One-Pass and Tree-Shaped Tableaux for LTL

Nicola Gigante

Course in Automatic Systems Verification
University of Udine, Italy
15 May 2019

Linear Temporal Logic

Linear Temporal Logic (LTL) is a propositional modal logic interpreted over infinite, discrete, linear orders.

$\mathsf{X}\,\alpha$	α will be true at the next state.
$\alpha\,\mathcal{U}\,\beta$	β will eventually be true, and α always holds until then.
$\mathsf{F}\,\beta \equiv \top\,\mathcal{U}\,\beta$	β will eventually be true.
$\mathsf{G}\,\beta \equiv \neg\,\mathsf{F}\neg\beta$	β will always be true.

Tableau methods for LTL satisfiability

LTL satisfiability is the problem of checking whether there exists a model that satisfies a given LTL formula.

- PSPACE-complete problem.
- Algorithmic solutions:
  - (Büchi) Automata-based
  - Tableau methods
  - Temporal resolution
  - Reduction to model checking
  - ...

Tableau methods for LTL satisfiability

You have already seen a tableau method for LTL satisfiability, which is graph-shaped:

- A graph structure is built from the closure of the formula
- Each node is an atom, a set of locally consistent formulae
- A path in the graph corresponds to a potential model of the formula
- A strongly connected components decomposition of the graph is used to check whether there exists a fulfilling path.

A One-Pass Tree-Shaped Tableau for LTL

Today we'll look at a different method, which is one-pass and tree-shaped.

- The built structure is a tree instead of a graph.
- A single pass is sufficient to build a branch of the tree and determine its acceptance or rejection.
- Very simple rule-based structure:
  - Easy to extend to LTL extensions
  - Easy to parallelize

How it works

How it works

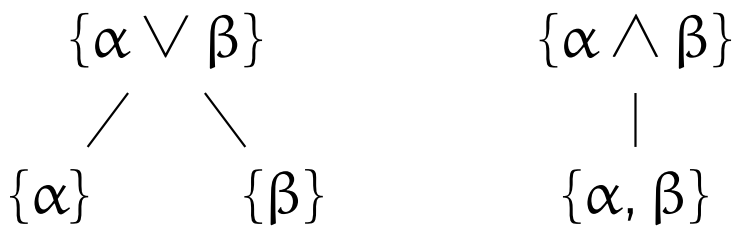
The tableau for ϕ is a tree where each node is labeled by a set of formulae, with the root labeled with $\{\phi\}$.

- The formula starts in Negated Normal Form.
- At each step some rules are applied to a leaf, depending on the contents of the label, possibly generating new children for the current node.
- Some rules can accept a branch, others can reject it.
- If the complete tree contains at least an accepted branch, the formula is satisfiable.

Expansion rules

Expansion rules are applied to a node until no other expansion rule can be applied anymore:

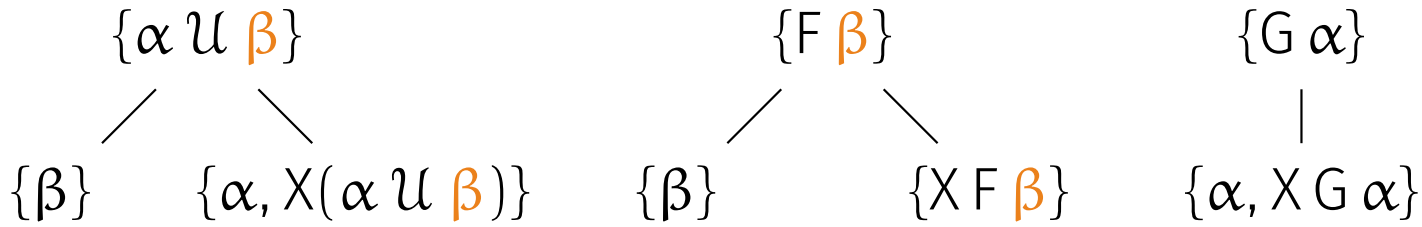
- Boolean connectives handled just like in classical propositional tableau.



Expansion rules

Expansion rules are applied to a node until no other expansion rule can be applied anymore:

- Common expansion rules handle temporal operators:



- β is called an eventuality.
- $X(\alpha \mathcal{U} \beta)$ is an X-eventuality.

Expansion rules

Rule	ϕ	$\Gamma_1(\phi)$	$\Gamma_2(\phi)$
Disjunction	$\alpha \vee \beta$	$\{\alpha\}$	$\{\beta\}$
Until	$\alpha \mathcal{U} \beta$	$\{\beta\}$	$\{\alpha, \mathsf{X}(\alpha \mathcal{U} \beta)\}$
Release	$\alpha \mathcal{R} \beta$	$\{\alpha, \beta\}$	$\{\beta, \mathsf{X}(\alpha \mathcal{R} \beta)\}$
Eventually	$\mathsf{F}\,\beta$	$\{\beta\}$	$\{\mathsf{X}\,\mathsf{F}\,\beta\}$
Conjunction	$\alpha \wedge \beta$	$\{\alpha, \beta\}$	
Always	$\mathsf{G}\,\alpha$	$\{\alpha, \mathsf{X}\,\mathsf{G}\,\alpha\}$	

Advancing to the next temporal step

Eventually we reach a poised node, labelled only of elementary formulae: propositions or tomorrows.

We can step to the next temporal state by the STEP rule:

$$\{\ldots, \mathrm{X}\,\alpha, \ldots\}$$
$$\downarrow$$
$$\{\alpha\}$$

The poised node give us the evaluation for the current state.

Before stepping, however, we must check whether the current branch has to be rejected or accepted.

Contradictions

If a label contains contradictions, we reject the branch.

$$\{\ldots, p, \ldots, \neg p, \ldots\}$$

$$\times$$

Acceptance and contradictions

If a STEP rule results into an empty label, we're done: the branch is accepted.

$$\{\ldots, p, \neg q, r, \ldots\}$$
$$\downarrow$$
$$\{\}$$
$$\checkmark$$

Finding periodic models - LOOP rule

Some formulae (*e.g.*, $\mathsf{G}\,\mathsf{F}\,p$) require to satisfy infinitely often the same request, thus the labels may never become empty.

- these formulae will have infinite periodic models
- the LOOP rule accepts such branches

Finding periodic models - LOOP rule

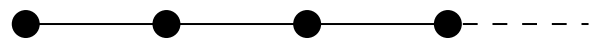
Let $\overline{u} = \langle u_0, \ldots, u_k \rangle$ be a branch with a poised leaf u_k:

LOOP rule

The branch is accepted if there is another position $i < k$ such that $\Gamma(u_j) = \Gamma(u_k)$, and all the X-eventualities requested in u_j are fulfilled in $\overline{u}_{[j+1,k]}$.

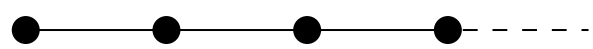
Example

$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$

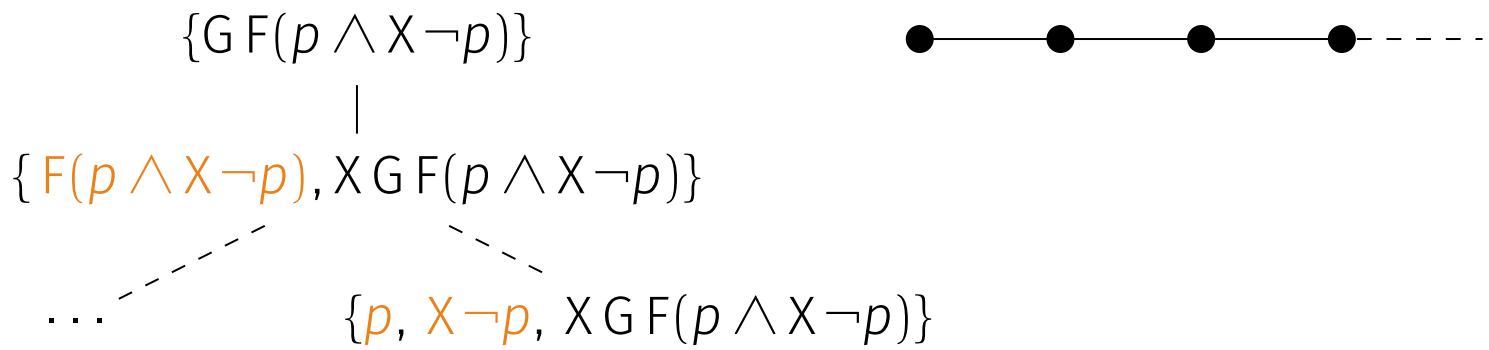


Example

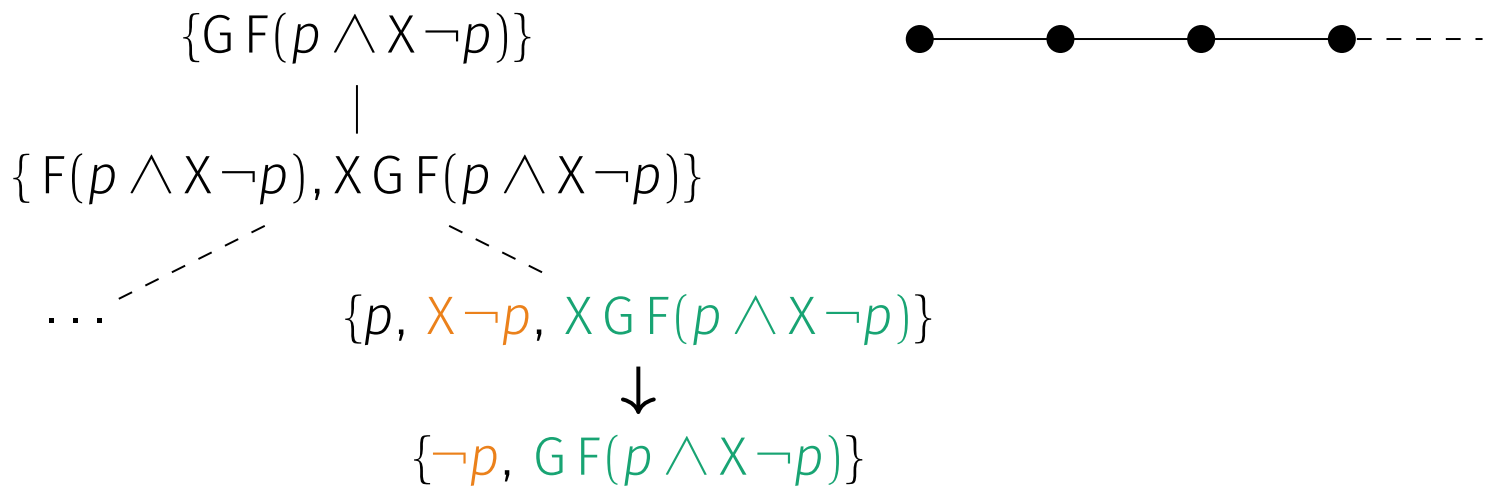
$$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$
$$|$$
$$\{\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$$



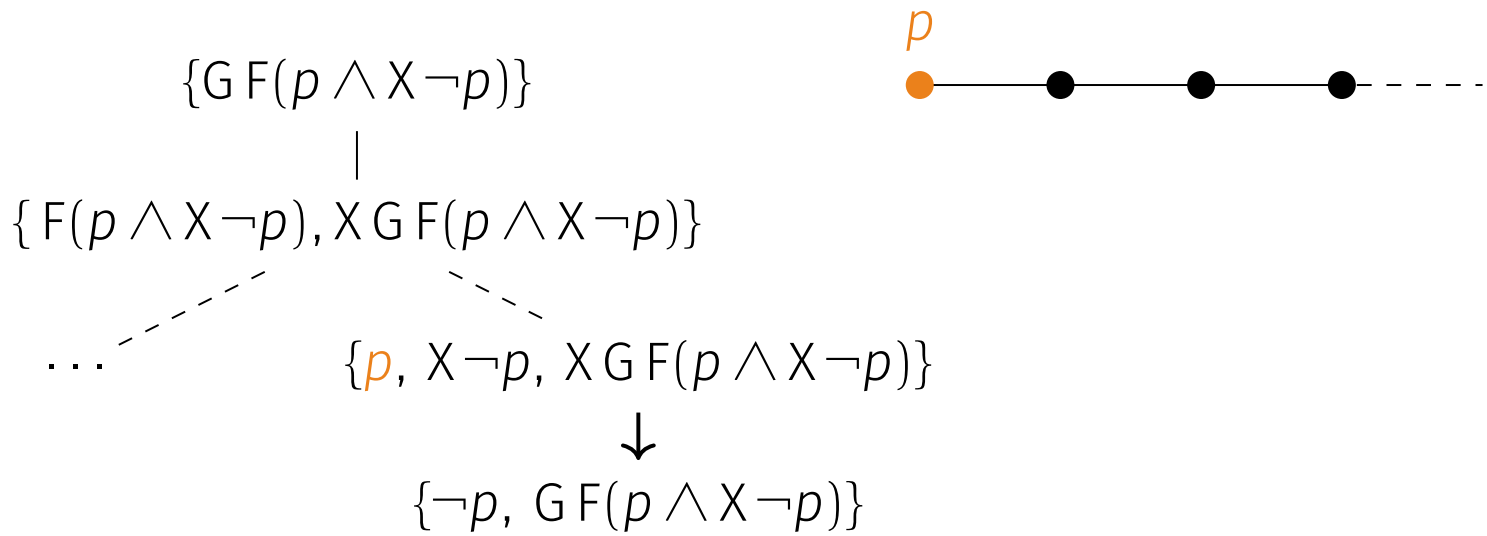
Example



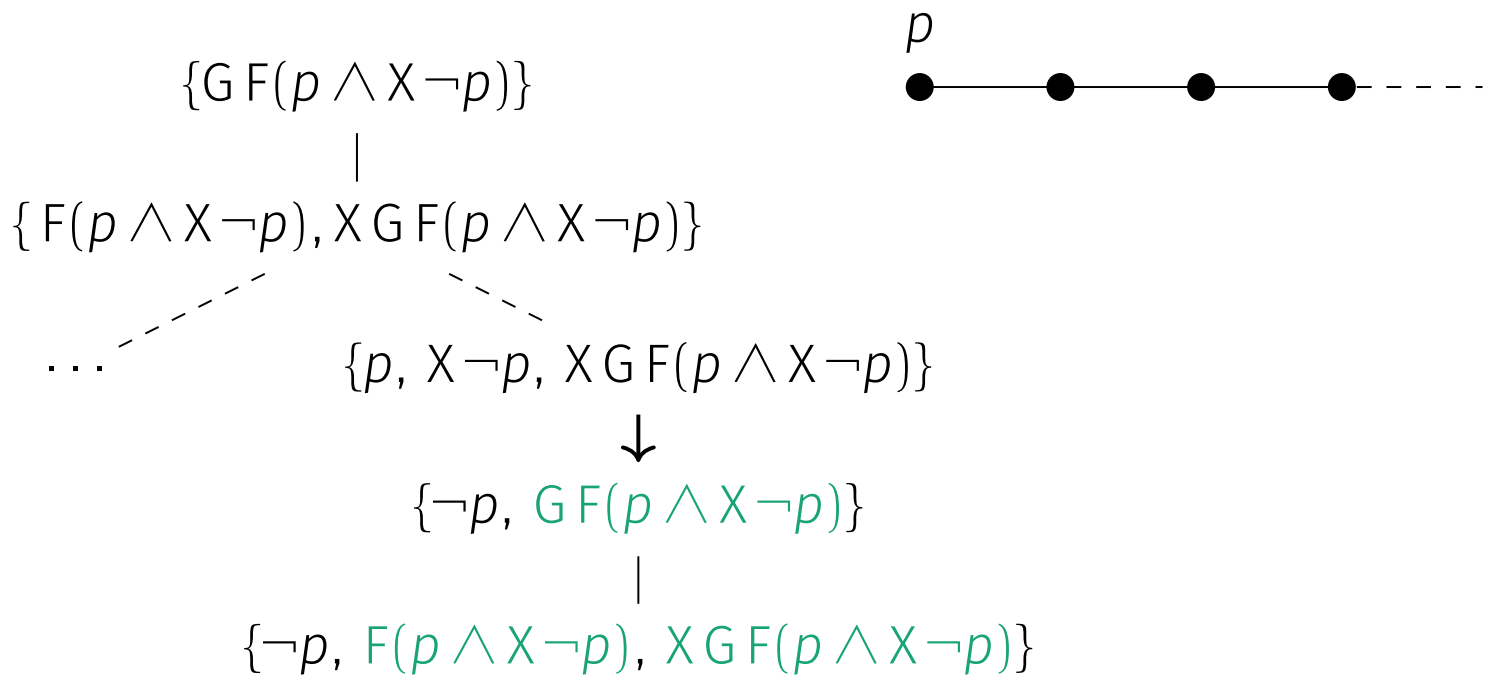
Example



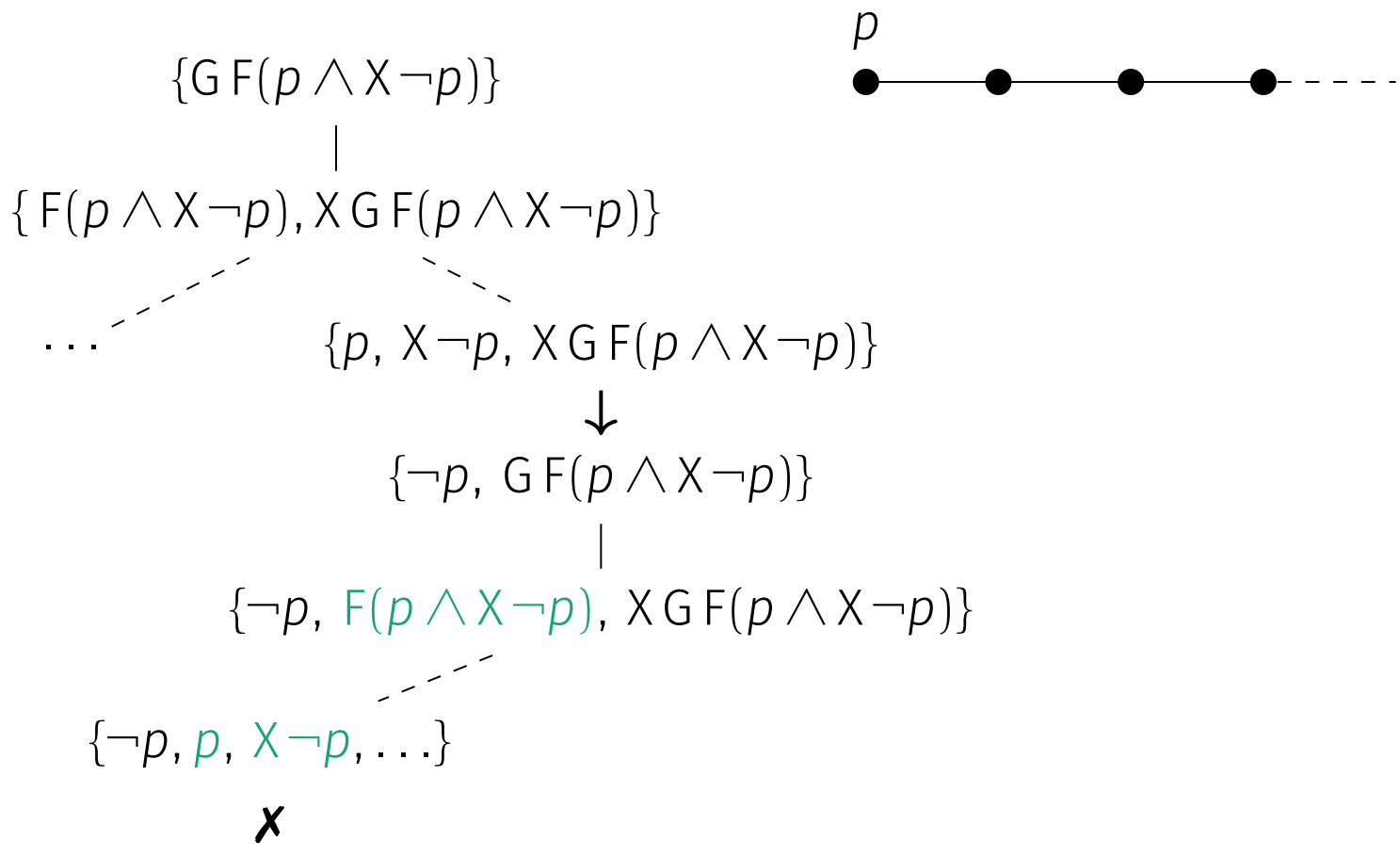
Example



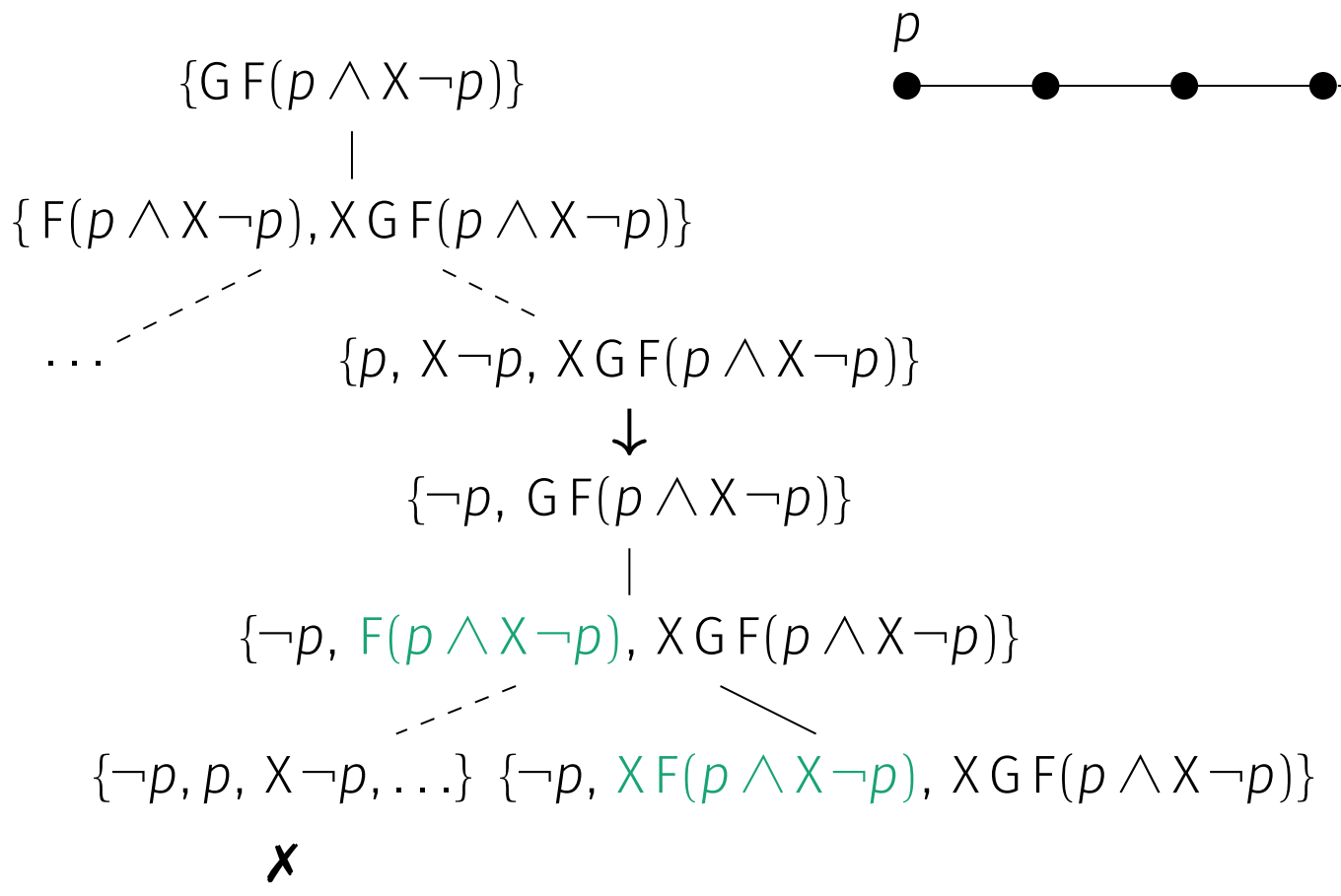
Example


$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\ldots$
$\{p, \mathsf{X}\neg p, \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, \mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
p

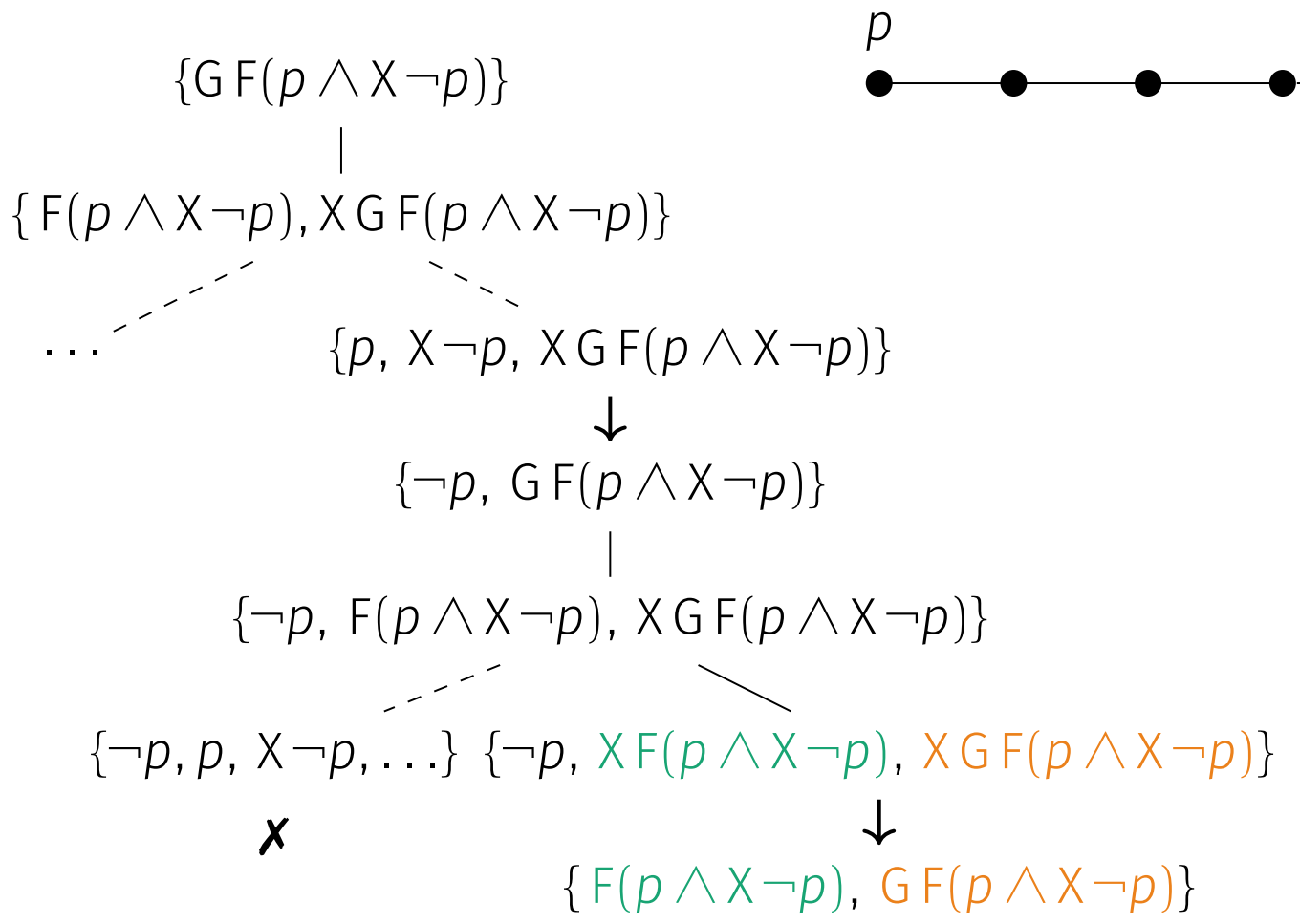
Example


$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\ldots$
$\{p, \mathsf{X}\neg p, \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, \mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, p, \mathsf{X}\neg p, \ldots\}$
✗
p

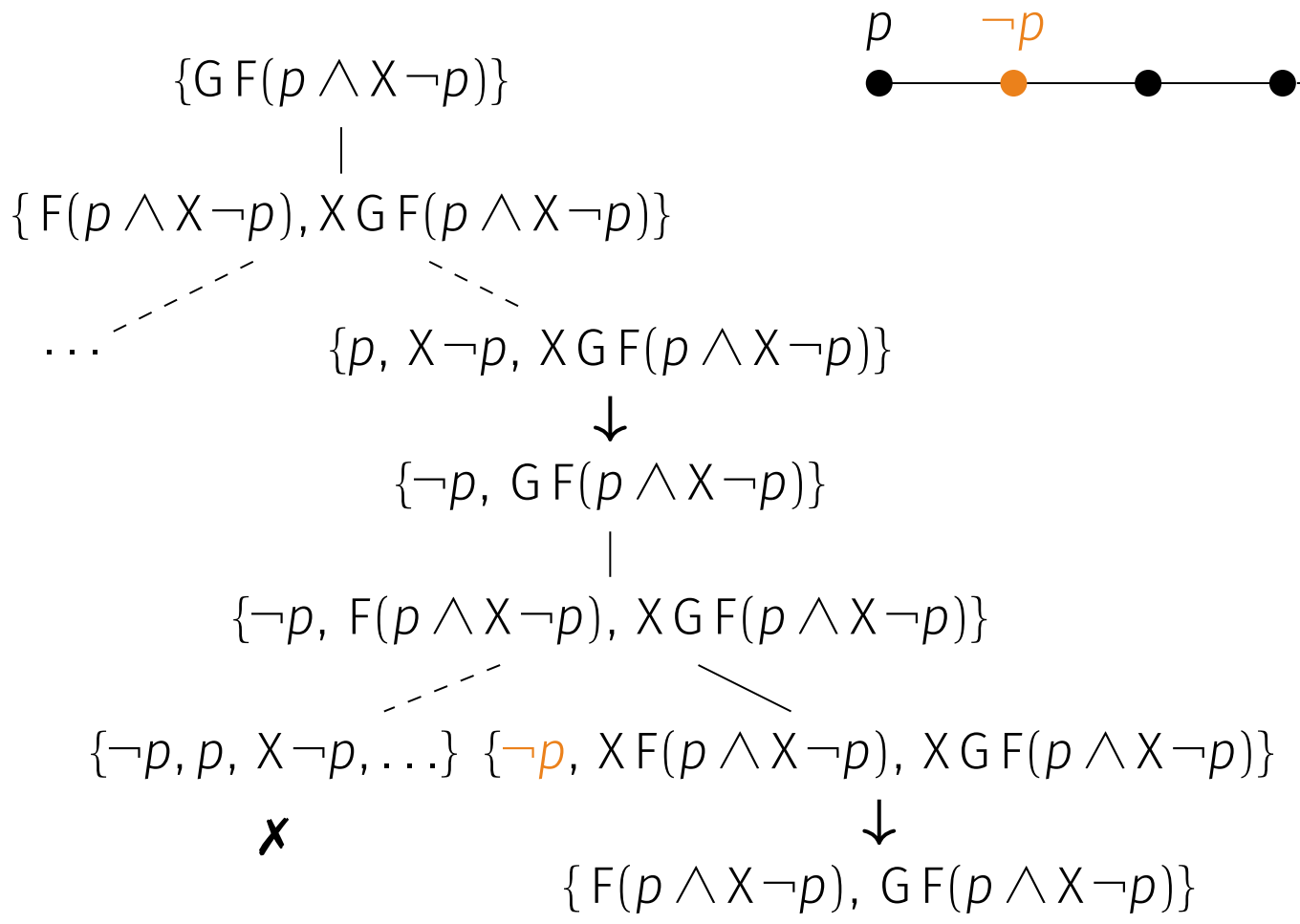
Example


p
{G F(p ∧ X¬p)}
{ F(p ∧ X¬p), X G F(p ∧ X¬p)}
...
{p, X¬p, X G F(p ∧ X¬p)}
{¬p, G F(p ∧ X¬p)}
{¬p, F(p ∧ X¬p), X G F(p ∧ X¬p)}
{¬p, p, X¬p, ...}
{¬p, X F(p ∧ X¬p), X G F(p ∧ X¬p)}
✗

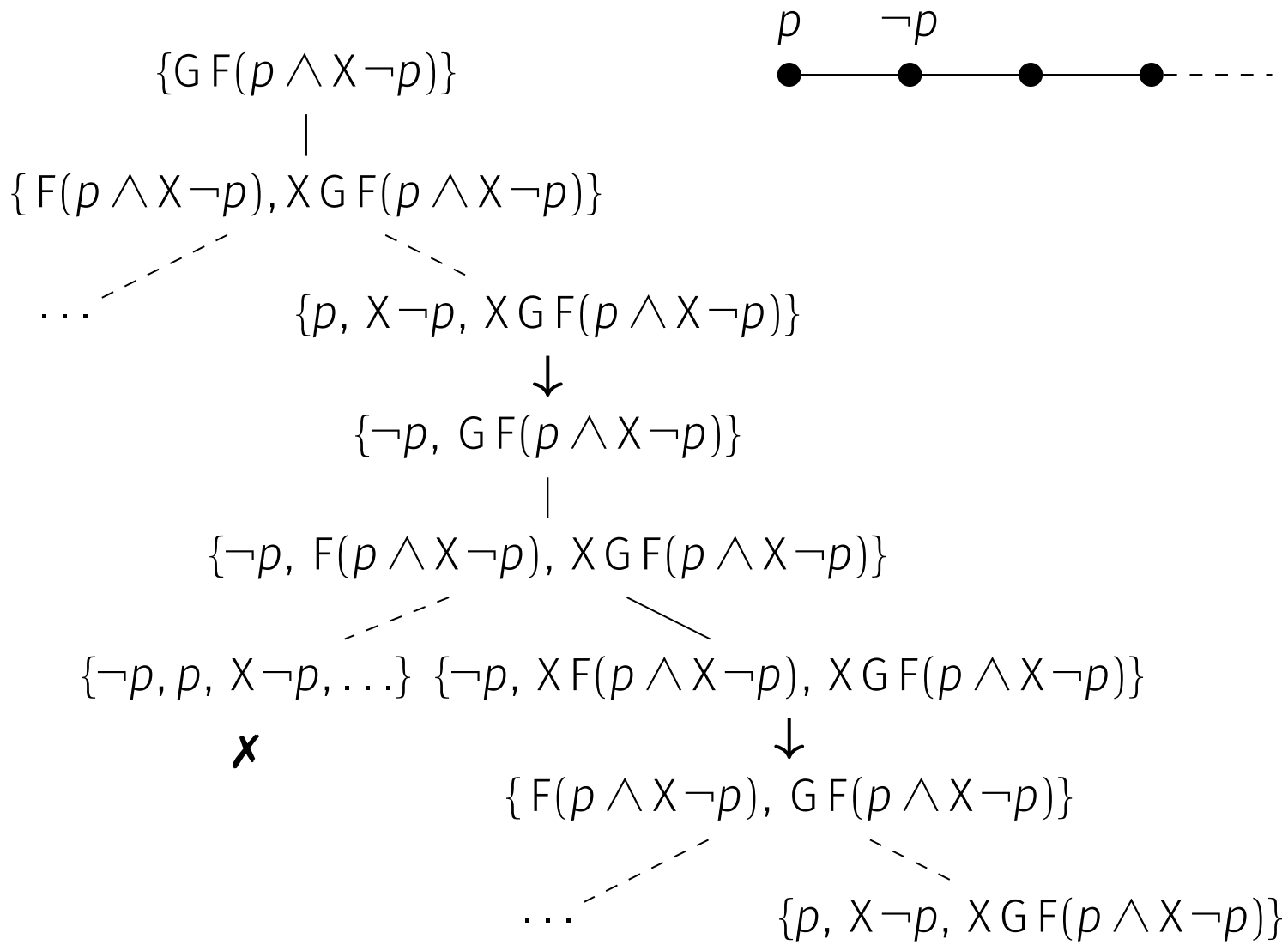
Example


$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\ldots$
$\{p, \mathsf{X}\neg p, \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, \mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, p, \mathsf{X}\neg p, \ldots\}$
$\{\neg p, \mathsf{X}\,\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
✗
$\{\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
p

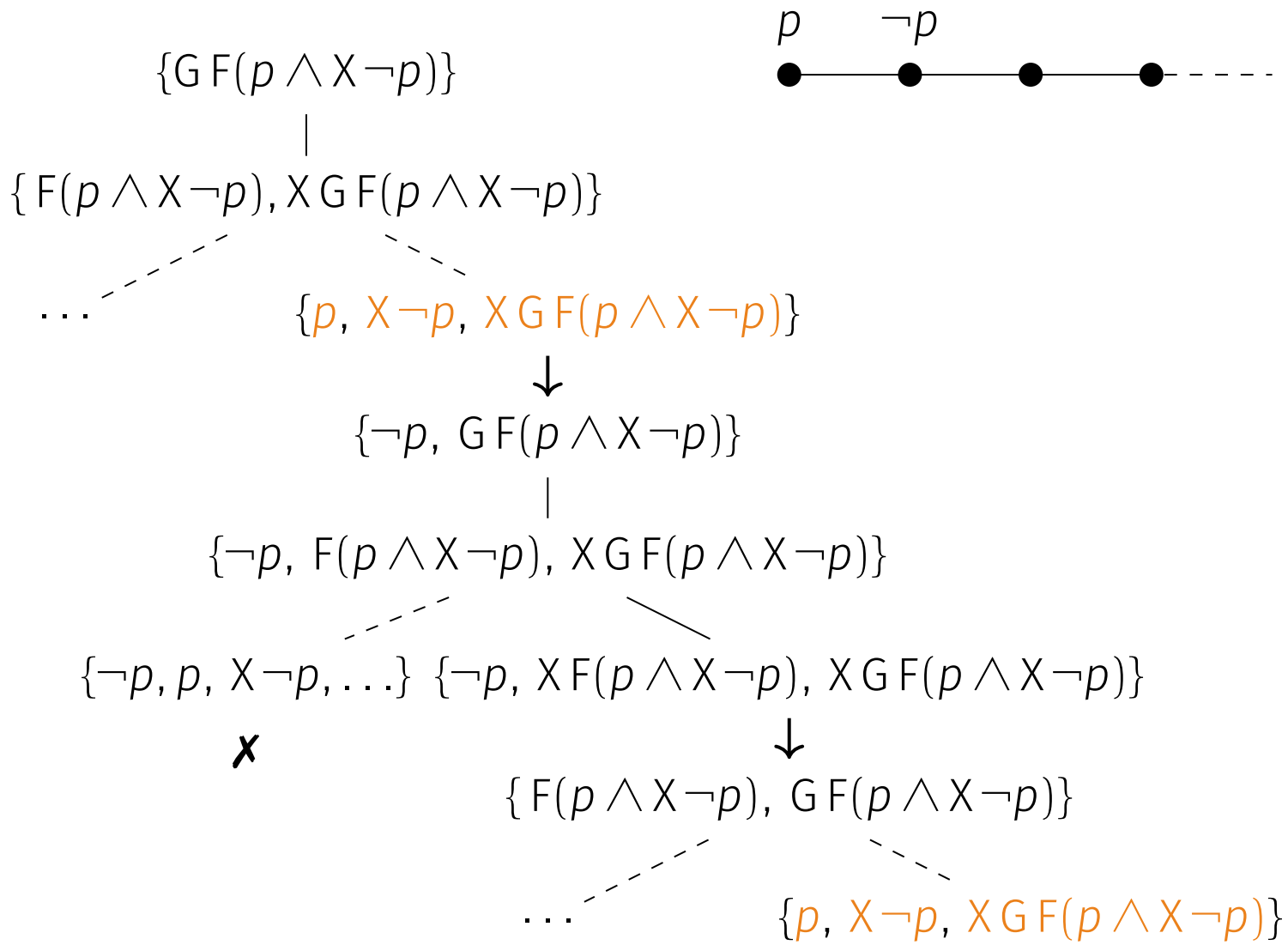
Example


$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\ldots$
$\{p, \mathsf{X}\neg p, \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, \mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, p, \mathsf{X}\neg p, \ldots\}$
$\{\neg p, \mathsf{X}\,\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
✗
$\{\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
p
$\neg p$

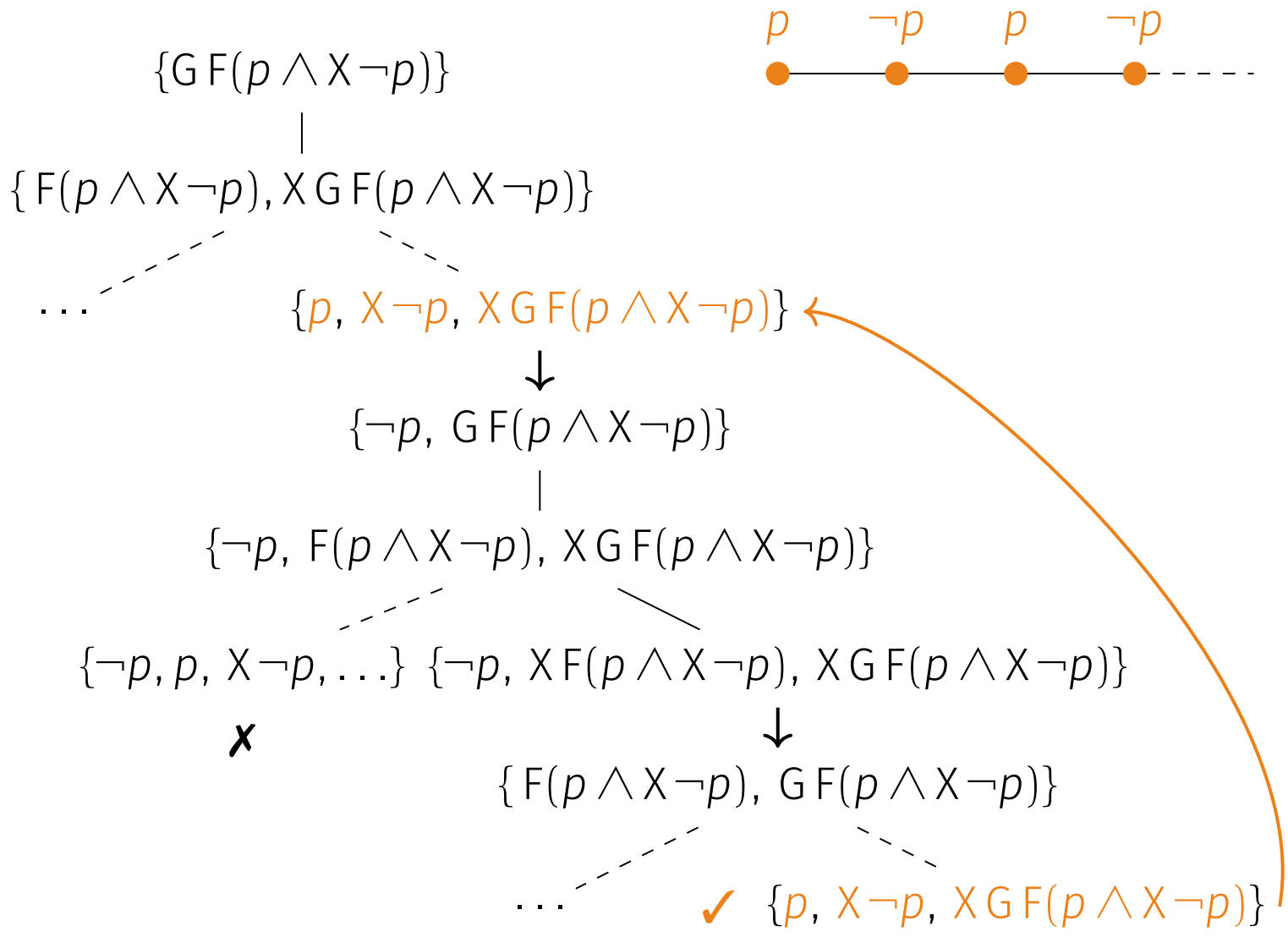
Example


$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\ldots$
$\{p, \mathsf{X}\neg p, \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, \mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, p, \mathsf{X}\neg p, \ldots\}$
$\{\neg p, \mathsf{X}\,\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
✗
$\{\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\ldots$
$\{p, \mathsf{X}\neg p, \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
p
$\neg p$

Example


p $\neg p$
$\{\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\ldots$
$\{p, \mathsf{X}\neg p, \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, \mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\{\neg p, p, \mathsf{X}\neg p, \ldots\}$
$\{\neg p, \mathsf{X}\,\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
✗
$\{\mathsf{F}(p \wedge \mathsf{X}\neg p), \mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$
$\ldots$
$\{p, \mathsf{X}\neg p, \mathsf{X}\,\mathsf{G}\,\mathsf{F}(p \wedge \mathsf{X}\neg p)\}$

Example


p ¬p p ¬p
{G F(p ∧ X¬p)}
{ F(p ∧ X¬p), X G F(p ∧ X¬p)}
...
{p, X¬p, X G F(p ∧ X¬p)}
{¬p, G F(p ∧ X¬p)}
{¬p, F(p ∧ X¬p), X G F(p ∧ X¬p)}
{¬p, p, X¬p, ...}
{¬p, X F(p ∧ X¬p), X G F(p ∧ X¬p)}
✗
{ F(p ∧ X¬p), G F(p ∧ X¬p)}
...
✓ {p, X¬p, X G F(p ∧ X¬p)}

Unrealizable eventualities

Something is still missing. Consider the following formula:

$$\mathsf{G}\neg p \wedge q \,\mathcal{U}\, p$$

- It is unsatisfiable, but not because of propositional contradictions.
- The requested eventuality is unrealizable.

Unrealizable eventualities - PRUNE rule

In these cases we have to stop postponing the eventuality to guarantee termination. Let $\overline{u} = \langle u_0, \ldots, u_k \rangle$ be a branch with a poised leaf u_k:

PRUNE rule

The branch is rejected if:

1. there are other two positions $i < j < k$ such that $\Gamma(u_i) = \Gamma(u_j) = \Gamma(u_k)$, and
2. all the X-eventualities fulfilled in $\overline{u}_{[j+1,k]}$ are fulfilled in $\overline{u}_{[i+1,j]}$ as well.

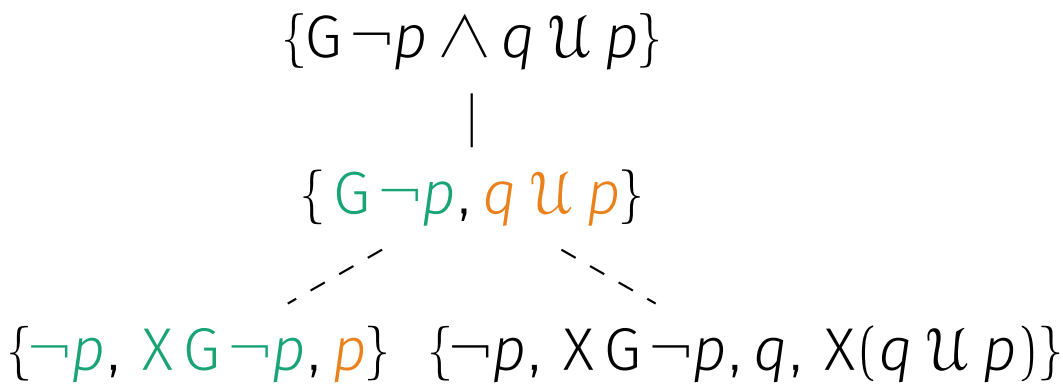
Example - unsatisfiable formula

$$\{\mathsf{G}\,\neg p \wedge q\,\mathcal{U}\,p\}$$

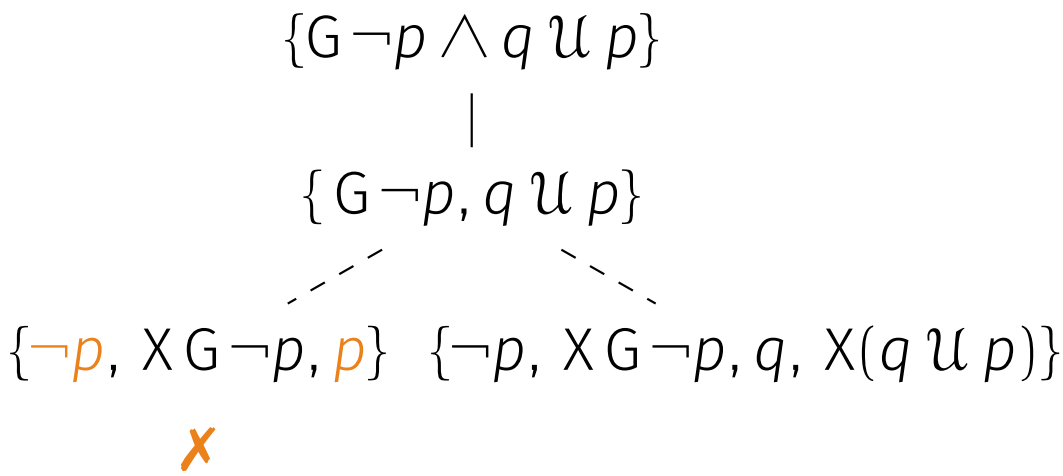
Example - unsatisfiable formula

$$\{\mathsf{G}\,\neg p \wedge q\,\mathcal{U}\,p\}$$
$$|$$
$$\{\,\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$$

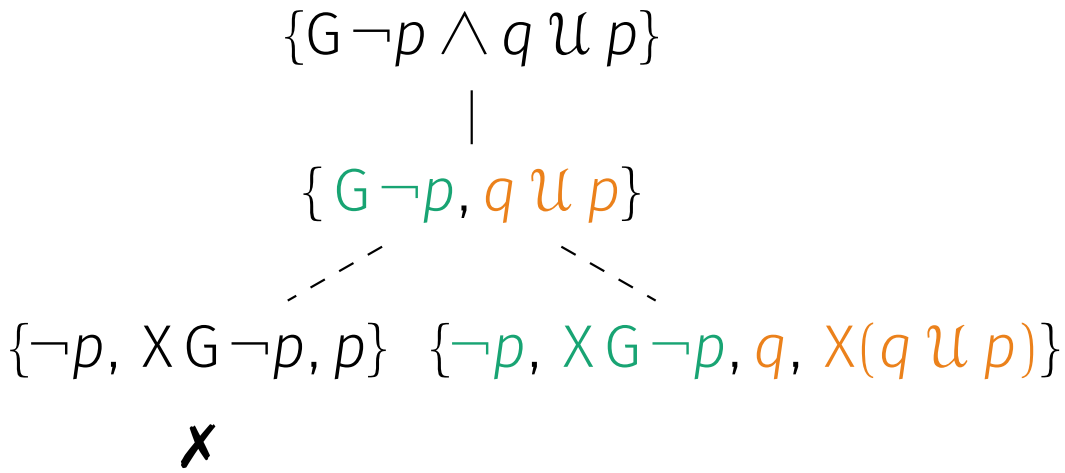
Example - unsatisfiable formula



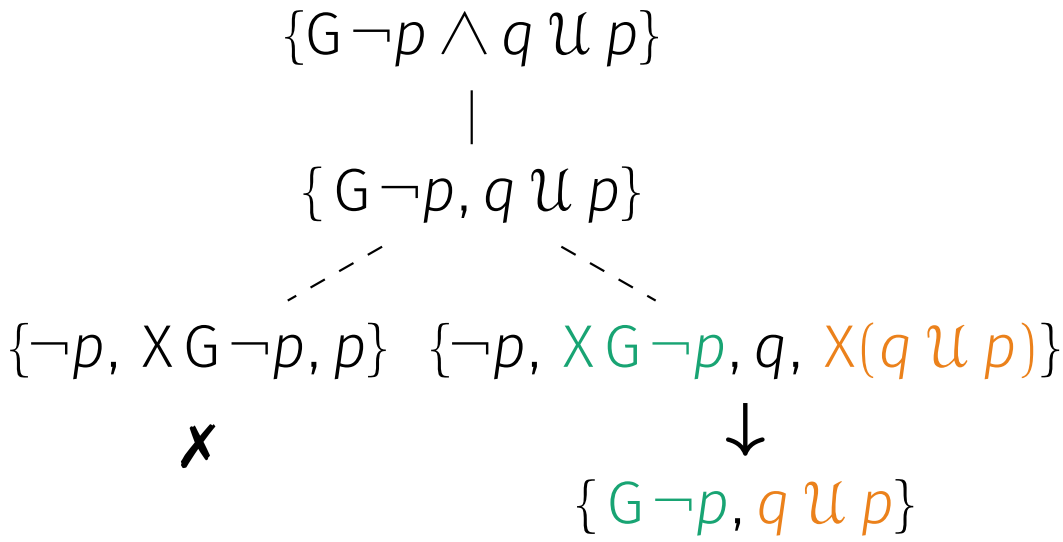
Example - unsatisfiable formula



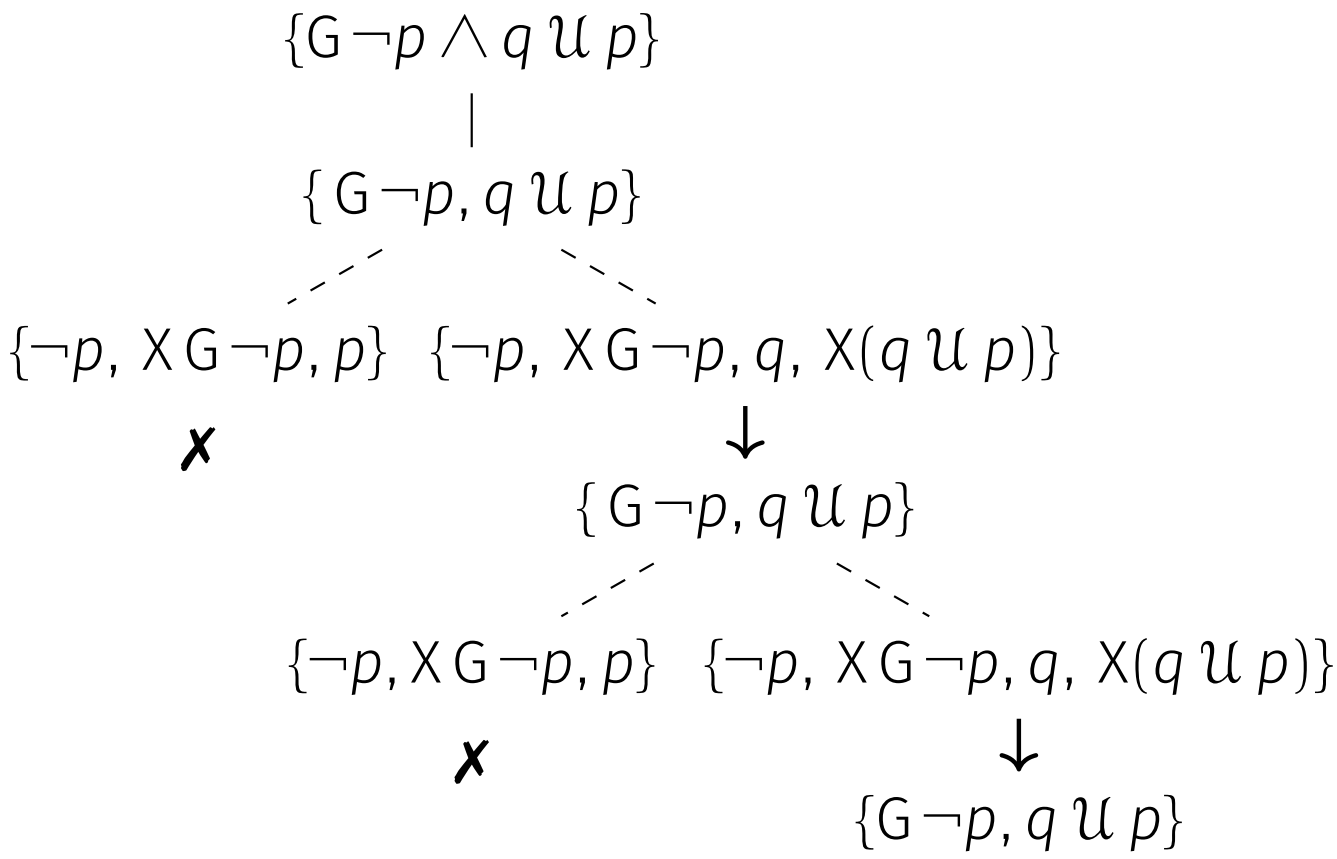
Example - unsatisfiable formula



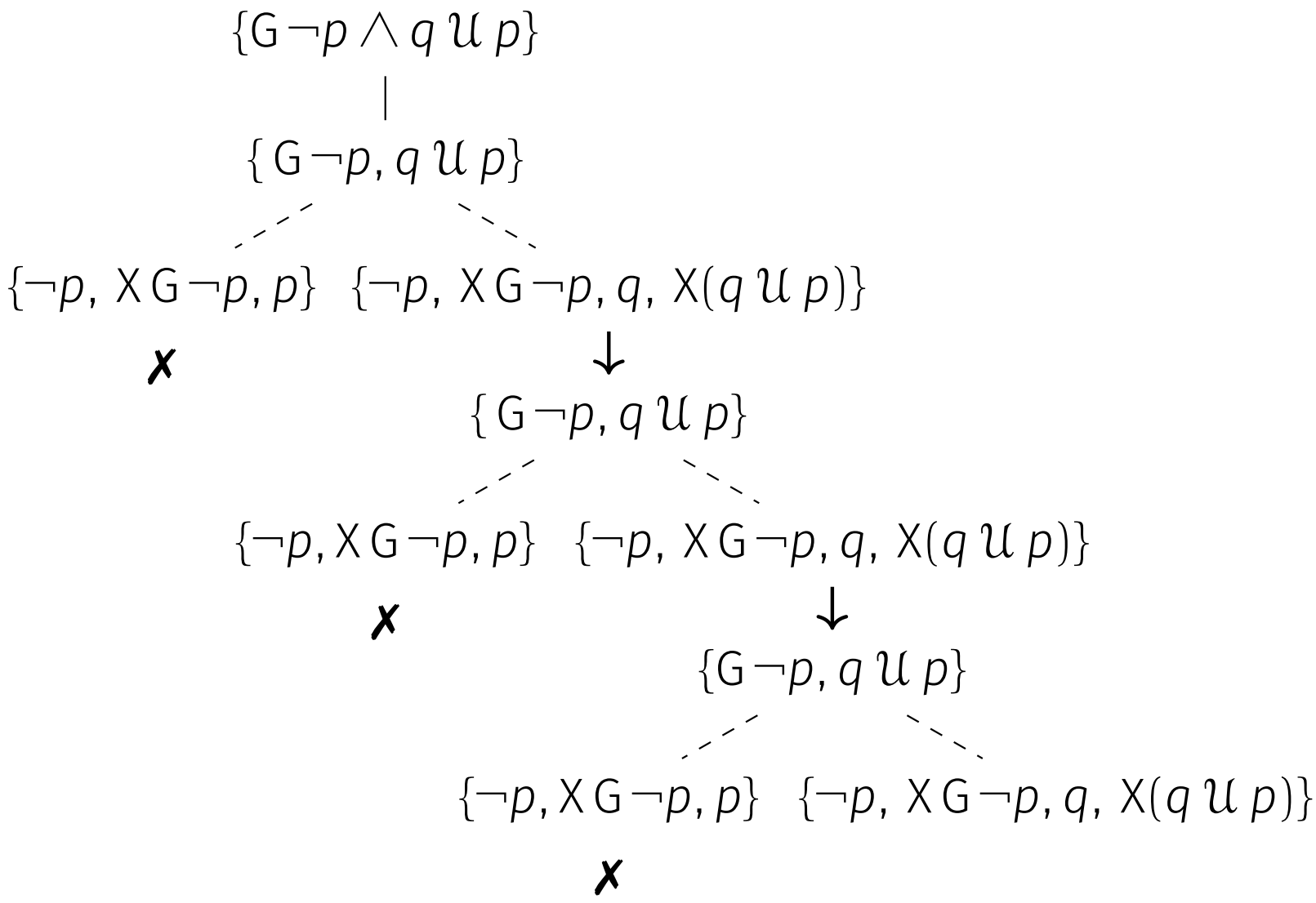
Example - unsatisfiable formula



Example - unsatisfiable formula



Example - unsatisfiable formula



Example - unsatisfiable formula

$$\{\mathsf{G}\,\neg p \wedge q\,\mathcal{U}\,p\}$$

$$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$$

$$\{\neg p,\ \mathsf{X}\,\mathsf{G}\,\neg p, p\} \qquad \{\neg p,\ \mathsf{X}\,\mathsf{G}\,\neg p, q,\ \mathsf{X}(q\,\mathcal{U}\,p)\}$$

✗

$$\downarrow$$

$$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$$

$$\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, p\} \qquad \{\neg p,\ \mathsf{X}\,\mathsf{G}\,\neg p, q,\ \mathsf{X}(q\,\mathcal{U}\,p)\}$$

✗

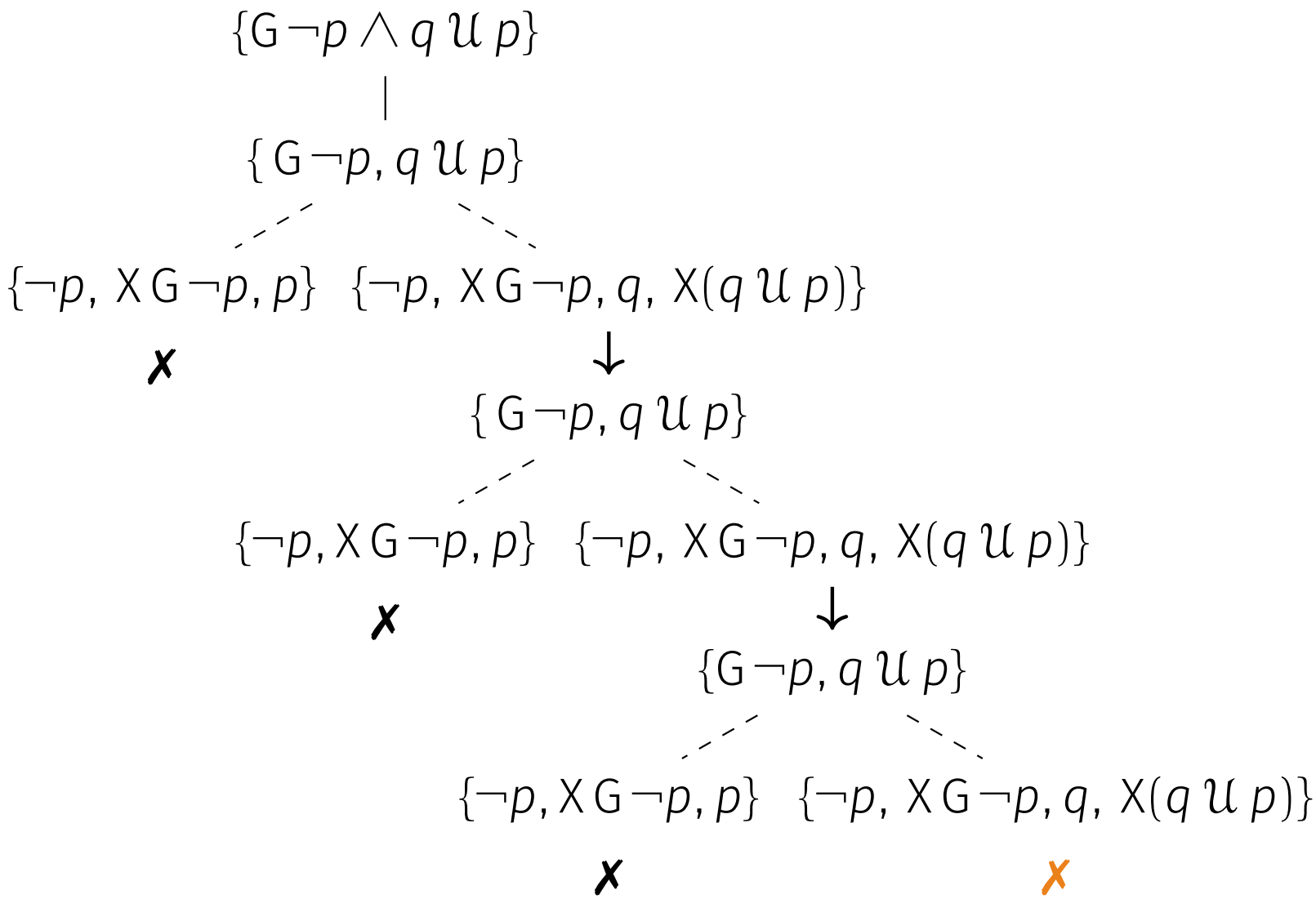
$$\downarrow$$

$$\{\mathsf{G}\,\neg p, q\,\mathcal{U}\,p\}$$

$$\{\neg p, \mathsf{X}\,\mathsf{G}\,\neg p, p\} \qquad \{\neg p,\ \mathsf{X}\,\mathsf{G}\,\neg p, q,\ \mathsf{X}(q\,\mathcal{U}\,p)\}$$

✗

Example - unsatisfiable formula



How it works - summary

To summarize:

- When to accept a branch?
  - When the label is empty
  - When we are looping while satisfying all the eventualities
- When to reject a branch?
  - When a label is contradictory
  - When we are looping but unable to satisfy all the eventualities

How it works - summary

To summarize:

- When to accept a branch?
  - When the label is empty
  - When we are looping while satisfying all the eventualities
- When to reject a branch?
  - When a label is contradictory
  - When we are looping but unable to satisfy all the eventualities

Proofs

We can briefly overview how to prove that the method is sound and complete.

Soudness

If the complete tableau for ϕ contains an accepted branch, then ϕ is satisfiable.

Completeness

If ϕ is satisfiable, then the complete tableau for ϕ contains at least an accepted branch.

Soundness

The soundness proof builds a model from the accepted branch:

- let $\overline{u} = \langle u_0, \ldots, u_n \rangle$ be an accepted branch (u_n is ticked)
- let $\overline{\pi} = \langle \pi_0, \ldots, \pi_m \rangle$ be the subsequence of its poised nodes
- if $\overline{u}$ is accepted then either $\Gamma(\pi_m)$ is empty or there is a π_k with $\Gamma(\pi_k) = \Gamma(\pi_m)$ that triggered the LOOP rule.
- we extend $\overline{\pi}$ ad infinitum to:

$$\Pi = \pi_{[0,k]}(\pi_{[k+1,m]})^{\omega}$$

(let $k = m - 1$ in the EMPTY case)

Soundness

The soundness proof builds a model from the accepted branch:

- then, we can extract a model $\overline{\sigma} = \langle \sigma_0, \sigma_1, \ldots \rangle$ from Π:
  - $\sigma, i \models p$ if and only if $p \in \Gamma(\Pi_i)$
- we can prove by structural induction that if $\psi \in \Gamma(\Pi_i)$, then $\sigma, i \models \psi$:
  - if $p \in \Gamma(\Pi_i)$, it holds by definition
  - if $\mathrm{X}\psi \in \Gamma(\Pi_i)$, either $\psi \in \Gamma(\Pi_{i+1})$ by the STEP rule, or $\Pi_i = \Pi_m = \Pi_k$ because of the LOOP rule, hence $\psi \in \Gamma(\Pi_{i+1}) = \Gamma(\Pi_{k+1})$ again because of the STEP rule.
  - all other cases follow the expansion rules, e.g. if $\alpha \vee \beta \in \Gamma(\Pi_i)$, then either $\alpha \in \Gamma(\Pi_i)$ or $\beta \in \Gamma(\Pi_i)$ by construction
- since $\phi \in \Pi_0$, $\overline{\sigma} \models \phi$. CVD

Completeness

Completeness

The completeness proof revolves around the PRUNE rule.

Atoms and pre-models

We need a different but related concept of atom:

Definition

Let $X \subseteq \mathcal{C}(\phi)$ be a set of formulae. An atom generated by X is a minimal set of formulae $\Delta \subseteq \mathcal{C}(\phi)$ such that:

- $X \subseteq \Delta$
- if $\psi \in \Delta$, then $\Gamma_1(\psi) \subseteq \Delta$ or $\Gamma_2(\psi) \subseteq \Delta$

Atoms and pre-models

A pre-models is a sequence of atoms that abstracts a set of models of a formula.

Definition

A pre-model generated by a set $X \subseteq \mathcal{C}(\phi)$ is an infinite sequence of atoms $\overline{\Delta} = \langle \Delta_0, \Delta_1, \ldots \rangle$ such that:

- Δ_0 is generated by X;
- for all $i \geqslant 0$, Δ_{i+1} is generated by $X(\Delta_i)$ where $X(\Delta) = \{\psi \mid \mathsf{X}\psi \in \Delta\}$;
- if $\alpha \,\mathcal{U}\, \beta \in \Delta_i$, then there is a $k \geqslant k$ such that $\beta \in \Delta_k$, and $\alpha \in \Delta_j$ for all $i \leqslant j < k$.

A pre-model for ϕ is a pre-model generated by $\{\phi\}$.

Atoms and pre-models

Pre-models abstract models for a formula:

- A set of models for ϕ can be extracted by a pre-model $\overline{\Delta}$
- A pre-model $\overline{\Delta}$ can be built from a model $\overline{\sigma}$ for ϕ such that if $\psi \in \Delta_i$ then $\overline{\sigma}, i \models \psi$

Finding the accepted branch

The tableau nodes can be used to build atoms:

- get the sequence $\overline{\pi} = \langle \pi_0, \ldots, \pi_m \rangle$ of a branch's poised nodes
- $\Delta(\pi_i)$, the atom of π_i, is the union of all the nodes that got expanded to reach π_i.
- $\Delta(\pi_i)$ is generated by $X(\Gamma(\pi_{i-1}))$.

Finding the accepted branch

So suppose a model for ϕ exists:

- make a pre-model $\overline{\Delta}$ for ϕ
- use $\overline{\Delta}$ as a guide to traverse the tableau for ϕ:
  - find a branch $\overline{u} = \langle u_0, \ldots, u_n \rangle$ *s.t.* if $\Delta(u_i) = \Delta_i$ for all $i \geqslant 0$
- the branch cannot be rejected by the CONTRADICTION rule, because otherwise we would have $\{p, \neg p\} \subseteq \Delta_n$.
- so either the branch is accepted, and we are done, or it has been rejected by the PRUNE rule.

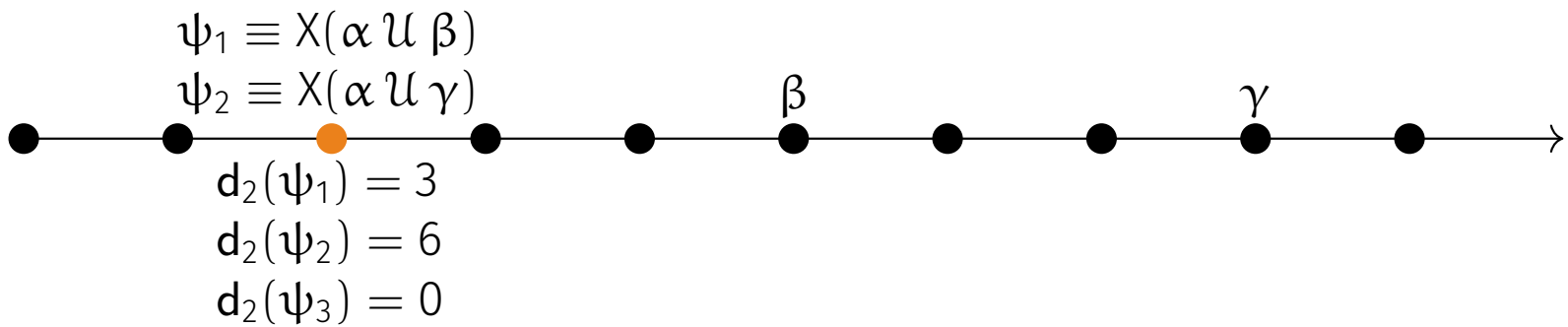
Let's show the latter cannot happen.

Greedy pre-models

Let $\overline{\Delta} = \{\Delta_0, \Delta_1, \ldots\}$ be a pre-model for ϕ. For each

X-eventuality $\psi \equiv X(\alpha \,\mathcal{U}\, \beta) \in \mathcal{C}(\phi)$ and position $i \geqslant 0$, let:

$$\mathbf{d}_i(\psi) = \begin{cases} 0 & \text{if } \psi \text{ is not requested at position } i \\ n & \text{if } \psi \text{ is requested at position } i \\ & \text{and fulfilled at } j > i \text{ such that } n = j - 1 \end{cases}$$

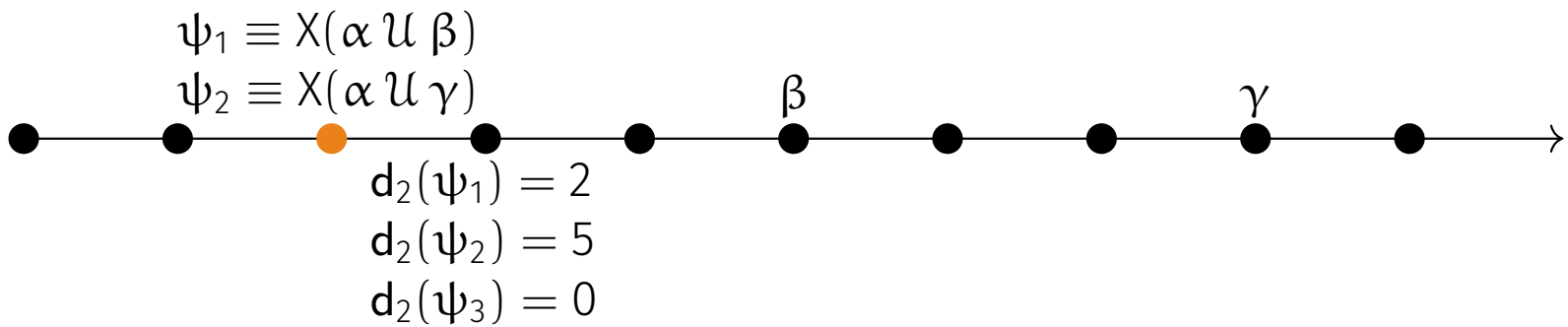


Greedy pre-models

Let $\overline{\Delta} = \{\Delta_0, \Delta_1, \ldots\}$ be a pre-model for ϕ. For each

X-eventuality $\psi \equiv X(\alpha \,\mathcal{U}\, \beta) \in \mathcal{C}(\phi)$ and position $i \geqslant 0$, let:

$$\mathbf{d}_i(\psi) = \begin{cases} 0 & \text{if } \psi \text{ is not requested at position } i \\ n & \text{if } \psi \text{ is requested at position } i \\ & \text{and fulfilled at } j > i \text{ such that } n = j - 1 \end{cases}$$

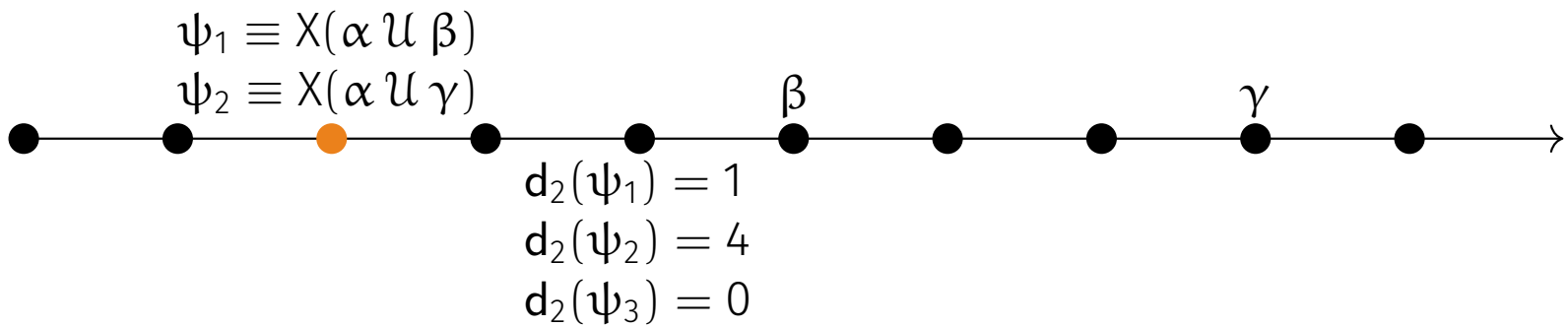


Greedy pre-models

Let $\overline{\Delta} = \{\Delta_0, \Delta_1, \ldots\}$ be a pre-model for ϕ. For each

X-eventuality $\psi \equiv X(\alpha \,\mathcal{U}\, \beta) \in \mathcal{C}(\phi)$ and position $i \geqslant 0$, let:

$$\mathbf{d}_i(\psi) = \begin{cases} 0 & \text{if } \psi \text{ is not requested at position } i \\ n & \text{if } \psi \text{ is requested at position } i \\ & \text{and fulfilled at } j > i \text{ such that } n = j - 1 \end{cases}$$

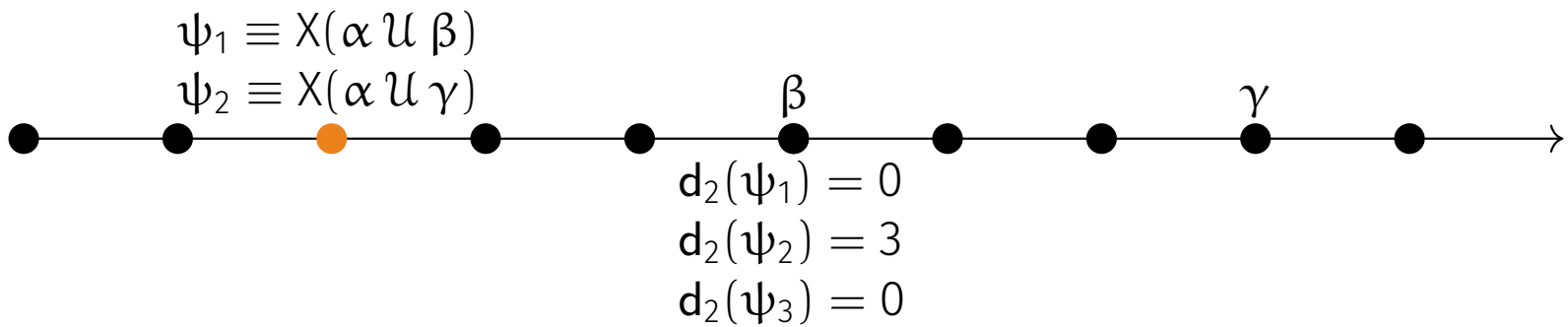


Greedy pre-models

Let $\overline{\Delta} = \{\Delta_0, \Delta_1, \ldots\}$ be a pre-model for ϕ. For each

X-eventuality $\psi \equiv X(\alpha \,\mathcal{U}\, \beta) \in \mathcal{C}(\phi)$ and position $i \geqslant 0$, let:

$$\mathbf{d}_i(\psi) = \begin{cases} 0 & \text{if } \psi \text{ is not requested at position } i \\ n & \text{if } \psi \text{ is requested at position } i \\ & \text{and fulfilled at } j > i \text{ such that } n = j - 1 \end{cases}$$



Greedy pre-models

With delays, we can define a preorder on pre-models.

Given two premodels $\overline{\Delta}$ and $\overline{\Delta}'$:

- $\mathbf{d}_i \leqslant \mathbf{d}'_i$ iff $\mathbf{d}_i(\psi) \leqslant \mathbf{d}'_i(\psi)$ for all X-eventualities ψ
- $\Delta_i \preceq \Delta'_i$ iff $\mathbf{d}_i \leqslant \mathbf{d}'_i$
- the order on pre-models is defined lexicographically:

  $\overline{\Delta} \preceq \overline{\Delta}'$ iff $\Delta_j \leqslant \Delta'_j$ for some $j \geqslant 0$ with $\Delta_i = \Delta'_i$ for all $i < j$.

A greedy pre-model is a minimal element of this preorder.

Existence of greedy pre-models

If there is a pre-model $\overline{\Delta}$ for ϕ, then there is a greedy one $\overline{\Delta}' \preceq \overline{\Delta}$.

$$
\begin{array}{ccccccc}
\Delta_0^0 & \Delta_1^0 & \Delta_2^0 & \cdots & \Delta_n^0 & \cdots \\
\upharpoonright & & & & & \\
\Delta_0^1 & \Delta_1^1 & \Delta_2^1 & \cdots & \Delta_n^1 & \cdots \\
\upharpoonright & & & & & \\
\Delta_0^2 & \Delta_1^2 & \Delta_2^2 & \cdots & \Delta_n^2 & \cdots \\
\wr\!\| & \upharpoonright & & & & \\
\Delta_0^3 & \Delta_1^3 & \Delta_2^3 & \cdots & \Delta_n^3 & \cdots \\
\wr\!\| & \wr\!\| & \upharpoonright & & & \\
\Delta_0^4 & \Delta_1^4 & \Delta_2^4 & \cdots & \Delta_n^4 & \cdots \\
\wr\!\| & \wr\!\| & \wr\!\| & & \upharpoonright & \\
\Delta_0^5 & \Delta_1^5 & \Delta_2^5 & \cdots & \Delta_n^5 & \cdots
\end{array}
$$

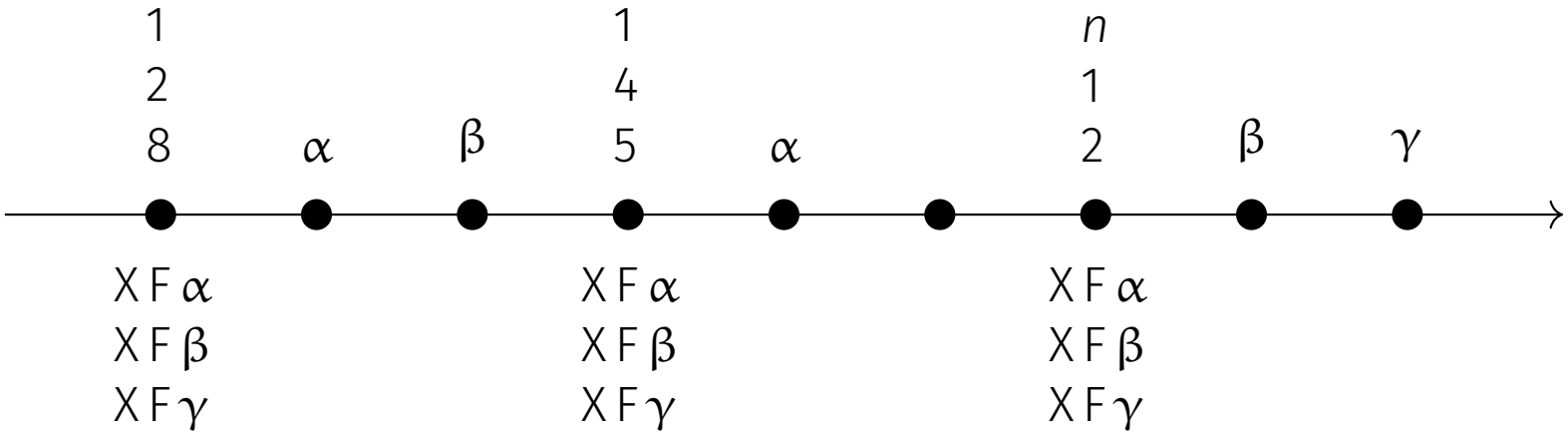
- An infinite descending chains of pre-models might exists
- but, for any m the prefix of length m stabilizes sooner or later
- hence the limit of the sequence is a pre-model, and is greedy.

Completeness: Endgame

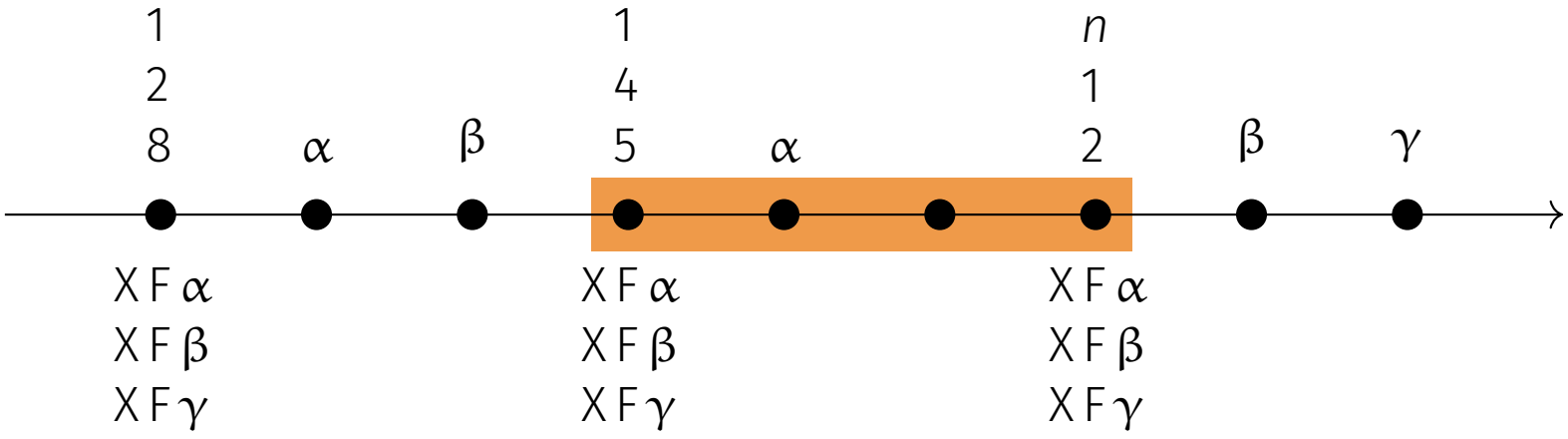
Greedy pre-models allow us to prove completeness:

- Traverse the tableau tree as described before, but suppose to do it with a greedy pre-model $\overline{\Delta}$ for the formula ϕ
- Recall that we found a branch $\overline{u}$ with poised nodes $\overline{\pi}$ such that $\Delta(\pi_i) = \Delta_i$
- Recall that by construction, it can only have been crossed by the PRUNE rule, if at all
- Let's show this cannot happen.

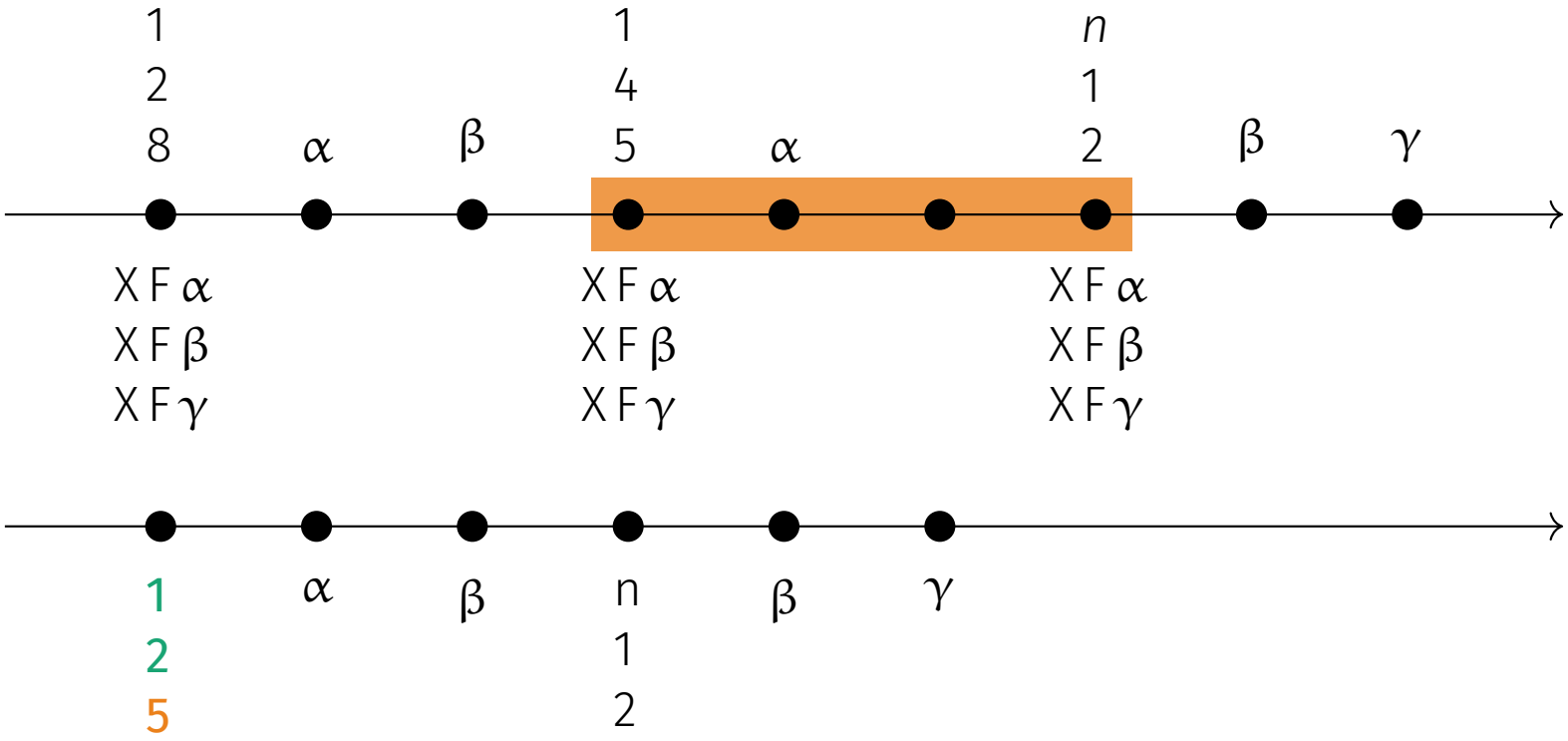
Completeness: Endgame



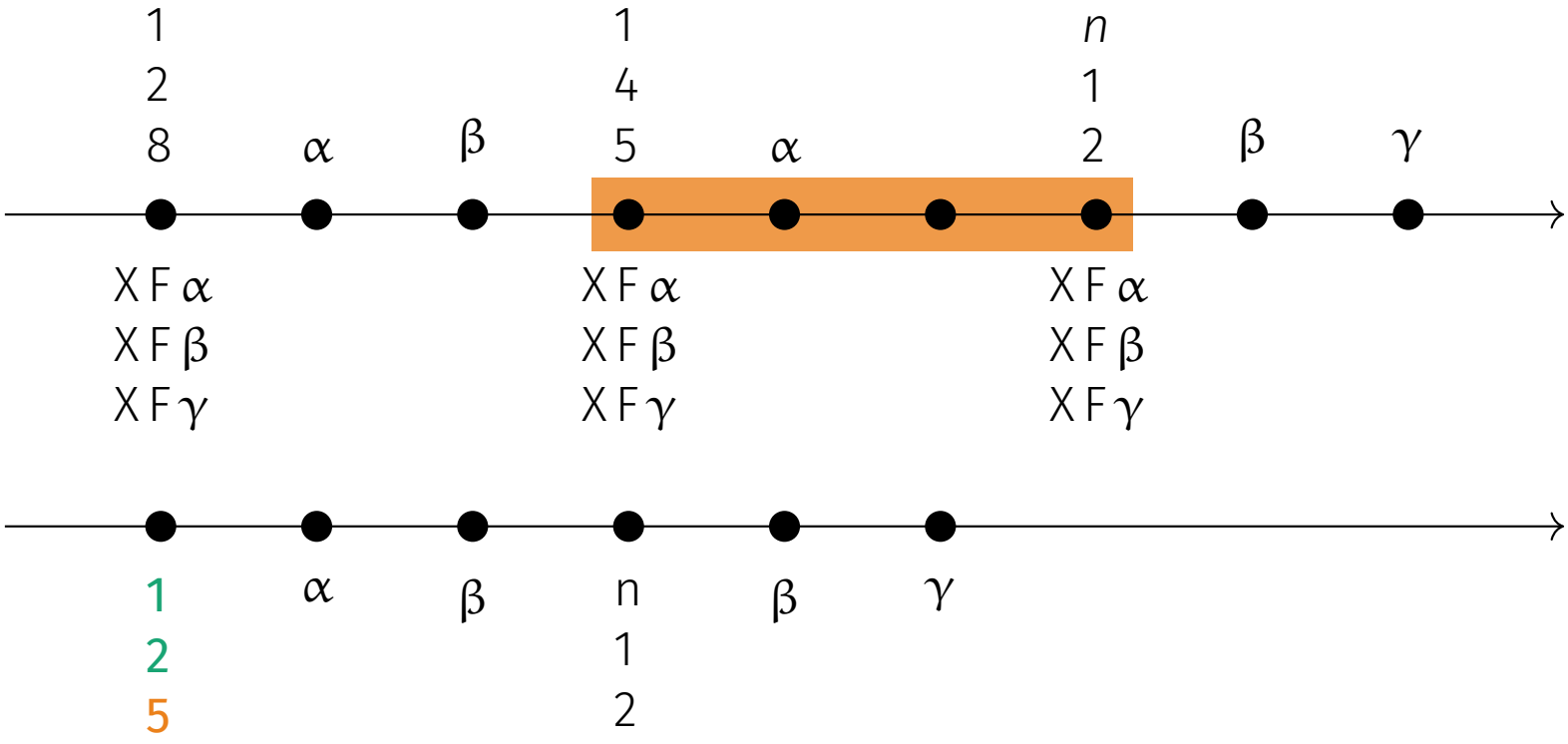
Completeness: Endgame



Completeness: Endgame

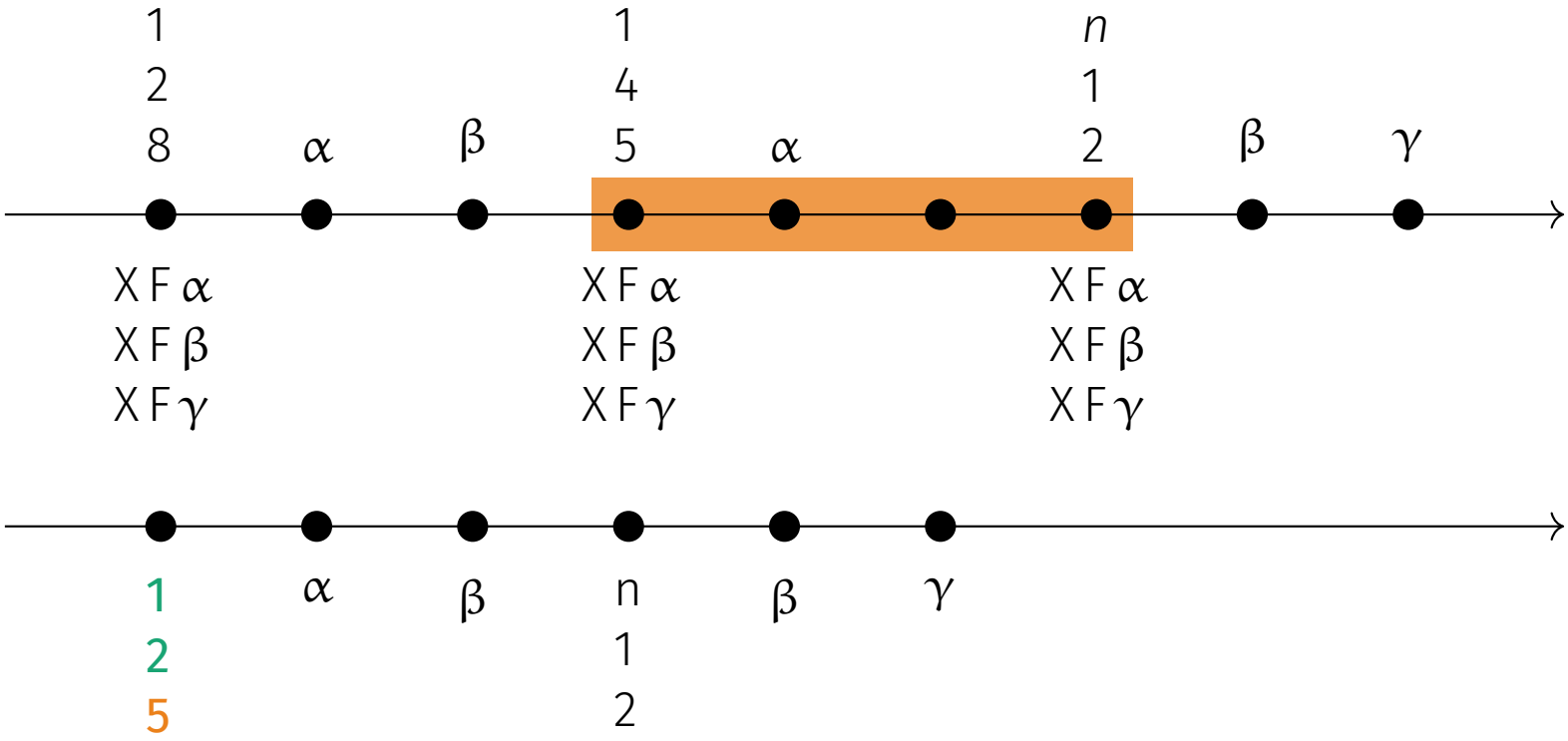


Completeness: Endgame



If this happens, we can find a $\overline{\Delta}' \preceq \overline{\Delta}$.
Hence $\overline{\Delta}$ is not greedy.

Completeness: Endgame



Hence, starting with a greedy pre-model,
we cannot find a rejected branch. CVD.

The end

Questions?

Appendix

Why three occurrences?

Consider this formula:

$$\phi \equiv p \wedge \mathsf{G}(p \longleftrightarrow \mathsf{X}\neg p) \wedge \mathsf{G}\,\mathsf{F}\,q_1 \wedge \mathsf{G}\,\mathsf{F}\,q_2 \wedge \\ \mathsf{G}\neg(q_1 \wedge q_2) \wedge \mathsf{G}(q_1 \longrightarrow \neg p) \wedge \mathsf{G}(q_2 \longrightarrow \neg p)$$

Why three occurrences?

Consider this formula:

$$\phi \equiv p \wedge \mathsf{G}(p \longleftrightarrow \mathsf{X}\neg p) \wedge \mathsf{G}\,\mathsf{F}\,q_1 \wedge \mathsf{G}\,\mathsf{F}\,q_2 \wedge \mathsf{G}\neg(q_1 \wedge q_2) \wedge \mathsf{G}(q_1 \longrightarrow \neg p) \wedge \mathsf{G}(q_2 \longrightarrow \neg p)$$

And its tableau:

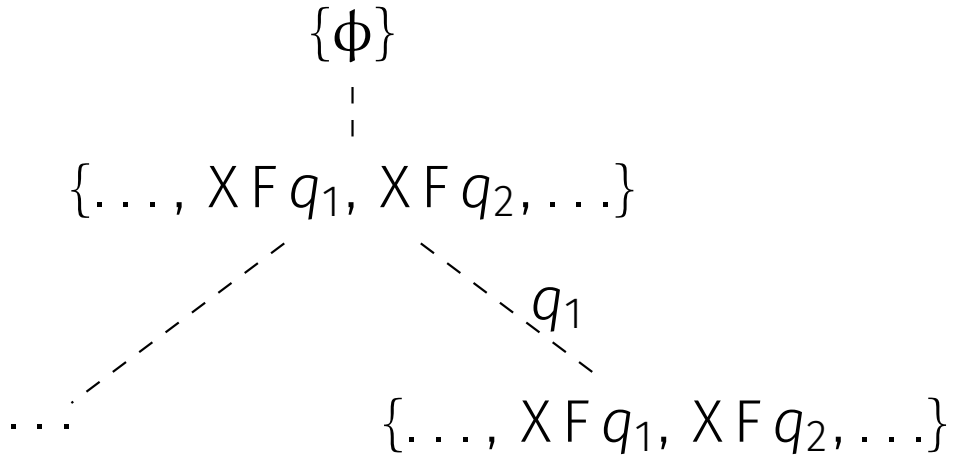
$$\{\phi\}$$

Why three occurrences?

Consider this formula:

$$\phi \equiv p \wedge \mathsf{G}(p \longleftrightarrow \mathsf{X} \neg p) \wedge \mathsf{G}\,\mathsf{F}\, q_1 \wedge \mathsf{G}\,\mathsf{F}\, q_2 \wedge \mathsf{G} \neg (q_1 \wedge q_2) \wedge \mathsf{G}(q_1 \longrightarrow \neg p) \wedge \mathsf{G}(q_2 \longrightarrow \neg p)$$

And its tableau:

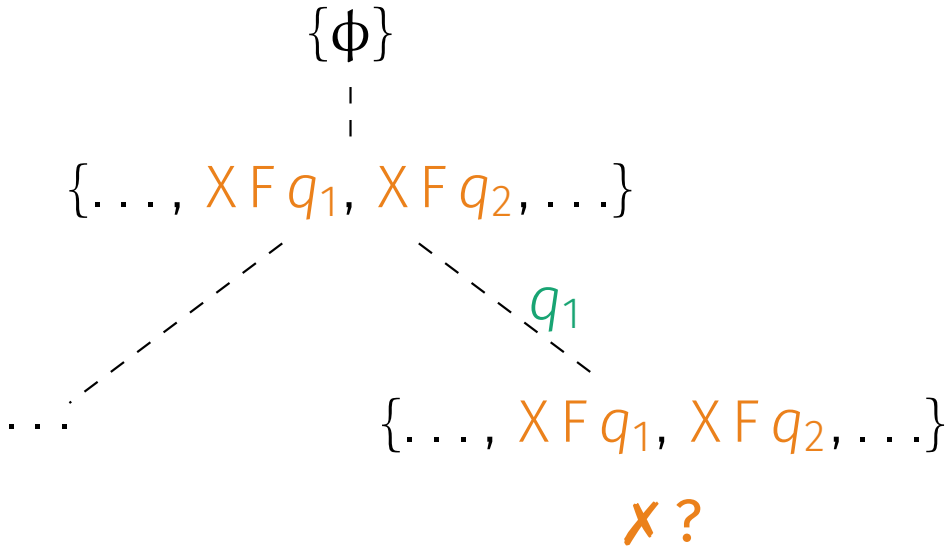


Why three occurrences?

Consider this formula:

$$\phi \equiv p \wedge \mathsf{G}(p \longleftrightarrow \mathsf{X}\neg p) \wedge \mathsf{G}\,\mathsf{F}\,q_1 \wedge \mathsf{G}\,\mathsf{F}\,q_2 \wedge \mathsf{G}\neg(q_1 \wedge q_2) \wedge \mathsf{G}(q_1 \longrightarrow \neg p) \wedge \mathsf{G}(q_2 \longrightarrow \neg p)$$

And its tableau:

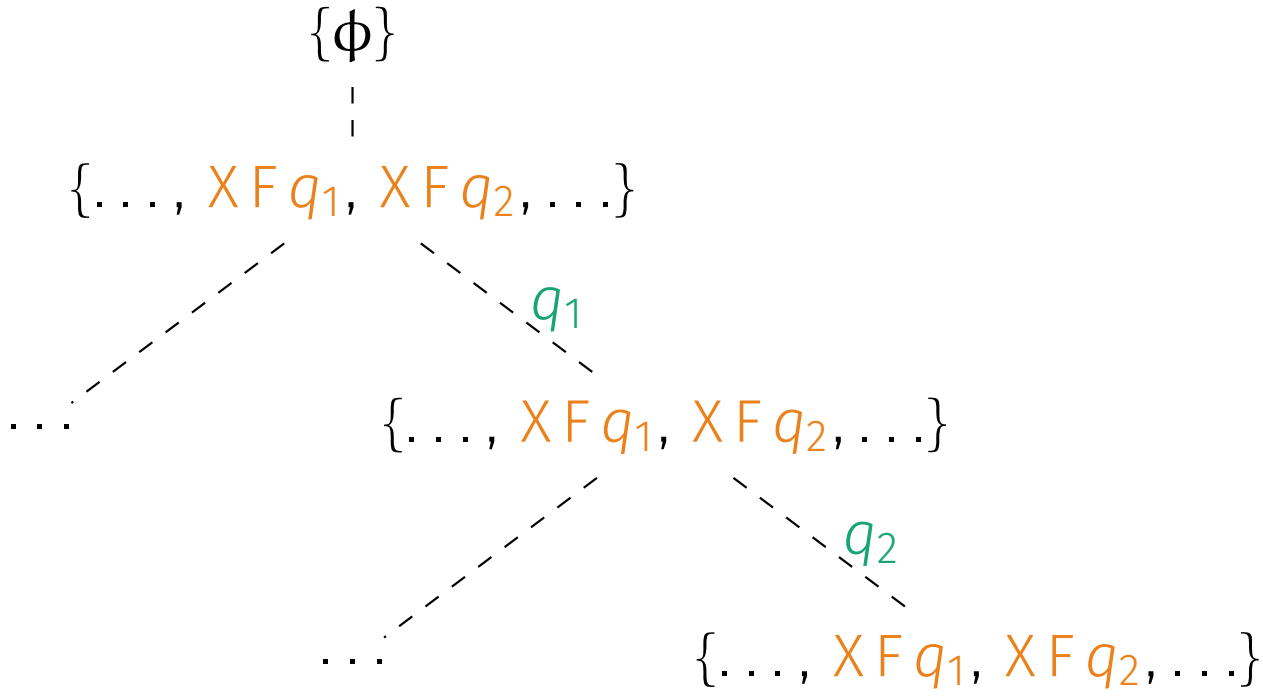


Why three occurrences?

Consider this formula:

$$\phi \equiv p \wedge \mathsf{G}(p \longleftrightarrow \mathsf{X}\neg p) \wedge \mathsf{G}\,\mathsf{F}\,q_1 \wedge \mathsf{G}\,\mathsf{F}\,q_2 \wedge \mathsf{G}\neg(q_1 \wedge q_2) \wedge \mathsf{G}(q_1 \longrightarrow \neg p) \wedge \mathsf{G}(q_2 \longrightarrow \neg p)$$

And its tableau:



Why three occurrences?

Consider this formula:

$$\phi \equiv p \wedge \mathsf{G}(p \longleftrightarrow \mathsf{X}\neg p) \wedge \mathsf{G}\,\mathsf{F}\,q_1 \wedge \mathsf{G}\,\mathsf{F}\,q_2 \wedge \mathsf{G}\neg(q_1 \wedge q_2) \wedge \mathsf{G}(q_1 \longrightarrow \neg p) \wedge \mathsf{G}(q_2 \longrightarrow \neg p)$$

And its tableau: