Example: \( \varphi_0 : \Diamond p \)

Tableau \( T_{\varphi_0} \):

\[
\begin{array}{c}
A_1 : \{ p, \Diamond p, \Diamond p \} \\
A_2 : \{ \neg p, \Diamond p, \Diamond p \} \\
A_3 : \{ p, \neg \Diamond p, \Diamond p \} \\
A_4 : \{ \neg p, \neg \Diamond p, \neg \Diamond p \}
\end{array}
\]

Promising Formula

In \( T \Diamond p \), a path can start and stay forever in atom \( A_2 \).
But \( A_2 \) includes \( \Diamond p \), i.e., \( A_2 \) promises that \( p \) will eventually happen, but it is never fulfilled in the path.
We want to exclude these paths.

The idea is that if a path contains an atom that includes a promising formula, then the path should fulfill the promise.

A formula \( \psi \in \Phi_{\varphi} \) is said to promise the formula \( r \)
if \( \psi \) is one of the forms:

\[
\begin{align*}
\Diamond r & \quad \Box \ Diamond r & \quad \neg \Box r & \quad \neg((\neg r) \lor p) \\
\approx \Diamond r \wedge \ldots & \approx \Diamond r & \approx \Diamond r \wedge \ldots
\end{align*}
\]

Example:

\( \varphi_1 : \Box p \land \Diamond \neg p \)

\( \Phi_{\varphi_1} : \{ \varphi_1, \Box p, \Diamond \neg p, \Box \neg p, \Box \Diamond p, \Diamond p, p \} \)

Only 2 promising formulas in \( \Phi_{\varphi} \)

\( \psi_1 : \neg \Box p \) promises \( r_1 : \neg p \)
\( \psi_2 : \Diamond \neg p \) promotes \( r_2 : \neg p \)
Promise Fulfillment

Property:
Let \( \sigma \) be an arbitrary model of \( \varphi \),
and \( \psi \in \Phi_\varphi \) a formula that promises \( r \).
If \( (\sigma, j) \vdash \psi \) then \( (\sigma, k) \vdash r \) for some \( k \geq j \)

Proof: Follows from the semantics of temporal formulas.

Claim: (promise fulfillment by models)
Let \( \sigma \) be an arbitrary model of \( \varphi \),
and \( \psi \in \Phi_\varphi \) a formula that promises \( r \).
Then \( \sigma \) contains infinitely many positions \( j \geq 0 \)
such that
\[
(\sigma, j) \models \neg \psi \text{ or } (\sigma, j) \models r
\]

Proof:
1. Assume \( \sigma \) contains infinitely many \( \psi \)-positions.
   Then \( \sigma \) must contain infinitely many \( r \)-positions,
since \( \psi \) promises \( r \).
2. Assume \( \sigma \) contains finitely many \( \psi \)-positions.
   Then it contains infinitely many \( \neg \psi \)-positions.

Tableau \( T \Diamond p \)

- \( A_1^+ : \{p, \Diamond \Diamond p, \Diamond p\} \)
- \( A_2^- : \{\neg p, \Diamond \Diamond p, \Diamond p\} \)
- \( A_3^+ : \{p, \neg \Diamond \Diamond p, \Diamond p\} \)
- \( A_4^- : \{\neg p, \neg \Diamond \Diamond p, \neg \Diamond p\} \)
- \( A_2^- : \{\neg p, \Diamond \Diamond p, \Diamond p\} \)

Fulfilling Atoms

Definition: Atom \( A \) fulfills \( \psi \in \Phi_\varphi \)
(which promises \( r \))
if \( \neg \psi \in A \) or \( r \in A \).

Example: In \( T \Diamond p \):
Only one promising formula:
\( \psi : \Diamond p \) promises \( r : p \)
\( A_1^+ : \{p, \Diamond \Diamond p, \Diamond p\} \)
fulfills \( \Diamond p \) since \( p \in A_1 \)
\( A_3^+ : \{p, \neg \Diamond \Diamond p, \Diamond p\} \)
fulfills \( \Diamond p \) since \( p \in A_3 \)
\( A_4^- : \{\neg p, \neg \Diamond \Diamond p, \neg \Diamond p\} \)
fulfills \( \Diamond p \) since \( \neg \Diamond p \in A_4 \)
But
\( A_2^- : \{\neg p, \Diamond \Diamond p, \Diamond p\} \)
does not fulfill \( \Diamond p \) since \( \Diamond p, \neg p \in A_2 \)

Fulfilling Paths

Definition: A path \( \pi : A_0, A_1, \ldots \) is fulfilling if
for every promising formula \( \psi \in \Phi_\varphi \)
it contains infinitely many \( A_j \) that fulfill \( \psi \).

Example: In \( T \Diamond p \):
\( A_2^- \), \( A_2^- \), \( A_2^+ \), \( A_3^+ \), \( A_4^+ \), \ldots
\( A_2^-, A_3^-, A_1^+, A_1^+, A_1^+, A_1^+, \ldots \)
are fulfilling paths, but
\( A_2^-, A_2^-, A_2^-, A_2^-, A_2^-, A_2^-, \ldots \)
is not a fulfilling path.
Example: (Cont’d)

- path $(A^{-}_{7})^{\omega}$ not fulfilling.
- path $(A^{++}_{2})^{\omega}$ is fulfilling.
- path $(A^{++}_{2}, A^{-}_{3})^{\omega}$ is fulfilling.
- path $A^{++}_{4}, (A^{++}_{5})^{\omega}$ is fulfilling.
- For arbitrary $m$, path
  \[ \pi: (A^{++}_{2}, A^{-}_{3})^{m}, A^{++}_{4}, (A^{++}_{5})^{\omega} \]
  is fulfilling.

Example:

\[ \varphi_{1}: \square p \land \Diamond \neg p \]

$T_{\varphi_{1}}$ in Fig 5.3

There are two promising formulas in $\Phi$:

- $\psi_{1} : \neg \square p$ promises $r_{1} : \neg p$
- $\psi_{2} : \Diamond \neg p$ promises $r_{2} : \neg p$

- $A^{++}_{0} : \{ \neg p, \neg \square p, \Diamond \neg p, \ldots \}$
- $A^{++}_{1} : \{ p, \neg \square p, \Diamond \neg p, \ldots \}$
- $A^{++}_{2} : \{ \neg p, \neg \square p, \Diamond \neg p, \ldots \}$
- $A^{-}_{3} : \{ p, \neg \square p, \Diamond \neg p, \ldots \}$
- $A^{++}_{4} : \{ \neg p, \neg \square p, \Diamond \neg p, \ldots \}$
- $A^{++}_{5} : \{ p, \neg \square p, \Diamond \neg p, \ldots \}$
- $A^{++}_{6} : \{ \neg p, \neg \square p, \Diamond \neg p, \ldots \}$
- $A^{++}_{7} : \{ p, \neg \square p, \Diamond \neg p, \ldots \}$

Models vs. fulfilling paths

Claim 2 (model $\rightarrow$ fulfilling path):
If
\[ \pi_{\sigma} : A_{0}, A_{1}, \ldots \]

is a path induced by a model $\sigma$ of $\varphi$,
then $\pi_{\sigma}$ is fulfilling.

Claim 3 (fulfilling path $\rightarrow$ model):

If
\[ \pi_{\sigma} : A_{0}, A_{1}, \ldots \]

is a fulfilling path in $T_{\varphi}$,
then there exists a model $\sigma$ of $\varphi$ that induces $\pi_{\sigma}$.
**Proposition 1** (satisfiability by path)

Formula $\varphi$ is satisfiable

iff

the tableau $T_\varphi$ contains a fulfilling path

$\pi : A_0, A_1, A_2, \ldots$ such that $\varphi \in A_0$

**Proof:**

($\Leftarrow$) $\pi : A_0, A_1, \ldots$ is a fulfilling path in $T_\varphi$ with

$\varphi \in A_0$

Then, by Claim 3, there exists model $\sigma$ such that

$\forall j \geq 0, \forall p \in \Phi_\varphi : (\sigma, j) \models p$ if $p \in A_j$

Since $\varphi \in A_0$, $(\sigma, 0) \models \varphi$ and thus $\sigma \models \varphi$.

($\Rightarrow$) $\sigma \models \varphi$. Then by Claims 1, 2, there exists a fulfilling

path $\pi_\sigma$ in $T_\varphi$ that is induced by $\sigma$.

Since $(\sigma, 0) \models \varphi$, by the definition of induced, $\varphi \in A_0$.

**Examples**

In the examples below we use the following optimization:

A path starting in $A$ can only visit nodes that are reachable from $A$ in $T_\varphi$. So we only need to consider nodes that are reachable from nodes labeled by atoms $A$ such that $\varphi \in A$.

**Example:**

$\varphi_1 : \Box p \land \neg \Box p$

$\Phi_{\varphi_1} = \{ \varphi_1, \Box p, \neg \Box p, \neg p, \Box \neg p, \neg \Box \neg p \}$

$\neg \Box p$ and $\neg \Box \neg p$ promise $\neg p$.

Basic formulas:

$\{ p, \Box p, \neg \Box p \} \rightarrow 8$ atoms

There is only one atom s.t. $\varphi_1 \in A$:

$A_7 : \{ p, \Box p, \neg \Box p, \neg p, \varphi_1 \}$

Any successor of $A_7$ requires $\Box p$, $\neg \Box p$, and therefore $\varphi_1$.

So the only successor is $A_7$ itself, and the part of $T_{\varphi_1}$ reachable from $A_7$ is

which has the infinite path $A_7^+$.

However, $A_7^+$ does not fulfill the promising formula $\Box \neg p$, and thus $A_7^+$ is not a fulfilling path. Hence, $\varphi_1$ is not satisfiable.
Strongly Connected Subgraphs (scs’s)

Definitions

• A subgraph $S \subseteq T_\varphi$ is called strongly connected subgraph (scs) if for every 2 distinct atoms $A, B \in S$
  which there exists a path from $A$ to $B$

Note: a single-node subgraph is an scs

• A single-node scs is called transient ("bad") if it is not connected to itself

• A non-transient ("good") scs $S$ is fulfilling if every promising formula $\psi \in \Phi_\varphi$ is fulfilled by some atom $A \in S$, i.e.
  $\neg \psi \in A$ or $r \in A$

• scs $S$ is $\varphi$-reachable if there exist a path and $k \geq 0$
  $B_0, B_1, \ldots, B_k, \ldots$
  such that $\varphi \in B_0$ and $B_k \in S$.

Example: In $T_\square p$,

$\{A_1^+, \{A_2^+, A_5^-\}, \{A_4^+\}\}$ are fulfilling
$\{A_2^-\}$ is not fulfilling
All scss are $(\square p)$-reachable.
$A_3$ is a transient scs. All others are good scss.

Example: In $T_\varphi_1$ (Fig. 5.3),

$\{A_4\}$ transient scs
$\{A_5\}$ good scs
$\{A_7\}$ is the only $\varphi_1$-reachable scs

$\{A_2^{++,}A_3^--\} ~ \{A_5^{++}\}$ fulfilling scs’s
$\{A_1^{++,}\} ~ \{A_7^+\}$ scs’s but not fulfilling

Why scs’s?

In general a tableau may have infinitely many paths, so we cannot directly determine whether there are any fulfilling paths.

What needs to hold?

• When does a graph have an infinite path?
  $\rightarrow$ it must have a non-transient scs.

• When is such an infinite path induced by a model of $\varphi$?
  $\rightarrow$ scs must be $\varphi$-reachable,
  i.e., reachable from a node labeled by $A$, s.t. $\varphi \in A$
  $\rightarrow$ scs must be fulfilling,
  i.e., for every promising formula $\psi \in \Phi_\varphi$ the scs must have at least one atom that fulfills $\psi$. 
Proposition (satisfiability by scs)

Formula $\phi$ is satisfiable

iff

the tableau $T_{\phi}$ contains a $\phi$-reachable fulfilling scs

The number of scs’s in a graph is finite, but may be exponential in the size of the graph!

Example: $\varphi_0 : \Box p$

In $T_{\varphi_0}$, the fulfilling SCS’s

$\{A_1^+ \} \{A_1^+, A_2^- \} \{A_4^+ \}$

are reachable from an initial node.

Thus, $\varphi_0 : \Box p$ is satisfiable.

Satisfying models:

$p^\omega (p, \neg p)^\omega p, (\neg p)^\omega$.

Maximal Strongly Connected Subgraphs (mscs’s)

Definition: An scs is maximal (mscs) if it is not properly contained in any larger scs

Example: In $T_{\varphi_1}$ (Fig. 5.3),

$\{A_2, A_3\}$ not MSCS

$\{A_2\}$ MSCS

In fact, it is sufficient to determine whether there exists a fulfilling reachable MSCS in $T_{\phi}$. The number of MSCS in $T_{\phi}$ is bounded by $|T_{\phi}|$.

Decomposition into mscs’s

There exists an efficient algorithm [Hopcroft&Tarjan] to decompose $T_{\phi}$ into subgraphs $G_1, \ldots, G_N$ such that

- each $G_i$ is an mscs (and therefore disjoint)

- $G_1 \cup \ldots \cup G_N = T_{\phi}$

- whenever there is an edge from a node in $G_i$ to a node in $G_j$ then $i \leq j$. 

Algorithm SAT

(check satisfiability of arbitrary temporal formula $\varphi$)

- construct $T_{\varphi}$

- construct $T^-_{\varphi}$ by removing all atoms that are not reachable from $\varphi$-atom

- decompose $T^-_{\varphi}$ into mscs’s $U_1, \ldots, U_k$

- check whether $U_1, \ldots, U_k$ is fulfilling:

  - if some $U_i$ is fulfilling: $\varphi$ is satisfiable.
    A model is defined by the path leading from a $\varphi$-atom to $U_i$ and staying in $U_i$ forever from then on.

  - if no $U_i$ is fulfilling: $\varphi$ is not satisfiable.
Proposition (satisfiability and mscs)

Formula $\varphi$ is satisfiable iff

The tableau $T_{\neg \varphi}$ contains a $\varphi$-reachable fulfilling mscs

Check validity of $\varphi$

Apply algorithm SAT to $\neg \varphi$

Algorithm reports success:

$\neg \varphi$ is satisfiable $\Rightarrow$ $\varphi$ is not valid

(the produced $\sigma$ is a counterexample)

Algorithm reports failure:

$\neg \varphi$ is unsatisfiable $\Rightarrow$ $\varphi$ is valid

Example: Check satisfiability of $\varphi_1$: $\Box p \land \Diamond \neg p$

$T_{\varphi_1} (\text{Fig 5.3})$

$T_{\neg \varphi_1} = \{A_7^+, A_7^-\}$ mscs of $T_{\neg \varphi_1} = \{A_7^+, A_7^-\}$

Corresponding nonfulfilling

Example:

$\psi_1 = \neg \varphi_1$: $\neg (\Box p \land \Diamond \neg p)$

$T_{\psi_1} (\text{Fig 5.3})$

$T_{\neg \psi_1}$: all atoms

mscs's:

- $\{A_0\}, \{A_4\}, \{A_6\}$ transient
- $\{A_1^+, A_7^-\}$ non-fulfilling
- $\{A_2^+, A_3^-\}, \{A_5^+\}$ fulfilling

For $A_5^+$:

$A_5^+$ model $(p; T)^\omega$

For $\{A_2^+, A_3^-\}$:

$(A_2, A_3)^\omega$ model $(p; F)(p; T)^\omega$

each satisfies $\psi_1$

Example: Check satisfiability of $\varphi_2$: $\Box (\neg at_{\ell_2} \lor \Diamond at_{\ell_3})$

$\varphi_2^+: \{ \Box p_2, \quad \Box \Box p_2, \quad p_2, \at_{\ell_2}, \at_{\ell_3}, \Box \at_{\ell_3}, \quad \Box \at_{\ell_3}, \at_{\ell_3} \}$

$\varphi_2$-reachable atoms

$\{ \Box p_2, \quad \Box \Box p_2, \quad p_2, \at_{\ell_2}, \at_{\ell_3}, \Box \at_{\ell_3}, \quad \Box \at_{\ell_3}, \at_{\ell_3} \}$

8 possibilities

Fixed:

One promising formula in $\Phi$: $\Diamond at_{\ell_3}$ (and $\neg \Box p_2$)

$A_5^+$:

$A_5^+$:

$A_1^+$:

$A_1^+$:

$A_2^+$:

$A_2^+$:

$A_3^+$:

$A_3^+$:

$A_4^+$:

$A_4^+$:
Example: (Cont’d)

Atom #8
{ □p2, ◇p2, ◇ at−ℓ2, ¬ at−ℓ3, ¬ ◇ at−ℓ3, \ldots }

is not considered since
¬ at−ℓ2 ∨ ◇ at−ℓ3 and at−ℓ2 → ◇ at−ℓ3

Tableau $T\varphi_2$ (Fig 5.4) = $T_{\neg\varphi_2}$

formula ◇ at−ℓ3 promising at−ℓ3

Decomposition to mscs’s

$\{A_1^-, A_3^+, A_4^-, A_6^+\} \{A_2^+\} \{A_5^+\}$

fulfilling mscs’s: $\{A_0^+\}, \{A_1^-, A_3^+, A_4^-, A_6^+\}$

(\{A_2\} and \{A_5\} are transient)

$\varphi_2$ is satisfiable

model (by $A_0^\omega$)

$\langle at−ℓ2: f, at−ℓ3: f \rangle^\omega$

Pruning the tableau

Definition: mscs $S$ is terminal if
there are no edges leading from
atoms of $S$ to atoms outside $S$

Example: Consider $\psi_1 = \neg\varphi_1 : \neg (\Box p \land \neg p)$
In $T_{\psi_1}$ (same as $T_{\varphi_1}$, Fig 5.3, except for initial nodes)

{A_1} {A_5} {A_7} are terminal mscs’s

{A_6} {A_2, A_3} are not

After constructing $T_{\varphi}$, remove useless atoms:

- Remove an mscs that is not $\varphi$-reachable.
- Remove a terminal mscs that is not fulfilling.

Iterate until no further atoms can be removed.
Fig. 5.3: Tableau $T_{\psi_1}$ for formula

$\psi_1: \neg(\Box p \land \lozenge \neg p)$.

A++

2: \{ ¬p, ¬/BE/BCp, /BE/BD¬p, ¬/BCp, /BD¬p, ¬ϕ₁ \}

A+-

7: \{ p, /BE/BCp, /BE/BD¬p, /BCp, /BD¬p, ϕ₁ \}

13-34

Example:

ϕ₃: /BC/BD(x = 3)

Φ⁺

ϕ₃: { ϕ₃, /BD(x = 3), x = 3, /BE/BD(x = 3), /BEϕ₃ }

Fulfilling msc's: \{ A⁺⁺₂, A⁺⁺₃ \}, \{ A⁺⁺₅ \}

$\psi_1: \neg(\Box p \land \lozenge \neg p)$ is satisfiable.

Pruned Tableau $T_{\psi_1}$ for

$\psi_1: \neg(\Box p \land \lozenge \neg p)$

A++

1: \{ x ≠ 3, ¬/BE/BD(x = 3), ¬/BEϕ₃, ¬/BD(x = 3), ¬ϕ₃ \}

13-35

Fig. 5.6. Pruned tableau $T_{ψ₃}$

ϕ₃: /BC/BD(x = 3)

A⁺⁺₁: x = 3

ϕ₃, /BC/BD(x = 3), /BCϕ₃, /BDϕ₃

A⁻⁻₁: x ≠ 3

The $ϕ₃$-reachable msc's: \{ A⁺⁺₀, A⁺⁺₁ \}

\{ A⁺⁺₀, A⁻⁻₁ \} is fulfilling.

Therefore, $ϕ₃$ is satisfiable.

Model (by $(A₀, A₁)ω$): $(⟨x: 3⟩, ⟨x: 0⟩)ω$

arbitrary $x ≠ 3$