cases}$$

Atomic Propositions

Quantifier Prefix (e.g. $\forall x \exists y$)

Binding Prefix (e.g. $(\alpha,x)(\beta,y)$)

Strategies are first-class citizen

Sentences

Boolean Connectives

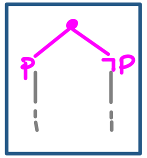
Temporal Operators

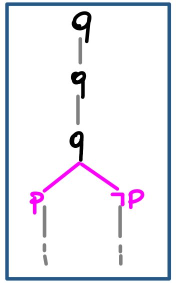
$\ast$ SL[1G] generalises ATL* to strategy quantifiers with arbitrary alternation

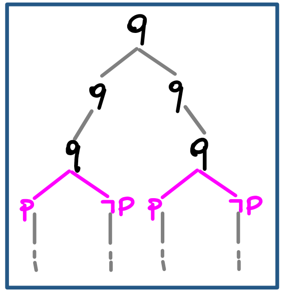
$\ast$ Examples of simple properties:

Bounded Fork Models

- $\varphi \triangleq \exists x\exists y(\alpha,x)(\alpha,\beta).Xp \wedge \exists x\exists y(\alpha,x)(\alpha,\beta).X\neg p \equiv EXp \wedge EX\neg p$
- $\exists x\exists y(\alpha,x)(\alpha,\beta).qU\varphi \equiv EqU\varphi$
- $\forall x\forall y(\alpha,x)(\alpha,\beta).\varphi Rq \equiv A\varphi Rq$

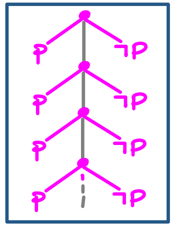


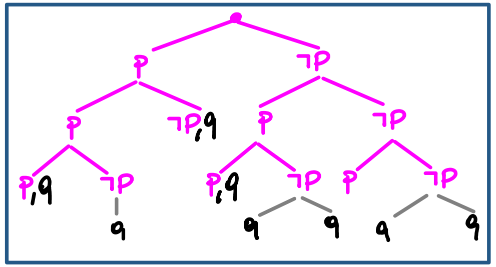




All paths have a finite number of bifurcations

- $\exists x\exists y(\alpha,x)(\alpha,\beta).G\varphi \equiv EG\varphi$
- $\forall x\forall y(\alpha,x)(\alpha,\beta).\varphi Uq \equiv A\varphi Uq$





One Goal Strategy Logic

* $SL[1G]$: $\begin{cases} \varphi ::= p \mid \neg\varphi \mid \varphi\wedge\varphi \mid \varphi\vee\varphi \mid \wp b.\psi \\ \psi ::= \varphi \mid \neg\psi \mid \psi\wedge\psi \mid \psi\vee\psi \mid X\psi \mid \psi U\psi \mid \psi R\psi \end{cases}$

Atomic Propositions

Quantifier Prefix (e.g. $\forall x \exists y$)

Binding Prefix (e.g. $(\alpha,x)(\beta,y)$)

Strategies are first-class citizen

Sentences

Boolean Connectives

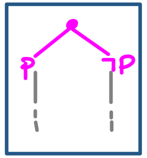
Temporal Operators

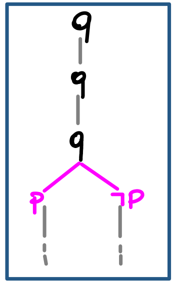
* $SL[1G]$ generalises ATL* to strategy quantifiers with arbitrary alternation

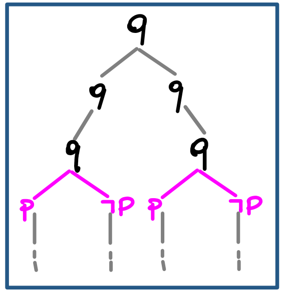
* Examples of simple properties:

Bounded Fork Models

- $\varphi \triangleq \exists x \exists y (\alpha,x)(\alpha,\beta).Xp \wedge \exists x \exists y (\alpha,x)(\alpha,\beta).X\neg p \equiv EXp \wedge EX\neg p$
- $\exists x \exists y (\alpha,x)(\alpha,\beta).q U \varphi \equiv E q U \varphi$
- $\forall x \forall y (\alpha,x)(\alpha,\beta).\varphi R q \equiv A \varphi R q$



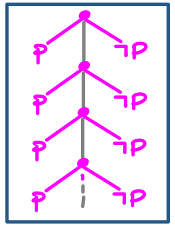




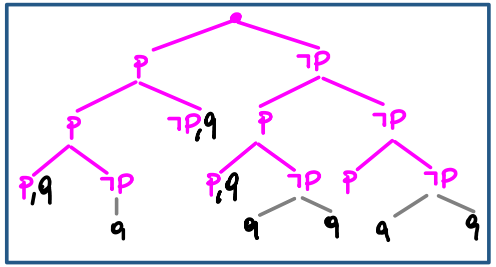
All paths have a finite number of bifurcations

Unbounded Fork Models

- $\exists x \exists y (\alpha,x)(\alpha,\beta).G\varphi \equiv EG\varphi$
- $\forall x \forall y (\alpha,x)(\alpha,\beta).\varphi U q \equiv A\varphi U q$



Paths with bifurcations of infinite/unbounded length



One Goal Strategy Logic

Atomic Propositions

Quantifier Prefix (e.g. $\forall x \exists y$)

Binding Prefix (e.g. $(\alpha,x)(\beta,y)$)

Strategies are first-class citizen

* SL[1G]:
$$\begin{cases} \varphi ::= p \mid \neg\varphi \mid \varphi\wedge\varphi \mid \varphi\vee\varphi \mid \wp\flat.\psi \\ \psi ::= \varphi \mid \underbrace{\neg\psi \mid \psi\wedge\psi \mid \psi\vee\psi}_{\text{Boolean Connectives}} \mid \underbrace{X\psi \mid \psi U\psi \mid \psi R\psi}_{\text{Temporal Operators}} \end{cases}$$

Sentences

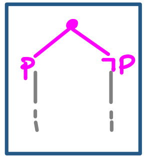
Satisfiability & Model Checking: Decidable in 2ExpTime-c!

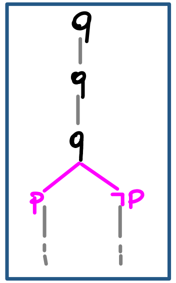
* SL[1G] generalises ATL* to strategy quantifiers with arbitrary alternation

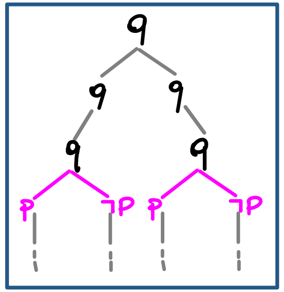
* Examples of simple properties:

Bounded Fork Models

- $\varphi \triangleq \exists x\exists y(\alpha,x)(\beta,y).Xp \wedge \exists x\exists y(\alpha,x)(\beta,y).X\neg p \equiv EXp \wedge EX\neg p$
- $\exists x\exists y(\alpha,x)(\beta,y).qU\varphi \equiv EqU\varphi$
- $\forall x\forall y(\alpha,x)(\beta,y).\varphi Rq \equiv A\varphi Rq$



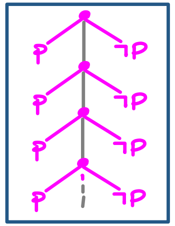




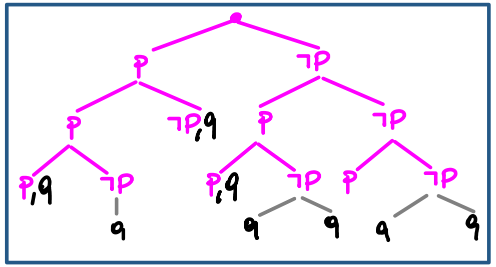
All paths have a finite number of bifurcations

Unbounded Fork Models

- $\exists x\exists y(\alpha,x)(\beta,y).G\varphi \equiv EG\varphi$
- $\forall x\forall y(\alpha,x)(\beta,y).\varphi Uq \equiv A\varphi Uq$



Paths with bifurcations of infinite/unbounded length



Non-Recurrent Fragments

$$\overbrace{\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp b.\eta \mid \wp b.\psi}^{SL^{\emptyset}[14]}$$

$$\eta ::= \varphi \mid \eta \wedge \eta \mid \eta \vee \eta \mid \psi \,U\, \varphi \mid \varphi \,R\, \psi \mid X \eta \mid X \psi$$

Non-Trivial Sentences are Forbidden

$$\psi ::= p \mid \neg \psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid \psi \,U\, \psi \mid \psi \,R\, \psi \mid X \psi$$

Classic LTL over Atomic Propositions

Non-Recurrent Fragments

$SL^{\emptyset}[1G]$

$$\varphi := p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp b.\eta \mid \wp b.\psi$$

$$\eta := \varphi \mid \eta \wedge \eta \mid \eta \vee \eta \mid \psi \mathbin{U} \varphi \mid \varphi \mathbin{R} \psi \mid X\eta \mid X\psi$$

Non-trivial sentences are forbidden

$$\psi := p \mid \neg \psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid \psi \mathbin{U} \psi \mid \psi \mathbin{R} \psi \mid X\psi$$

Classic LTL over Atomic Propositions

$WSL^{\emptyset}[1G]$

$$\varphi := p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp b.\eta$$

$$\eta := \varphi \mid \eta \wedge \eta \mid \eta \vee \eta \mid \beta \mathbin{U} \varphi \mid \varphi \mathbin{R} \beta \mid X\eta$$

No temporal operator is allowed

$$\beta := p \mid \neg \beta \mid \beta \wedge \beta \mid \beta \vee \beta$$

Boolean formulae over Atomic Propositions

Non-Recurrent Fragments

$SL^{\emptyset}[1G]$

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp\flat.\eta \mid \wp\flat.\psi$$

$$\eta ::= \varphi \mid \eta \wedge \eta \mid \eta \vee \eta \mid \psi \, U \varphi \mid \varphi R \psi \mid X\eta \mid X\psi$$

Non-trivial sentences are forbidden

$$\psi ::= p \mid \neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid \psi U \psi \mid \psi R \psi \mid X\psi$$

Classic LTL over Atomic Propositions

$WSL^{\emptyset}[1G]$

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp\flat.\eta$$

$$\eta ::= \varphi \mid \eta \wedge \eta \mid \eta \vee \eta \mid \beta U \varphi \mid \varphi R \beta \mid X\eta$$

No temporal operator is allowed

$$\beta ::= p \mid \neg\beta \mid \beta \wedge \beta \mid \beta \vee \beta$$

Boolean formulae over Atomic Propositions

	$SL[1G]$	$SL^{\emptyset}[1G]$	$WSL^{\emptyset}[1G]$
$EG\varphi$, $A\varphi U q$			
$A[(qRp)U\varphi]$, $E[\varphi R(pUq)]$			
$E[(q \vee t)U\varphi]$, $A[\varphi R(s \wedge t)]$			

$\boxed{\varphi \triangleq EXp \wedge EX\neg p}$

Non-Recurrent Fragments

$SL^{\emptyset}[1G]$

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp\flat.\eta \mid \wp\flat.\psi$$

$$\eta ::= \varphi \mid \eta \wedge \eta \mid \eta \vee \eta \mid \psi U \varphi \mid \varphi R \psi \mid X\eta \mid X\psi$$

Non-trivial sentences are forbidden

$$\psi ::= p \mid \neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid \psi U \psi \mid \psi R \psi \mid X\psi$$

Classic LTL over Atomic Propositions

$WSL^{\emptyset}[1G]$

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp\flat.\eta$$

$$\eta ::= \varphi \mid \eta \wedge \eta \mid \eta \vee \eta \mid \beta U \varphi \mid \varphi R \beta \mid X\eta$$

No temporal operator is allowed

$$\beta ::= p \mid \neg\beta \mid \beta \wedge \beta \mid \beta \vee \beta$$

Boolean formulae over Atomic Propositions

	$SL[1G]$	$SL^{\emptyset}[1G]$	$WSL^{\emptyset}[1G]$
$EG\varphi,\ A\varphi U q$	✓	✗	✗
$A[(qRp)U\varphi],\ E[\varphi R(pUq)]$			
$E[(q \vee t)U\varphi],\ A[\varphi R(s \wedge t)]$			

$$\boxed{\varphi \triangleq EXp \wedge EX\neg p}$$

Non-Recurrent Fragments

$SL^{\emptyset}[1G]$

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp b.\eta \mid \wp b.\psi$$

$$\eta ::= \varphi \mid \eta \wedge \eta \mid \eta \vee \eta \mid \psi U \varphi \mid \varphi R \psi \mid X\eta \mid X\psi$$

Non-trivial sentences are forbidden

$$\psi ::= p \mid \neg \psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid \psi U \psi \mid \psi R \psi \mid X\psi$$

Classic LTL over Atomic Propositions

$WSL^{\emptyset}[1G]$

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp b.\eta$$

$$\eta ::= \varphi \mid \eta \wedge \eta \mid \eta \vee \eta \mid \beta U \varphi \mid \varphi R \beta \mid X\eta$$

No temporal operator is allowed

$$\beta ::= p \mid \neg \beta \mid \beta \wedge \beta \mid \beta \vee \beta$$

Boolean formulae over Atomic Propositions

	$SL[1G]$	$SL^{\emptyset}[1G]$	$WSL^{\emptyset}[1G]$
$EG\varphi,\ A\varphi U q$	✓	✗	✗
$A[(qRp)U\varphi],\ E[\varphi R(pUq)]$	✓	✓	✗
$E[(q\vee t)U\varphi],\ A[\varphi R(s\wedge t)]$			

$$\boxed{\varphi \triangleq EXp \wedge EX\neg p}$$

Non-Recurrent Fragments

$$\overbrace{\varphi := p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp\flat.\eta \mid \wp\flat.\psi}^{SL^{\emptyset}[1G]}$$

$$\eta := \varphi \mid \eta \wedge \eta \mid \eta \vee \eta \mid \psi U \varphi \mid \varphi R \psi \mid X\eta \mid X\psi$$

Non-trivial sentences are forbidden

$$\psi := p \mid \neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid \psi U \psi \mid \psi R \psi \mid X\psi$$

Classic LTL over Atomic Propositions

$$\overbrace{\varphi := p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp\flat.\eta}^{WSL^{\emptyset}[1G]}$$

$$\eta := \varphi \mid \eta \wedge \eta \mid \eta \vee \eta \mid \beta U \varphi \mid \varphi R \beta \mid X\eta$$

No temporal operator is allowed

$$\beta := p \mid \neg\beta \mid \beta \wedge \beta \mid \beta \vee \beta$$

Boolean formulae over Atomic Propositions

	$SL[1G]$	$SL^{\emptyset}[1G]$	$WSL^{\emptyset}[1G]$
$EG\varphi,\ A\varphi Uq$	✓	✗	✗
$A[(qRp)U\varphi],\ E[\varphi R(pUq)]$	✓	✓	✗
$E[(q\vee t)U\varphi],\ A[\varphi R(s\wedge t)]$	✓	✓	✓

$$\boxed{\varphi \triangleq EXp \wedge EX\neg p}$$

Non-Recurrent Fragments

$SL^{\emptyset}[1G]$

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp\flat.\eta \mid \wp\flat.\psi$$

$$\eta ::= \varphi \mid \eta \wedge \eta \mid \eta \vee \eta \mid \psi U \varphi \mid \varphi R \psi \mid X\eta \mid X\psi$$

Non-trivial sentences are forbidden

$$\psi ::= p \mid \neg \psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid \psi U \psi \mid \psi R \psi \mid X\psi$$

Classic LTL over Atomic Propositions

$WSL^{\emptyset}[1G]$

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \wp\flat.\eta$$

$$\eta ::= \varphi \mid \eta \wedge \eta \mid \eta \vee \eta \mid \beta U \varphi \mid \varphi R \beta \mid X\eta$$

No temporal operator is allowed

$$\beta ::= p \mid \neg \beta \mid \beta \wedge \beta \mid \beta \vee \beta$$

Boolean formulae over Atomic Propositions

	$SL[1G]$	$SL^{\emptyset}[1G]$	$WSL^{\emptyset}[1G]$
$EG\varphi,\ A\varphi U q$	✓	✗	✗
$A[(qRp)U\varphi],\ E[\varphi R(pUq)]$	✓	✓	✗
$E[(q \vee t)U\varphi],\ A[\varphi R(s \wedge t)]$	✓	✓	✓

$\boxed{\varphi \triangleq EXp \wedge EX\neg p}$

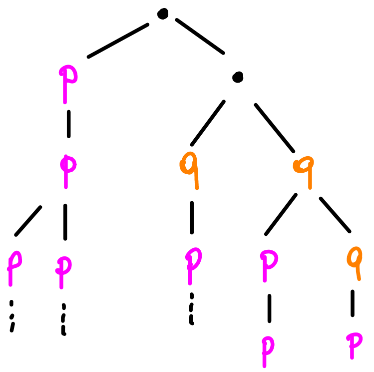
* $SL^{\emptyset}[1G]$ / $WSL^{\emptyset}[1G]$ are still powerful enough to specify finite horizon (state) properties (e.g. common in communication protocols)

Model Theoretic Analysis

Single-Time Satisfaction

$$\varphi = EX\,p \wedge EX\,\neg p \wedge AF\big[\underbrace{A(q\;U\;\underbrace{\exists G\,p}_{\varphi_2})}_{\varphi_1}\big]$$

$q \models \varphi$

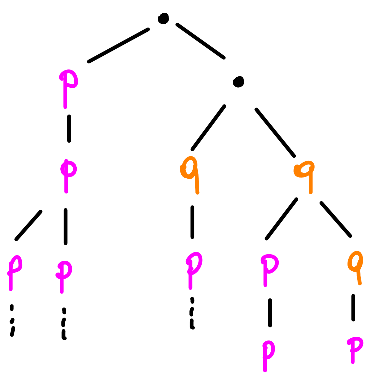


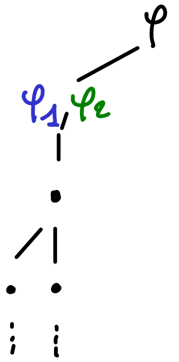
Model Theoretic Analysis

Single-Time Satisfaction

$$\varphi = EXp \wedge EX\neg p \wedge AF[\underbrace{A(q\,U\,\underbrace{\exists Gp}_{\varphi_2})}_{\varphi_1}]$$

$G \models \varphi$



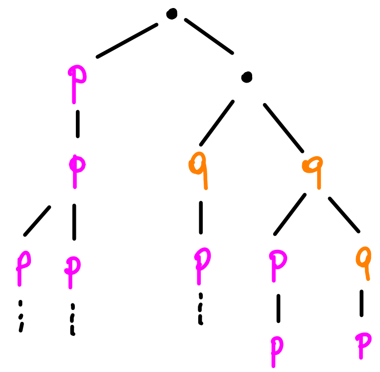


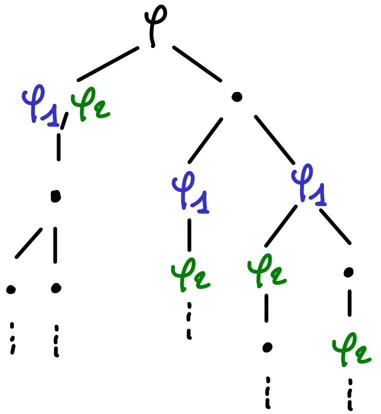
Model Theoretic Analysis

Single-Time Satisfaction

$$\varphi = EXp \wedge EX\neg p \wedge AF[\underbrace{A(q \, U \, \underbrace{\exists G p}_{\varphi_2})}_{\varphi_1}]$$

$G \models \varphi$



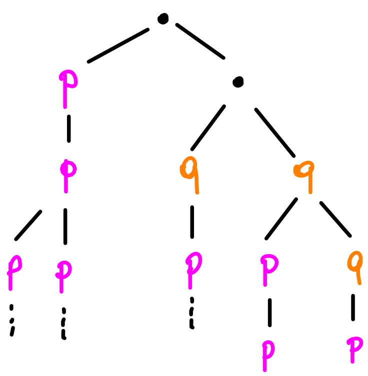


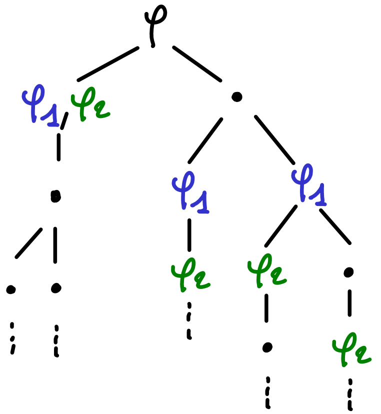
Model Theoretic Analysis

Single-Time Satisfaction

$$\varphi = EXp \wedge EX\neg p \wedge AF[\underbrace{A(q\ U\ \underbrace{\exists G p}_{\varphi_2})}_{\varphi_1}]$$

$G \models \varphi$





Normal Models

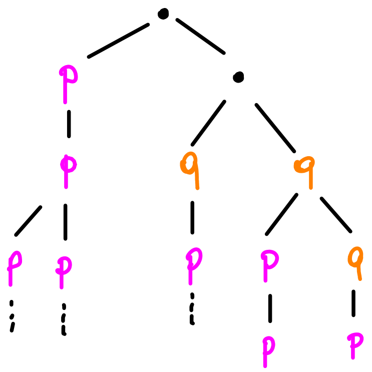
$$\varphi = [\forall x\, \exists y\, (\alpha, x)(\beta, y)\, \varphi_1] \wedge [\exists x\, \forall y\, (\alpha, x)(\beta, y)\, \varphi_2] \wedge [\exists x\, \forall z\, (\alpha, z)(\beta, x)\, \varphi_3]$$

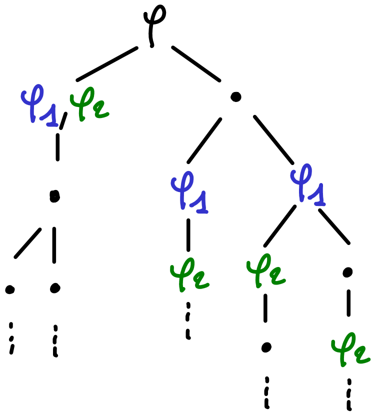
Model Theoretic Analysis

Single-Time Satisfaction

$$\varphi = EXp \wedge EX\neg p \wedge AF[\underbrace{A(q\, U\, \underbrace{\exists G p}_{\varphi_2})}_{\varphi_1}]$$

$G \models \varphi$





Normal Models

$$\varphi = [\forall x \exists y\, (\alpha, x)(\beta, y)\, \psi_1] \wedge [\exists x \forall y\, (\alpha, x)(\beta, y)\, \psi_2] \wedge [\exists x \forall z\, (\alpha, z)(\beta, x)\, \psi_3]$$

Skolemisation

$$\big\downarrow$$

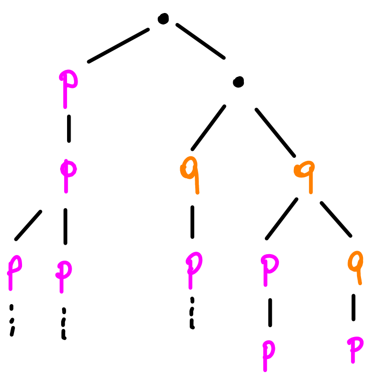
$$Sk(\varphi) = [\forall x\, (\alpha, x)(\beta, f(x))\, \psi_1] \wedge [\forall y\, (\alpha, c_1)(\beta, y)\, \psi_2] \wedge [\forall z\, (\alpha, z)(\beta, c_2)\, \psi_3]$$

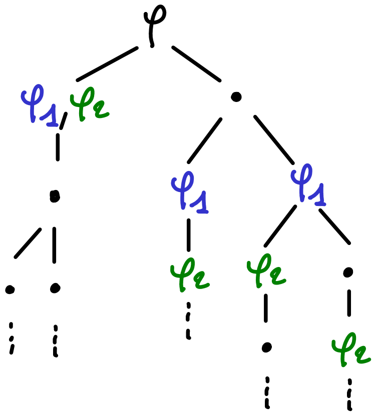
Model Theoretic Analysis

Single-Time Satisfaction

$$\varphi = EX\,p \wedge EX\,\neg p \wedge AF\Big[\underbrace{A\big(q\;U\;\underbrace{\exists G\,p}_{\varphi_2}\big)}_{\varphi_1}\Big]$$

$G \models \varphi$





Normal Models

$$\varphi = \big[\forall x\,\exists y\,(\alpha,x)(\beta,y)\,\psi_1\big] \wedge \big[\exists x\,\forall y\,(\alpha,x)(\beta,y)\,\psi_2\big] \wedge \big[\exists x\,\forall z\,(\alpha,z)(\beta,x)\,\psi_3\big]$$

Skolemisation

$$\downarrow$$

$$sk(\varphi) = \big[\forall x\,\underbrace{(\alpha,x)(\beta,f(x))}\,\psi_1\big] \wedge \big[\forall y\,\underbrace{(\alpha,c_1)(\beta,y)}\,\psi_2\big] \wedge \big[\forall z\,(\alpha,z)(\beta,c_2)\,\psi_3\big]$$

unify

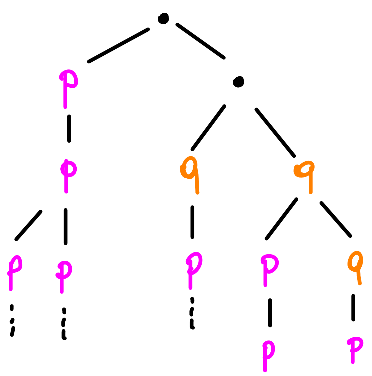
$$(\alpha,c_1)(\beta,f(c_1))$$

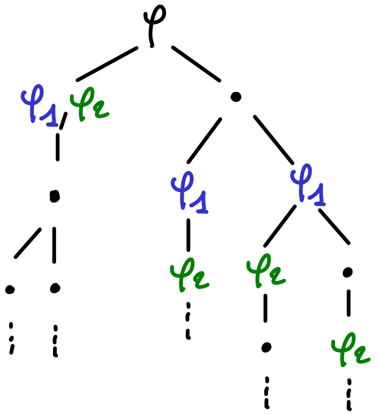
Model Theoretic Analysis

Single-Time Satisfaction

$$\varphi = EXp \wedge EX\neg p \wedge AF\big[\underbrace{A(q\;U\;\underbrace{\exists G p}_{\varphi_2})}_{\varphi_1}\big]$$

$G \models \varphi$





Normal Models

$$\varphi = [\forall x \exists y\,(\alpha,x)(\beta,y)\,\psi_1] \wedge [\exists x \forall y\,(\alpha,x)(\beta,y)\,\psi_2] \wedge [\exists x \forall z\,(\alpha,z)(\beta,x)\,\psi_3]$$

Skolemisation

$\downarrow$

$$Sk(\varphi) = [\forall x\,(\alpha,x)(\beta,f(x))\,\psi_1] \wedge [\forall y\,(\alpha,c_1)(\beta,y)\,\psi_2] \wedge [\forall z\,(\alpha,z)(\beta,c_2)\,\psi_3]$$

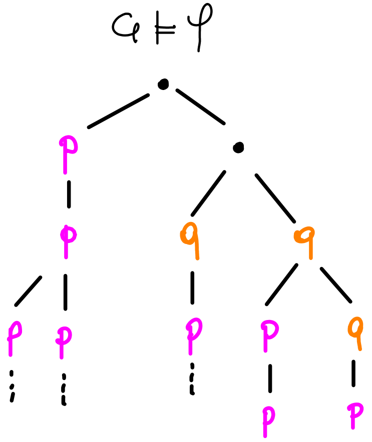
unify
$(\alpha,c_1)(\beta,f(c_1))$

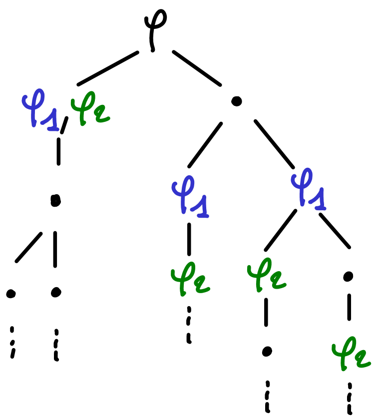
unify
$(\alpha,c_1)(\beta,c_2)$

Model Theoretic Analysis

Single-Time Satisfaction

$$\varphi = EXp \wedge EX\neg p \wedge AF[\underbrace{A(q\,U\,\underbrace{\exists Gp}_{\varphi_2})}_{\varphi_1}]$$





Normal Models

$$\varphi = [\forall x \exists y\, (\alpha, x)(\beta, y)\, \psi_1] \wedge [\exists x \forall y\, (\alpha, x)(\beta, y)\, \psi_2] \wedge [\exists x \forall z\, (\alpha, z)(\beta, x)\, \psi_3]$$

Skolemisation

do not unify

$$Sk(\varphi) = [\forall x\, \underbrace{(\alpha, x)(\beta, f(x))}\, \psi_1] \wedge [\forall y\, \underbrace{(\alpha, c_1)(\beta, y)}\, \psi_2] \wedge [\forall z\, \underbrace{(\alpha, z)(\beta, c_2)}\, \psi_3]$$

unify

$(\alpha, c_1)(\beta, f(c_1))$

unify

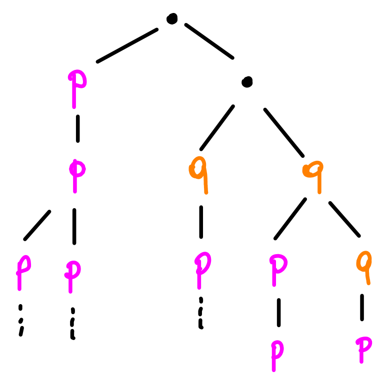
$(\alpha, c_1)(\beta, c_2)$

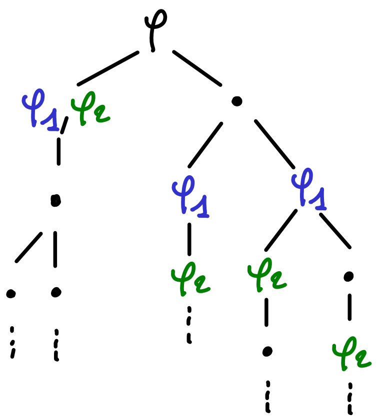
Model Theoretic Analysis

Single-Time Satisfaction

$$\varphi = EX\,p \wedge EX\,\neg p \wedge AF\big[\underbrace{A\big(q\,U\,\underbrace{\exists G\,p}_{\varphi_2}\big)}_{\varphi_1}\big]$$

$q \models \varphi$





Normal Models

$$\varphi = \big[\forall x \exists y\,(\alpha,x)(\beta,y)\,\psi_1\big] \wedge \big[\exists x \forall y\,(\alpha,x)(\beta,y)\,\psi_2\big] \wedge \big[\exists x \forall z\,(\alpha,z)(\beta,x)\,\psi_3\big]$$

Skolemisation

$\downarrow$

do not unify

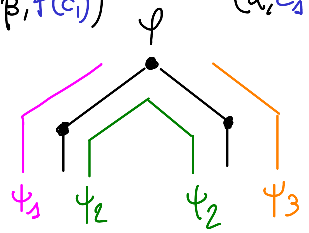
$$Sk(\varphi) = \big[\forall x\,(\alpha,x)(\beta,f(x))\,\psi_1\big] \wedge \big[\forall y\,(\alpha,c_1)(\beta,y)\,\psi_2\big] \wedge \big[\forall z\,(\alpha,z)(\beta,c_2)\,\psi_3\big]$$

unify

$(\alpha,c_1)(\beta,f(c_1))$

unify

$(\alpha,c_1)(\beta,c_2)$

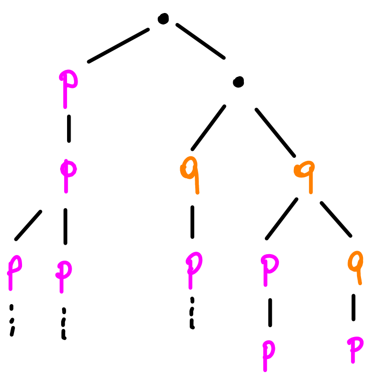


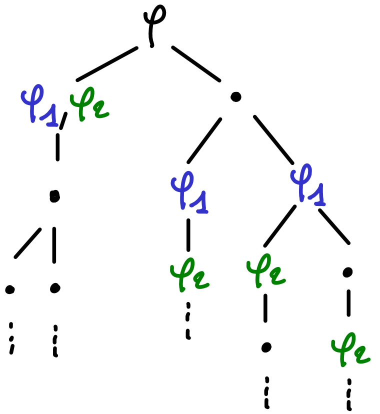
Model Theoretic Analysis

Single-Time Satisfaction

$$\varphi = EX\,p \wedge EX\,\neg p \wedge AF[\underbrace{A(q\,U\,\underbrace{\exists G\,p}_{\varphi_2})}_{\varphi_1}]$$

$\mathcal{G} \models \varphi$





Normal Models

$$\varphi = [\forall x \exists y\, (\alpha, x)(\beta, y)\, \psi_1] \wedge [\exists x \forall y\, (\alpha, x)(\beta, y)\, \psi_2] \wedge [\exists x \forall z\, (\alpha, z)(\beta, x)\, \psi_3]$$

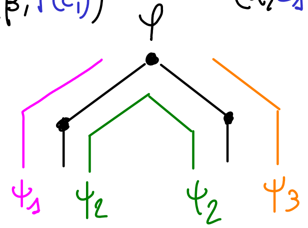
Skolemisation

do not unify

$$sk(\varphi) = [\forall x\, (\alpha, x)(\beta, f(x))\, \psi_1] \wedge [\forall y\, (\alpha, c_1)(\beta, y)\, \psi_2] \wedge [\forall z\, (\alpha, z)(\beta, c_2)\, \psi_3]$$

unify $(\alpha, c_1)(\beta, f(c_1))$

unify $(\alpha, c_1)(\beta, c_2)$



* Every $SL^{\emptyset}[1G]$ / $WSL^{\emptyset}[1G]$ satisfiable sentence has a **k-fork normal model** (every branch has at most k forks), with $k \leq |\varphi|$

New Classes of Automata

Bounded-Fork Tree Automata

* A Nondeterministic Tree Automaton is K-fork, for some $K \in \mathbb{N}$, if

Bounded-Fork Tree Automata

* A Nondeterministic Tree Automaton is K-fork, for some $K \in \mathbb{N}$, if the set of states Q can be partitioned in $K+1$ sets

$$Q_K \quad Q_{K-1} \cdots Q_1 \quad Q_0$$

Bounded-Fork Tree Automata

* A Nondeterministic Tree Automaton is K-fork, for some $K \in \mathbb{N}$, if the set of states Q can be partitioned in $K+1$ sets

Initial state $\rightarrow$ $Q_K \quad Q_{K-1} \cdots Q_1 \quad Q_0$

Bounded-Fork Tree Automata

* A Nondeterministic Tree Automaton is K-fork, for some $K \in \mathbb{N}$, if the set of states Q can be partitioned in $K+1$ sets

Initial state $\longrightarrow$ $Q_K \quad Q_{K-1} \cdots Q_1 \quad Q_0$ $\longleftarrow$ Acceptance condition

Bounded-Fork Tree Automata

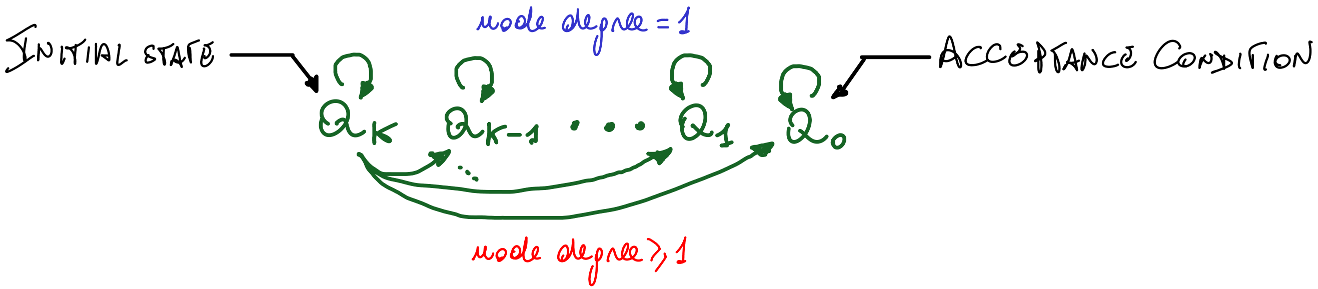
* A Nondeterministic Tree Automaton is K-fork, for some $K \in \mathbb{N}$, if the set of states Q can be partitioned in $K+1$ sets

node degree = 1

Initial state $\longrightarrow$ $Q_K \quad Q_{K-1} \quad \cdots \quad Q_1 \quad Q_0$ $\longleftarrow$ Acceptance condition

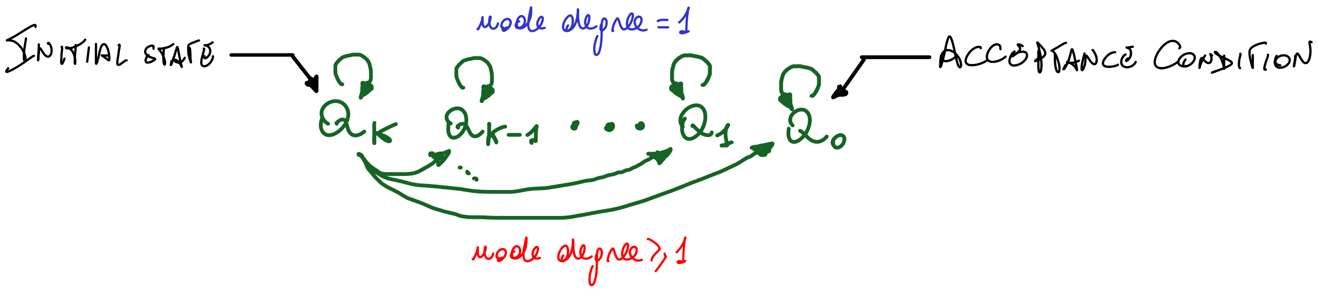
Bounded-Fork Tree Automata

* A Nondeterministic Tree Automaton is K-fork, for some $K \in \mathbb{N}$, if the set of states Q can be partitioned in $K+1$ sets



Bounded-Fork Tree Automata

* A Nondeterministic Tree Automaton is K-fork, for some $K \in \mathbb{N}$, if the set of states Q can be partitioned in $K+1$ sets



* Emptiness Problem: $\mathrm{SPACE}\left(K \cdot \log n + (\log n)^2 + \log m\right) \,/\, \mathrm{ATIME}[K\text{-ALT}]\left(\log n + \log m\right)$

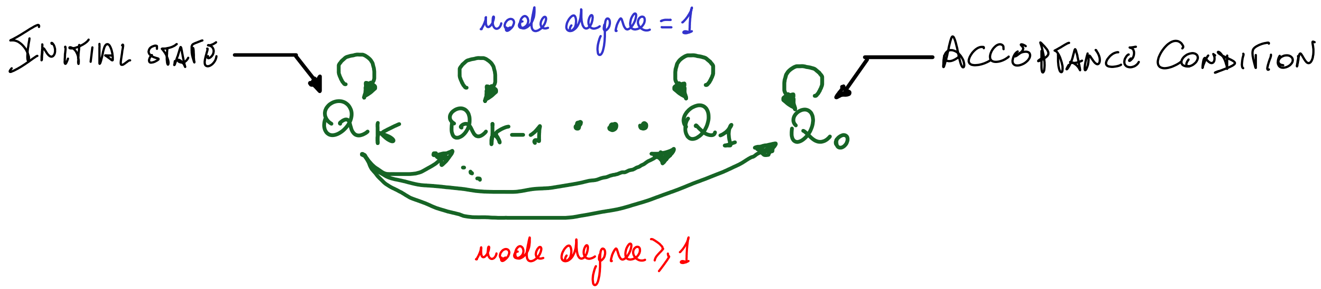
↳ For Büchi Bounded-Fork NTA

K : bound on the number of forks
n : number of states
m : number of transitions

Bounded-Fork Tree Automata

* A Nondeterministic Tree Automaton is **K-fork**, for some $K \in \mathbb{N}$, if

  the set of states Q can be partitioned in $K+1$ sets



* Emptiness Problem: $\mathrm{SPACE}\left(K \cdot \log n + (\log n)^2 + \log m\right) / \mathrm{ATIME}[K\text{-ALT}](\log n + \log m)$

  For Büchi Bounded-Fork NTA

  K: bound on the number of forks
  n: number of states
  m: number of transitions

* Recall that the emptiness of Büchi NTA is PTIME-C in the size of the automaton (Vardi, Wolper '86)

Prefix-Deterministic Word Automata

* Safraless approach to Satisfiability: Find a SUITABLE Nondeterministic Word Automaton to run over a tree

  + A NWA W is a Good-for-Game Automaton if $wot(W)$ accepts all trees T whose branches are accepted by W

    Word-on-tree: NTA running W on all branches

Prefix-Deterministic Word Automata

* Safraless approach to Satisfiability: Find a SUITABLE Nondeterministic Word Automaton to run over a tree

  + A NWA W is a Good-for-Game Automaton if wot(W) accepts all trees T whose branches are accepted by W

    Word-on-tree: NTA running W on all branches

* A Nondeterministic Word Automaton W is prefix deterministic, if

Prefix-Deterministic Word Automata

* Safraless approach to Satisfiability: Find a SUITABLE Nondeterministic Word Automaton to run over a tree

  + A NWA W is a Good-for-Game Automaton if $\mathrm{wot}(W)$ accepts all trees T whose branches are accepted by W

    Word-on-tree: NTA running W on all branches

* A Nondeterministic Word Automaton W is prefix deterministic, if its transition function δ contains a deterministic transition subfunction $\tilde{\delta} \subseteq \delta$ s.t. every finite prefix ρ of a run $\pi = \rho \cdot \pi'$ is compatible with $\tilde{\delta}$, i.e., every finite execution of W can be simulated deterministically

Prefix-Deterministic Word Automata

* Safraless approach to Satisfiability: Find a SUITABLE Nondeterministic Word Automaton to run over a tree

  + A NWA W is a Good-for-Game Automaton if $wot(W)$ accepts all trees T whose branches are accepted by W
    Word-on-tree: NTA running W on all branches

* A Nondeterministic Word Automaton W is prefix deterministic, if its transition function δ contains a deterministic transition subfunction $\tilde{\delta} \subseteq \delta$ s.t. every finite prefix ρ of a run $\pi = \rho \cdot \pi'$ is compatible with $\tilde{\delta}$, i.e., every finite execution of W can be simulated deterministically

* Every PD-NWA is a Good-for-Game Automaton w.r.t. the class of bounded-fork trees

Prefix-Deterministic Word Automata

* Safraless approach to Satisfiability: Find a SUITABLE Nondeterministic Word Automaton to run over a tree
    + A NWA W is a Good-for-Game Automaton if wot(W) accepts all trees T whose branches are accepted by W
      Word-on-tree: NTA running W on all branches

* A Nondeterministic Word Automaton W is prefix deterministic, if
  its transition function δ contains a deterministic transition subfunction $\tilde{\delta} \subseteq \delta$ s.t. every finite prefix ρ of a run $\pi = \rho \cdot \pi'$ is compatible with $\tilde{\delta}$, i.e., every finite execution of W can be simulated deterministically

* Every PD-NWA is a Good-for-Game Automaton w.r.t. the class of bounded-fork trees

* For every LTL/WLTL formula φ there exists a Büchi PD-NWA W_φ with $2^{O(2^{|\varphi|})}$ / $2^{O(|\varphi|^3)}$ many states
  Fragment of LTL underlying $WSL^{\emptyset}[1G]$ — $w \in \mathcal{L}(W_\varphi)$ iff $w \models \varphi$

The Satisfiability Problem

Automata Theoretic Approach to Sat

* For every $SL^{\emptyset}[1G]$ / $WSL^{\emptyset}[1G]$ sentence φ we build a bounded-fork NTA $N_\varphi = D_{NM} \times N_{PF}$ s.t.

Automata Theoretic Approach to Sat

* For every $SL^{\emptyset}[1G]$ / $WSL^{\emptyset}[1G]$ sentence φ we build a bounded-fork NTA $N_\varphi = D_{NM} \times N_{PF}$ s.t.

  + D_{NM} : Safety Deterministic Tree Automaton verifying that the accepted tree T is a K-fork normal model

    Exponential in the number of binding prefixes in φ

    number of subsentences in φ

Automata Theoretic Approach to Sat

* For every $SL^{\emptyset}[1G]$/$WSL^{\emptyset}[1G]$ sentence φ we build a bounded-fork NTA $N_\varphi = D_{NM} \times N_{PF}$ s.t.

  + D_{NM} : Safety Deterministic Tree Automaton verifying that the accepted tree T is a K-fork normal model

    Exponential in the number of binding prefixes in φ — number of subsentences in φ

  + N_{PF} : Product of word-on-trees transformations of the Büchi PD-NWAs W_ψ for all LTL/WLTL formulae ψ in φ

    Doubly/Singly exponential in the total length of the LTL/WLTL formulae in φ

Automata Theoretic Approach to Sat

* For every $SL^{\emptyset}[1G]$ / $WSL^{\emptyset}[1G]$ sentence φ we build a bounded-fork NTA $N_\varphi = D_{NM} \times N_{PF}$ s.t.

  + D_{NM} : Safety Deterministic Tree Automaton verifying that the accepted tree T is a K-fork normal model

    └ Exponential in the number of binding prefixes in φ

    number of subsentences in φ

  + N_{PF} : Product of word-on-trees transformations of the Büchi PD-NWAs W_ψ for all LTL / WLTL formulae ψ in φ

    └ Doubly/Singly exponential in the total length of the LTL / WLTL formulae in φ

* The Satisfiability Problem of $SL^{\emptyset}[1G]$ / $WSL^{\emptyset}[1G]$ can be solved in AExpTime[k-alt] / PSpace !

Conclusions

Some Results

* Some Non-Recurrent fragments of SL[1G] have simpler satisfiability problems than model-checking problems

	SAT		MC
SL[1G]	2EXPTIME-C	=	2EXPTIME-C
└ $SL^{\emptyset}$[1G]	AEXPTIME[k-alt]	≤	2EXPTIME-C
└ $WSL^{\emptyset}$[1G]	PSPACE-C	<	2EXPTIME-C

Some Results

* Some Non-Recurrent fragments of SL[1G] have simpler satisfiability problems than model-checking problems

	SAT		MC
SL[1G]	2EXPTIME-c	$=$	2EXPTIME-c
$SL^{\emptyset}$[1G]	AEXPTIME[k-alt]	$\leq$	2EXPTIME-c
$WSL^{\emptyset}$[1G]	PSPACE-c	$<$	2EXPTIME-c
ATL^{*}	2EXPTIME-c	$=$	2EXPTIME-c
$ATL^{*\emptyset}$	AEXPTIME[k-alt]	$\leq$	2EXPTIME-c
$WATL^{*\emptyset}$	PSPACE-c	$<$	2EXPTIME-c

Some Results

* Some Non-Recurrent fragments of SL[1G] have simpler satisfiability problems than model-checking problems

	SAT		MC
SL[1G]	2ExpTime-c	=	2ExpTime-c
$SL^{\emptyset}$[1G]	AExpTime[k-dt]	$\leq$	2ExpTime-c
$WSL^{\emptyset}$[1G]	PSpace-c	<	2ExpTime-c
ATL^{*}	2ExpTime-c	=	2ExpTime-c
$ATL^{*\emptyset}$	AExpTime[k-dt]	$\leq$	2ExpTime-c
$WATL^{*\emptyset}$	PSpace-c	<	2ExpTime-c
CTL^{*}	2ExpTime-c	>	PSpace-c
$CTL^{*\emptyset}$	AExpTime[k-dt]	>	PSpace-c
$WCTL^{*\emptyset}$	PSpace-c	=	PSpace-c

Some Results

* Some Non-Recurrent fragments of SL[1G] have simpler *satisfiability problems* than *model-checking problems*

	SAT		MC
SL[1G]	2EXPTIME-c	=	2EXPTIME-c
└ $SL^{\emptyset}$[1G]	AEXPTIME[k-alt]	≤	2EXPTIME-c
└ $WSL^{\emptyset}$[1G]	PSPACE-c	<	2EXPTIME-c
ATL^{*}	2EXPTIME-c	=	2EXPTIME-c
└ $ATL^{*\emptyset}$	AEXPTIME[k-alt]	≤	2EXPTIME-c
└ $WATL^{*\emptyset}$	PSPACE-c	<	2EXPTIME-c
CTL^{*}	2EXPTIME-c	>	PSPACE-c
└ $CTL^{*\emptyset}$	AEXPTIME[k-alt]	>	PSPACE-c
└ $WCTL^{*\emptyset}$	PSPACE-c	=	PSPACE-c
SL[CG]	?		2EXPTIME-c
└ FSL[CG]	PSPACE-c	<	2EXPTIME-c

Some Results

* Some Non-Recurrent fragments of SL[1G] have simpler satisfiability problems than model-checking problems

	SAT		MC
SL[1G]	2ExpTime-c	=	2ExpTime-c
$SL^{\emptyset}$[1G]	AExpTime[k-dt]	≤	2ExpTime-c
$WSL^{\emptyset}$[1G]	PSpace-c	<	2ExpTime-c
ATL^{*}	2ExpTime-c	=	2ExpTime-c
$ATL^{*\emptyset}$	AExpTime[k-dt]	≤	2ExpTime-c
$WATL^{*\emptyset}$	PSpace-c	<	2ExpTime-c
CTL^{*}	2ExpTime-c	>	PSpace-c
$CTL^{*\emptyset}$	AExpTime[k-dt]	>	PSpace-c
$WCTL^{*\emptyset}$	PSpace-c	=	PSpace-c
SL[CG]	?		2ExpTime-c
FSL[CG]	PSpace-c	<	2ExpTime-c

Few Open Problems

* Is the satisfiability of $SL^{\emptyset}$[1G]/$ATL^{*\emptyset}$/$CTL^{*\emptyset}$ AExpTime[k-dt]-c?

* What is the most expressive PSpace fragment of $SL^{\emptyset}$[1G]?

* Are SL[CG]/$SL^{\emptyset}$[1G] decidable?

* What are the most expressive fragments of SL with a satisfiability problem simpler than the model-checking one?

Thank You Very Much