Logic Games and Automata – Memory Log and FAQ

1 Memory Log

Question 1.1 How is LTL defined, what is the syntax and semantics?

Question 1.2 How did we solve LTL model checking?
Answer. Model checking for LTL can be done effectively using an automaton approach because LTL is language invariant. For every φ and every $\alpha \in 2^{\sigma^\omega}$ we construct a Büchi automaton $\mathcal{B}$ such that $\alpha \models \varphi \iff \alpha \in \mathcal{L}(\mathcal{B})$.

First you transform the given formula into an equivalent reduced formula, i.e. we consider only formulas without $\bot, \vee, \mathbf{F}, \mathbf{G}, \mathbf{B}$. That means only propositional symbols, $\top$, the Boolean connectives $\wedge$ and $\neg$ and temporal operators $\mathbf{U}$ and $\mathbf{X}$ can appear in a formula. Give this formula, we then construct a non-deterministic Büchi-automaton that, at each position of a run, guesses the set of subformulas that are currently true and then checks whether the guess was correct. For formulas of the form $\mathbf{X}\varphi$ we can check locally via a simple transition whether φ holds at the next step. For every formula of the form $\varphi\mathbf{U}\psi$ we encode this information in the accepting condition of our generalized Büchi automaton. Then construct the appropriate automaton for the transition system B_S, compute the intersection with $B_{\neg\varphi}$ and test for emptiness by checking the existence of a reachable strongly connected component. The construction takes exponential time in the size of the formula and it can be shown that LTL model checking is PSPACE complete.

Question 1.3 How do we compute the intersection of the languages of two NBAs?

Question 1.4 Roughly how did we translate $\varphi \in$ LTL into a NBA?

Question 1.5 How did we check if $\mathcal{L}(B) \neq \emptyset$?

Question 1.6 Explain which of the logics we discussed are stronger/weake than each other?

Question 1.7 How did we define games?

Question 1.8 How did we define parity games?

Question 1.9 How did we prove that parity games have a positional winning strategy?

Question 1.10 How did we prove that parity games are determined?

2 Transition systems

Question 2.1 What is the model checking problem? What is its input and what is the output? How is it connected to real-world situation when we have a piece of software or hardware and we want to know whether is has some desired property?
Answer. The Model checking problem of a logic $L[\sigma]$ takes as an input a σ-TS S, a state $v \in V^S$ or a run $\alpha \in \mathcal{P}(S)$ and a formula $\varphi \in L[\sigma]$. The output is the decision whether $(S, s) \models \varphi$ or $(S, \alpha) \models \varphi$. A real world application of this is checking if a system has some property that can be described by a formula. A classical example would be whether a run in a system has the ability to run into a deadlock.

Question 2.2 Define bisimulation. When are two transition systems bisimilar?
Answer. $\mathcal{S}$, $\mathcal{T}$ be two σ-TSs. A *bisimulation* is a relation $R \subseteq V^{\mathcal{S}} \times V^{\mathcal{T}}$ such that $R \neq \emptyset$ and for all $(s,t) \in R$ hold:

(B1) $\beta(s) = \beta(t)$

(B2) For all $s' \in N(s)$ there exists $t' \in N(t)$ such that $(s', t') \in R$

(B3) For all $t' \in N(t)$ there exists $s' \in N(s)$ such that $(s', t') \in R$

Let $s_0 \in V^{\mathcal{S}}, t_0 \in V^{\mathcal{T}}$. (S, s_0) and (T, t_0) are called bisimilar (written $(S, s_0) \sim (T, t_0)$), iff there exists such a bisimulation R with $(s_0, t_0) \in R$.
reference def 1.11

Question 2.3 Is union of two bisimulations a bisimulation?
Answer. Yes.

Proof. Let $B := B_1 \cup B_2$ the union of two bisumulations. (B1) clearly holds. For (B2) let $s \in B_i$, clearly for a $s' \in N(s) \cap B_i$ the required t' exists via (B2) of B_i. So suppose $s' \in N(s) \setminus B_i = N(s) \cap B_{\bar{i}}$ but then the required t' exists via (B2) of $B_{\bar{i}}$. (B3) is analogous. □

Question 2.4 We say that two states are bisimilar if there exists a bisimulation between them. Is the relation of being bisimilar an equivalence?
Answer. Yes.

Proof. Bisimulation is clearly reflexive, take $B := \{(s, s) : s \in V\}$. Bisimulation is symmetric, take $B^{-1} := \{(t, s) : (s, t) \in B\}$ and swap (B2) and (B3). Bisimulation is transitive, take $B := B_1 \circ B_2$. □

Question 2.5 What does it mean for two runs (of a system or two different systems) to be indistinguishable?
Answer. Let $\mathcal{T}$ be a σ-TS. A *run* of $\mathcal{T}$ is an infinite word $\alpha \in (V^{\mathcal{T}})^\omega$ such that $(\alpha_0, \alpha_{i+1}) \in E^{\mathcal{T}}$ for all $i \in \mathbb{N}$. Two runs $\alpha = a_1 a_2 \dots, \gamma = c_1 c_2 \dots$ are indistinguishable if $\beta(\alpha) = \beta(a_1)\beta(a_2) \dots = \beta(\gamma)$.

Question 2.6 We say that that system S is similar to system T if for each run in T there is an indistinguishable run in S. Is it true that if S is similar to T and at the same time T is similar to S then S and T are bisimilar? Prove your answer.
Answer. No. Counter example: typical tree example, in which $\mathcal{S}_1$ makes decision of branching before $\mathcal{S}_2$. $\mathcal{S}_1, \mathcal{S}_2$ are similiar, because no matter what $\mathcal{S}_i$ chooses, the other one can always replicate the run, which is not possible when their turns are right after the move of the other TS.

Question 2.7 How does one algorithmically solve the problem of determining whether two transition systems are bisimilar?
Answer. By solving the bisimulation game. (Chapter 2.2)

3 Games

Question 3.1 Define game, play and strategy and winning strategy.
Answer. A game $\mathcal{G} := (V, V_0, E, v_0, \Omega)$ consists of an arena $\mathcal{A} := (V, V_0, E)$, an initial position $v_0 \in V$ and a winning condition $\Omega \subseteq V^\omega$.
A play on $\mathcal{A}$ is a maximal walk $\bar{v}$ in the digraph (V, E). $\bar{v}$ is finite if ends in a leaf or infinite. $\mathcal{P}(\mathcal{A}) := \{\bar{v} \in V^* \cup V^\omega \mid \bar{v} \text{ is a play no } \mathcal{A}\}$. $\mathcal{P}(\mathcal{A}, v_0)$ denotes all plays that start in v_0.
A strategy is a partial mapping $f : V^* V_\rho \to V$ such that $f(\bar{v})$ is definied for all walks $\bar{v} = v_0, \dots, v_n$ with $v_n \in V_\rho$, $N(v_n) \neq \emptyset$, $f(\bar{v}) \in N(v_n)$

Question 3.2 What does one have to do to prove that a game is winning for one of the players?
Answer. Show that a winning strategy exists.

Question 3.3 What does it mean for a game to be determined? Are all games determined?
Answer. A game is determined if a player has a winning strategy. Not all games are determined. Ultrafilters can be used to define non determined games.

Question 3.4 What are simple games? How does one solve them algorithmically?
Answer. Simple games are games where the winning condition is either V^ω or $\emptyset$. To solve a simple game start by marking all vertices in V_0 with no successors as part of W_1. We then add all nodes $v \in V_0$ to W_1 if $N(v) \subseteq W_1$ and all $v \in V_1$ if $N(v) \cap W_1 \neq \emptyset$ until completion.

4 Modal Logic

Question 4.1 Define syntax and semantics of modal logic.
Answer. Let σ be a signature, then ML[σ] contains

- the operators $\wedge, \vee, \neg, \top, \bot$ are contained with their usual semantics
- $P \in \text{ML}[\sigma]$ for all $P \in \sigma$ with $(S, s) \models P \iff s \in P$.
- If $\psi \in \text{ML}[\sigma]$, then $\Diamond\psi, \Box\psi \in \text{ML}[\sigma]$
  - $(S, s) \models \Diamond\psi \iff \exists s' \in N(s).(S, s') \models \psi$,
  - $(S, s) \models \Box\psi \iff \forall s' \in N(s).(S, s') \models \psi$

Question 4.2 What does it mean for two ML formulas to be equivalent?
Answer. $\psi_1 \equiv \psi_2$ if for all σ-TS S and $s \in V^S$ $(S, s) \models \psi_1 \iff (S, s) \models \psi_2$.

Question 4.3 What does it mean for a logic to be bisimulation invariant?
Answer. Let $L[\sigma]$ be a logic, let S_1, S_2 be two σ-TS, let $s_1 \in V^{S_1}, s_2 \in V^{S_2}$ with

$$(S_1, s_1) \sim (S_2, s_2) \Longrightarrow ((S_1, s_1) \models \varphi \iff (S_2, s_2) \models \varphi)$$

Question 4.4 How does one algorithmically solve the model checking problem for modal logic?
Answer. By solving the game (Def. 3.12).

Question 4.5 How would you prove that there is no ML formula which is true in a transition system if and only if the transition system contains a cycle?
Answer. Let $\psi \in \text{ML}[\sigma]$ be a formula that is true iff (S, s) is contained in a cycle. Then the tree (T, t) unraveling of (S, s) is bisimilar, but contains no cycle.

5 Temporal Logics

Question 5.1 Define syntax and semantics of LTL. Make sure that you understand the difference between formula being satisfied by a run and by a state.
Answer. **X, F, G, U, B**. $(S, \alpha) \models \varphi$ if the path satisfies the formula, and $(S, s) \models \varphi$ if all $\alpha \in \mathcal{P}(S, s)$ satisfy φ. **B**efore has the semantics that if there is a point v_i on a run $\alpha = v_0 v_1 \ldots$ at which ψ_2 holds, then for all $j < i$ ψ_1 holds.

Question 5.2 Missing img(18)

Question 5.3 Define syntax and semantics of CTL. CTL is the "extension" of LTL by existential path quantification E. However CTL does not allow nested uses of $\circ \in \{G, F, X\}$ such as $\mathbf{A} \circ \circ\, \varphi$. This is again allowed in CTL^*, hence every LTL formula has an equivalent CTL^* formula.

Question 5.4 Missing img(20)

Question 5.5 What does it formally mean for one logic to be stronger or more expressive than another?
Answer. That there is a formula in the stronger logic that can not be expressed in the weaker logic.

Question 5.6 How does one prove that one logic is not stronger than another? In other words how does one usually show that there are properties expressible in one logic which are not expressible in another?
Answer. By finding a formula that is not expressible in the other logic and claiming that the proof for this is rather involved. In the case of ML $\not\subset$ LTL, we used that fact $\mathcal{L}(S,s) = \mathcal{L}(T,t) \implies (S,s) \equiv_{\text{LTL}} \mathcal{L}(T,t)$, and that ML can distinguish this, take for example the classic branching example.

Question 5.7 Is LTL a stronger (more expressive) logic than CTL or vice versa?
Answer. No. CTL is not stronger that LTL: See $\mathbf{FG}P \in$ LTL and $\mathbf{AFAG}P \in$ CTL. LTL is not stronger than CTL, take $\mathbf{E} \circ \varphi$, $\varphi \in CTL$, $\circ \in \{\mathbf{X,F,G,U,B}\}$ But LTL is weaker than CTL*.

Question 5.8 Give a diagram of which logic is stronger than which and highlight which logics are incomparable.

Question 5.9 Can you classify the complexity of the Model-Checking problem for each of the logics we discussed in the lecture? How does this relate to the diagram you just gave? (What about the Modal-mu calculus in this scenario?)

6 Automata on Infinite Words

Question 6.1 What is the difference between finite automaton and transition system?

Question 6.2 Define Büchi automaton and generalized Büchi automaton.
Answer. A Büchi Automata is a tuple $\mathcal{A} := (Q, \Sigma, q_0, \Delta, F)$. $\mathcal{A}$ accepts a run ρ if $\text{Inf}(\rho) \cap F \neq \emptyset$.
A generalised Büchi Automata is a tuple $\mathcal{B} := (Q, \Sigma, q_0, \Delta, F_1, \ldots, F_k)$. $\mathcal{B}$ accepts a run ρ if $\text{Inf}(\rho) \cap F_i \neq \emptyset$ for all $1 \leq i \leq k$.

Question 6.3 How does one translate a transition system into a Büchi automaton?
Answer. See Definition 5.11. Summary: Let $\mathcal{S}$ be a σ-TS and $\mathcal{A}$ the constructed Büchi-Automaton.

1. $Q = V^{\mathcal{S}} \cup \{q_0\}$,
2. $\Sigma = 2^{\sigma}$,
3. $\Delta := \{(s, \beta(s'), s') \mid (s \in V^{\mathcal{S}} ands' \in N(s) or (s = q_0, s' = s_0)\}$,
4. $F = Q$

Question 6.4 Missing img(29)

Question 6.5 Are Büchi recognizable languages closed under: a union, b intersection, c complement

Question 6.6 Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two Büchi recognizable languages. How does one construct an automaton which accepts the language $\mathcal{L}_1 \cup \mathcal{L}_2$?
Answer. Let $\mathcal{A}_1, \mathcal{A}_2$ be the corresponding Büchi Automata. Remove q_0 in both. Then $\mathcal{A} := \mathcal{A}_1 \cup \mathcal{A}_2$ and q_0 with $\Delta := \{(q_0, \beta(s_1), s_1), (q_0, \beta(s_1), s_1)\} \cup \Delta_1 \cup \Delta_2$

Question 6.7 Does there exist to every Büchi automaton an equivalent deterministic Büchi automaton? If the answer is yes, how does one construct it? If the answer is no, what is the example of a Büchi recognizable language for which there is no deterministic Büchi automaton?
Answer. No, non-determinism yields a more powerful automaton scheme. One way to see this is the language $\mathcal{L} = (0+1)^* 0^\omega$ which is obviously Büchi recognizable.
Assume there is a deterministic automaton $\mathcal{B}$ with $|Q|$ states that accepts $\mathcal{L}$. Consider $\alpha := (\beta)^\omega \notin \mathcal{L}$ with $\beta := 0^{|Q|+1}1$ and for any i we have $\gamma_i := \underbrace{\beta^i}_{\gamma} 0^\omega \in \mathcal{L}$.

Since γ is a prefix of α, the automaton must have the same unique run up until this point by determinism. Since $\gamma_i \in \mathcal{L}$ for any i, the automaton must visit an accepting state when parsing the $|Q|+1$ zeros of β because it would otherwise never visit an accepting state in even longer stretches of zeros (by determinism) and would thus not accept all γ_i. Since β occurs infinitely often in α, the automaton accepts $\alpha \notin \mathcal{L}$ which is a contradiction.

Question 6.8 Characterize the languages accepted by deterministic Büchi automata.
Answer. A language $L \subseteq \Sigma^\omega$ is deterministically Büchi-recognisable if and only if there is a regular language $K \subseteq \Sigma^*$ such that $L = \lim K$.

Proof: Let $B = (Q, \Sigma, q_0, \delta, F)$ be a DBA with $L(B) = L$. Let $A = (Q, \Sigma, q_0, \delta, F)$ be the corresponding deterministic finite automaton (DFA), i.e., A is B considered as a DFA. Then, for $\alpha \in \Sigma^\omega$, we have:

$$\begin{aligned}\alpha \in L(B) &\iff \text{the unique run } \rho \text{ of } B \text{ on } \alpha \text{ visits } q \in F \text{ infinitely often}\\ &\iff \text{there are infinitely many } i > 0 \text{ such that the run of } A \text{ on } \alpha_0 \ldots \alpha_i \text{ ends in } q \in F\\ &\iff \text{there are infinitely many } i > 0 \text{ with } \alpha_0 \ldots \alpha_i \in L(A)\\ &\iff \alpha \in \lim(L(A)).\end{aligned}$$

Question 6.9 How does one test whether the language accepted by a Büchi automaton is empty? Where is this used?
Answer. Convert Büchi Automaton into a generalized Büchi Automaton. Then Run "NBA Emtpiness Test". Algorithm:

1. Compute $\mathcal{G}(\mathcal{B})$
2. Compute the SCCs of $\mathcal{G}(\mathcal{B})$
3. Compute the SCCs having a state of F
4. Compute which SCCs are reachable from q_0. If there is one, accept, otherwise reject.

Question 6.10 Consider transition system $\mathcal{T}$ and LTL formula φ such that $(\mathcal{T}, t_0) \not\models \varphi$. Does this imply that $(\mathcal{T}, t_0) \models \neg\varphi$?
Answer. No, the semantics of LTL are universal, meaning $(\mathcal{T}, t_0)$ satisfies φ if and only if all paths $\alpha \in \mathcal{P}(\mathcal{T}, t_0)$ satisfy φ. Obviously this doesn't mean when there exists a path that does not satisfy the formula that all paths don't satisfy the formula.

Question 6.11 Given an LTL formula φ and transition system $\mathcal{T}$, how does one verify whether $\mathcal{T} \models \varphi$? Describe each step in as much detail as possible.
Answer. Let $\mathcal{S}$ be a σ-TS, $v_0 \in V^{\mathcal{S}}$, $\varphi \in LTL$. Algorithm

1. Construct an NBA $\mathcal{B}_{\neg\varphi}$ for $\neg\varphi$ of size $2^{\mathcal{O}(|\varphi|)}$
2. Construct an NBA $\mathcal{B}_{\mathcal{S}}$ for (S, s_0) of size $\mathcal{O}(||\mathcal{S}||)$
3. Construct the NBA $\mathcal{B}$ with $\mathcal{L} = \mathcal{B}_{\neg\varphi} \cap \mathcal{B}_{\mathcal{S}}$ of size $\mathcal{O}(||\mathcal{B}_{\mathcal{S}} \cdot ||\mathcal{B}_{\neg\varphi}||)$
4. Test in linear time whether $\mathcal{L}(\mathcal{B}) = \emptyset$, if yes accept, otherwise reject.

Question 6.12 Why is proving co-NP-hardness of LTL model-checking more natural than proving NP-hardness?
Answer. As one can see above the satisfiability problem is solved by checking $\neg\varphi$, which yields in co-NP.

Question 6.13 What is the difference between generalized Büchi automaton and Muller automaton?
Answer. Let Q be the set of states, then the accepting condition is given as $\mathcal{F} \subseteq 2^Q$. Muller automaton accepts a run ρ if $\text{Inf}(\rho) \in \mathcal{F}$. Hence in contrast to (generalised) Büchi automata, where for any accepting run it must hold for all $F \in \mathcal{F}$ that $\text{Inf}(\rho) \cap F \neq \emptyset$, Muller automata list explicitly which runs are accepting and reject all other runs. Note that one can safely remove all supersets $F' \supset F$ from $\mathcal{F}$ if both $F, F' \in \mathcal{F}$.

7 Modal μ-calculus

Question 7.1 Define monotone function and fixed point
Answer. Let M be a set. $F : 2^M \to 2^M$ a function.

- F is called monotone if $F(X) \subseteq F(Y)$ for $X, Y \in M$ with $X \subseteq Y$
- $X \in M$ is called fixed point, if $F(X) = X$

Question 7.2 What does the Knaster-Tarski theorem say? Can you prove it?
Answer. Let M be a set, $F : 2^M \to 2^M$ a function. F has an unique least/ greatest fixed point namely

- $\mu(F) := \bigcap\{X \in M \mid F(X) \subseteq X\}$
- $\nu(F) := \bigcup\{X \in M \mid X \subseteq F(X)\}$

Proof. We only proof for $\mu(F)$. $\nu(F)$ is analogous. Let $A := \{X \in M \mid F(X) \subseteq X\}, Y = \bigcap_{X \in A} X$. For all $X \in A$ we know $Y \subseteq X$. By monotonicity $F(Y) \subseteq F(X) = X$. Hence $F(Y) \subseteq \bigcap_{X \in A} X = Y$.
$F(Y) \subseteq Y$ implies $F(F(Y)) \subseteq F(Y)$, thus $F(Y) \in A$, thus $Y \subseteq F(Y)$. Hence $F(Y) = Y$. Proof for unique smallest fixpoint is by assumption of a smaller one which yields a contradiction. □

Question 7.3 How does one compute least and largest fixed points? Can you prove it?
Answer. We can computer them by induction. For $\mu(F)$ start with $X^0 := \emptyset$, $X^{\alpha+1} := F(X^\alpha)$, which yields $X^0 \subseteq X^1 \subseteq X^2 \subseteq \ldots \subseteq X^\omega \subseteq \mu(F)$. Case for $\nu(F)$ follows analogously by starting in M instead of $\emptyset$.

Question 7.4 Define syntax and semantics of modal μ-calculus.
Answer. Syntax see definition 6.15. Semantics see definition 6.17

Question 7.5 Why does one have to be careful with the use of negation in modal μ-calculus?
Answer. Having negation yield in anti-monotonous functions F which then yield no fixed point anymore. More precisely

- For all $X \in Symb \setminus neg(\varphi)$, $F^{\mathcal{S}}_{\varphi,X}$ is monotone.
- For all $X \in Symb \setminus pos(\varphi)$, $F^{\mathcal{S}}_{\varphi,X}$ is anti-monotone.

Question 7.6 Given a fixed formula and a fixed transition system, can you give us the states that satisfy the formula?
Answer. Shouldn't be to hard, just do the recursion until a fixed point is reached. Check for μ or ν.

Question 7.7 Can you interpret the following μ-calculus formulae $\mu X(X \vee \Diamond P), \mu X(X \wedge \Box P), \nu X(Q \wedge \Diamond X), \nu Y(Y \vee \Diamond P)$ etc.
Answer.

- $\mu X(X \vee \Diamond P)$. $X^0 = \emptyset$, $F^{\mathcal{S}}_{\varphi,X}(X^0)$ denotes all the vertices that has a neighbor in which P holds. In the next iteration we get all vertices of X^0 and all the vertices which fullfil $\Diamond P$, which are already in X^0, thus we've reached already a fixed point
- $\mu X(X \wedge \Box P)$. $X^0 = \emptyset$. We can already see, that this yields $\emptyset$ over all, since X starts empty. Thus $\emptyset$ is the fixed point
- $\nu X(Q \wedge \Diamond X)$. $X^0 = V^{\mathcal{S}}$. In the first iteration we have all vertices there are. In the second we have all vertices $s \in Q^{\mathcal{S}}$. In the third we have all vertices that are in $Q^{\mathcal{S}}$ and whos neighbors are in the result of the second iteration, and so on... This leaves with a set of vertices from which we can reach Q by going in some direction which is equivalent to $\psi = \mathbf{EG}Q \in \text{CTL}$. Thus the set of vertices in which ψ holds are part of the fixed point.
- $\nu Y(Y \vee \Diamond P)$. Lame. We always remain in $V^{\mathcal{S}}$.

8 Parity Games

Question 8.1 Define parity games.
Answer. See definition 7.1

Question 8.2 Explain and define the translation from modal logic to simple games and from modal μ-calculus to parity games.
Answer.

- Modal Logic $\rightarrow$ simple games. Let $\mathcal{S}$ be a σ-TS and let $\varphi \in \text{ML}$. The arena $\mathcal{A}(\mathcal{S},\varphi) := (V, V_0, E)$ is defined as follows:
  - $V := \{(\psi, s) \mid \psi \in cl(\varphi), s \in V^{\mathcal{S}}\}$
  - $V_0 := \{(\psi, s) \in V \mid \psi = \Diamond\chi$ or $\psi = \chi_1 \vee \chi_2$ or ψ is literal with $(\mathcal{S}, s) \not\models \psi\}$
  - $E := \{(\psi, s), (\psi', s') \mid (\psi = \Box\chi$ or $\psi = \Diamond\chi$ and $\psi' = \chi$ or $s' \in N(s)) or (\psi = \chi_1 \wedge \chi_2$ or $\psi = \chi_1 \vee \chi_2)$ and $s' = s$ and $\psi' = \chi_1$ or $\psi' = \chi_2\}$
- Modal μ-calculus $\rightarrow$ parity games. Let $\mathcal{S}$ be a σ-TS, $\varphi \in L_\mu[\sigma]$. The arena $\mathcal{A}(\mathcal{S},\varphi) := (V, V_0, E)$ is defined as follows:
  - $V := \{(\psi, s) \mid \psi \in cl(\varphi), s \in V^{\mathcal{S}}\}$
  - $V_0 := \{(\psi, s) \in V \mid \psi = \Diamond\chi$ or $\psi = \chi_1 \vee \chi_2$ or $\psi \in \{P, \neg P, \bot\}$ with $(\mathcal{S}, s) \not\models \psi\}$
  - $((\psi, s), (\psi', s')) \in E$, if
    - $\psi = \circ\chi \in \{\Diamond, \Box\}$ and $\psi' = \chi$ and $s' \in N(s)$
    - $\psi = \chi_1 \circ \chi_2, \circ \in \{\wedge, \vee\}$ and $\psi' \in \{\chi_1, \chi_2\}$ and $s' = s$
    - $\psi = \mu X.\chi$ or $\psi = \nu X.\chi$ and $\psi' = \chi$ and $s' = s$

* $\psi = X$ for some $X \in Symb \setminus free(\varphi)$ and $\psi = D_\varphi(X)$ and $s' = s$

Question 8.3 What is an attractor? What are traps? What is a dominion? Give a formal definition
Answer. Let $U \subseteq V$. We define $Attr_\rho(U)$ inductively.

- $\text{Attr}^0_\rho(U) = U$
- $\text{Attr}^{\alpha+1}_\rho(U) := \text{Attr}^\alpha_\rho(U) \cup \{u \in V_\rho \mid \exists v \in N(v) \cap \text{Attr}^\alpha_\rho\} \cup \{u \in V_{\bar{\rho}} \mid \forall v \in N(v) \subseteq \text{Attr}^\alpha_\rho\}$

That means there is a (attractor) strategy such that for all $u \in \text{Attr}^\alpha_\rho$ Player ρ is able to get into U.
$V \setminus \text{Attr}^\alpha_\rho$ is a ρ-trap, which means Player ρ is not able to leave the trap, otherwise there would be a vertex v with which Player ρ will in U, than v is in the attractor. That's why this is called a trap.
Dominions are needed for the involved Parity-Games proof. A dominion for player ρ is a subset of the winning region in which player ρ can force the game to stay entirely within the dominion while still winning it.

Question 8.4 Explain the recursive algorithm for solving parity games given in the lecture notes.

Question 8.5 Can you sketch the idea behind the quasi-polynomial Zielonka algorithm?
Answer. The quasi-polynomial algorithm is essentially 3 copies of the exponential time algorithm. However, we pass additional precision parameters $p_\rho, p_{1-\rho}$ artificially restricting the maximum depth of the search tree. The idea is, when recursively looking for finer and finer dominions (recall that Zielonka recursively computes parts of the winning region we called W_1'' in the proof of positional determinacy), there can ever only be 1 part that is bigger than half of the entire winning region.

Question 8.6 How do parity games relate to the μ-calculus?
Answer. They can be used to do model checking on L_μ. Parity games can also be modeled by a L_μ formula and a transition system. Hence the coincide exactly in their expressiveness.

Question 8.7 Describe how Model Checking for the μ-calculus can be reduced to solving parity games? What are the implications on the running time?
Answer. A game with size $\mathcal{O}(||S|| \cdot |\varphi|)$ can be constructed. Hence the running time is $(||S|| \cdot |\varphi|)^{\mathcal{O}(\log(|S| \cdot |\varphi|))}$

9 Notable Formulas

9.1 CTL* $\subseteq L_\mu$

$\nu X.P \wedge \Diamond\Diamond X \in L_\mu$ but there is no equivalent formula in CTL*.

9.2 FO $\neq$ CTL*

$\mathbf{EF}P \in$ CTL* but there is no equivalent formula in FO as the formula expresses a global property. But on the other hand FO can detect circles, which all bisimulation invariant logics, and hence CTL*, cannot.

9.3 $LTL \neq CTL$

$\mathbf{FG}P \in$ LTL has no equivalent formula in CTL (CTL does not allow nesting of F, G, X until another A or E). $\mathbf{EX}P \in$ CTL has not equivalent formula LTL.

9.4 LTL, CTL $\subseteq$ CTL*

By definition.

9.5 ML$^\infty$ $\subseteq L_\mu$

For both logics to be equivalent it has to hold that for each TS (S, s) and every formula $\varphi \in L_\mu$ there has to be a $\psi \in$ ML$^\infty$ such that

$$(S, s) \models \varphi \iff (S, s) \models \psi$$

Since (S, s) has an infinitely large tree unraveling (T, t) such that $(S, s) \sim (T, t)$ there has to be an limit ordinal such that ψ can describe properties of L_μ, but since there are infinitely many infinitely large TS that's not possible, thus ML$^\infty$ $\subset L_\mu$. Proof of other direction is rather involved and left as an excercise for the reader :)

9.6 ML $\not\subseteq$ LTL

Let $\varphi = \Diamond\psi \in$ ML, then there is no LTL formula. (see 9.3). Another way to see this is the classic branching example of 2 non-bisimilar transition systems that have the same language. Since we know that LTL is language invariant, LTL cannot differentiate between the two systems. ML, however, can do so easily by $\Diamond(\Diamond P \wedge \Diamond\neg P)$. Note that the LTL formula $\mathbf{X}(\mathbf{X}P \wedge \mathbf{X}\neg P)$ is unsatisfiable.

10 Translation of formulas

10.1 CTL $\Longrightarrow L_\mu$

normal stuff as usual. Let $\varphi \in$ CTL, $\hat{\varphi} \in L_\mu$

1. $\varphi = \mathbf{EG}\psi$ let $\hat{\varphi} = \nu X.(\hat{\psi} \wedge \Diamond X)$,
2. $\varphi = \mathbf{E}(\psi \mathbf{U} \chi)$ let $\hat{\varphi} = \mu X.(\hat{\chi} \vee (\hat{\psi} \wedge \Diamond X))$.

10.2 ML $\Longrightarrow$ FO

We can recursively construct a formula $\hat{\varphi}(x)$ with a free variable to simulate the modalities. Boolean combinations are trivial. Let $\varphi \in \text{ML}, \hat{\varphi} \in \text{FO}$, $(\mathcal{S}, s)$ be a σ-TS.

1. Let $\varphi = \Box\psi$, then $\hat{\varphi}(x) := \forall y(E(x,y) \wedge \hat{\psi}(x))$.
2. Let $\varphi = \Diamond\psi$, then $\hat{\varphi}(x) := \exists y(E(x,y) \wedge \hat{\psi}(x))$.

10.3 ML $\Longrightarrow$ ML$^\infty$

common-part is trivial, except $\wedge, \vee$. Let $\varphi \in \text{ML}, \hat{\varphi} \in \text{ML}^\infty$
$\varphi = \psi_1 \wedge \psi_2$ let $\hat{\varphi} = \bigwedge\{\psi_1, \psi_2\}$ and $\varphi = \psi_1 \vee \psi_2$ let $\hat{\varphi} = \bigvee\{\psi_1, \psi_2\}$

10.4 LTL $\Longrightarrow$ CTL*

For each $\varphi \in \text{LTL}, \text{A}\varphi \in \text{CTL}^*$.

Note that FO, ML formulas have only finite reach, while $\text{LTL}, \text{CTL}, \text{CTL}^*, L_\mu, \text{ML}^\infty$ have infinitely large reachability.

11 Important Algorithm

11.1 Solve Simple Games

Assume all infinite games are won by player 0, otherwise change the roles of the players. Then player 1 can only win by forcing player 0 into a deadlock in which player 0 has to move. Hence the algorithm that computes the winning region of player 1 simply has to compute the player-1-attractor of all such vertices. This is possible in linear time.

11.2 Automata Based Model Checking (s.61)

11.3 Emptiness Test on NBA

11.4 MC(CTL)

11.5 Solve parity game on finite arena in exp-time

11.6 Solve parity game on finite arena in quasi-poly-time

12 Important Proves (that are (non)-trivial)

12.1 Correctness of Bisimulation Game (2.17)

Theorem 12.1 *Let $\mathcal{S}, \mathcal{T}$ be TS, $s_0 \in V^{\mathcal{S}}, t_0 \in V^{\mathcal{T}}$ such that $\beta(s_0) = \beta(t_0)$. Then*

$$(S, s_0) \sim (T, t_0) \iff \textit{Player 0 wins the corresponding bisimulation game}$$

PROOF. $\Longrightarrow$: Let $R : (\mathcal{S}, s_0) \sim (\mathcal{T}, t_0)$ be a bisim. We are going to define w.s. f. Let $\pi := (A, s_0, t_0), \ldots, (A, s, t), (S, s', t)$ with $(s, t) \in R$. Then there is a $t' \in \mathcal{T}$ with $(s', t') \in R$, define $f(\pi) := (A, s', t')$. Analogous for $\pi := (A, s_0, t_0), \ldots, (A, s, t), (S, s, t')$. In case $(s, t) \notin R$, then $f(\pi)$ undefined.
As $(s_0, t_0) \in R$, each play consistent with f is either infinite (we win by winning cond.) or finite, for which there exists a (A, s, t) such that $(s, t) \notin R$, by which $N(s) = N(t) = \emptyset$, we also win here.
$\Longleftarrow$: Let f be a w.s. for Player 0. We define

$$R := \{(s, t) \mid \text{ there exists a walk consistent with } f \text{ from } (A, s_0, t_0) \text{ to } (A, s, t)\}$$

12.2 Simple Games are positional determined (2.23)

PROOF.

12.3 MC(ML) (3.14)

Definition 12.2 Let $\mathcal{S}$ a TS, $\varphi \in ML$. Arena $\mathcal{A} := (V, V_0, E)$ defined as follows

- $V := \{(\psi, s) \mid \psi \in cl(\varphi), s \in V^{\mathcal{S}}\}$
- $V_0 := \{(\psi, s) \mid \psi = \Diamond\chi$ or $\psi = \chi_1 \vee \chi_2$ or ψ is literal such that $(\mathcal{S}, s) \not\models \psi\}$
- $E := \{((\psi, s), (\psi', s)) \mid if$
  - $\psi = \circ\chi, \circ \in \{\Box, \Diamond\}, \psi' = \chi, s' \in N(s)$ or
  - $\psi = \chi_1 \circ \chi_2, \circ \in \{\wedge, \vee\}, \psi' = \chi_1$ or $\psi' = \chi_2$ and $s' = s\}$

Theorem 12.3 *Let $\mathcal{S}$ a TS, $s \in V^{\mathcal{S}}$, $\varphi \in \mathrm{ML}$. Then*

$(\mathcal{S}, s) \models \varphi \iff$ *Player 0 wins the corresponding "ML" game*$(\mathcal{A}(\mathcal{S}, \varphi), (\varphi, s_0), V^\omega)$

Proof. Let $s \in V^{\mathcal{S}}, \psi \in cl(\varphi)$. By induction over φ.

- ψ is literal. (ψ, s) has no successor, thus Player 0 wins iff $(\mathcal{S}, s) \models \psi$ (by definition)
- $\psi = \chi_1 \vee \chi_2$. Assume $(\mathcal{S}, s) \models \psi$. (ψ, s) has 2 successors. Wlog. $(\mathcal{S}, s) \models \chi_1$, by ind.hyp. Player 0 wins from $(\mathcal{S}, \chi_1)$, thus also from $(\mathcal{S}, \psi)$. Assume P_0 wins $(\mathcal{A}, (\psi, s), V^\omega)$. Wlog. P_0 wins from (χ_1, s) (by assumption), then $(\mathcal{S}, s) \models \chi_1$ (by ind.hyp.), thus $(\mathcal{S}, s) \models \psi$
- $\psi = \Box\chi$. Assume $(\mathcal{S}, s) \models \psi$. Let $s' \in N(s)$, then $(\mathcal{S}, s') \models \chi$, by ind.hyp. P_0 wins from $(\mathcal{S}, \chi)$, thus also from $(\mathcal{S}, \psi)$. Assume P_0 wins $(\mathcal{A}, (\psi, s), V^\omega)$. Wlog. P_0 wins (χ, s') for all $s' \in N(s)$ (by assumption), then $(\mathcal{S}, s') \models \chi$ (by ind.hyp.), thus $(\mathcal{S}, s) \models \Box\chi$. Note that this is already similar to $\psi = \chi_1 \vee \chi_2$! □

12.4 ML is bisim. invariant (3.17)

Theorem 12.4 *Modal logic is bisim. invariant: For all $\varphi \in \mathrm{ML}$ and all TS $\mathcal{S}, \mathcal{T}$ with $s \in V^{\mathcal{S}}, t \in V^{\mathcal{T}}$*

$$(\mathcal{S}, s) \sim (\mathcal{T}, t) \iff ((\mathcal{S}, s) \models \varphi \iff (\mathcal{T}, t) \models \varphi)$$

Proof. Let $(\mathcal{S}, s) \sim (\mathcal{T}, t), \varphi \in \mathrm{ML}$. Proof by induction over φ.

- φ atomic. We have $(\mathcal{S}, s) \models \varphi \iff (\mathcal{T}, t) \models \varphi$, when $\beta(s) = \beta(t)$.
- $\varphi = \psi_1 \wedge \psi_2$. Thus $(\mathcal{S}, s) \models \psi_1$ and $(\mathcal{S}, s) \models \psi_2$. $(\mathcal{T}, t) \models \psi_i, i \in \{1, 2\}$ (by ind.hyp). Thus $(\mathcal{T}, t) \models \psi_1 \wedge \psi_2$.
- $\Diamond$ analogously to 3.14. For $\Box\psi$ consider $\neg\Diamond\neg\psi$. □

12.5 Hennessy & Millner (3.26)

Theorem 12.5 *Let $\mathcal{S}, \mathcal{T}$ be finite σ-TS, $s \in V^{\mathcal{S}}, t \in V^{\mathcal{T}}$. Then*

$$(\mathcal{S}, s) \equiv_{\mathrm{LTL}} (\mathcal{T}, t) \iff (\mathcal{S}, s) \sim (\mathcal{T}, t)$$

Proof. $\Longleftarrow$: Assume $(\mathcal{S}, s) \sim (\mathcal{T}, t)$. Then $(\mathcal{S}, s) \sim_n (\mathcal{T}, t)$ for all $n \in \mathbb{N}$, thus ML with modal-depth can fully describe $(\mathcal{S}, s_0), (\mathcal{T}, t)$.
$\Longrightarrow$: Assume $(\mathcal{S}, s) \equiv_{\mathrm{LTL}} (\mathcal{T}, t)$. ToDo □

12.6 Modal Logic has finite model property (3.27)

Theorem 12.6 *Each satisfiable formular $\varphi \in$ ML has a model with $\leq 2^{|\varphi|}$ states.*

Proof. (Simple one) Let $\varphi \in$ ML and let $(\mathcal{S}, s) \models \varphi$ be its model.

1. Construct tree unravelling (T, ω) of $(\mathcal{S}, s)$
2. Cut $(\mathcal{T}, \omega)$ at depth $n := md(\varphi)$, obtain $(\mathcal{T}', \omega)$. Then $(\mathcal{T}, \omega) \sim_n (\mathcal{T}', \omega)$ (sufficient for φ, as it cannot look further)
3. Prune $(\mathcal{T}', \omega)$. □

12.7 LTL is bisim. invariant (4.11)

Theorem 12.7 *Let $(S, s_0), (T, t_0)$ be TS. Then*

$$[(S, s_0) \sim (T, t_0) \Longrightarrow] \; \mathcal{L}(\mathcal{S}, s_0) = \mathcal{L}(\mathcal{T}, t_0) \Longrightarrow (S, s_0) \equiv_{\mathrm{LTL}} (T, t_0)$$

Proof. We show $(S, s_0) \not\models \varphi \iff (T, t_0) \not\models \varphi$. Let $\alpha^{\mathcal{S}} \in \mathcal{P}(\mathcal{S}, s_0), \alpha^{\mathcal{T}} \in \mathcal{P}(\mathcal{T}, t_0)$ with $\beta(\alpha^{\mathcal{S}}) = \beta(\alpha^{\mathcal{T}})$, thus $(\mathcal{S}, s_0) \models \varphi \iff (\mathcal{T}, t_0) \models \varphi$ for all $\varphi \in$ LTL. From this if there is a $\psi \in$ LTL with $(\mathcal{S}, s_0) \not\models \psi$, then there is $\alpha \in \mathcal{P}(\mathcal{S}, s_0)$ with $(\mathcal{S}, \alpha) \not\models \psi$, since $\mathcal{L}(\mathcal{S}, s_0) = \mathcal{L}(T, t_0)$ $(\mathcal{T}, t_0)$ has the same path, thus $(\mathcal{T}, t_0) \not\models \psi$, hence $(T, t_0) \not\models \psi$. □

12.8 büchi-rec. languages are closed under complement (5.41)

Theorem 12.8

$$\mathcal{L}(\mathcal{B})^c = \bigcup_{\substack{K,J \text{ equivalence classes of } \approx_{\mathcal{B}} \\ KJ^\omega \not\subseteq \mathcal{L}(\mathcal{B})}} KJ^\omega$$

Proof. $\subseteq$: Let $\alpha \in \mathcal{L}(\mathcal{B})^c$. Furthermore observe that $\approx_{\mathcal{B}}$ is an equivalence (and more importantly a congruence) with finitely many classes on Σ^*. By the [technical lemma] we thus know there exist equivalence classes $K, J \subseteq \Sigma^*$ of $\approx_{\mathcal{B}}$ such that $\alpha \in KJ^\omega$. Since $\alpha \notin \mathcal{L}(\mathcal{B})$ we have $KJ^\omega \not\subseteq \mathcal{L}(\mathcal{B})$. Thus α is in the (exhaustive) union over all such choices K, J of equivalence classes.

$\supseteq$: Let $\alpha \in \bigcup_{\substack{K,J \text{ equivalence classes of } \approx_{\mathcal{B}} \\ KJ^\omega \not\subseteq \mathcal{L}(\mathcal{B})}} KJ^\omega$. Let K, J be the corresponding equivalence classes such that $\alpha \in KJ^\omega$. Assume for the sake of contradiction that $\alpha \notin \mathcal{L}^c(\mathcal{B})$. Then $\alpha \in \mathcal{L}(\mathcal{B})$ and hence $KJ^\omega \cap \mathcal{L}(\mathcal{B}) \neq \emptyset$. By [small, not so technical lemma] it must then be the case that $KJ^\omega \subseteq \mathcal{L}(\mathcal{B})$. But then the choices of K, J contradict the condition under the union. □

12.9 $\alpha \models \varphi \iff \alpha \in \mathcal{B}$ (5.50)

Theorem 12.9 *Let σ a signature, $\Sigma := 2^\sigma$. For every $\varphi \in \mathrm{LTL}[\sigma]$ there is a NBA $\mathcal{B}$ with $Q^{\mathcal{B}} \leq |\varphi| \cdot 2^{2|\varphi|}$ such that for all $\alpha \in \Sigma^\omega$*

$$\alpha \models \varphi \iff \alpha \in \mathcal{L}(\mathcal{B})$$

12.10 NBA emptiness test (5.51)

12.11 LTL is co-NP hard (5.54)

Algorith:

1. Compute $\mathcal{G}(\mathcal{B})$
2. Compute SCCs of $\mathcal{G}$
3. Compute SCCs having F
4. Check if q_0 can reach such SCCs, if yes, then accept, otherwise reject

12.12 Knaster-Tarski (6.2)

12.13 L_μ is bisim. invariant (6.31)

PROOF. Assume $\varphi \in L_\mu$ with $(\mathcal{S}, s) \models \varphi$ and $(\mathcal{T}, t) \not\models \varphi$. Select $\alpha \in On$ with $\alpha \geq \max\{|V|^{\mathcal{S}}, |V|^{\mathcal{T}}\}$. Then there is $\psi \in \mathrm{ML}^\infty$ equivalent to φ. Thus $(\mathcal{S}, s) \models \varphi$ and $(\mathcal{T}, t) \not\models \varphi$, which contradicts ML^∞ is bisim. invariant for fixed classes of TS. □

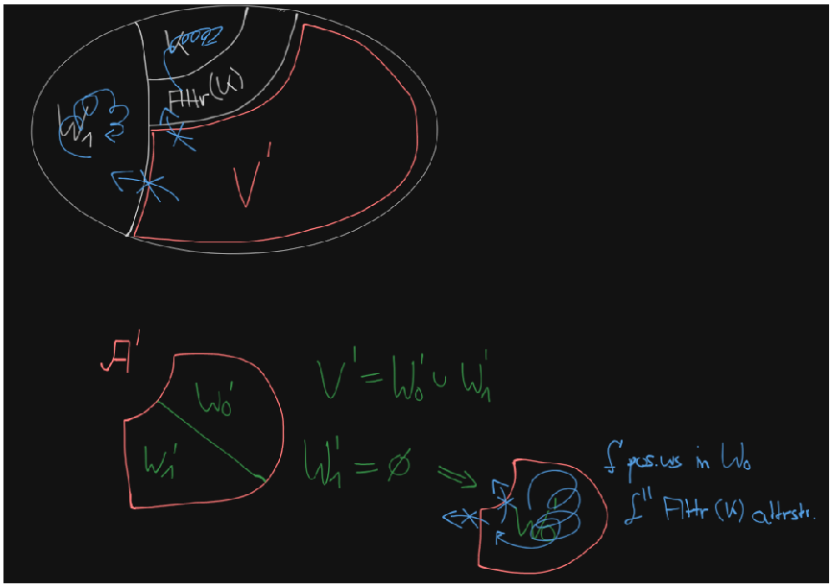
12.14 w.s. for all $v \in W_\rho$ (7.11)

Theorem 12.10 *Let $\rho \in \{0, 1\}$. There is a pos.ws. f for ρ such that f is a ws. for all $v \in W_\rho$ in the game.*

PROOF. We know that every $v \in W_\rho$ has a pos.ws. f_v. We want to merge all of those into a single ws. f. Problem is that there might be vertices such that $f_v(v) \neq f_{v'}(v)$. We need to eliminate those. Be $\{f_{v_1}, \dots f_{|W_\rho|}\}$ be the set of all pos.ws., then there is a well ordering of those. We always take the smallest f_{v_i} in this set and add it to f. Done. □

12.15 parity games are pos. determined (7.12)

Theorem 12.11 $V = W_0 \cup W_1$



Proof.

12.16 MC(L_μ) (7.34,7.35,7.38)