Linear Temporal Logic (LTL)

Temporal logic

- Writing property NBAs requires training in automata theory
- We search for a more intuitive (but still formal) description language: Temporal Logic.
- Temporal logic extends propositional logic with temporal operators like always and eventually.
- Linear Temporal Logic (LTL) is a temporal logic interpreted over linear structures.

Linear Temporal Logic (LTL)

- We are given:
  - A set AP of atomic propositions (names for basic properties)
  - A valuation assigning to each atomic proposition a set of configurations (intended meaning: the set of configurations that satisfy the property).

Example

```
1  while x = 1 do
2     if y = 1 then
3        x ← 0
4     y ← 1 − x
5  end
```

- AP : at_1, at_2, ... , at_5, x=0, x=1, y=0, y=1
- $V(at_i) = \{[\ell, x, y] \in C \mid \ell = i\}$ for every $i \in \{1, \ldots, 5\}$
- $V(x{=}0) = \{[\ell, x, y] \in C \mid x = 0\}$

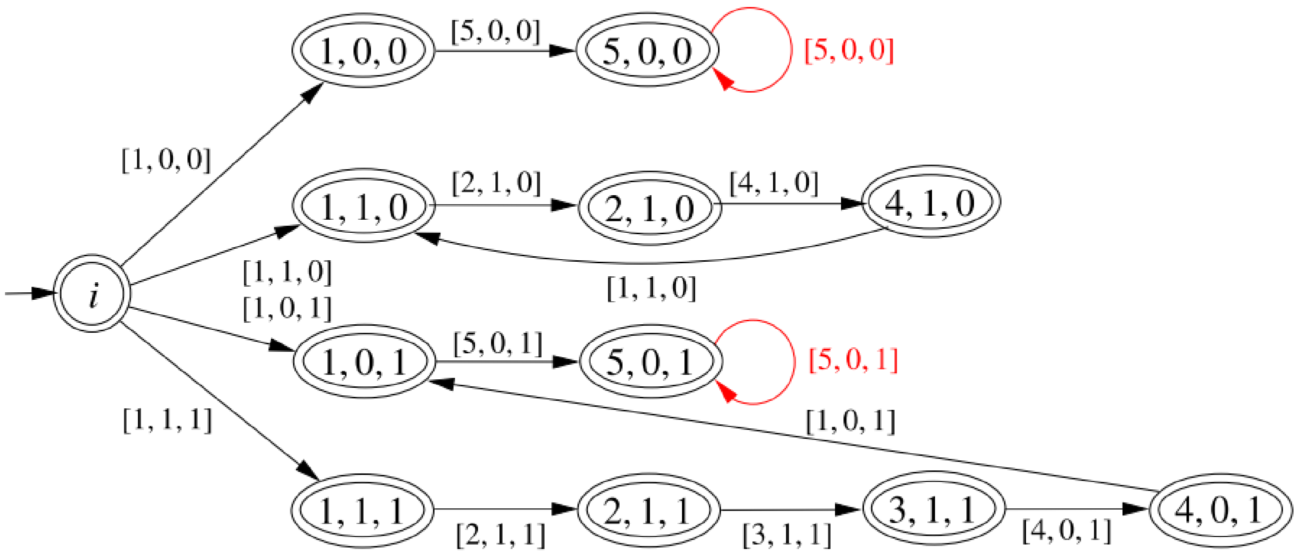
Computations

- A computation is an infinite sequence of subsets of AP.
- Examples for $AP = \{p, q\}$

$$\emptyset^{\omega} \quad (\{p\}\{p,q\})^{\omega} \quad \{p\}\,\{p,q\}\,\emptyset\,\emptyset\,\{p\}^{\omega}$$

- We map every possible execution to a computation by mapping each configuration to the set of atomic propositions it satisfies.
- A computation is executable if some ω-execution maps to it.

Example



ω-executions:

e_1 = [1,0,0] [5,0,0]$^\omega$

e_2 = ([1,1,0] [2,1,0] [4,1,0])$^\omega$

e_3 = [1,0,1][5,0,1]$^\omega$

e_4 = [1,1,1][2,1,1][3,1,1][4,0,1][1,0,1][5,0,1]$^\omega$

From executions to computations

e_1 = [1,0,0] [5,0,0]$^\omega$

e_2 = ([1,1,0] [2,1,0] [4,1,0])$^\omega$

σ_1 = {at1, x=0, y=0} {at5, x=0, y=0}$^\omega$

σ_2 = ({at1, x=0, y=0} {at2, x=1, y=0} {at4, x=1, y=0})$^\omega$

Syntax of LTL

- Given: set AP of atomic propositions, valuation assigning to each atomic proposition a set configurations.
- The formulas of LTL are given by the syntax:

$$\varphi ::= \mathbf{true} \mid p \mid \neg\varphi_1 \mid \varphi_1 \wedge \varphi_2 \mid \mathrm{X}\varphi_1 \mid \varphi_1 \mathrm{U} \varphi_2$$

where $p \in AP$

Semantics of LTL

- Formulas are interpreted on computations (executable or not).
- The satisfaction relation $\sigma \vDash \varphi$ is given by:

$$\sigma \vDash \mathbf{true}$$

$$\sigma \vDash p \text{ iff } p \in \sigma(0)$$

$$\sigma \vDash \neg\varphi \text{ iff not } \sigma \vDash \varphi$$

$$\sigma \vDash \varphi_1 \wedge \varphi_2 \text{ iff } \sigma \vDash \varphi_1 \text{ and } \sigma \vDash \varphi_2$$

$$\sigma \vDash \mathrm{X}\varphi \text{ iff } \sigma^1 \vDash \varphi$$

$$\sigma \vDash \varphi_1 \mathrm{U} \varphi_2 \text{ iff there is } k \geq 0 \text{ s.t.: } \sigma^k \vDash \varphi_2 \text{ and }$$
$$\sigma^i \vDash \varphi_1 \text{ for all } 0 \leq i \leq k$$

Abbreviations

- The boolean abbreviations **false**, $\vee$, $\rightarrow$, $\leftrightarrow$ etc. are defined as usual.
- Fφ $:=$ **true** U φ (eventually φ).

  According to the semantics:

  $\sigma \vDash \mathrm{F}\varphi$ iff there is $k \geq 0$ s. t. $\sigma^k \vDash \varphi$
- Gφ $:= \neg$ F$\neg\varphi$ (always φ or globally φ).

  According to the semantics:

  $\sigma \vDash \mathrm{G}\varphi$ iff $\sigma^k \vDash \varphi$ for every $k \geq 0$
- Gφ $:= \neg$ F$\neg\varphi$ (always φ or globally φ).

  According to the semantics:

  $\sigma \vDash \mathrm{G}\varphi$ iff $\sigma^k \vDash \varphi$ for every $k \geq 0$

Getting used to LTL

- Express in natural language $\mathsf{FG}p$, $\mathsf{GF}p$
- Are these pairs of formulas equivalent?

$\mathsf{FF}p$	$\mathsf{F}p$	$\mathrm{GG}p$	$\mathrm{G}p$
$\mathsf{FG}p$	$\mathsf{GF}p$	$\mathrm{FGF}p$	$\mathrm{GF}p$
$p \cup q$	$p \cup (p \wedge q)$		
$\mathrm{F}p$	$p \vee \mathrm{XF}p$	$\mathrm{F}p$	$p \wedge \mathrm{XF}p$
$\mathrm{G}p$	$p \vee \mathrm{XG}p$	$\mathrm{G}p$	$p \wedge \mathrm{XG}p$
$p \cup q$	$p \vee \mathsf{X}\,(p \cup q)$	$p \cup q$	$p \wedge \mathsf{X}\,(p \cup q)$
$p \cup q$	$q \vee \mathsf{X}\,(p \cup q)$	$p \cup q$	$q \wedge \mathsf{X}\,(p \cup q)$
$p \cup q$	$q \vee (p \wedge \mathsf{X}\,(p \cup q)$	$p \cup q$	$q \wedge (p \vee \mathsf{X}\,(p \cup q)$

Expressing properties of a program

- AP : at_1, at_2, ... , at_5, x=0, x=1, y=0, y=1

  $V(\text{at}_\text{i}) = \{[\ell, x, y] \in C \mid \ell = i\}$ for every $i \in \{1, \ldots, 5\}$

  V(x=0)= $\{[\ell, x, y] \in C \mid$ x=0$\}$
- φ_0 = x=1 ∧ X y=1 ∧ X X at3
- φ_1 = F x=0
- φ_2 = x=0 U at5
- φ_3 = y=1 ∧ F(x=0 ∧ at5) ∧ ¬(F (y=0 ∧ X y=1))

Expressing properties of Lamport's algorithm

- $AP = \{ NC_0, T_0, C_0, NC_1, T_1, C_1, M_0, M_1 \}$

  Valuation as expected.

- Mutual exclusion: $\mathsf{G}\ (\neg C_0 \vee \neg C_1)$
- Finite waiting: $\mathsf{G}\ (T_0 \rightarrow \mathsf{F} C_0) \wedge \mathsf{G}\ (T_1 \rightarrow \mathsf{F} C_1)$

Expressing properties of Lamport's algorithm

- Finite waiting: G $(T_0 \rightarrow \mathsf{F} C_0) \wedge$ G $(T_1 \rightarrow \mathsf{F} C_1)$
  - The property G $(T_0 \rightarrow \mathsf{F} C_0)$ does not hold because of $[0,0, nc_0, nc_1]\ [1,0, t_0, nc_1]\ [1,1, m\ t_0, t_1]^\omega$
  - Not a problem of the algorithm, but of the specification!
- Fairness assumption: both processes execute infinitely many actions

  (Usually a weaker assumption is used: if a process can execute actions infinitely often, then it executes infinitely many actions.)

- Reformulation: in every fair full execution, if a process is trying to access the critical section, it will eventually access it.

Expressing properties of Lamport's algorithm

- How can we represent the fairness condition?
  - Enrich the notion of configuration: pair (c, i) where c is a configuration as before, and i is the index of the process that made the last move.
  - Let M_0 and M_1 be the sets of configurations with indics 0 and 1, respectively.
- Fair finite waiting:

$$(\mathrm{GF}\, M_0 \wedge \mathrm{GF}\, M_1) \rightarrow \big(\mathrm{G}(T_0 \rightarrow \mathrm{F} C_0) \wedge \mathrm{G}(T_1 \rightarrow \mathrm{F} C_1)\big)$$

Lamport's algorithm

- Bounded overtaking:

$$\mathsf{G}\left(T_0 \rightarrow \left(\neg C_1 \,\mathsf{U}\, \left(C_1 \mathsf{U} \left(\neg C_1 \mathsf{U}\, C_0\right)\right)\right)\right)$$

Whenever T_0 holds, the computation continues with

a (possibly empty) interval at which $\neg C_1$ holds,

followed by

a (possibly empty) interval at which C_1 holds,

followed by

a point at which C_0 holds.

From formulas to NBAs

- Given: set AP of atomic propositions
- Language $L(\varphi)$ of a formula φ : set of computations satisfying φ.
- Examples for $AP = \{p, q\}$
  - $L(\mathrm{F}p)$ = computations $s_1 s_2 s_3 \ldots$ such that $p \in s_i$ for some $i \geq 1$
  - $L(\mathrm{G}(p \wedge q)) = \{ \{p, q\}^\omega \}$
- $L(\varphi)$ is an ω-language over the alphabet 2^{AP}
- For $AP = \{p, q\}$ we get $2^{AP} = \{\emptyset, \{p\}, \{q\}, \{p, q\}\}$

NBAs for some formulas

$$AP = \{p, q\}$$

- $\mathrm{F}p$
- $\mathrm{G}p$
- $p \ \mathrm{U} \ q$
- $\mathrm{GF}p$

From LTL formulas to NGAs

We present an algorithm that takes a formula φ over a fixed set AP of atomic propositions as input and returns a NGA A_φ such that $L(A_\varphi) = L(\varphi)$.

Closure of a formula

- Define neg(ψ) $= \begin{cases} \psi \text{ if } \varphi = \neg\psi \\ \neg\varphi \text{ otherwise} \end{cases}$
- The closure $cl(\varphi)$ of φ is the set containing ψ and neg(ψ) for every subformula ψ of φ
- Example:

$$cl(p\,\mathrm{U}\neg q) = \{p, \neg p, \neg q, q, p\mathrm{U}\neg q, \neg(p\mathrm{U}\neg q)\}$$

Satisfaction sequence

- The satisfaction sequence of a computation $s_0 s_1 s_2 \ldots$ with respect to φ is the sequence $\alpha_0 \alpha_1 \alpha_2 \ldots$ where α_i contains the formulas of $cl(\varphi)$ satisfied by $s_i s_{i+1} s_{i+2} \cdots$

Satisfaction sequence

- The satisfaction sequence of a computation $s_0 s_1 s_2 \ldots$ with respect to φ is the sequence $\alpha_0 \alpha_1 \alpha_2 \ldots$ where α_i contains the formulas of $cl(\varphi)$ satisfied by $s_i s_{i+1} s_{i+2} \cdots$
- The satisfaction sequence of $\{p\}^\omega$ w.r.t. $p\ U\ q$ is:

Satisfaction sequence

- The satisfaction sequence of a computation $s_0 s_1 s_2 \ldots$ with respect to φ is the sequence $\alpha_0 \alpha_1 \alpha_2 \ldots$ where α_i contains the formulas of $cl(\varphi)$ satisfied by $s_i s_{i+1} s_{i+2} \cdots$
- The satisfaction sequence of $\{p\}^\omega$ w.r.t. $p\ U\ q$ is:

$$\{p, \neg q, \neg(p\ U\ q)\}^\omega$$

Satisfaction sequence

- The satisfaction sequence of a computation $s_0 s_1 s_2 \ldots$ with respect to φ is the sequence $\alpha_0 \alpha_1 \alpha_2 \ldots$ where α_i contains the formulas of $cl(\varphi)$ satisfied by $s_i s_{i+1} s_{i+2} \cdots$
- The satisfaction sequence of $\{p\}^\omega$ w.r.t. $p\ U\ q$ is:

$$\{p, \neg q, \neg(p\ U\ q)\}^\omega$$

- The satisfaction sequence of $(\{p\}\{q\})^\omega$ w.r.t. $p\ U\ q$ is:

Satisfaction sequence

- The satisfaction sequence of a computation $s_0 s_1 s_2 \ldots$ with respect to φ is the sequence $\alpha_0 \alpha_1 \alpha_2 \ldots$ where α_i contains the formulas of $cl(\varphi)$ satisfied by $s_i s_{i+1} s_{i+2} \cdots$
- The satisfaction sequence of $\{p\}^\omega$ w.r.t. $p\ U\ q$ is:

$$\{p, \neg q, \neg(p\ U\ q)\}^\omega$$

- The satisfaction sequence of $(\{p\}\{q\})^\omega$ w.r.t. $p\ U\ q$ is:

$$(\ \{p, \neg q, p\ U\ q\}\ \{\neg p, q, p\ U\ q\}\ )^\omega$$

Atoms

- Intuition: an atom is a maximal set of formulas of $cl(\varphi)$ that "could be simultaneously true by looking only at $\neg$ and $\wedge$"

Atoms

- Intuition: an atom is a maximal set of formulas of $cl(\varphi)$ that "could be simultaneously true by looking only at $\neg$ and $\wedge$"
- A set $\alpha \subseteq cl(\varphi)$ is an atom if it satisfies the following two conditions:
  - For every $\psi \in cl(\varphi)$, exactly one of ψ and $\text{neg}(\psi)$ belong to α
  - For every $\psi_1 \wedge \psi_2 \in cl(\varphi)$, $\psi_1 \wedge \psi_2 \in \alpha$ iff $\psi_1 \in \alpha$ and $\psi_2 \in \alpha$

Atoms

- Intuition: an atom is a maximal set of formulas of $cl(\varphi)$ that "could be simultaneously true by looking only at $\neg$ and $\wedge$"
- A set $\alpha \subseteq cl(\varphi)$ is an atom if it satisfies the following two conditions:
  - For every $\psi \in cl(\varphi)$, exactly one of ψ and neg(ψ) belong to α
  - For every $\psi_1 \wedge \psi_2 \in cl(\varphi)$, $\psi_1 \wedge \psi_2 \in \alpha$ iff $\psi_1 \in \alpha$ and $\psi_2 \in \alpha$
- Examples of atoms for $\varphi = \neg(p \wedge q)\ U\ Fp$:

Atoms

- Intuition: an atom is a maximal set of formulas of $cl(\varphi)$ that "could be simultaneously true by looking only at ¬ and ∧"
- A set $\alpha \subseteq cl(\varphi)$ is an atom if it satisfies the following two conditions:
  - For every $\psi \in cl(\varphi)$, exactly one of ψ and $\text{neg}(\psi)$ belong to α
  - For every $\psi_1 \wedge \psi_2 \in cl(\varphi)$, $\psi_1 \wedge \psi_2 \in \alpha$ iff $\psi_1 \in \alpha$ and $\psi_2 \in \alpha$
- Examples of atoms for $\varphi = \neg(p \wedge q)\ U\ Fp$:

$$\{\neg p, \neg q, \neg(p \wedge q), Fp, \varphi\} \quad \{p, q, (p \wedge q), \neg Fp, \neg\varphi\}$$

Atoms

- Intuition: an atom is a maximal set of formulas of $cl(\varphi)$ that "could be simultaneously true by looking only at ¬ and ∧"
- A set $\alpha \subseteq cl(\varphi)$ is an atom if it satisfies the following two conditions:
  - For every $\psi \in cl(\varphi)$, exactly one of ψ and neg(ψ) belong to α
  - For every $\psi_1 \wedge \psi_2 \in cl(\varphi)$, $\psi_1 \wedge \psi_2 \in \alpha$ iff $\psi_1 \in \alpha$ and $\psi_2 \in \alpha$
- Examples of atoms for $\varphi = \neg(p \wedge q)\ U\ Fp$:

  $\{\neg p, \neg q, \neg(p \wedge q), Fp, \varphi\}$ $\{p, q, (p \wedge q), \neg Fp, \neg\varphi\}$
- Examples of non-atoms for $\varphi = \neg(p \wedge q)\ U\ Fp$:

Atoms

- Intuition: an atom is a maximal set of formulas of $cl(\varphi)$ that "could be simultaneously true by looking only at $\neg$ and $\wedge$"
- A set $\alpha \subseteq cl(\varphi)$ is an atom if it satisfies the following two conditions:
  - For every $\psi \in cl(\varphi)$, exactly one of ψ and neg(ψ) belong to α
  - For every $\psi_1 \wedge \psi_2 \in cl(\varphi)$, $\psi_1 \wedge \psi_2 \in \alpha$ iff $\psi_1 \in \alpha$ and $\psi_2 \in \alpha$
- Examples of atoms for $\varphi = \neg(p \wedge q)\ U\ Fp$:

$$\{\neg p, \neg q, \neg(p \wedge q), Fp, \varphi\} \quad \{p, q, (p \wedge q), \neg Fp, \neg\varphi\}$$

- Examples of non-atoms for $\varphi = \neg(p \wedge q)\ U\ Fp$:

$$\{p, q, p \wedge q, Fp\} \quad \{p \wedge q, Fp, \varphi\}$$

Hintikka sequences

- A pre-Hinttika sequence for φ is a sequence $\alpha_0\alpha_1\alpha_2\ldots$ of subsets of $cl(\varphi)$ satisfying the following conditions for every $i \geq 0$:
  - For every $X\psi \in cl(\varphi)$:
    $X\psi \in \alpha_i$ iff $\psi \in \alpha_{i+1}$
  - For every $\psi_1 U\,\psi_2 \in cl(\varphi)$:
    $\psi_1 U\,\psi_2 \in \alpha_i$ iff $\psi_2 \in \alpha_i$ or $\psi_1 \in \alpha_i$ and $\psi_1\,U\,\psi_2 \in \alpha_{i+1}$

Hintikka sequences

- A pre-Hinttika sequence for φ is a sequence $\alpha_0\alpha_1\alpha_2 \ldots$ of subsets of $cl(\varphi)$ satisfying the following conditions for every $i \geq 0$:
  - For every $\mathrm{X}\psi \in cl(\varphi)$:
    $\mathrm{X}\psi \in \alpha_i$ iff $\psi \in \alpha_{i+1}$
  - For every $\psi_1 U\, \psi_2 \in cl(\varphi)$:
    $\psi_1 U\, \psi_2 \in \alpha_i$ iff $\psi_2 \in \alpha_i$ or $\psi_1 \in \alpha_i$ and $\psi_1\, U\, \psi_2 \in \alpha_{i+1}$
- A pre-Hinttika sequence is a Hinttika sequence if it also satisfies for every $i \geq 0$:
  - For every $\psi_1 U\, \psi_2 \in cl(\varphi)$: if $\psi_1 U\, \psi_2 \in \alpha_i$ then there exists $j \geq i$ such that $\psi_2 \in \alpha_j$

Hintikka sequences: An example

- Let $\varphi = \neg(p \wedge q)\ U\ (r \wedge s)$. Which of the following are pre-Hintikka and Hintikka sequences ?

Hintikka sequences: An example

- Let $\varphi = \neg(p \wedge q)\ U\ (r \wedge s)$. Which of the following are pre-Hintikka and Hintikka sequences ?
    1. $\{p, \neg q, r, s, \varphi\}^{\omega}$

Hintikka sequences: An example

- Let $\varphi = \neg(p \wedge q)\ U\ (r \wedge s)$. Which of the following are pre-Hintikka and Hintikka sequences ?

  1. $\{p, \neg q, r, s, \varphi\}^{\omega}$

  2. $\{\neg p, r, \neg\varphi\}^{\omega}$

Hintikka sequences: An example

- Let $\varphi = \neg(p \wedge q)\ U\ (r \wedge s)$. Which of the following are pre-Hintikka and Hintikka sequences ?
  1. $\{p, \neg q, r, s, \varphi\}^\omega$
  2. $\{\neg p, r, \neg\varphi\}^\omega$
  3. $\{\neg p, q, \neg r, (r \wedge s), \neg\varphi\}^\omega$

Hintikka sequences: An example

- Let $\varphi = \neg(p \wedge q)\ U\ (r \wedge s)$. Which of the following are pre-Hintikka and Hintikka sequences ?

  1. $\{p, \neg q, r, s, \varphi\}^{\omega}$
  2. $\{\neg p, r, \neg\varphi\}^{\omega}$
  3. $\{\neg p, q, \neg r, (r \wedge s), \neg\varphi\}^{\omega}$
  4. $\{p, q, (p \wedge q), r, s, (r \wedge s), \neg\varphi\}$

Hintikka sequences: An example

- Let $\varphi = \neg(p \land q)\ U\ (r \land s)$. Which of the following are pre-Hintikka and Hintikka sequences ?
  1. $\{p, \neg q, r, s, \varphi\}^{\omega}$
  2. $\{\neg p, r, \neg\varphi\}^{\omega}$
  3. $\{\neg p, q, \neg r, (r \land s), \neg\varphi\}^{\omega}$
  4. $\{p, q, (p \land q), r, s, (r \land s), \neg\varphi\}$
  5. $\{p, \neg q, \neg(p \land q), \neg r, s, \neg(r \land s), \varphi\}^{\omega}$

Hintikka sequences: An example

- Let $\varphi = \neg(p \wedge q)\ U\ (r \wedge s)$. Which of the following are pre-Hintikka and Hintikka sequences ?

1. $\{p, \neg q, r, s, \varphi\}^\omega$
2. $\{\neg p, r, \neg \varphi\}^\omega$
3. $\{\neg p, q, \neg r, (r \wedge s), \neg \varphi\}^\omega$
4. $\{p, q, (p \wedge q), r, s, (r \wedge s), \neg \varphi\}$
5. $\{p, \neg q, \neg(p \wedge q), \neg r, s, \neg(r \wedge s), \varphi\}^\omega$
6. $\{p, q, (p \wedge q), r, s, (r \wedge s)\ \varphi\}^\omega$

Main theorem

- Definition: A Hintikka sequence $\alpha_0\alpha_1\alpha_2 \ldots$ extends a computation $s_0s_1s_2 \ldots$ if $s_i \cap cl(\varphi) = \alpha_i \cap AP$ for every $i \geq 0$.
- Theorem: Every computation $s_0s_1s_2 \ldots$ can be extended to a unique Hintikka sequence, and this extension is equal to the satisfaction sequence.

Strategy for the NGA of a formula

- Let σ be a computation over AP.

Strategy for the NGA of a formula

- Let σ be a computation over AP.
- We have: $\sigma \models \varphi$

iff φ belongs to the first set of the satisfaction sequence for σ

iff φ belongs to the first set of the Hintikka sequence for σ

Strategy for the NGA of a formula

- Let σ be a computation over AP.
- We have: $\sigma \models \varphi$

  iff φ belongs to the first set of the satisfaction sequence for σ

  iff φ belongs to the first set of the Hintikka sequence for σ

- Strategy: design the NGA so that for every σ
  - The runs on σ correspond to the pre-Hintikka sequences $\alpha_0 \alpha_1 \alpha_2 \ldots$ that extend σ and satisfy $\varphi \in \alpha_0$
  - A run is accepting iff its corresponding pre-Hintikka sequence is also a Hintikka sequence.

The NGA A_φ

The NGA A_φ

- Alphabet: 2^{AP}

The NGA A_φ

- Alphabet: 2^{AP}
- States: atoms of φ.

The NGA A_φ

- Alphabet: 2^{AP}
- States: atoms of φ.
- Initial states: atoms containing φ.

The NGA A_φ

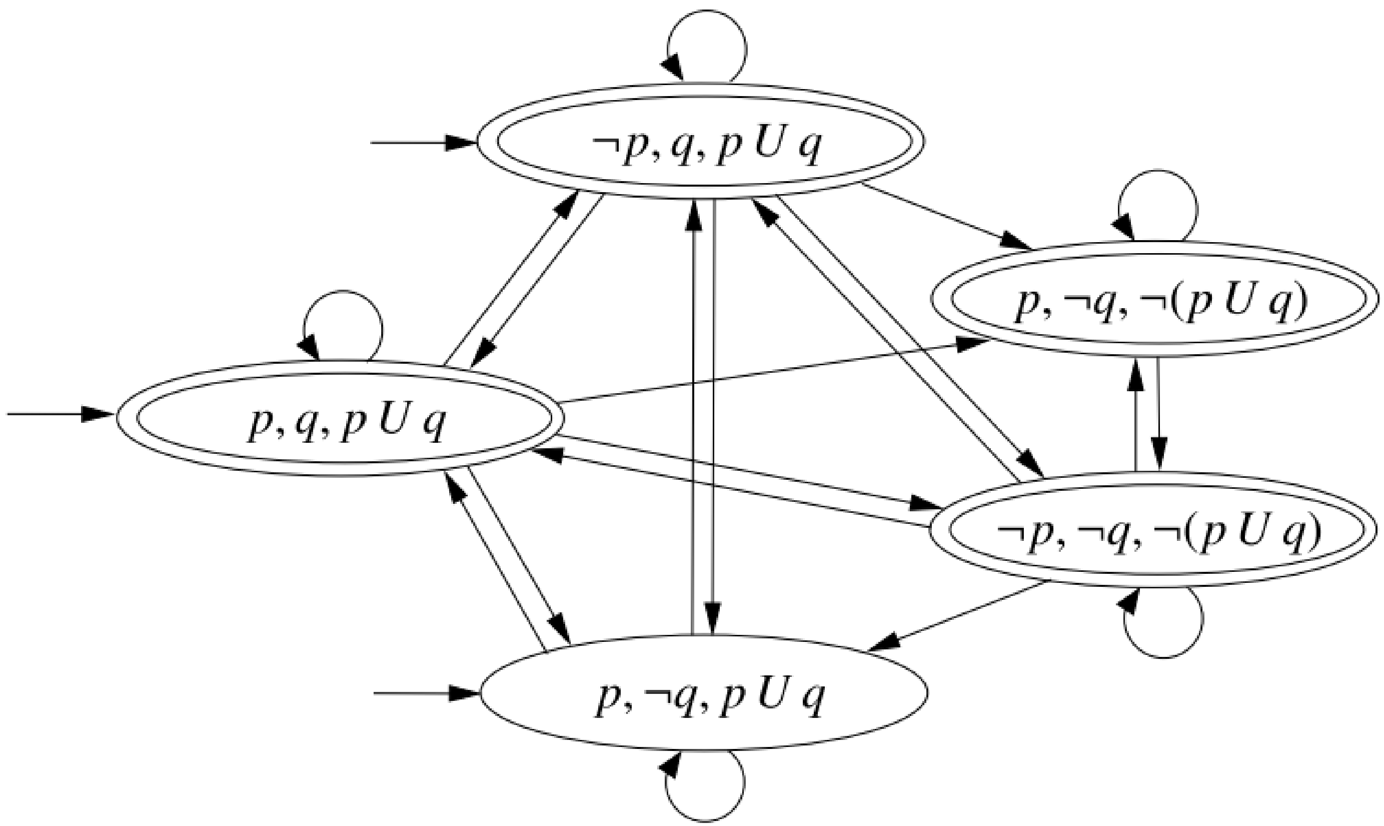
- Alphabet: 2^{AP}
- States: atoms of φ.
- Initial states: atoms containing φ.
- Transitions: triples $\alpha \xrightarrow{s} \beta$ such that $\alpha \cap AP = s$ and α, β satisfies the conditions of a pre-Hintikka sequence.

The NGA A_φ

- Alphabet: 2^{AP}
- States: atoms of φ.
- Initial states: atoms containing φ.
- Transitions: triples $\alpha \xrightarrow{s} \beta$ such that $\alpha \cap AP = s$ and α, β satisfies the conditions of a pre-Hintikka sequence.
- Sets of accepting states: A set $F_{\psi_1 U \psi_2}$ for every until-subformula $\psi_1 U \psi_2$ of φ.

  $F_{\psi_1 U \psi_2}$ contains the atoms α such that $\psi_1 U \psi_2 \notin \alpha$ or $\psi_2 \in \alpha$.

Example: The NGA $A_{p\ U\ q}$



(Labels of transitions omitted. The label of a transition from atom α is the set $\{p \in AP \mid p \in \alpha\}$. There is only one set of accepting states.)

Some observations

- All transitions leaving a state carry the same label.
- For every computation $s_0 s_1 s_2 \ldots$ satisfying φ there is a unique accepting run $\alpha_0 \xrightarrow{s_0} \alpha_1 \xrightarrow{s_1} \alpha_2 \xrightarrow{s_2} \cdots$, namely the one such that $\alpha_0 \alpha_1 \alpha_2 \ldots$ is the satisfaction sequence for $s_0 s_1 s_2 \ldots$.
- The sets of computations accepted from each initial state are pairwise disjoint.
- The number of states is bounded by $2^{|\varphi|}$.