A Decision Procedure for CTL* Based on Tableaux and Automata

Oliver Friedmann[1] Markus Latte[1] Martin Lange[2]

[1] Dept. of Computer Science, Ludwig-Maximilians-University, Munich, Germany

[2] Dept. of Electrical Engineering and Computer Science, University of Kassel, Germany

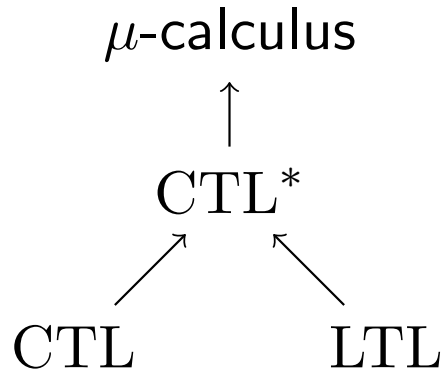
IJCAR

Edinburgh, 16–19 July 2010

Introduction to CTL*

Origin: Emerson and Halpern '86

- supersedes the branching-time logic CTL and the linear-time logic LTL



- applied to specify and verify reactive and agent-based systems
- also applied to program synthesis
- however: decision procedures difficult to obtain
- worst case runtime: doubly exponential
  - lower bound: Vardi and Stockmeyer '85
  - upper bound: Emerson and Sistla '84; Emerson and Jutla '00

Decision procedures

Emerson-Jutla Method ('84)

- emptiness test of a tree automaton accepting all models
- drawbacks: no implementation, unintuitive proof structure, constant branching degree

Reynolds' Tableaux ('09)

- exhaustive tableau-search restricted by small model property
- drawbacks: fairly slow in practice, no intrinsic detection of unfulfilled eventualities

Our System

- existence of infinite tableaux with global conditions
- drawbacks: requires automata deterministation for checking global conditions

Table of contents

Syntax of CTL*

Negation normal form

$$\psi ::= q \mid \neg q \mid \psi \wedge \psi \mid \psi \vee \psi \mid \mathtt{X}\psi \mid \psi \mathtt{U} \psi \mid \psi \mathtt{R} \psi \mid \mathtt{E}\psi \mid \mathtt{A}\psi$$

where $q \in \mathcal{P}$ are propositional constants

Syntax of CTL*

Negation normal form

$$\psi ::= q \mid \neg q \mid \psi \wedge \psi \mid \psi \vee \psi \mid \mathtt{X}\psi \mid \cancel{\psi \mathtt{U} \psi} \mid \cancel{\psi \mathtt{R} \psi} \mid \mathtt{E}\psi \mid \mathtt{A}\psi$$

where $q \in \mathcal{P}$ are propositional constants

This talk: replace fixpoints $\psi \mathtt{U} \psi$, $\psi \mathtt{R} \psi$ by $\mathtt{F}\psi$, $\mathtt{G}\psi$.

Syntax of CTL*

Negation normal form

$$\psi ::= q \mid \neg q \mid \psi \wedge \psi \mid \psi \vee \psi \mid \mathtt{X}\psi \mid \mathtt{F}\psi \mid \mathtt{G}\psi \mid \mathtt{E}\psi \mid \mathtt{A}\psi$$

where $q \in \mathcal{P}$ are propositional constants

This talk: replace fixpoints $\psi \mathtt{U} \psi$, $\psi \mathtt{R} \psi$ by $\mathtt{F}\psi$, $\mathtt{G}\psi$.

Interpretation

Transition systems

TS $\mathcal{T} = (\mathcal{S}, \rightarrow, \lambda)$ with

- $(\mathcal{S}, \rightarrow)$ directed, total graph
- $\lambda : \mathcal{S} \rightarrow 2^{\mathcal{P}}$ labeling function

Interpretation

Transition systems

TS $\mathcal{T} = (\mathcal{S}, \rightarrow, \lambda)$ with

- $(\mathcal{S}, \rightarrow)$ directed, total graph
- $\lambda : \mathcal{S} \rightarrow 2^{\mathcal{P}}$ labeling function

Path π: sequence $(s_i)_{i \in \mathbb{N}} = s_0, s_1, \ldots$ of states respecting edges

Interpretation

Transition systems

TS $\mathcal{T} = (\mathcal{S}, \rightarrow, \lambda)$ with

- $(\mathcal{S}, \rightarrow)$ directed, total graph
- $\lambda : \mathcal{S} \rightarrow 2^{\mathcal{P}}$ labeling function

Path π: sequence $(s_i)_{i \in \mathbb{N}} = s_0, s_1, \ldots$ of states respecting edges

Notations: $\pi^i = s_i, s_{i+1}, \ldots$

Semantics

Semantics of Formulas

- $\mathcal{T}, \pi \models q$ iff $q \in \lambda(\pi(0))$
- $\mathcal{T}, \pi \models \neg q$ iff $q \notin \lambda(\pi(0))$
- $\mathcal{T}, \pi \models \psi_1 \wedge \psi_2$ iff $\mathcal{T}, \pi \models \psi_1$ and $\mathcal{T}, \pi \models \psi_2$
- $\mathcal{T}, \pi \models \psi_1 \vee \psi_2$ iff $\mathcal{T}, \pi \models \psi_1$ or $\mathcal{T}, \pi \models \psi_2$
- $\mathcal{T}, \pi \models \mathtt{X}\psi$ iff $\mathcal{T}, \pi^1 \models \psi$
- $\mathcal{T}, \pi \models \mathtt{F}\psi$ iff $\mathcal{T}, \pi^i \models \psi$ for some $i \in \mathbb{N}$
- $\mathcal{T}, \pi \models \mathtt{G}\psi$ iff $\mathcal{T}, \pi^i \models \psi$ for all $i \in \mathbb{N}$
- $\mathcal{T}, \pi \models \mathtt{E}\psi$ iff $\mathcal{T}, \widetilde{\pi} \models \psi$ for some $\widetilde{\pi}$ with $\pi(0) = \widetilde{\pi}(0)$
- $\mathcal{T}, \pi \models \mathtt{A}\psi$ iff $\mathcal{T}, \widetilde{\pi} \models \psi$ for all $\widetilde{\pi}$ with $\pi(0) = \widetilde{\pi}(0)$

A Tableau for CTL*

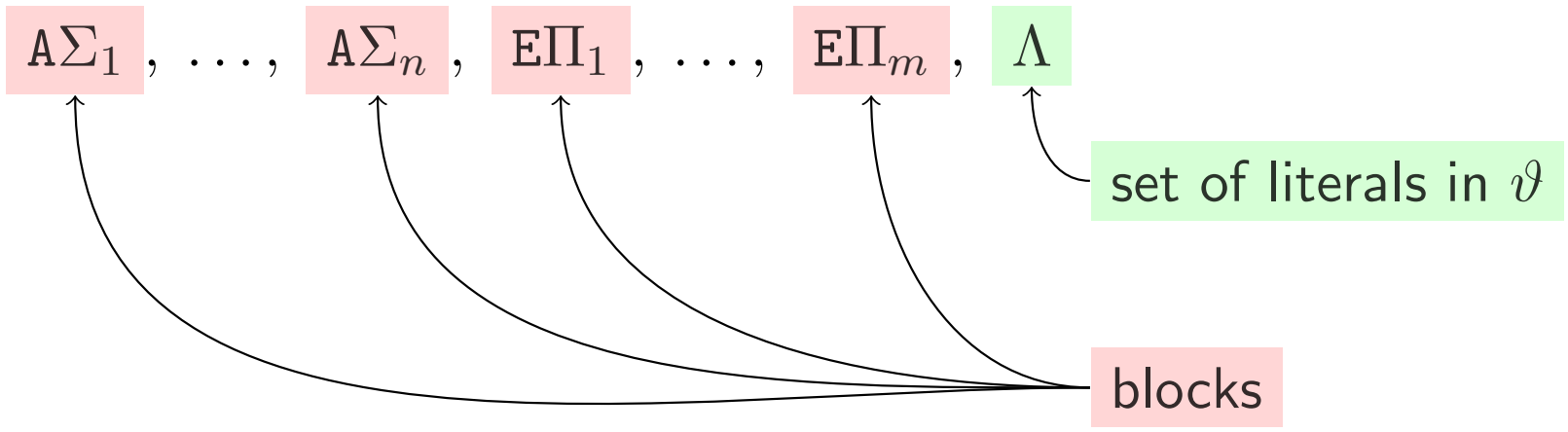
A tableau for ϑ is a tree which imitates a potential model of ϑ.

A Tableau for CTL*

A tableau for ϑ is a tree which imitates a potential model of ϑ.

A pre-tableau for a formula ϑ is an infinite tree s.th.

- it is finitely branching,
- each node is labelled with a goal (as a set),



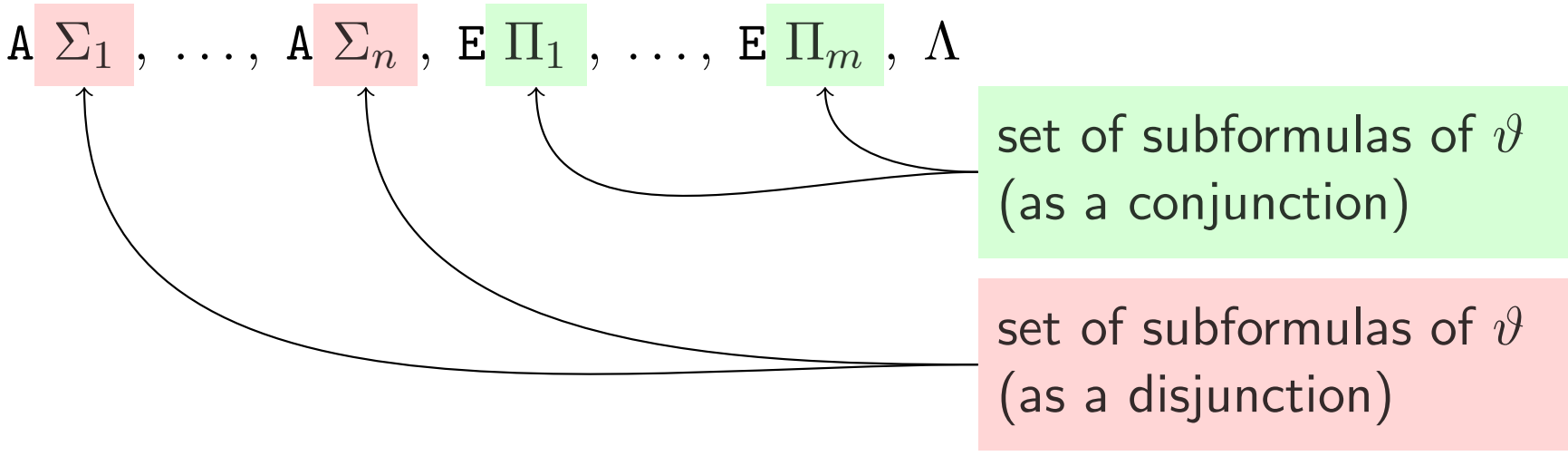
Example: $\{\mathtt{A}\{\neg p \vee q\},\ \mathtt{E}\{\mathtt{X}p, \mathtt{F}q\},\ \neg p,\ \neg q\}$.
Sloppy writing: $\mathtt{A}(\neg p \vee q)$ or $\mathtt{E}(\mathtt{X}p, \Pi)$, e.g.

A Tableau for CTL*

A tableau for ϑ is a tree which imitates a potential model of ϑ.

A pre-tableau for a formula ϑ is an infinite tree s.th.

- it is finitely branching,
- each node is labelled with a goal (as a set),



Example: $\{\mathtt{A}\{\neg p \vee q\},\ \mathtt{E}\{\mathtt{X}p, \mathtt{F}q\},\ \neg p,\ \neg q\}$.
Sloppy writing: $\mathtt{A}(\neg p \vee q)$ or $\mathtt{E}(\mathtt{X}p, \Pi)$, e.g.

A Tableau for CTL*

A tableau for ϑ is a tree which imitates a potential model of ϑ.

A pre-tableau for a formula ϑ is an infinite tree s.th.

- it is finitely branching,
- each node is labelled with a goal (as a set),

$$\underbrace{\mathtt{A}\Sigma_1,\ \ldots,\ \mathtt{A}\Sigma_n,\ \mathtt{E}\Pi_1,\ \ldots,\ \mathtt{E}\Pi_m,\ \Lambda}$$

$$\bigwedge_{i=1}^{n} \mathtt{A}\Big(\bigvee \Sigma_i\Big)\ \wedge\ \bigwedge_{i=1}^{m} \mathtt{E}\Big(\bigwedge \Pi_i\Big)\ \wedge\ \bigwedge \Lambda$$

Example: $\{\mathtt{A}\{\neg p \vee q\},\ \mathtt{E}\{\mathtt{X}p, \mathtt{F}q\},\ \neg p,\ \neg q\}$.
Sloppy writing: $\mathtt{A}(\neg p \vee q)$ or $\mathtt{E}(\mathtt{X}p, \Pi)$, e.g.

A Tableau for CTL*

A tableau for ϑ is a tree which imitates a potential model of ϑ.

A pre-tableau for a formula ϑ is an infinite tree s.th.

- it is finitely branching,
- each node is labelled with a goal (as a set),

  $\mathtt{A}\Sigma_1,\ \ldots,\ \mathtt{A}\Sigma_n,\ \mathtt{E}\Pi_1,\ \ldots,\ \mathtt{E}\Pi_m,\ \Lambda$
- nodes are locally consistent, i.e.
  - does not contain a literal together with its negation, and
  - does not contain $\mathtt{A}\emptyset$.
- root is labelled with $\mathtt{E}\{\vartheta\}$,
- nodes follow the following rules ...

Rules of the Tableau – Logical Rules

Notations: Trees are shown botanically correct.
Symbol $\ldots \mid \ldots$ separates alterative premisses.

Rules for Boolean connectives

$$(\mathtt{E}\vee)\ \frac{\mathtt{E}(\varphi,\Pi),\Phi \quad\mid\quad \mathtt{E}(\psi,\Pi),\Phi}{\mathtt{E}(\varphi\vee\psi,\Pi),\Phi} \qquad (\mathtt{E}\wedge)\ \frac{\mathtt{E}(\varphi,\psi,\Pi),\Phi}{\mathtt{E}(\varphi\wedge\psi,\Pi),\Phi}$$

$$(\mathtt{A}\vee)\ \frac{\mathtt{A}(\varphi,\psi,\Sigma),\Phi}{\mathtt{A}(\varphi\vee\psi,\Sigma),\Phi} \qquad (\mathtt{A}\wedge)\ \frac{\mathtt{A}(\varphi,\Sigma),\mathtt{A}(\psi,\Sigma),\Phi}{\mathtt{A}(\varphi\wedge\psi,\Sigma),\Phi}$$

$$(\mathtt{Ett})\ \frac{\Phi}{\mathtt{E}\emptyset,\Phi}$$

Rules for literals

$$(\mathtt{El})\ \frac{\ell\ ,\mathtt{E}\Pi,\Phi}{\mathtt{E}(\ \ell\ ,\Pi),\Phi}$$

$$(\mathtt{Al})\ \frac{\ell\ ,\Phi \quad\mid\quad \mathtt{A}\Sigma,\Phi}{\mathtt{A}(\ \ell\ ,\Sigma),\Phi}$$

Rules of the Tableau – Logical Rules

Notations: Trees are shown botanically correct.
Symbol $\ldots \mid \ldots$ separates alterative premisses.

Rules for Boolean connectives

$$(\mathtt{E}\vee)\ \frac{\mathtt{E}(\varphi,\Pi),\Phi \quad\mid\quad \mathtt{E}(\psi,\Pi),\Phi}{\mathtt{E}(\varphi\vee\psi,\Pi),\Phi} \qquad (\mathtt{E}\wedge)\ \frac{\mathtt{E}(\varphi,\psi,\Pi),\Phi}{\mathtt{E}(\varphi\wedge\psi,\Pi),\Phi}$$

$$(\mathtt{A}\vee)\ \frac{\mathtt{A}(\varphi,\psi,\Sigma),\Phi}{\mathtt{A}(\varphi\vee\psi,\Sigma),\Phi} \qquad (\mathtt{A}\wedge)\ \frac{\mathtt{A}(\varphi,\Sigma),\mathtt{A}(\psi,\Sigma),\Phi}{\mathtt{A}(\varphi\wedge\psi,\Sigma),\Phi}$$

$$(\mathtt{Ett})\ \frac{\Phi}{\mathtt{E}\emptyset,\Phi}$$

Rules for literals and path quantifiers

$$(\mathtt{El})\ \frac{\ell,\mathtt{E}\Pi,\Phi}{\mathtt{E}(\ell,\Pi),\Phi} \qquad (\mathtt{EE})\ \frac{\mathtt{E}\varphi,\mathtt{E}\Pi,\Phi}{\mathtt{E}(\mathtt{E}\varphi,\Pi),\Phi} \qquad (\mathtt{EA})\ \frac{\mathtt{A}\varphi,\mathtt{E}\Pi,\Phi}{\mathtt{E}(\mathtt{A}\varphi,\Pi),\Phi}$$

$$(\mathtt{Al})\ \frac{\ell,\Phi \quad\mid\quad \mathtt{A}\Sigma,\Phi}{\mathtt{A}(\ell,\Sigma),\Phi} \qquad (\mathtt{AE})\ \frac{\mathtt{E}\varphi,\Phi \quad\mid\quad \mathtt{A}\Sigma,\Phi}{\mathtt{A}(\mathtt{E}\varphi,\Sigma),\Phi} \qquad (\mathtt{AA})\ \frac{\mathtt{A}\varphi,\Phi \quad\mid\quad \mathtt{A}\Sigma,\Phi}{\mathtt{A}(\mathtt{A}\varphi,\Sigma),\Phi}$$

Rules of the Tableau – Temporal Rules

Characterisation as fixed points

$$\mathtt{F}\varphi \leftrightarrow \varphi \vee \mathtt{X}(\mathtt{F}\varphi) \qquad \mathtt{G}\varphi \leftrightarrow \varphi \wedge \mathtt{X}(\mathtt{G}\varphi).$$

Corresponding rules

$$(\mathtt{EF})\ \frac{\mathtt{E}(\psi, \Pi), \Phi \quad | \quad \mathtt{E}(\mathtt{X}(\mathtt{F}\psi), \Pi), \Phi}{\mathtt{E}(\mathtt{F}\psi, \Pi), \Phi} \qquad (\mathtt{EG})\ \frac{\mathtt{E}(\psi, \mathtt{X}(\mathtt{G}\psi), \Pi), \Phi}{\mathtt{E}(\mathtt{G}\psi, \Pi), \Phi}$$

$$(\mathtt{AF})\ \frac{\mathtt{A}(\psi, \mathtt{X}(\mathtt{F}\psi), \Sigma), \Phi}{\mathtt{A}(\mathtt{F}\psi, \Sigma), \Phi} \qquad (\mathtt{AG})\ \frac{\mathtt{A}(\psi, \Sigma), \mathtt{A}(\mathtt{X}(\mathtt{G}\psi), \Sigma), \Phi}{\mathtt{A}(\mathtt{G}\psi, \Sigma), \Phi}$$

Rules of the Tableau – Successor Rules

Successor Rules

$$(\mathtt{X}_1)\ \frac{\mathtt{E}\Pi_1, \mathtt{A}\Sigma_1, \ldots, \mathtt{A}\Sigma_m \quad \ldots \quad \mathtt{E}\Pi_n, \mathtt{A}\Sigma_1, \ldots, \mathtt{A}\Sigma_m \qquad (n > 0)}{\mathtt{EX}\Pi_1, \ldots, \mathtt{EX}\Pi_n, \mathtt{AX}\Sigma_1, \ldots, \mathtt{AX}\Sigma_m, \Lambda}$$

$$(\mathtt{X}_0)\ \frac{\mathtt{A}\Sigma_1, \ldots, \mathtt{A}\Sigma_m}{\mathtt{AX}\Sigma_1, \ldots, \mathtt{AX}\Sigma_m, \Lambda}$$

Notation: $\mathtt{X}\Gamma := \{\mathtt{X}\gamma \mid \gamma \in \Gamma\}$.

Note: rule $(\mathtt{X}_0)$ ensures that the intended model is total.

Rules of the Tableau – Successor Rules

Successor Rules

$$(\mathtt{X}_1)\ \frac{\mathtt{E}\Pi_1, \mathtt{A}\Sigma_1, \ldots, \mathtt{A}\Sigma_m \quad \ldots \quad \mathtt{E}\Pi_n, \mathtt{A}\Sigma_1, \ldots, \mathtt{A}\Sigma_m \qquad (n > 0)}{\mathtt{EX}\Pi_1, \ldots, \mathtt{EX}\Pi_n, \mathtt{AX}\Sigma_1, \ldots, \mathtt{AX}\Sigma_m, \Lambda}$$

$$(\mathtt{X}_0)\ \frac{\mathtt{A}\Sigma_1, \ldots, \mathtt{A}\Sigma_m}{\mathtt{AX}\Sigma_1, \ldots, \mathtt{AX}\Sigma_m, \Lambda}$$

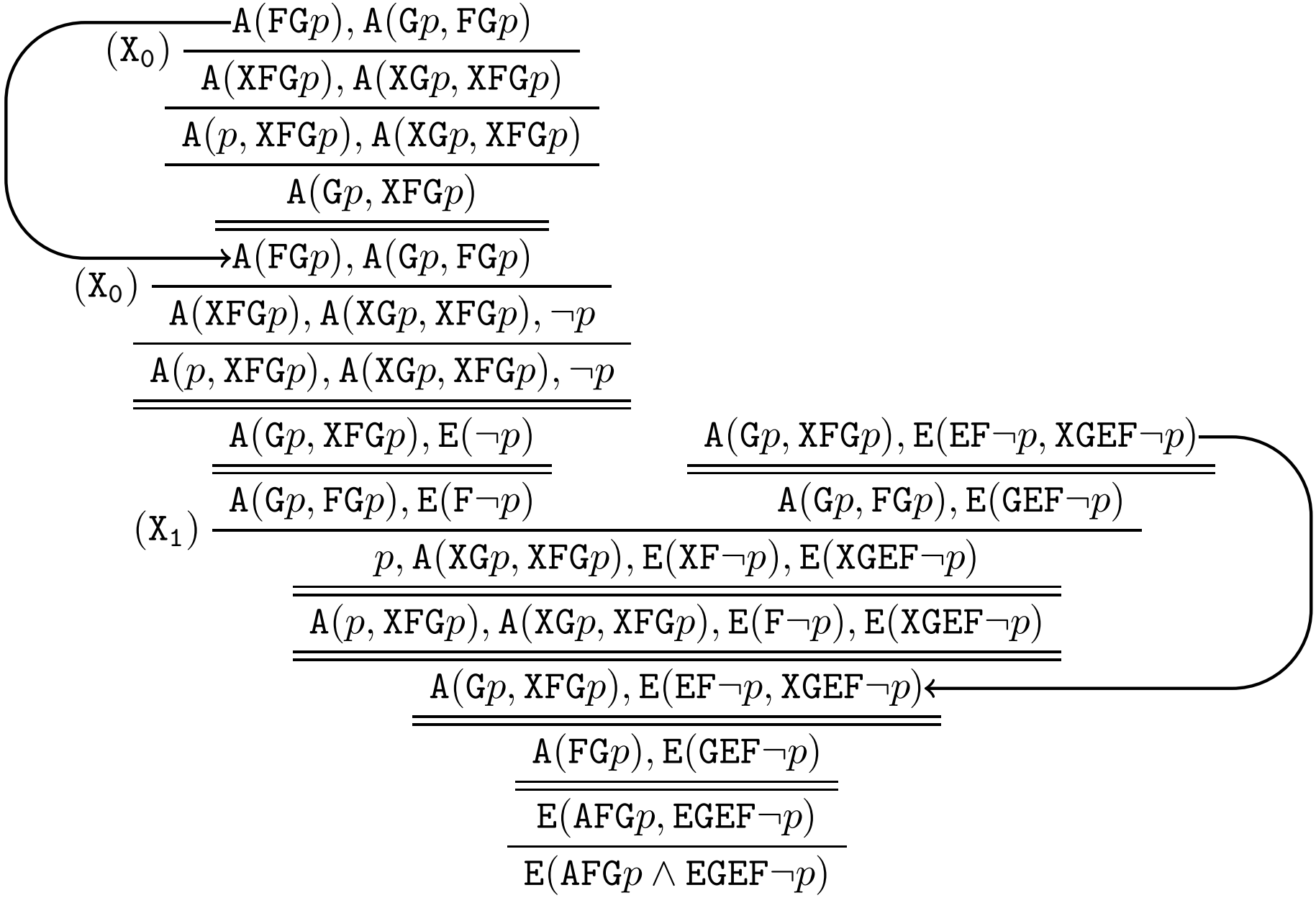
Notation: $\mathtt{X}\Gamma := \{\mathtt{X}\gamma \mid \gamma \in \Gamma\}$.

Note: rule $(\mathtt{X}_0)$ ensures that the intended model is total.

Lemma

Every infinite branch of a pre-tableau contains infinitely many applications of rules $(\mathtt{X}_0)$ or $(\mathtt{X}_1)$.

Pre-Tableau for $\mathtt{AFG}p \wedge \mathtt{EGEF}\neg p$

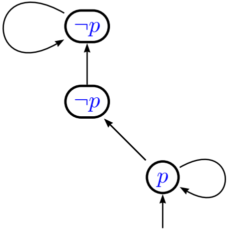


Pre-Tableau for $\mathtt{AFG}p \wedge \mathtt{EGEF}\neg p$

$$\mathtt{A}(\mathtt{FG}p), \mathtt{A}(\mathtt{G}p, \mathtt{FG}p)$$
$(\mathtt{X}_0)$
$$\mathtt{A}(\mathtt{XFG}p), \mathtt{A}(\mathtt{XG}p, \mathtt{XFG}p)$$
$$\mathtt{A}(p, \mathtt{XFG}p), \mathtt{A}(\mathtt{XG}p, \mathtt{XFG}p)$$
$$\mathtt{A}(\mathtt{G}p, \mathtt{XFG}p)$$
$$\mathtt{A}(\mathtt{FG}p), \mathtt{A}(\mathtt{G}p, \mathtt{FG}p)$$
$(\mathtt{X}_0)$
$$\mathtt{A}(\mathtt{XFG}p), \mathtt{A}(\mathtt{XG}p, \mathtt{XFG}p), \neg p$$
$$\mathtt{A}(p, \mathtt{XFG}p), \mathtt{A}(\mathtt{XG}p, \mathtt{XFG}p), \neg p$$
$$\mathtt{A}(\mathtt{G}p, \mathtt{XFG}p), \mathtt{E}(\neg p)$$
$$\mathtt{A}(\mathtt{G}p, \mathtt{FG}p), \mathtt{E}(\mathtt{F}\neg p)$$

$$\mathtt{A}(\mathtt{G}p, \mathtt{XFG}p), \mathtt{E}(\mathtt{EF}\neg p, \mathtt{XGEF}\neg p)$$
$$\mathtt{A}(\mathtt{G}p, \mathtt{FG}p), \mathtt{E}(\mathtt{GEF}\neg p)$$

$(\mathtt{X}_1)$
$$p, \mathtt{A}(\mathtt{XG}p, \mathtt{XFG}p), \mathtt{E}(\mathtt{XF}\neg p), \mathtt{E}(\mathtt{XGEF}\neg p)$$
$$\mathtt{A}(p, \mathtt{XFG}p), \mathtt{A}(\mathtt{XG}p, \mathtt{XFG}p), \mathtt{E}(\mathtt{F}\neg p), \mathtt{E}(\mathtt{XGEF}\neg p)$$
$$\mathtt{A}(\mathtt{G}p, \mathtt{XFG}p), \mathtt{E}(\mathtt{EF}\neg p, \mathtt{XGEF}\neg p)$$
$$\mathtt{A}(\mathtt{FG}p), \mathtt{E}(\mathtt{GEF}\neg p)$$
$$\mathtt{E}(\mathtt{AFG}p, \mathtt{EGEF}\neg p)$$
$$\mathtt{E}(\mathtt{AFG}p \wedge \mathtt{EGEF}\neg p)$$



$\not\models \mathtt{AFG}p \wedge \mathtt{EGEF}\neg p$

Pre-Tableau for $\mathtt{AFG}p \wedge \mathtt{EGEF}\neg p$

$$\mathtt{A}(\mathtt{FG}p), \mathtt{A}(\mathtt{G}p, \mathtt{FG}p)$$

$$(\mathtt{X}_0)\ \overline{\mathtt{A}(\mathtt{XFG}p), \mathtt{A}(\mathtt{XG}p, \mathtt{XFG}p)}$$

$$\overline{\mathtt{A}(p, \mathtt{XFG}p), \mathtt{A}(\mathtt{XG}p, \mathtt{XFG}p)}$$

$$\overline{\mathtt{A}(\mathtt{G}p, \mathtt{XFG}p)}$$

$$\overline{\overline{\mathtt{A}(\mathtt{FG}p), \mathtt{A}(\mathtt{G}p, \mathtt{FG}p)}}$$

$$(\mathtt{X}_0)\ \overline{\mathtt{A}(\mathtt{XFG}p), \mathtt{A}(\mathtt{XG}p, \mathtt{XFG}p), \neg p}$$

$$\overline{\mathtt{A}(p, \mathtt{XFG}p), \mathtt{A}(\mathtt{XG}p, \mathtt{XFG}p), \neg p}$$

$$\overline{\overline{\mathtt{A}(\mathtt{G}p, \mathtt{XFG}p), \mathtt{E}(\neg p)}} \qquad \mathtt{A}(\mathtt{G}p, \mathtt{XFG}p), \mathtt{E}(\mathtt{EF}\neg p, \mathtt{XGEF}\neg p)$$

$$\overline{\overline{\mathtt{A}(\mathtt{G}p, \mathtt{FG}p), \mathtt{E}(\mathtt{F}\neg p)}} \qquad \overline{\overline{\mathtt{A}(\mathtt{G}p, \mathtt{FG}p), \mathtt{E}(\mathtt{GEF}\neg p)}}$$

$$(\mathtt{X}_1)\ \overline{p, \mathtt{A}(\mathtt{XG}p, \mathtt{XFG}p), \mathtt{E}(\mathtt{XF}\neg p), \mathtt{E}(\mathtt{XGEF}\neg p)}$$

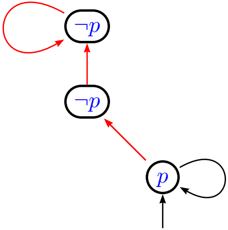
$$\overline{\overline{\mathtt{A}(p, \mathtt{XFG}p), \mathtt{A}(\mathtt{XG}p, \mathtt{XFG}p), \mathtt{E}(\mathtt{F}\neg p), \mathtt{E}(\mathtt{XGEF}\neg p)}}$$

$$\overline{\overline{\mathtt{A}(\mathtt{G}p, \mathtt{XFG}p), \mathtt{E}(\mathtt{EF}\neg p, \mathtt{XGEF}\neg p)}}$$

$$\overline{\overline{\mathtt{A}(\mathtt{FG}p), \mathtt{E}(\mathtt{GEF}\neg p)}}$$

$$\overline{\overline{\mathtt{E}(\mathtt{AFG}p, \mathtt{EGEF}\neg p)}}$$

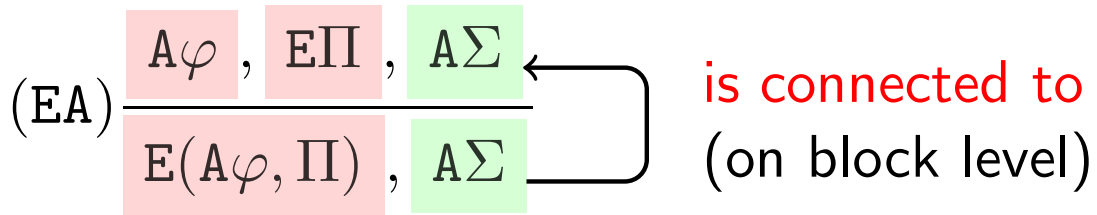
$$\overline{\mathtt{E}(\mathtt{AFG}p \wedge \mathtt{EGEF}\neg p)}$$



$\not\models \mathtt{AFG}p \wedge \mathtt{EGEF}\neg p$

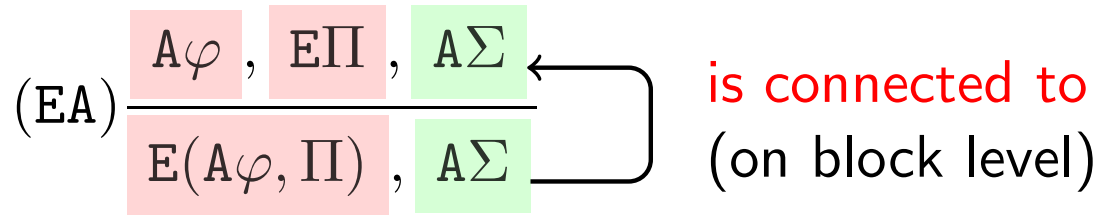
Connection Relations

- Connection on the block level.
  Example:

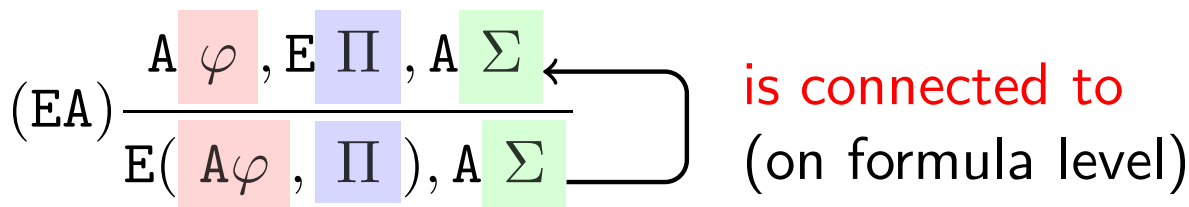


Connection Relations

- Connection on the block level.
  Example:



- Connection on the formula level.
  Example:



Traces and Threads

Traces

- A trace is an infinite sequence of connected blocks.
- A trace is an E- resp. A- trace iff the block quantifier eventually remains E resp. A.

Thread

- A thread is an infinite sequence of connected formulas.
- A thread is an F- resp. G-thread iff there is some ψ s.t. the thread finally alternates between $\mathtt{F}\psi$ or $\mathtt{XF}\psi$ (resp. G...).

Traces and Threads

Traces

- A trace is an infinite sequence of connected blocks.
- A trace is an $\mathtt{E}$- resp. $\mathtt{A}$- trace iff the block quantifier eventually remains $\mathtt{E}$ resp. $\mathtt{A}$.

Thread

- A thread is an infinite sequence of connected formulas.
- A thread is an $\mathtt{F}$- resp. $\mathtt{G}$-thread iff there is some ψ s.t. the thread finally alternates between $\mathtt{F}\psi$ or $\mathtt{XF}\psi$ (resp. $\mathtt{G}\ldots$).

Lemma

- Any trace is either an $\mathtt{E}$- or an $\mathtt{A}$-trace.
- Any thread is either an $\mathtt{F}$- or a $\mathtt{G}$-thread.

Tableau

Pre-tableaux are insufficient – an *informal* dicussion

- In the intended model
  - every formula on a F-thread is false, and
  - every formula on a G-thread is true.
- Blocks in an E-trace is understood as a conjunction.
  - Avoid F-threads.
- Blocks in an A-trace is understood as a disjunction.
  - Assure a G-thread.

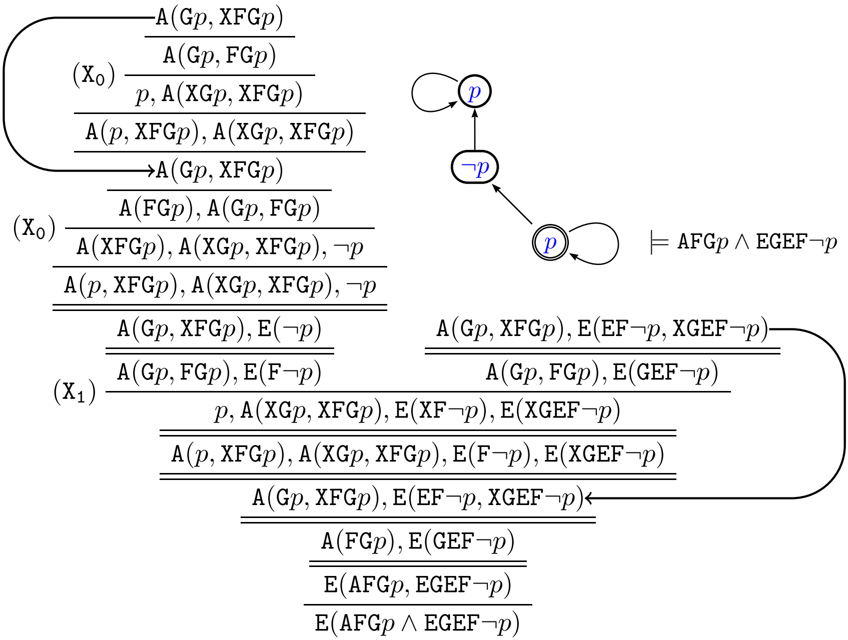
Definiton

A tableau for ϑ is a pre-tableau for ϑ iff on every branch we have

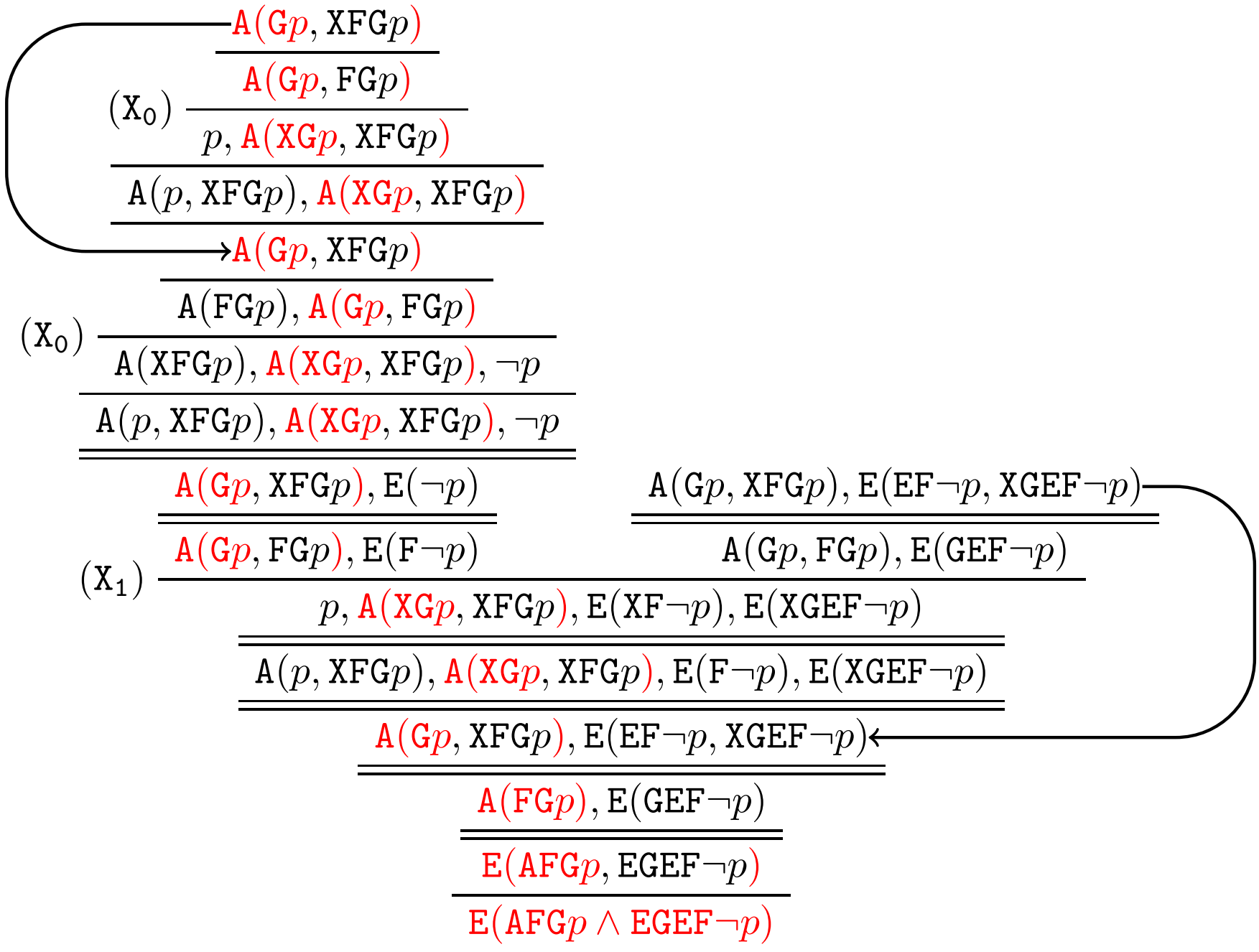
- every E-trace does not contain an F-thread, and
- every A-trace contains a G-thread.

Such traces and branches are called good.

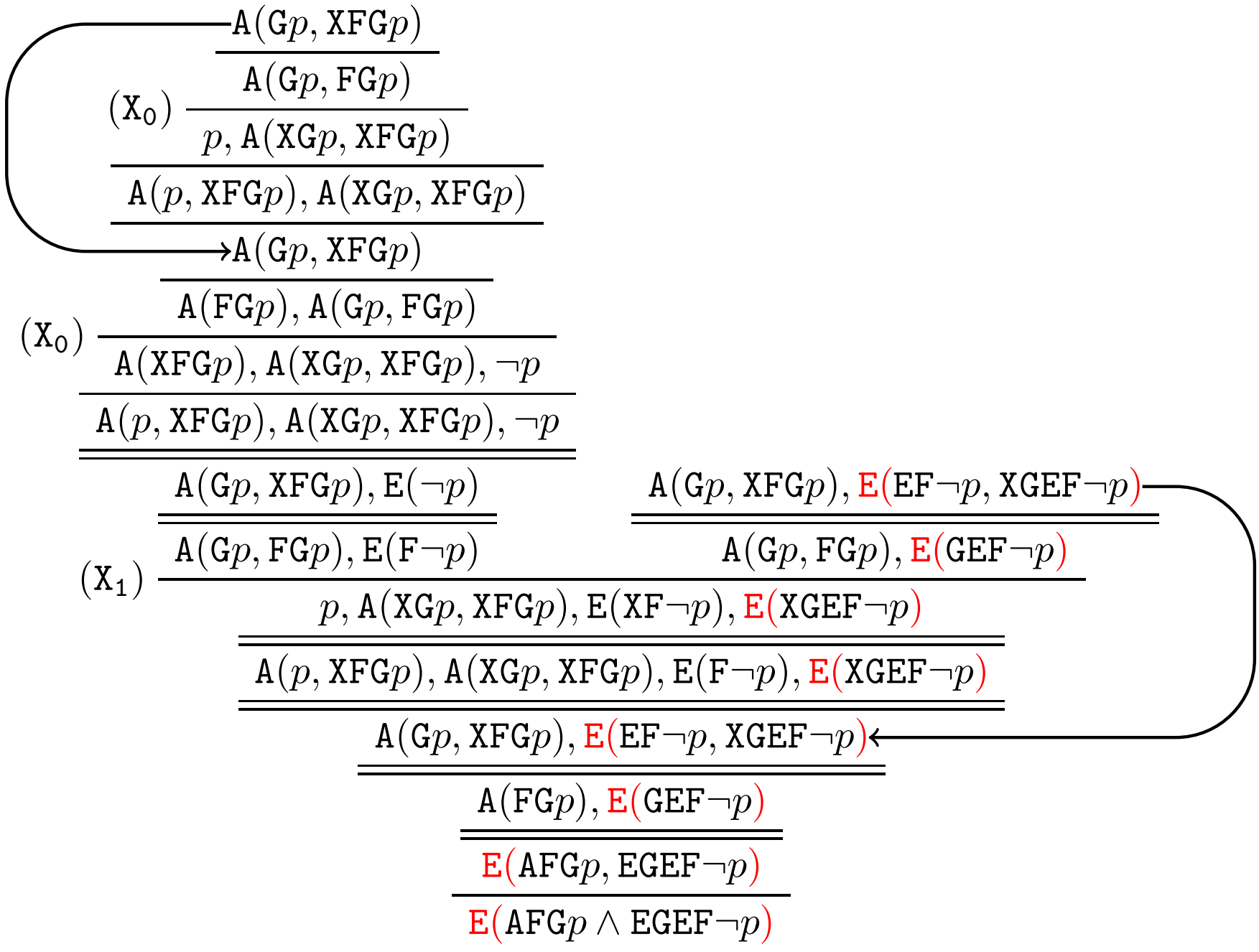
Successful Tableau for $\mathtt{AFG}p \wedge \mathtt{EGEF}\neg p$



Successful Tableau for $\mathtt{AFG}p \wedge \mathtt{EGEF}\neg p$



Successful Tableau for $\mathtt{AFG}p \wedge \mathtt{EGEF}\neg p$



Soundness

Theorem

If there is a tableau for CTL^*-formula ϑ then ϑ is satisfiable.

Proof sketch.

Collapse given tableau to an interpretation and show that this interpretation models ϑ. $\square$

Completeness

Theorem

If a CTL^*-formula ϑ is satisfiable then it has a tableau.

Proof sketch.

- Construct a pre-tableaux for ϑ by unrolling a model of ϑ s.th. the unraveled model satisfied the current goal.
- For rules with alternatives premises: prefer premises decomposing the principal formula obeying the model.
- Hence, $\mathtt{F}\varphi$ is rewritten to φ as soon as possible, for instance. □

Completeness

Theorem

If a CTL^*-formula ϑ is satisfiable then it has a tableau.

Proof sketch.

- Construct a pre-tableaux for ϑ by unrolling a model of ϑ s.th. the unraveled model satisfied the current goal.
- For rules with alternatives premises: prefer premises decomposing the principal formula obeying the model.
- Hence, $\mathtt{F}\varphi$ is rewritten to φ as soon as possible, for instance. □

Remark

- Completeness proof does not use the small model property. (That is, replace fixed points by approximants.)
- We have not use any automata or game theory so far.

Decision Procedure

there is a tableau for ϑ

Given a CTL^*-formula ϑ, decide whether ~~ϑ is satisfiable~~.

Decision Procedure

Given a CTL^*-formula ϑ, decide whether ~~ϑ is satisfiable~~ there is a tableau for ϑ.

Idea: treat a tableau as a parity game.

Tableau $\mapsto$ Game

Observation

The property separating tableaux from pre-tableaux is ω-regular.

Tableau $\mapsto$ Game

Observation

The property separating tableaux from pre-tableaux is ω-regular.

Game

Given an appropriate deterministic ω-automaton $\mathcal{A}$.

- States: pairs of goals for ϑ and $\mathcal{A}$'s of state.
- Proponent chooses the rule application.
- Opponent chooses the premise whenever the rule application is branching.
- Additionally, edges respect the transition relation of $\mathcal{A}$.
- Turn $\mathcal{A}$'s acceptance condition into that of the game.

Tableau $\mapsto$ Game

Observation

The property separating tableaux from pre-tableaux is ω-regular.

Game

Given an appropriate deterministic ω-automaton $\mathcal{A}$.

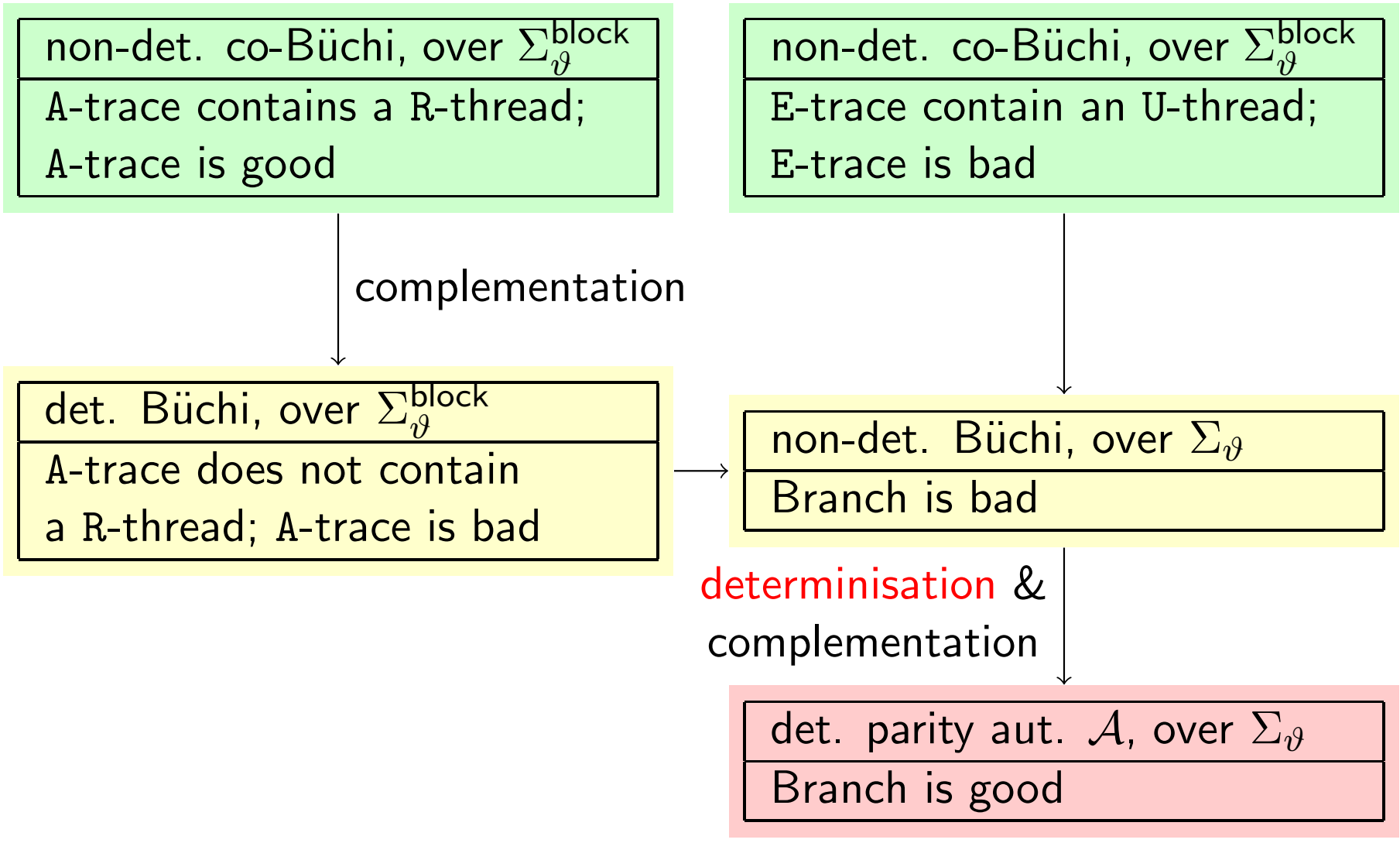
- States: pairs of goals for ϑ and $\mathcal{A}$'s of state.
- Proponent chooses the rule application.
- Opponent chooses the premise
  whenever the rule application is branching.
- Additionally, edges respect the transition relation of $\mathcal{A}$.
- Turn $\mathcal{A}$'s acceptance condition into that of the game.

Property

Proponent has a winning strategy for $\mathtt{E}\{\vartheta\}$
iff
ϑ has a tableau.

Tableau $\mapsto$ Game — Automata

Alphabet $\Sigma_\vartheta^{\text{block}} := \Sigma_\vartheta \times 2^{Sub(\vartheta)}$.
Second component marks an A- or E-block in the first component.



Conclusion

Summary

- Tableau is sound and complete for CTL^*.
- Rules are "natural".
- Correctness proof relies on neither automata theory nor game theory nor the small model property.
- The decision procedure uses game theory as a back-end and can benefit from it.
- Complexity of the decision procedure optimal.

Implementation

- `http://www.tcs.ifi.lmu.de/mlsolver/`.

Future Work

- Implement Emerson's procedure (as a comparison).
- Find a way to avoid or to reduce determinisation.
- Find a proof system for CTL^* with natural axioms and rules —as opposed to existing ones.