From Modal μ-Calculus to Alternating Tree Automata using Parity Games

Automata, Logics and Infinite games

M.Fareed Arif

TU Dresden

Sept. 09, 2010

Outline

- Preliminaries
- Model Checking
- μ-*Calculus*
- Syntax & Semantics
- Kripke Model $(\mathcal{K})$
- Correspondance:($L_\mu \equiv L_\mathcal{A}$)
- $\varphi \in L_\mu \rightsquigarrow \mathcal{A}(\varphi)$
- $L_{\mathcal{A}(\varphi)} \rightsquigarrow \mathcal{G} = (\mathcal{A}(\varphi), \mathcal{K}, s_I)$
- Completeness Proof
- Complexity Issues
- Solving Parity by reduction to SAT
- Appendix

Motivation

- Complexity Results.
- Model Checking Problem Reduction to Acceptance Problem.
- Satisfiability in μ-*Calculus* to Emptiness Problem.
- Easy Completeness Proof.
- The advantage of the automaton model is its ability to deal with arbitrary branching in a much simpler way as compare to the one proposed by Janin and Walukiewicz [Panagiotis 2000].

Preliminaries

A *Transitionsystem* (i.e., *KripkeStructure*) over $\mathcal{P}$ is a tripe $\mathcal{K} = (S, R, \lambda)$ where

Kripke Structure $\mathcal{K}$:

- $\mathcal{P}$ be a set of atomic propositions (properties) and for any propositional interpretation $\mathcal{I} : \mathcal{P} \rightarrow \{true, false\}$.
- S is a set called **states** (worlds), universe of $\mathcal{K}$,
- $R \subseteq S \times S$ is a transition relation and
- $\lambda : S \rightarrow 2^{\mathcal{P}}$ is a mapping (i.e., $\lambda(s_i) = p_i$ for every $p_i \in \mathcal{P}$). $\lambda(s_i) = p_i$ if p_i is true in s_i and $\neg p_i$ if p_i is false in s_i.

$\lambda : S \rightarrow 2^{\mathcal{P}}$ regards transition systems as labeled directed graphs. For every $s \in S$, we denote

$$sR = \{s' \in S | (s, s') \in R\}, Rs = \{s' \in S | (s', s) \in R\}$$

Alternating Tree Automaton

An alternating tree automata is a device which accepts or rejects pointed transition systems by parsing the paths.

Alternating Tree Automata($\mathcal{A}$):

An alternating tree is a tuple $\mathcal{A} = \{Q, q_I, \delta, \Omega\}$ where

- Q is a finite set of states of the automaton,
- $q_I \in Q$ is a state called the initial state,
- $\delta : Q \rightarrow TC^Q$ is a transition function which maps every state $q \in Q$ to a transition condition TC where all the transition conditions TC over Q are defined by:
  - 0 and 1 are transition conditions over Q.
  - $p, \neg p$ are transition conditions over Q, for every $p \in \mathcal{P}$.
  - $q, \Box q, \Diamond q$ are transition conditions over Q, for every $q \in Q$.
  - $q_1 \wedge q_2, q_1 \vee q_2$ are transition conditions over Q, for every $q_1, q_2 \in Q$.
- $\Omega : Q \rightarrow \omega$ is called priority function (coloring function) which assigns color to states of $\mathcal{A}$.

Alternating Tree Automata

Word Problem

The word problem is to decide whether a given alternating tree automaton $\mathcal{A} = \{Q, q_I, \delta, \Omega\}$ accepts a *given* finite pointed transition system $(\mathcal{K}, s_I)$.

Emptiness Problem

The emptiness problem is to show that an alternating tree automaton $\mathcal{A} = \{Q, q_I, \delta, \Omega\}$ accepts if $\mathcal{A}$ accepts at least one transition system (i.e., $\mathcal{K}$).

Parity Games

A parity game is infinite two-person games on directed graphs along with winning play strategies for a certain player [Rene 2002].

Game $\mathcal{G}$:

A game is composed of an arena and a winding condition. Let $\mathcal{A}$ be an arena then the pair $\mathcal{G} = (\mathcal{A}, Win)$ is called a **game** where $Win \subseteq V^{\omega}$ is a winning set where ω is infinite supply of intergers.

Arena $\mathcal{A}$:

An **arena** is a triple

$$\mathcal{A} = (V_0, V_1, E)$$

Parity Games

Play π:

We define a play in the arena $\mathcal{A}$ as followed:

- a finite play $\pi = v_0 v_1 \ldots v_l \in V^+$ with $v_{i+1} \in v_i E$ for all $i < l$ and $v_l E = \emptyset$ represents a dead-end, a prefix of this finite play is $\rho(\pi) = v_0 v_1 \ldots v_k$ for $k \leq l$.
- an infinite play $\pi = v_0 v_1 \ldots v_l \in V^\omega$ with $v_{i+1} \in v_i E$ for all i $\in \omega$, a prefix for this infinite play is $\rho(\pi) = v_0 v_1 \ldots v_k$ for $k \geq 0$

Parity Games

Winning Set

To define the winning conditions for Players (Player 0, Player 1) are as followed:

Player σ is declared the winner of a play π in the game $\mathcal{G}$ iff

- π is a finite play $\pi = v_0 v_1 \ldots v_l \in V^+$ and v_l is a $\bar{\sigma}$-Vertex where Player $\bar{\sigma}$ can not move anymore (i.e., v_l is a dead-end, $v_l E = \emptyset$) or
- π is an infinite play and $\pi \in Win$.

Conversely, Player $\bar{\sigma}$ wins play π if Player σ does not win π.

Parity Games

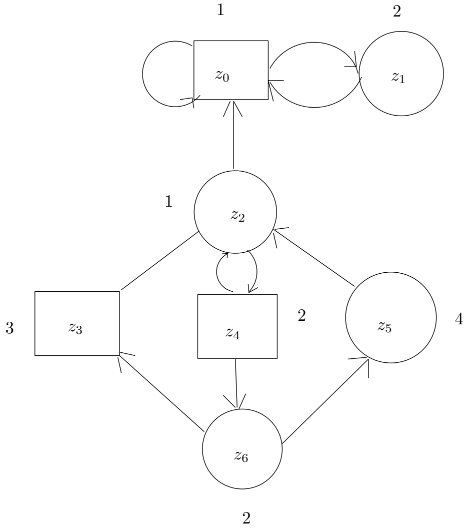
Coloring Function

The coloring function $\chi : V \rightarrow C$ color vertices of arena $\mathcal{A}$ where C is a finite set of colors (priorities)(i.e, $C \subseteq \mathbb{N}$) and it extends to an infinite play $\pi = v_0 v_1 \ldots$ as $\chi(\pi) = \chi(v_0)\chi(v_1)\ldots$.

Let *Win* is an acceptance condition for an automaton then $W_\chi(Win)$ is the winning set consisting of all infinite plays π where $\chi(\pi)$ is accepted according to *Win*.

- Parity conditions or Colour set C is a finite subset of integers and $Inf(\chi(\pi))$ be the set of colors that occurs infinitely often in $\chi(\pi)$ then for,
- Max-parity condition: $\pi \in W_\chi(Win)$ iff $max(Inf(\chi(\pi)))$ is even.
- Min-parity condition: $\pi \in W_\chi(Win)$ iff $min(Inf(\chi(\pi)))$ is even.

Example Game



Parity Games

Example:

Let $(G) = (\mathcal{A}, Win)$ where $\mathcal{A} = (V_0, V_1, E)$ such that $V_0 = \{z_1, z_2, z_5, z_5\}$ (circles), $V_1 = \{z_0, z_3, z_4\}$ (squares), Coloring set $C = \{1, 2, 3, 4\}$ and $\chi(z_4) = 2$ as shows in figure; winning set of condition $Win = \{\{1, 2\}, \{1, 2, 3, 4\}\}$. In a possible infinite play in this is $\pi = z_6 z_3 z_2 z_4 z_2 z_4 z_6 z z_5 (z_2 z_4)^\omega$. According to Muller acceptance condition (i.e., $\pi \in W_\chi(Win)$ iff $Inf(\chi(\pi)) \in \mathcal{A}$) this play π is winning for Player 0 because $\chi(\pi) = 23121224(12)^\omega$ where $Inf(\chi(\pi)) = \{1, 2\} \in Win$. For play $\pi' = (z_2 z_4 z_6 z_3)^\omega$ yields $\chi(\pi') = (1223)^\omega$ and $Inf(\chi(\pi')) = \{1, 2, 3\} \notin Win$, hence π' is winning for Player 1. Regarding parity conditions this play is a loss for player 0 because $min(Inf(\chi(\pi)) = \{1\})$ is odd., hence a win for the opponent.

Modal μ-Calculus L_μ

The set L_μ is a set of inductively defined modal μ-Calculus formulas:

Syntax of μ-Calculus L_μ

- $\bot, \top \in L_\mu$.
- For every atomic proposition $p \in \mathcal{P}$; $p, \neg p \in L_\mu$.
- If $\varphi, \psi \in L_\mu$, then $\varphi \circ \psi \in L_\mu$ where $\circ \in \{\vee, \wedge\}$.
- If $\varphi \in L_\mu$, then $\Box\varphi, \Diamond\varphi \in L_\mu$.
- If $p \in \mathcal{P}$, $\varphi \in L_\mu$, and p occurs only positively in φ then $\mu p\varphi, \nu p\varphi \in L_\mu$.

Modal μ-Calculus L_μ

The set L_μ is a set of inductively defined modal μ-Calculus formulas:

Syntax of μ-Calculus L_μ

- $\bot, \top \in L_\mu$.
- For every atomic proposition $p \in \mathcal{P}$; $p, \neg p \in L_\mu$.
- If $\varphi, \psi \in L_\mu$, then $\varphi \circ \psi \in L_\mu$ where $\circ \in \{\vee, \wedge\}$.
- If $\varphi \in L_\mu$, then $\Box\varphi, \Diamond\varphi \in L_\mu$.
- If $p \in \mathcal{P}$, $\varphi \in L_\mu$, and p occurs only positively in φ then $\mu p \varphi, \nu p \varphi \in L_\mu$.

e.g a:- $(\varphi \in L_\mu)$

$\varphi = \nu p_1(\mu p_2(p \vee \Diamond p_2) \wedge \Box p_1)$.

Modal μ-Calculus L_μ

The set L_μ is a set of inductively defined modal μ-Calculus formulas:

Syntax of μ-Calculus L_μ

- $\bot, \top \in L_\mu$.
- For every atomic proposition $p \in \mathcal{P}$; $p, \neg p \in L_\mu$.
- If $\varphi, \psi \in L_\mu$, then $\varphi \circ \psi \in L_\mu$ where $\circ \in \{\vee, \wedge\}$.
- If $\varphi \in L_\mu$, then $\Box\varphi, \Diamond\varphi \in L_\mu$.
- If $p \in \mathcal{P}$, $\varphi \in L_\mu$, and p occurs only positively in φ then $\mu p \varphi, \nu p \varphi \in L_\mu$.

e.g a:- $(\varphi \in L_\mu)$

$\varphi = \nu p_1(\mu p_2(p \vee \Diamond p_2) \wedge \Box p_1)$.

e.g b:- $(\psi, \varphi \in L_\mu)$

$\psi = \mu p_1((p_2 \wedge p_0) \vee p_1)$

Modal μ-Calculus L_μ

The set L_μ is a set of inductively defined modal μ-Calculus formulas:

Syntax of μ-Calculus L_μ

- $\bot, \top \in L_\mu$.
- For every atomic proposition $p \in \mathcal{P}$; $p, \neg p \in L_\mu$.
- If $\varphi, \psi \in L_\mu$, then $\varphi \circ \psi \in L_\mu$ where $\circ \in \{\vee, \wedge\}$.
- If $\varphi \in L_\mu$, then $\Box\varphi, \Diamond\varphi \in L_\mu$.
- If $p \in \mathcal{P}$, $\varphi \in L_\mu$, and p occurs only positively in φ then $\mu p \varphi, \nu p \varphi \in L_\mu$.

e.g a:- ($\varphi \in L_\mu$)

$\varphi = \nu p_1(\mu p_2(p \vee \Diamond p_2) \wedge \Box p_1)$.

e.g b:- ($\psi, \varphi \in L_\mu$)

$\psi = \mu p_1((p_2 \wedge p_0) \vee p_1)$

$\varphi = \nu p_2(\Diamond\psi)$

Model Checking

Model checking in μ-Calculus

Given a finite rooted Kripke structure $(\mathcal{K}, s_I)$ and an L_μ formula φ, determine whether $(\mathcal{K}, s_I) \models \varphi$.

Satisfiability in μ-Calculus

Given an L_μ formula φ, determine whether there *exists* a pointed Kripke structure $(\mathcal{K}, s_I)$ such that $\mathcal{K} \models \varphi$.

Modal μ-Calculus L_μ

The set $free(\varphi)$ of free variables of an L_μ formula φ is defined inductively as follows:

Free variable set for ($\varphi \in L_\mu$)

- $free(\bot) = free(\top) = \emptyset$,
- $free(p) = free(\neg p) = \{p\}$,
- $free(\varphi \circ \psi) = free(\varphi) \cup free(\psi)\}$ where $\circ \in \{\wedge, \vee\}$,
- $free(\Box\varphi) = free(\Diamond\varphi) = free(\varphi)$,
- $free(\mu p\varphi) = free(\nu p\varphi) = free(\varphi)/\{p\}$.

Modal μ-Calculus L_μ

The set $free(\varphi)$ of free variables of an L_μ formula φ is defined inductively as follows:

Free variable set for $(\varphi \in L_\mu)$

- $free(\bot) = free(\top) = \emptyset$,
- $free(p) = free(\neg p) = \{p\}$,
- $free(\varphi \circ \psi) = free(\varphi) \cup free(\psi)\}$ where $\circ \in \{\wedge, \vee\}$,
- $free(\Box\varphi) = free(\Diamond\varphi) = free(\varphi)$,
- $free(\mu p\varphi) = free(\nu p\varphi) = free(\varphi)/\{p\}$.

e.g a:- $(\varphi \in L_\mu)$

$\varphi = \nu p_1(\mu p_2(p \vee \Diamond p_2) \wedge \Box p_1)$.

Modal μ-Calculus L_μ

For every formula $\varphi \in L_\mu$, subformula of φ defined as follows:

Subformula(s) of ($\varphi \in L_\mu$)

- φ is subformula of $\varphi \in L_\mu$,
- φ, ψ is subformula of $\varphi \circ \psi \in L_\mu$ where $\circ \in \{\wedge, \vee\}$,
- φ is subformula of $\Box\varphi, \Diamond\varphi, \mu p\varphi, \nu p\varphi \in L_\mu$.

Modal μ-Calculus L_μ

A *Transitionsystem* (i.e., *KripkeStructure*) over $\mathcal{P}$ is a tripe $\mathcal{K} = (S, R, \lambda)$ where

Kripke Structure $\mathcal{K}$:

- $\mathcal{P}$ be a set of atomic propositions (properties) and for any propositional interpretation $\mathcal{I} : \mathcal{P} \rightarrow \{true, false\}$.
- S is a set called **states** (worlds), universe of $\mathcal{K}$,
- $R \subseteq S \times S$ is a transition relation and
- $\lambda : S \rightarrow 2^{\mathcal{P}}$ is a mapping (i.e., $\lambda(s_i) = p_i$ for every $p_i \in \mathcal{P}$). $\lambda(s_i) = p_i$ if p_i is true in s_i and $\neg p_i$ if p_i is false in s_i.

$\lambda : S \rightarrow 2^{\mathcal{P}}$ regards transition systems as labeled directed graphs. For every $s \in S$, we denote

$$sR = \{s' \in S | (s, s') \in R\}, Rs = \{s' \in S | (s', s) \in R\}$$

Modal μ-Calculus L_μ

The formulas of modal μ-Calculus are interpreted in Kripke structures $\mathcal{K}$ such that for every Kripke structure $\mathcal{K}$ and every $\varphi \in L_\mu$, where $\kappa : \mathcal{P} \rightarrow 2^S$ is defined as:

Semantics of μ-Calculus L_μ

- $||\bot||_\mathcal{K} = \emptyset$, $||\top||_\mathcal{K} = S$,
- $||p||_\mathcal{K} = \kappa(p)$, $||\neg p||_\mathcal{K} = S/\kappa(p)$,
- $||\varphi_1 \vee \varphi_2||_\mathcal{K} = ||\varphi_1||_\mathcal{K} \cup ||\varphi_2||_\mathcal{K}$,
- $||\varphi_1 \wedge \varphi_2||_\mathcal{K} = ||\varphi_1||_\mathcal{K} \cap ||\varphi_2||_\mathcal{K}$,
- $||\Box\varphi||_\mathcal{K} = \{s \in S | sR \subseteq ||\varphi||_\mathcal{K}\}$,
- $||\Diamond\varphi||_\mathcal{K} = \{s \in S | sR \cap ||\varphi||_\mathcal{K} \neq \emptyset\}$.

Modal μ-Calculus L_μ

Semantics of Fixed-point operators (i.e., μ, ν)

For a set of states $S' \subseteq S$ and $\mathcal{K}[p \mapsto S']$ denoted as followed:

$$\mathcal{K}[p \mapsto S'] = (S, E, \kappa[p \mapsto S'])$$

where $\kappa[p \mapsto S']$ is given as followed:

$$\mathcal{K}[p \mapsto S']p' = \begin{cases} S' & \text{if } p' = p \\ \kappa(p) & \text{if } p' \neq p \end{cases}$$

Modal μ-Calculus L_μ

The semantics of the fixed-point operators is now defined as:

Semantics of μ-Calculus L_μ

- $||\mu p \varphi||_{\mathcal{K}} = \bigcap\{S' \subseteq S| \ \ ||\varphi||_{\mathcal{K}[p \mapsto S']} \subseteq S'\}$
- $||\nu p \varphi||_{\mathcal{K}} = \bigcup\{S' \subseteq S| \ \ ||\varphi||_{\mathcal{K}[p \mapsto S']} \supseteq S'\}$

Modal μ-Calculus L_μ

Pointed transition system $(\mathcal{K}, s_I)$

A *pointed transition system* (i.e, a *rooted kripke structure*) is a pair $(\mathcal{K}, s_I)$ in a transition system $\mathcal{K} = (S, R, \kappa)$ with an initial state $s_I \in S$ [Daniel 2002].

- A pointed Kripke structure $(\mathcal{K}, s_I)$ is a model of $\varphi \in L_\mu$, denoted by $\mathcal{K} \models \varphi$ if $s_I \in ||\varphi||_{\mathcal{K}}$.
- Aditionally, $\varphi \equiv \psi$ if for all Kripke models $(\mathcal{K}, s_I)$, we have $(\mathcal{K}, s_I) \models \varphi$ iff $(\mathcal{K}, s_I) \models \psi$.

Correspondance:($L_\mu \leftrightsquigarrow L_\mathcal{A}$)

Theorem:

Let φ be an arbitrary L_μ formula. Then φ and $\mathcal{A}(\varphi)$ are equivalent, that is:

$$||\varphi|| = ||\mathcal{A}(\varphi)||.$$

Correspondance:($L_\mu \equiv L_{\mathcal{A}}$)

Consider an alternating tree automaton $\mathcal{A} = (Q, q_I, \delta, \Omega)$ and a $\varphi \in L_\mu$ formula then alternating tree automaton $\mathcal{A}(\varphi)$ is defined by:

Translation ($\varphi \in L_\mu \rightsquigarrow \mathcal{A}(\varphi)$)

- Q is the set which contains for each subformula ψ of φ (including φ itself), a state denoted by $\langle\psi\rangle$,
- the initial state is given by $q_I = \langle\varphi\rangle$.
- $\delta(\langle\bot\rangle) = 0$,
- $\delta(\langle\top\rangle) = 1$,
- $\delta(\langle p\rangle) = \begin{cases} p & \textit{if } p \in \textit{free}(\varphi) \\ \langle\varphi_p\rangle & \textit{if } p \notin \textit{free}(\varphi) \end{cases}$
- $\delta(\langle\neg p\rangle) = \neg p$,
- $\delta(\langle\psi_1 \wedge \psi_2\rangle) = \langle\psi_1\rangle \wedge \langle\psi_2\rangle$, $\delta(\langle\psi_1 \vee \langle\psi_2\rangle) = \langle\psi_1\rangle \vee \langle\psi_2\rangle$,

Correspondance:($L_\mu \equiv L_{\mathcal{A}}$)

Translation ($\varphi \in L_\mu \rightsquigarrow \mathcal{A}(\varphi)$)

- $\delta(\langle \Diamond\psi \rangle) = \Diamond\langle\psi\rangle$,
- $\delta(\langle \Box\psi \rangle) = \Box\langle\psi\rangle$,
- $\delta(\langle \mu p\psi \rangle) = \langle\psi\rangle$,
- $\delta(\langle \nu p\psi \rangle) = \langle\psi\rangle$.

Ω

- $\Omega =$ The smallest odd number greater or equal to $\alpha(\psi)$ - 1 where $\psi \in \mathcal{F}_\mu$,
- $\Omega =$ The smallest even number greater or equal to $\alpha(\psi)$ - 1 where $\psi \in \mathcal{F}_\nu$,
- $\Omega = 0$ for $\psi \notin \mathcal{F}_\nu$.

$L_{\mathcal{A}(\varphi)} \rightsquigarrow \mathcal{G} = (\mathcal{A}(\varphi), \mathcal{K}, s_I)$

$\mathcal{G} = (\mathcal{A}(\varphi), \mathcal{K}, s_I)$

A vertex $v = (\langle\psi\rangle, s)$ belongs to Player 0 iff

- $\psi = \perp$,
- $\psi = p$, $p \in free(\varphi)$, $s \notin \kappa(p)$,
- $\psi = \neg p$, $p \in free(\varphi)$, $s \in \kappa(p)$,
- $\psi = p$, $p \notin free(\varphi)$,
- $\psi = \eta p \psi'$ where $\eta \in \{\mu, \nu\}$,
- $\psi = \psi_1 \vee \psi_2$ for some $\psi_1, \psi_2 \in L_\mu$,
- $\psi = \Diamond\psi'$.

$L_{\mathcal{A}(\varphi)} \rightsquigarrow \mathcal{G} = (\mathcal{A}(\varphi), \mathcal{K}, s_I)$

$\mathcal{G} = (\mathcal{A}(\varphi), \mathcal{K}, s_I)$

A vertex $v = (\langle\psi\rangle, s)$ belongs to Player 1 iff

- $\psi = \top$,
- $\psi = p,\ p \in \mathit{free}(\varphi),\ s \in \lambda(s)$,
- $\psi = \neg p,\ p \in \mathit{free}(\varphi),\ s \notin \lambda(s)$,
- $\psi = p,\ p \in \mathit{free}(\varphi)$,
- $\psi = \psi_1 \wedge \psi_2$ for some $\psi_1, \psi_2 \in L_\mu$,
- $\psi = \Box\psi'$.

In parity Game $\mathcal{G}$, the edge relation $E^{\mathcal{G}}$ is defined as:

$$E^{\mathcal{G}} = \begin{cases} \{(\langle\psi'\rangle, s) | \langle\psi'\rangle \in \delta(\langle\psi\rangle)\} & \mathit{if}\ \psi \neq \Diamond\psi', \Box\psi' \\ \{(\langle\psi'\rangle, s') | \langle\psi'\rangle \in \delta(\langle\psi\rangle), s' \in sR\} & \mathit{if}\ \psi = \Diamond\psi', \Box\psi' \end{cases}$$

For μ-formula the priority is odd and for a ν-formula priority is even.

Correspondance:($L_\mu \leftrightsquigarrow L_{\mathcal{A}}$)

Model Checking Problem Reduction to Acceptance Problem

The alternating tree automaton $\mathcal{B}$ accepts $(\mathcal{K}, s_I)$ if and only if Player 0 has a winning strategy in the parity game $\mathcal{G}(\mathcal{T}) = (\mathcal{K}, \mathcal{B}, s_I)$ (i.e., $\mathcal{T} = (V_0, V_1, q_I \times s_I, \Omega)$).

Satisfiability in μ-Calculus to Emptiness Problem

The automaton $\mathcal{C}$ accepts a pointed Kripke structure $(\mathcal{K}, s_I)$ if and only if Player 0 wins the game $\mathcal{T}$.

Completeness of Translation

Completeness Theorem:

Let φ be an arbitrary L_μ formula. Then for every pointed transition system $(\mathcal{K}, s)$ the following holds:

$$(\mathcal{K}, s) \models \varphi \ \texttt{iff} \ (\mathcal{K}, s) \in L(\mathcal{A}(\varphi))$$

Complexity Bounds

The Model-Checking problem for μ-Calculus, is solvable in time:

$$\mathcal{O}(ln(\frac{2nkn}{b})^{\lfloor b/2 \rfloor})$$

where k is the number of worlds of the Kripke structure, l is the size of accessability relation, n is the number of subformulas and b is the alternation depth.

Complexity Class:

The model checking is in $UP \cap co\text{-}UP$.

In complexity theory, UP ("Unambiguous Non-deterministic Polynomial-time") is the complexity class of decision problems solvable in polynomial time on a non-deterministic Turing machine with at most one accepting path for each input. UP contains P and is contained in NP.

Complexity Bounds

Complexity Class

The satisfiability of μ-Calculus is in *EXPTIME*.

Motivation

Solve parity games (encoding winning strategy) as a reduciton to SAT [Martin 2005].

- Comparision with other techniques for parity games like Omega, as well as model cherks for the modal μ-*calculus* like SMV.
- Reduction of Parity to decide fragment of SAT to check whether it can be proved in polynomial time.

THANKS

Completeness of Translation

Completeness Theorem:

Let φ be an arbitrary L_μ formula. Then for every pointed transition system $(\mathcal{K}, s)$ the following holds:

$$(\mathcal{K}, s) \models \varphi \ \texttt{iff} \ (\mathcal{K}, s) \in L(\mathcal{A}(\varphi))$$

Proof of Completeness

Proof:

? Case ($\varphi = \top$):

Proof of Completeness

Proof:

- ✔ Case ($\varphi = \top$): : Clearly, every Kripke structure $(\mathcal{K}, s_I)$ is a model of φ. Thus every pointed transition system is accepted by $\mathcal{A}(\varphi)$. The initial state of game $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is a $Vertex - 1$ and is dead-end. Hence, every game in this play is won by Player 0.

Proof of Completeness

Proof:

- ✔ Case ($\varphi = \top$): : Clearly, every Kripke structure $(\mathcal{K}, s_I)$ is a model of φ. Thus every pointed transition system is accepted by $\mathcal{A}(\varphi)$. The initial state of game $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is a $Vertex - 1$ and is dead-end. Hence, every game in this play is won by Player 0.
- ? Case ($\varphi = \bot$):

Proof of Completeness

Proof:

- ✔ Case ($\varphi = \top$): : Clearly, every Kripke structure $(\mathcal{K}, s_I)$ is a model of φ. Thus every pointed transition system is accepted by $\mathcal{A}(\varphi)$. The initial state of game $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is a $Vertex - 1$ and is dead-end. Hence, every game in this play is won by Player 0.
- ✔ Case ($\varphi = \bot$): : Then for the complement case $(\mathcal{K}, s) \not\models \bot$ and from proposition 1 it follows that automata $\mathcal{A}$ does not contain any succeeding run for $\varphi = \bot$ (i.e., $(\mathcal{K}, s) \notin L(\mathcal{A}(\varphi))$.

Proof of Completeness

Proof:

- ✔ Case ($\varphi = \top$): : Clearly, every Kripke structure $(\mathcal{K}, s_I)$ is a model of φ. Thus every pointed transition system is accepted by $\mathcal{A}(\varphi)$. The initial state of game $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is a $Vertex - 1$ and is dead-end. Hence, every game in this play is won by Player 0.
- ✔ Case ($\varphi = \bot$): : Then for the complement case $(\mathcal{K}, s) \not\models \bot$ and from proposition 1 it follows that automata $\mathcal{A}$ does not contain any succeeding run for $\varphi = \bot$ (i.e., $(\mathcal{K}, s) \notin L(\mathcal{A}(\varphi))$.
- ✔ Case ($\varphi = p$):

Proof of Completeness

Proof:

- ✔ Case ($\varphi = \top$): : Clearly, every Kripke structure $(\mathcal{K}, s_I)$ is a model of φ. Thus every pointed transition system is accepted by $\mathcal{A}(\varphi)$. The initial state of game $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is a $Vertex - 1$ and is dead-end. Hence, every game in this play is won by Player 0.
- ✔ Case ($\varphi = \bot$): : Then for the complement case $(\mathcal{K}, s) \not\models \bot$ and from proposition 1 it follows that automata $\mathcal{A}$ does not contain any succeeding run for $\varphi = \bot$ (i.e., $(\mathcal{K}, s) \notin L(\mathcal{A}(\varphi))$.
- ✔ Case ($\varphi = p$): : Let $(\mathcal{K}, s_I) \models \varphi$ if $s_I \in \kappa(p)$. Thus in $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is $vertex - 1$ and a deadend as well, therefore $(\mathcal{K}, s_I) \in L(\mathcal{A}(\varphi))$. Similarly, if $(\mathcal{K}, s_I) \not\models \varphi$ if $\kappa(p) \in s_I$ then we have $s_I \notin \kappa(p)$, thus $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is $vertex - 0$ and a dead-end. Therefore, $(\mathcal{K}, s_I) \notin L(\mathcal{A}(\varphi))$.

Proof of Completeness

Proof:

- ✔ Case ($\varphi = \top$): : Clearly, every Kripke structure $(\mathcal{K}, s_I)$ is a model of φ. Thus every pointed transition system is accepted by $\mathcal{A}(\varphi)$. The initial state of game $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is a $Vertex - 1$ and is dead-end. Hence, every game in this play is won by Player 0.
- ✔ Case ($\varphi = \bot$): : Then for the complement case $(\mathcal{K}, s) \not\models \bot$ and from proposition 1 it follows that automata $\mathcal{A}$ does not contain any succeeding run for $\varphi = \bot$ (i.e., $(\mathcal{K}, s) \notin L(\mathcal{A}(\varphi))$.
- ✔ Case ($\varphi = p$): : Let $(\mathcal{K}, s_I) \models \varphi$ if $s_I \in \kappa(p)$. Thus in $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is $vertex - 1$ and a deadend as well, therefore $(\mathcal{K}, s_I) \in L(\mathcal{A}(\varphi))$. Similarly, if $(\mathcal{K}, s_I) \not\models \varphi$ if $\kappa(p) \in s_I$ then we have $s_I \notin \kappa(p)$, thus $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is $vertex - 0$ and a dead-end. Therefore, $(\mathcal{K}, s_I) \notin L(\mathcal{A}(\varphi))$.
- ✔ Case ($\varphi = \neg p$):

Proof of Completeness

Proof:

- ✔ Case ($\varphi = \top$): : Clearly, every Kripke structure $(\mathcal{K}, s_I)$ is a model of φ. Thus every pointed transition system is accepted by $\mathcal{A}(\varphi)$. The initial state of game $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is a $Vertex - 1$ and is dead-end. Hence, every game in this play is won by Player 0.
- ✔ Case ($\varphi = \bot$): : Then for the complement case $(\mathcal{K}, s) \not\models \bot$ and from proposition 1 it follows that automata $\mathcal{A}$ does not contain any succeeding run for $\varphi = \bot$ (i.e., $(\mathcal{K}, s) \notin L(\mathcal{A}(\varphi))$.
- ✔ Case ($\varphi = p$): : Let $(\mathcal{K}, s_I) \models \varphi$ if $s_I \in \kappa(p)$. Thus in $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is $vertex - 1$ and a deadend as well, therefore $(\mathcal{K}, s_I) \in L(\mathcal{A}(\varphi))$. Similarly, if $(\mathcal{K}, s_I) \not\models \varphi$ if $\kappa(p) \in s_I$ then we have $s_I \notin \kappa(p)$, thus $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is $vertex - 0$ and a dead-end. Therefore, $(\mathcal{K}, s_I) \notin L(\mathcal{A}(\varphi))$.
- ✔ Case ($\varphi = \neg p$): : Similar to the previous case.

Proof of Completeness

Proof:

- ✔ Case ($\varphi = \top$): : Clearly, every Kripke structure $(\mathcal{K}, s_I)$ is a model of φ. Thus every pointed transition system is accepted by $\mathcal{A}(\varphi)$. The initial state of game $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is a $Vertex - 1$ and is dead-end. Hence, every game in this play is won by Player 0.
- ✔ Case ($\varphi = \bot$): : Then for the complement case $(\mathcal{K}, s) \not\models \bot$ and from proposition 1 it follows that automata $\mathcal{A}$ does not contain any succeeding run for $\varphi = \bot$ (i.e., $(\mathcal{K}, s) \notin L(\mathcal{A}(\varphi))$.
- ✔ Case ($\varphi = p$): : Let $(\mathcal{K}, s_I) \models \varphi$ if $s_I \in \kappa(p)$. Thus in $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is $vertex - 1$ and a deadend as well, therefore $(\mathcal{K}, s_I) \in L(\mathcal{A}(\varphi))$. Similarly, if $(\mathcal{K}, s_I) \not\models \varphi$ if $\kappa(p) \in s_I$ then we have $s_I \notin \kappa(p)$, thus $\mathcal{G}(\mathcal{A}(\varphi), \mathcal{K}, s_I)$ is $vertex - 0$ and a dead-end. Therefore, $(\mathcal{K}, s_I) \notin L(\mathcal{A}(\varphi))$.
- ✔ Case ($\varphi = \neg p$): : Similar to the previous case.

Proof of Completeness

Proof:

? Case ($\varphi = \psi_1 \wedge \psi_2$):

Proof of Completeness

Proof:

- Case $(\varphi = \psi_1 \wedge \psi_2)$: :

$$
\begin{aligned}
(\mathcal{K}, s_I) \models \psi_1 \wedge \psi_2 &= (\mathcal{K}, s_I) \models \psi_1 \wedge \psi_2 && (1)\\
&= s_I \in ||\psi_1||_{\mathcal{K}} \cap s_I \in ||\psi_2||_{\mathcal{K}} && (2)\\
&= (\mathcal{K}, s_I) \models \psi_1 \;\; \texttt{and} \;\; (\mathcal{K}, s_I) \models \psi_2 && (3)\\
&= (\mathcal{K}, s_I) \in L(\mathcal{A}(\psi_1)) \cap L(\mathcal{A}(\psi_2)) && (4)\\
&= (\mathcal{K}, s_I) \in L(\mathcal{A}(\psi_1 \wedge \psi_2)) && (5)
\end{aligned}
$$

Proof of Completeness

Proof:

- Case $(\varphi = \psi_1 \wedge \psi_2)$: :

$$
\begin{aligned}
(\mathcal{K}, s_I) \models \psi_1 \wedge \psi_2 &= (\mathcal{K}, s_I) \models \psi_1 \wedge \psi_2 && (1)\\
&= s_I \in ||\psi_1||_{\mathcal{K}} \cap s_I \in ||\psi_2||_{\mathcal{K}} && (2)\\
&= (\mathcal{K}, s_I) \models \psi_1 \ \texttt{and} \ (\mathcal{K}, s_I) \models \psi_2 && (3)\\
&= (\mathcal{K}, s_I) \in L(\mathcal{A}(\psi_1)) \cap L(\mathcal{A}(\psi_2)) && (4)\\
&= (\mathcal{K}, s_I) \in L(\mathcal{A}(\psi_1 \wedge \psi_2)) && (5)
\end{aligned}
$$

? Case $\varphi = \psi_1 \vee \psi_2$:

Proof of Completeness

Proof:

- ✔ Case $(\varphi = \psi_1 \wedge \psi_2)$: :

$$
\begin{aligned}
(\mathcal{K}, s_I) \models \psi_1 \wedge \psi_2 &= (\mathcal{K}, s_I) \models \psi_1 \wedge \psi_2 && (1)\\
&= s_I \in ||\psi_1||_{\mathcal{K}} \cap s_I \in ||\psi_2||_{\mathcal{K}} && (2)\\
&= (\mathcal{K}, s_I) \models \psi_1 \;\texttt{and}\; (\mathcal{K}, s_I) \models \psi_2 && (3)\\
&= (\mathcal{K}, s_I) \in L(\mathcal{A}(\psi_1)) \cap L(\mathcal{A}(\psi_2)) && (4)\\
&= (\mathcal{K}, s_I) \in L(\mathcal{A}(\psi_1 \wedge \psi_2)) && (5)
\end{aligned}
$$

- ✔ Case $\varphi = \psi_1 \vee \psi_2$: : Similar to the previous case, *lema*.4 [Julia 2002] is used instead.

Proof of Completeness

Proof:

- ✔ Case ($\varphi = \psi_1 \wedge \psi_2$): :

$$
\begin{aligned}
(\mathcal{K}, s_I) \models \psi_1 \wedge \psi_2 &= (\mathcal{K}, s_I) \models \psi_1 \wedge \psi_2 && (1)\\
&= s_I \in ||\psi_1||_{\mathcal{K}} \cap s_I \in ||\psi_2||_{\mathcal{K}} && (2)\\
&= (\mathcal{K}, s_I) \models \psi_1 \ \texttt{and} \ (\mathcal{K}, s_I) \models \psi_2 && (3)\\
&= (\mathcal{K}, s_I) \in L(\mathcal{A}(\psi_1)) \cap L(\mathcal{A}(\psi_2)) && (4)\\
&= (\mathcal{K}, s_I) \in L(\mathcal{A}(\psi_1 \wedge \psi_2)) && (5)
\end{aligned}
$$

- ✔ Case $\varphi = \psi_1 \vee \psi_2$: : Similar to the previous case, *lema*.4 [Julia 2002] is used instead.

Proof of Completeness

Proof:

? Case $\varphi = \Box\psi$:

Proof of Completeness

Proof:

- Case $\varphi = \Box\psi$: :

$$
\begin{aligned}
(\mathcal{K}, s_I) \models \Box\varphi &= s_I \in ||\Box\psi||_{\mathcal{K}} && (6)\\
&= sR \in ||\psi||_{\mathcal{K}} && (7)\\
&= \forall s' \in sR.(\mathcal{K}, s') \models \psi. && (8)\\
&= \forall s' \in sR.(\mathcal{K}, s') \in L(\mathcal{A} && (9)\\
&= (\mathcal{K}, s') \in L(\mathcal{A}(\Box\psi)). && (10)
\end{aligned}
$$

Proof of Completeness

Proof:

- Case $\varphi = \Box\psi$: :

$$
\begin{aligned}
(\mathcal{K}, s_I) \models \Box\varphi &= s_I \in ||\Box\psi||_{\mathcal{K}} && (6)\\
&= sR \in ||\psi||_{\mathcal{K}} && (7)\\
&= \forall s' \in sR.(\mathcal{K}, s') \models \psi. && (8)\\
&= \forall s' \in sR.(\mathcal{K}, s') \in L(\mathcal{A} && (9)\\
&= (\mathcal{K}, s') \in L(\mathcal{A}(\Box\psi)). && (10)
\end{aligned}
$$

? Case $\varphi = \mu p \psi$:

Proof of Completeness

Proof:

✔ Case $\varphi = \Box\psi$: :

$$\begin{aligned} (\mathcal{K}, s_I) \models \Box\varphi &= s_I \in ||\Box\psi||_{\mathcal{K}} && (6) \\ &= sR \in ||\psi||_{\mathcal{K}} && (7) \\ &= \forall s' \in sR.(\mathcal{K}, s') \models \psi. && (8) \\ &= \forall s' \in sR.(\mathcal{K}, s') \in L(\mathcal{A} && (9) \\ &= (\mathcal{K}, s') \in L(\mathcal{A}(\Box\psi)). && (10) \end{aligned}$$

✔ Case $\varphi = \mu p \psi$: :

$(\mathcal{K}, s) \in L(\mathcal{A}(\mu p \psi))$ `iff Player 0 wins the game` $\mathcal{G} = (\mathcal{A}(\mu p \psi), \mathcal{K}$

.

Proof of Completeness

Proof:

✔ Case $\varphi = \Box\psi$: :

$$
\begin{aligned}
(\mathcal{K}, s_I) \models \Box\varphi &= s_I \in ||\Box\psi||_{\mathcal{K}} && (6)\\
&= sR \in ||\psi||_{\mathcal{K}} && (7)\\
&= \forall s' \in sR.(\mathcal{K}, s') \models \psi. && (8)\\
&= \forall s' \in sR.(\mathcal{K}, s') \in L(\mathcal{A} && (9)\\
&= (\mathcal{K}, s') \in L(\mathcal{A}(\Box\psi)). && (10)
\end{aligned}
$$

✔ Case $\varphi = \mu p \psi$: :

$(\mathcal{K}, s) \in L(\mathcal{A}(\mu p \psi))$ `iff Player 0 wins the game` $\mathcal{G} = (\mathcal{A}(\mu p \psi), \mathcal{K}$

.

For Further Reading

Martin Lange. 2005
Solving Parity Games by reduction to SAT.

Julia Zappe. 2002
Automata, Logics, and Infinite Games: Lecture Notes in Computer Science, "Modal μ-Calculus and Alternating Tree Automata"

Julia Zappe. 2002
Automata, Logics, and Infinite Games: Lecture Notes in Computer Science, "Modal μ-Calculus and Alternating Tree Automata"

Rene Mazala. 2002
Automata, Logics, and Infinite Games: Lecture Notes in Computer Science, "Infinite Games"

Daniel Kirsten. 2002
Automata, Logics, and Infinite Games: Lecture Notes in Computer Science, "Alternating Tree Automata and Parity Games"

Thomas Wilke. 2002

Modal μ-Calculus L_μ

For an arbitrary formula $\varphi \in L_\mu$, its alternation depth $\alpha(\varphi) : L_\mu \to \mathbb{N}$ is function defined inductively:

Fixed point Alternation(Syntactic alternation depth)

- $\alpha(\bot) = \alpha(\top) = \alpha(p) = \alpha(\neg p) = 0$,
- $\alpha(\varphi \circ \psi) = max\{\alpha(\varphi), \alpha(\psi)\}$ where $\circ \in \{\wedge, \vee\}$,
- $\alpha(\Box\varphi) = \alpha(\Diamond\varphi) = \alpha(\varphi)$,
- $\alpha(\mu p \varphi) = max\{1, \alpha(\psi)\} \cup \{\alpha(\nu p' \psi') + 1 | \nu p' \psi' \leq \psi, p \in free(\nu p' \psi')\}$,
- $\alpha(\nu p \varphi) = max\{1, \alpha(\psi)\} \cup \{\alpha(\mu p' \psi') + 1 | \mu p' \psi' \leq \psi, p \in free(\mu p' \psi')\}$.

Modal μ-Calculus L_μ

For an arbitrary formula $\varphi \in L_\mu$, its alternation depth $\alpha(\varphi) : L_\mu \to \mathbb{N}$ is function defined inductively:

Fixed point Alternation(Syntactic alternation depth)

- $\alpha(\bot) = \alpha(\top) = \alpha(p) = \alpha(\neg p) = 0$,
- $\alpha(\varphi \circ \psi) = max\{\alpha(\varphi), \alpha(\psi)\}$ where $\circ \in \{\wedge, \vee\}$,
- $\alpha(\Box\varphi) = \alpha(\Diamond\varphi) = \alpha(\varphi)$,
- $\alpha(\mu p \varphi) = max\{1, \alpha(\psi)\} \cup \{\alpha(\nu p'\psi') + 1 | \nu p'\psi' \leq \psi, p \in free(\nu p'\psi')\}$,
- $\alpha(\nu p \varphi) = max\{1, \alpha(\psi)\} \cup \{\alpha(\mu p'\psi') + 1 | \mu p'\psi' \leq \psi, p \in free(\mu p'\psi')\}$.

e.g a:- $(\varphi \in L_\mu)$

$\varphi = \nu p_1(\mu p_2(p \vee \Diamond p_2) \wedge \Box p_1)$.

Modal μ-Calculus L_μ

For an arbitrary formula $\varphi \in L_\mu$, its alternation depth $\alpha(\varphi) : L_\mu \to \mathbb{N}$ is function defined inductively:

Fixed point Alternation(Syntactic alternation depth)

- $\alpha(\bot) = \alpha(\top) = \alpha(p) = \alpha(\neg p) = 0$,
- $\alpha(\varphi \circ \psi) = max\{\alpha(\varphi), \alpha(\psi)\}$ where $\circ \in \{\wedge, \vee\}$,
- $\alpha(\Box\varphi) = \alpha(\Diamond\varphi) = \alpha(\varphi)$,
- $\alpha(\mu p \varphi) = max\{1, \alpha(\psi)\} \cup \{\alpha(\nu p' \psi') + 1 | \nu p' \psi' \leq \psi, p \in free(\nu p' \psi')\}$,
- $\alpha(\nu p \varphi) = max\{1, \alpha(\psi)\} \cup \{\alpha(\mu p' \psi') + 1 | \mu p' \psi' \leq \psi, p \in free(\mu p' \psi')\}$.

e.g a:- ($\varphi \in L_\mu$)

$\varphi = \nu p_1(\mu p_2(p \vee \Diamond p_2) \wedge \Box p_1)$.

$\alpha(\varphi) = 1$.

Modal μ-Calculus L_μ

For an arbitrary formula $\varphi \in L_\mu$, its alternation depth $\alpha(\varphi) : L_\mu \to \mathbb{N}$ is function defined inductively:

Fixed point Alternation(Syntactic alternation depth)

- $\alpha(\bot) = \alpha(\top) = \alpha(p) = \alpha(\neg p) = 0$,
- $\alpha(\varphi \circ \psi) = max\{\alpha(\varphi), \alpha(\psi)\}$ where $\circ \in \{\wedge, \vee\}$,
- $\alpha(\Box\varphi) = \alpha(\Diamond\varphi) = \alpha(\varphi)$,
- $\alpha(\mu p \varphi) = max\{1, \alpha(\psi)\} \cup \{\alpha(\nu p' \psi') + 1 | \nu p' \psi' \leq \psi, p \in free(\nu p' \psi')\}$,
- $\alpha(\nu p \varphi) = max\{1, \alpha(\psi)\} \cup \{\alpha(\mu p' \psi') + 1 | \mu p' \psi' \leq \psi, p \in free(\mu p' \psi')\}$.

e.g a:- ($\varphi \in L_\mu$)

$\varphi = \nu p_1(\mu p_2(p \vee \Diamond p_2) \wedge \Box p_1)$.

$\alpha(\varphi) = 1$.

e.g b:- ($\psi, \varphi \in L_\mu$)

$\psi = \mu p_1(p_2 \wedge p_0) \vee p_1)$

Modal μ-Calculus L_μ

For an arbitrary formula $\varphi \in L_\mu$, its alternation depth $\alpha(\varphi) : L_\mu \to \mathbb{N}$ is function defined inductively:

Fixed point Alternation(Syntactic alternation depth)

- $\alpha(\bot) = \alpha(\top) = \alpha(p) = \alpha(\neg p) = 0$,
- $\alpha(\varphi \circ \psi) = max\{\alpha(\varphi), \alpha(\psi)\}$ where $\circ \in \{\wedge, \vee\}$,
- $\alpha(\Box\varphi) = \alpha(\Diamond\varphi) = \alpha(\varphi)$,
- $\alpha(\mu p \varphi) = max\{1, \alpha(\psi)\} \cup \{\alpha(\nu p'\psi') + 1 | \nu p'\psi' \leq \psi, p \in free(\nu p'\psi')\}$,
- $\alpha(\nu p \varphi) = max\{1, \alpha(\psi)\} \cup \{\alpha(\mu p'\psi') + 1 | \mu p'\psi' \leq \psi, p \in free(\mu p'\psi')\}$.

e.g a:- ($\varphi \in L_\mu$)

$\varphi = \nu p_1(\mu p_2(p \vee \Diamond p_2) \wedge \Box p_1)$.

$\alpha(\varphi) = 1$.

e.g b:- ($\psi, \varphi \in L_\mu$)

$\psi = \mu p_1(p_2 \wedge p_0) \vee p_1)$

$\alpha(\psi) = 1$.

Modal μ-Calculus L_μ

For an arbitrary formula $\varphi \in L_\mu$, its alternation depth $\alpha(\varphi) : L_\mu \to \mathbb{N}$ is function defined inductively:

Fixed point Alternation(Syntactic alternation depth)

- $\alpha(\bot) = \alpha(\top) = \alpha(p) = \alpha(\neg p) = 0$,
- $\alpha(\varphi \circ \psi) = max\{\alpha(\varphi), \alpha(\psi)\}$ where $\circ \in \{\wedge, \vee\}$,
- $\alpha(\Box\varphi) = \alpha(\Diamond\varphi) = \alpha(\varphi)$,
- $\alpha(\mu p \varphi) = max\{1, \alpha(\psi)\} \cup \{\alpha(\nu p' \psi') + 1 | \nu p' \psi' \leq \psi, p \in free(\nu p' \psi')\}$,
- $\alpha(\nu p \varphi) = max\{1, \alpha(\psi)\} \cup \{\alpha(\mu p' \psi') + 1 | \mu p' \psi' \leq \psi, p \in free(\mu p' \psi')\}$.

e.g a:- ($\varphi \in L_\mu$)

$\varphi = \nu p_1(\mu p_2(p \vee \Diamond p_2) \wedge \Box p_1)$.

$\alpha(\varphi) = 1$.

e.g b:- ($\psi, \varphi \in L_\mu$)

$\psi = \mu p_1(p_2 \wedge p_0) \vee p_1)$

$\alpha(\psi) = 1$.

$\varphi = \nu p_2(\Diamond\psi)$

$\alpha(\varphi) = 2$.

Modal μ-Calculus L_μ

Tarski-Knaster Fix-point Theorem

Let $f : L \rightarrow L$ be a monotonic function on a complete lattice $(L, \leq, \sqcup, \sqcap)$ then for $A = \{y | f(y) \leq y\}$, $x = \sqcap A$ is the *least fixed point* of f.

Example

Given a set S, the power set of S, (i.e., $\mathcal{P}(S)$) is $(\mathcal{P}(S), \subseteq)$ is a lattice. For a given set $A \subseteq \mathcal{P}(S)$ of subsets such that maximal set $\sqcup A = \bigcup_{S' \in A} S'$ and minimal set $\sqcap A = \bigcap_{S' \in A} S'$:

$$\sqcup A = \{ \bigcup_{S' \in A} S' | A \supseteq \mathcal{P}(S')\}$$

$$\sqcap A = \{ \bigcap_{S' \in A} S' | A \subseteq \mathcal{P}(S')\}$$