

Propositional Logic Syntactic Sugar

$$\varphi \Leftrightarrow \psi := (\neg \varphi \vee \psi) \wedge (\neg \psi \vee \varphi) \quad \varphi \rightarrow \psi := \neg \varphi \vee \psi$$

$$\varphi \oplus \psi := (\varphi \wedge \neg \psi) \vee (\psi \wedge \neg \varphi) \quad \varphi \bar{\wedge} \psi := \neg(\varphi \wedge \psi)$$

$$(\alpha \Rightarrow \beta | \gamma) := (\neg \alpha \vee \beta) \wedge (\alpha \vee \gamma) \quad \varphi \bar{\vee} \psi := \neg(\varphi \vee \psi)$$

Distributivity: $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$
 $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$
De Morgan: $\neg(a \vee b) \equiv (\neg a \wedge \neg b)$
 $\neg(a \wedge b) \equiv (\neg a \vee \neg b)$
CNF: from truth table, take minterms that are 0.
Each minterm is built as an OR of the negated variables. E.g., $(0, 0, 1) \rightarrow (x \vee y \vee \neg z)$.

SAT SOLVERS
Satisfiability, Validity and Equivalence

$$\text{SAT}(\varphi) := \neg \text{VALID}(\neg \varphi) \quad \varphi \Leftrightarrow \psi := \text{VALID}(\varphi \leftrightarrow \psi)$$

$$\text{VALID}(\varphi) := (\varphi \Leftrightarrow 1) \quad \text{SAT}(\varphi) := \neg(\varphi \Leftrightarrow 0).$$

Sequent Calculus:

-**Validity:** start with $\{\} \vdash \phi$; valid iff $\Gamma \cap \Delta \neq \{\}$
FOR ALL leaves.
-**Satisfiability:** start with $\{\phi\} \vdash \{\}$; satisfiable iff $\Gamma \cap \Delta = \{\}$ for AT LEAST ONE leaf.
-Counterexample/sat variable assignment: var is true, if $x \in \Gamma$; false otherwise; "don't care", if variable doesn't appear.

OPER.	LEFT	RIGHT
NOT	$\frac{\neg \phi, \Gamma \vdash \Delta}{\Gamma \vdash \phi, \Delta}$	$\frac{\Gamma \vdash \neg \phi, \Delta}{\phi, \Gamma \vdash \Delta}$
AND	$\frac{\phi \wedge \psi, \Gamma \vdash \Delta}{\phi, \psi, \Gamma \vdash \Delta}$	$\frac{\Gamma \vdash \phi \wedge \psi, \Delta}{\Gamma \vdash \phi, \Delta \quad \Gamma \vdash \psi, \Delta}$
OR	$\frac{\phi \vee \psi, \Gamma \vdash \Delta}{\phi, \Gamma \vdash \Delta \quad \psi, \Gamma \vdash \Delta}$	$\frac{\Gamma \vdash \phi \vee \psi, \Delta}{\Gamma \vdash \phi, \psi, \Delta}$

Resolution Calculus $\frac{\{ \neg x \} \cup C_1 \quad \{ x \} \cup C_2}{C_1 \cup C_2}$

To prove unsatisfiability of given clauses in CNF: If we reach $\{\}$, the formula is unsatisfiable. E.g., $\{\{a\}, \{\neg a, b\}, \{\neg b\}\}$, we get: $\{a\} + \{\neg a, b\} \rightarrow \{b\}$; $\{b\} + \{\neg b\} \rightarrow \{\}$ (unsatisfiable).
To prove validity, prove UNSAT of negated formula.

Davis Putnam Procedure - proves SAT; To prove validity: prove unsatisfiability of negated formula.

- (1) Compute Linear Clause Form
(Don't forget to create the last clause $\{x_n\}$)
(2) Last variable has to be 1 (true) \rightarrow find implied variables.
(3) For remaining variables: assume values and compute newly implied variables.
(4) If contradiction reached: backtrack.

Linear Clause Forms (Computes CNF) - Bottom up (inside out) in the syntax tree: convert "operators and variables" into new variable. E.g., $\neg a \vee b$ becomes $x_1 \leftrightarrow \neg a$; $x_2 \leftrightarrow x_1 \vee b$. Use rules below to find CNF. Create last clause $\{X_n\}$

$$x \leftrightarrow \neg y \Leftrightarrow (\neg x \vee \neg y) \wedge (x \vee y)$$

$$x \leftrightarrow y_1 \wedge y_2 \Leftrightarrow (\neg x \vee y_1) \wedge (\neg x \vee y_2) \wedge (x \vee \neg y_1 \vee \neg y_2)$$

$$x \leftrightarrow y_1 \vee y_2 \Leftrightarrow (\neg x \vee y_1 \vee y_2) \wedge (x \vee \neg y_1) \wedge (x \vee \neg y_2)$$

$$x \leftrightarrow y_1 \rightarrow y_2 \Leftrightarrow (x \vee y_1) \wedge (x \vee \neg y_1 \vee \neg y_2) \wedge (\neg x \vee \neg y_1 \vee y_2)$$

$$x \leftrightarrow (y_1 \leftrightarrow y_2) \Leftrightarrow (x \vee y_1 \vee y_2) \wedge (x \vee \neg y_1 \vee \neg y_2) \wedge (\neg x \vee y_1 \vee \neg y_2) \wedge (\neg x \vee \neg y_1 \vee y_2)$$

$$x \leftrightarrow y_1 \oplus y_2 \Leftrightarrow (x \vee \neg y_1 \vee y_2) \wedge (x \vee y_1 \vee \neg y_2) \wedge (\neg x \vee y_1 \vee y_2) \wedge (\neg x \vee \neg y_1 \vee \neg y_2)$$

<pre> Compose(int x, BddNode ψ, α) int m; BddNode h, l; if x>label(ψ) then return ψ; elseif x=label(ψ) then return ITE(α, high(ψ), low(ψ)); else m=max{label(ψ), label(α)} (α_0, α_1) := Ops(α, m); (ψ_0, ψ_1) := Ops(ψ, m); h:=Compose(x, ψ_1, α_1); l:=Compose(x, ψ_0, α_0); return CreateNode(m, h, l) endif; end </pre>	<pre> ITE(BddNode i, j, k) int m; BddNode h, l; if i = 0 then return k elseif i=1 then return j elseif j=k then return k else m = max{label(i), label(j), label(k)} (i_0, i_1) := Ops(i, m); (j_0, j_1) := Ops(j, m); (k_0, k_1) := Ops(k, m); l:=ITE(i_0, j_0, k_0); h:=ITE(i_1, j_1, k_1); return CreateNode(m, h, l) endif; end </pre>
<pre> Constrain(Φ, β) if $\beta=0$ then ret 0 elseif $\Phi \in \{0, 1\} (\beta = 1)$ ret Φ else m=max{label(β), label(Φ)} (Φ_0, Φ_1) := Ops(Φ, m); (β_0, β_1) := Ops(β, m); if $\beta_0=0$ ret Constrain(Φ_1, β_1) elseif $\beta_1=0$ then ret Constrain(Φ_0, β_0) else l:=Constrain(Φ_0, β_0); h:=Constrain(Φ_1, β_1); ret CreateNode(m, h, l) endif; endif; end </pre>	<pre> Apply(\odot, Bddnode a, b) int m; BddNode h, l; if isLeaf(a)&isLeaf(b) then return Eval(\odot, label(a), label(b)); else m=max{label(a), label(b)} (a_0, a_1) := Ops(a, m); (b_0, b_1) := Ops(b, m); h:=Apply(\odot, a1, b1); l:=Apply(\odot, a0, b0); return CreateNode(m, h, l) endif; end </pre>
<pre> Restrict(Φ, β) if $\beta=0$ return 0 elseif $\Phi \in \{0, 1\} \vee (\beta = 1)$ return Φ else m=max{label(β), label(Φ)} (Φ_0, Φ_1) := Ops(Φ, m); (β_0, β_1) := Ops(β, m) if $\beta_0=0$ return Restrict(Φ_1, β_1) elseif $\beta_1=0$ return Restrict(Φ_0, β_0) elseif m=label(Φ) return CreateNode(m, Restrict(Φ_1, β_1), Restrict(Φ_0, β_0)) else return Restrict(Φ, Apply(\vee, β_0, β_1)) endif; endif; end </pre>	<pre> Exists(BddNode e, φ) if isLeaf(φ)&isLeaf(e) return φ; elseif label(e)>label(φ) return Exist(high(e), φ) elseif label(e)=label(φ) h=Exist(high(e), high(φ)) l=Exist(high(e), low(φ)) return Apply(\vee, l, h) else (label(e)<label(φ)) h:=Exists(e, high(φ)) l:=Exists(e, low(φ)) return CreateNode(label(φ), h, l) endif; end function. ----- ZDD: If positive cofactor = 0, redirect edge to negative cofactor. If variable not in the formula, add with both edges pointing to same node. ----- FDD: Positive Davio Decomposition. (Keep both edges to i if happens!) $\varphi = [\varphi]_x^0 \oplus x \wedge (\partial \varphi / \partial x)$ $(\partial \varphi / \partial x) := [\varphi]_x^0 \oplus [\varphi]_x^1$ </pre>

Local Model Checking (follow precedence!)

$\frac{s \vdash \varphi \wedge \psi}{\{s \vdash \varphi\} \wedge \{s \vdash \psi\}}$	$\frac{s \vdash \varphi \vee \psi}{\{s \vdash \varphi\} \vee \{s \vdash \psi\}}$
$\frac{s \vdash \varphi \perp \varphi}{\{s_1 \vdash \varphi\} \dots \{s_n \vdash \varphi\}}$	$\frac{s \vdash \varphi \Diamond \varphi}{\{s_1 \vdash \varphi\} \dots \{s_n \vdash \varphi\}}$
$\frac{s \vdash \varphi \Box \varphi}{\{s'_1 \vdash \varphi\} \dots \{s'_n \vdash \varphi\}}$	$\frac{s \vdash \varphi \Diamond \varphi}{\{s'_1 \vdash \varphi\} \dots \{s'_n \vdash \varphi\}}$
$\frac{s \vdash \varphi \mu x. \varphi \quad s \vdash \varphi \nu x. \varphi}{s \vdash \varphi}$	$\frac{s \vdash \varphi \quad \mathcal{D}\Phi(\text{replace w. initial form.})}{s \vdash \mathcal{D}\Phi(x)}$
$\{s_1 \dots s_n\} = suc_{\exists}^R(s)$	$\{s'_1 \dots s'_n\} = pre_{\exists}^R(s)$

Approximations and Ranks

If $(s, \mu x. \varphi)$ repeats \rightarrow return 0	$apx_0(\mu x. \varphi) := 0$
If $(s, \nu x. \varphi)$ repeats \rightarrow return 1	$apx_0(\nu x. \varphi) := 1$

Tarski-Knaster Theorem: μ : is starts $\bot \rightarrow$ least fixpoint $\blacklozenge \nu$: starts $\top \rightarrow$ greatest fixpoint

Quantif. $\exists x. \varphi := [\varphi]_x^1 \vee [\varphi]_x^0 \quad \forall x. \varphi := [\varphi]_x^1 \wedge [\varphi]_x^0$

Predecessor and Successor

$$\diamond := pre_{\exists}^R(Q) := \exists x'_1, \dots, x'_n. \varphi_R \wedge [\varphi_Q]_{x'_1, \dots, x'_n}^{x_1, \dots, x_n}$$

$$\Diamond := suc_{\exists}^R(Q) := [\exists x_1, \dots, x_n. \varphi_R \wedge \varphi_Q]_{x'_1, \dots, x'_n}^{x_1, \dots, x_n}$$

$$\square = pre_{\forall}^R(Q) := \forall x'_1, \dots, x'_n. \varphi_R \rightarrow [\varphi_Q]_{x'_1, \dots, x'_n}^{x_1, \dots, x_n}$$

$$\Box := suc_{\forall}^R(Q) := [\forall x_1, \dots, x_n. \varphi_R \rightarrow \varphi_Q]_{x'_1, \dots, x'_n}^{x_1, \dots, x_n}$$

Example: \square / \Box

$pre_{\forall}^R(\{S3, S4\}) = \{S0, S5\}$

$suc_{\forall}^R(\{S3, S4\}) = \{S2, S5\}$

$pre_{\forall}^R(Q = \{S_1, \dots, S_n\})$ for each node n in \mathcal{K} : if (n points to a node that is not in Q) $n \notin pre_{\forall}^R(Q)$ else $n \in pre_{\forall}^R(Q)$	$suc_{\forall}^R(Q = \{S_1, \dots, S_n\})$ for each node n in \mathcal{K} : if (n is pointed by a node that is not in Q) $n \notin suc_{\forall}^R(Q)$ else $n \in suc_{\forall}^R(Q)$
---	---

AUTOMATA
Automata types: G \rightarrow Safety; F \rightarrow Liveness;
FG \rightarrow Persistence/Co-Buchi; GF \rightarrow Fairness/Buchi.
Automaton Determinization
NDet_G \rightarrow Det_G: 1.Remove all states/edges that do not satisfy acceptance condition; 2.Use Subset construction (Rabin-Scott); 3.Acceptance condition will be the states where $\{\}$ is never reached.
{NDet_F (partial) or NDet_{prefix}} \rightarrow Det_{FG}: Breakpoint Construction.
NDet_F (total) \rightarrow Det_F: Subset Construction.
NDet_{FG} \rightarrow Det_{FG}: Breakpoint Construction.
NDet_{GF} \rightarrow {Det_{Rabin} or Det_{Streett}}: Safra Algorithm.

Boolean Operations on ω -Automata
Complement

$$\neg A_{\forall}(Q, \mathcal{I}, \mathcal{R}, \mathcal{F}) = A_{\exists}(Q, \mathcal{I}, \mathcal{R}, \neg \mathcal{F})$$

$$\neg A_{\exists}(Q, \mathcal{I}, \mathcal{R}, \mathcal{F}) = A_{\forall}(Q, \mathcal{I}, \mathcal{R}, \neg \mathcal{F})$$

Conjunction

$$(A_{\exists}(Q_1, \mathcal{I}_1, \mathcal{R}_1, \mathcal{F}_1) \wedge A_{\exists}(Q_2, \mathcal{I}_2, \mathcal{R}_2, \mathcal{F}_2)) = A_{\exists}(Q_1 \cup Q_2, \mathcal{I}_1 \wedge \mathcal{I}_2, \mathcal{R}_1 \wedge \mathcal{R}_2, \mathcal{F}_1 \wedge \mathcal{F}_2)$$

Disjunction

$$(A_{\exists}(Q_1, \mathcal{I}_1, \mathcal{R}_1, \mathcal{F}_1) \vee A_{\exists}(Q_2, \mathcal{I}_2, \mathcal{R}_2, \mathcal{F}_2)) = A_{\exists}\left(\begin{matrix} Q_1 \cup Q_2 \cup \{q\}, \\ (\neg q \wedge \mathcal{I}_1) \vee (q \wedge \mathcal{I}_2), \\ (\neg q \wedge \mathcal{R}_1 \wedge \neg q') \vee (q \wedge \mathcal{R}_2 \wedge q'), \\ (\neg q \wedge \mathcal{F}_1) \vee (q \wedge \mathcal{F}_2) \end{matrix}\right)$$

If both automata are totally defined,

$$(A_{\exists}(Q_1, \mathcal{I}_1, \mathcal{R}_1, \mathcal{F}_1) \vee A_{\exists}(Q_2, \mathcal{I}_2, \mathcal{R}_2, \mathcal{F}_2)) = A_{\exists}(Q_1 \cup Q_2, \mathcal{I}_1 \wedge \mathcal{I}_2, \mathcal{R}_1 \wedge \mathcal{R}_2, \mathcal{F}_1 \vee \mathcal{F}_2)$$

Eliminate Nesting - Acceptance condition **must** be an automata of the same type

$$A_{\exists}(Q^1, \mathcal{I}_1^1, \mathcal{R}_1^1, A_{\exists}(Q^2, \mathcal{I}_1^2, \mathcal{R}_1^2, \mathcal{F}_1)) = A_{\exists}(Q^1 \cup Q^2, \mathcal{I}_1^1 \wedge \mathcal{I}_1^2, \mathcal{R}_1^1 \wedge \mathcal{R}_1^2, \mathcal{F}_1))$$

Boolean Operations of G

(1) $\neg G\varphi = F\neg\varphi$ (2) $G\varphi \wedge G\psi = G[\varphi \wedge \psi]$
(3) $G\varphi \vee G\psi = A_{\exists}(\{p, q\}, p \wedge q, [p' \leftrightarrow p \wedge \varphi] \wedge [q' \leftrightarrow q \wedge \psi], G[p \vee q])$

Boolean Operations of F

(1) $\neg F\varphi = G\neg\varphi$ (2) $F\varphi \vee F\psi = F[\varphi \vee \psi]$
(3) $F\varphi \wedge F\psi = A_{\exists}(\{p, q\}, \neg p \wedge \neg q, [p' \leftrightarrow p \vee \varphi] \wedge [q' \leftrightarrow q \vee \psi], F[p \wedge q])$

Boolean Operations of FG

(1) $\neg FG\varphi = GF\neg\varphi$ (2) $FG\varphi \wedge FG\psi = FG[\varphi \wedge \psi]$
(3) $FG\varphi \vee FG\psi = A_{\exists}(\{q\}, \neg q, q' \leftrightarrow (q \Rightarrow \psi | \neg \varphi), FG[\neg q \vee \psi])$

Boolean Operations of GF

(1) $\neg GF\varphi = FG\neg\varphi$ (2) $GF\varphi \vee GF\psi = GF[\varphi \vee \psi]$

(3) $GF\varphi \wedge GF\psi = A_{\exists}(\{q\}, \neg q, q' \leftrightarrow (q \Rightarrow \neg \psi | \varphi), GF[q \wedge \psi])$

Transformation of Acceptance Conditions

Reduction of G
 $G\varphi = A_{\exists}(\{q\}, q, \varphi \wedge q \wedge q', Fq)$
 $G\varphi = A_{\exists}(\{q\}, q, q' \leftrightarrow q \wedge \varphi, FGq)$
 $G\varphi = A_{\exists}(\{q\}, q, q' \leftrightarrow q \wedge \varphi, GFq)$
Reduction of F

$F\varphi$ can **not** be expressed by $NDet_G$
 $F\varphi = A_{\exists}(\{q\}, \neg q, q' \leftrightarrow q \vee \varphi, FGq)$
 $F\varphi = A_{\exists}(\{q\}, \neg q, q' \leftrightarrow q \vee \varphi, GFq)$
Reduction of FG
 $FG\varphi$ can **not** be expressed by $NDet_G$
 $FG\varphi = A_{\exists}(\{q\}, \neg q, q \rightarrow \varphi \wedge q', Fq)$

$$FG\varphi = A_{\exists}\left(\begin{matrix} \{p, q\}, & \neg p \wedge \neg q, \\ \left[\begin{matrix} (p \rightarrow p') \wedge (p' \rightarrow p \vee \neg q) \wedge \\ (q' \leftrightarrow (p \wedge \neg q \vee \neg \varphi) \vee (p \wedge q)) \end{matrix} \right], & \\ G\neg q \wedge Fp \end{matrix}\right)$$

$$FG\varphi = A_{\exists}\left(\begin{matrix} \{p, q\}, & \neg p \wedge \neg q, \\ \left[\begin{matrix} (p \rightarrow p') \wedge (p' \rightarrow p \vee \neg q) \wedge \\ (q' \leftrightarrow (p \wedge \neg q \vee \neg \varphi) \vee (p \wedge q)) \end{matrix} \right], & \\ GF[p \wedge \neg q] \end{matrix}\right)$$

TEMPORAL LOGICS

- (S1) Pure LTL: AFGa
 - (S2) LTL + CTL: AFa
 - (S3) Pure CTL: AGEFa
 - (S4) CTL*: AFGa \vee AGEFa
- Remarks** *Beware of Finite Paths*
E and A quantify over infinite paths.
 $\triangleright A\varphi$ holds on every state that has no infinite path;
 $\triangleright E\varphi$ is false on states that have no infinite path;
A0 holds on states with only finite paths;
E1 is false on state with only finite paths;
 $\Diamond 0$ holds on states with no successor states;
 $\Diamond 1$ holds on states with successor states.

$F\varphi = \varphi \vee X F\varphi$ $G\varphi = \varphi \wedge X G\varphi$
 $[\varphi U \psi] = \psi \vee (\varphi \wedge X[\varphi U \psi])$
 $[\varphi B \psi] = \neg \psi \wedge (\varphi \vee X[\varphi B \psi])$
 $[\varphi W \psi] = (\psi \wedge \varphi) \vee (\neg \psi \wedge X[\varphi W \psi])$
LTL Syntactic Sugar: analog for past operators
 $G\varphi = \neg[1 \underline{U} (\neg \varphi)]$ $F\varphi = [1 \underline{U} \varphi]$
 $[\varphi W \psi] = \neg[(\neg \varphi \vee \neg \psi) \underline{U} (\neg \varphi \wedge \psi)]$
 $[\varphi W \psi] = [(\neg \psi) \underline{U} (\varphi \wedge \psi)]$ ($\neg \psi$ holds until $\varphi \wedge \psi$)
 $[\varphi B \psi] = \neg[(\neg \varphi) \underline{U} \psi]$
 $[\varphi B \psi] = [(\neg \psi) \underline{U} (\varphi \wedge \neg \psi)]$ (ψ can't hold when φ holds)
 $[\varphi U \psi] = \neg[(\neg \psi) \underline{U} (\varphi \wedge \neg \psi)]$
 $[\varphi U \psi] = [\varphi \underline{U} \psi] \vee G\varphi$
 $[\varphi \underline{U} \psi] = \neg[(\neg \psi) U (\neg \varphi \wedge \neg \psi)]$
 $[\varphi \underline{U} \psi] = \neg[(\neg \psi) W (\varphi \rightarrow \psi)]$
 $[\varphi \underline{U} \psi] = [\psi \underline{W} (\varphi \rightarrow \psi)]$
 $[\varphi \underline{U} \psi] = \neg[(\neg \varphi) B \psi]$ (φ doesn't matter when ψ holds)
 $[\varphi \underline{U} \psi] = [\psi \underline{B} (\neg \varphi \wedge \neg \psi)]$

CTL Syntactic Sugar: analog for past operators
Existential Operators

$EF\varphi = E[1 \underline{U} \varphi]$
 $EG\varphi = E[\varphi \underline{U} 0]$
 $E[\varphi U \psi] = E[\varphi \underline{U} \psi] \vee EG\varphi$
 $E[\varphi B \psi] = E[(\neg \psi) \underline{U} (\varphi \wedge \neg \psi)] \vee EG\neg \psi$
 $E[\varphi B \psi] = E[(\neg \psi) \underline{U} (\varphi \wedge \neg \psi)]$
 $E[\varphi B \psi] = E[(\neg \psi) \underline{U} (\varphi \wedge \neg \psi)]$
 $E[\varphi \underline{B} \psi] = E[(\neg \psi) \underline{U} (\varphi \wedge \neg \psi)]$
 $E[\varphi W \psi] = E[(\neg \psi) \underline{U} (\varphi \wedge \neg \psi)] \vee EG\neg \psi$
 $E[\varphi W \psi] = E[(\neg \psi) \underline{U} (\varphi \wedge \neg \psi)]$
 $E[\varphi W \psi] = E[(\neg \psi) \underline{U} (\varphi \wedge \neg \psi)]$
Universal Operators
 $AX\varphi = \neg EX\neg \varphi$
 $AG\varphi = \neg E[1 \underline{U} \neg \varphi]$

$AF\varphi = \neg EG\neg\varphi$
 $AF\varphi = \neg E[(\neg\varphi) \ U \ 0]$
 $A[\varphi \ U \ \psi] = \neg E[(\neg\psi) \ \underline{U} \ (\neg\varphi \wedge \neg\psi)]$
 $A[\varphi \ \underline{U} \ \psi] = \neg E[(\neg\psi) \ \underline{U} \ (\neg\varphi \wedge \neg\psi)] \wedge \neg EG\neg\psi$
 $A[\varphi \ \underline{U} \ \psi] = \neg E[(\neg\psi) \ \underline{U} \ (\neg\varphi \wedge \neg\psi)]$
 $A[\varphi \ \bar{B} \ \psi] = \neg E[(\neg\varphi) \ \underline{U} \ \psi]$
 $A[\varphi \ \bar{B} \ \psi] = \neg E[(\neg\varphi) \ \underline{U} \ \psi]$
 $A[\varphi \ \bar{B} \ \psi] = \neg E[(\neg\varphi \vee \psi) \ \underline{U} \ \psi] \wedge \neg EG(\neg\varphi \vee \psi)$
 $A[\varphi \ W \ \psi] = \neg E[(\neg\psi) \ \underline{U} \ (\neg\varphi \wedge \psi)]$
 $A[\varphi \ W \ \psi] = \neg E[(\neg\psi) \ \underline{U} \ (\neg\varphi \wedge \psi)] \wedge \neg EG\neg\psi$
 $A[\varphi \ W \ \psi] = \neg E[(\neg\psi) \ \underline{U} \ (\neg\varphi \wedge \psi)]$
CTL* to CTL - Existential Operators
 $EX\varphi = EXE\varphi$
 $EF\varphi = EFE\varphi$ $EF\bar{G}\varphi \equiv EFEG\varphi$
 $E[\varphi \ W \ \psi] = E[(E\varphi) \ W \ \psi]$
 $E[\varphi \ W \ \psi] = E[(E\varphi) \ W \ \psi]$
 $E[\psi \ \underline{U} \ \varphi] = E[\psi \ U \ E(\varphi)]$
 $E[\psi \ \underline{U} \ \varphi] = E[\psi \ U \ E(\varphi)]$
 $E[\varphi \ \bar{B} \ \psi] = E[(E\varphi) \ \bar{B} \ \psi]$
 $E[\varphi \ \bar{B} \ \psi] = E[(E\varphi) \ \bar{B} \ \psi]$
obs. $EGF\varphi \neq EGEF\varphi \rightarrow$ can't be converted
CTL* to CTL - Universal Operators
 $AX\varphi = AXA\varphi$
 $AG\varphi = AGA\varphi$
 $A[\varphi \ W \ \psi] = A[(A\varphi) \ W \ \psi]$
 $A[\varphi \ W \ \psi] = A[(A\varphi) \ W \ \psi]$
 $A[\varphi \ \underline{U} \ \psi] = A[A(\varphi) \ \underline{U} \ \psi]$
 $A[\varphi \ \underline{U} \ \psi] = A[A(\varphi) \ \underline{U} \ \psi]$
 $A[\psi \ \bar{B} \ \varphi] = A[\psi \ \bar{B} \ (E(\varphi))]$
 $A[\psi \ \bar{B} \ \varphi] = A[\psi \ \bar{B} \ (E(\varphi))]$
Weak Equivalences
 $*[\varphi U \psi] := [\varphi \underline{U} \psi] \vee G\varphi$ $*[\varphi B \psi] := [\varphi \underline{B} \psi] \vee G\neg\psi$
 $*\text{same to past version}$
 $[\varphi W \psi] := \neg[(\neg\varphi) \underline{W} \psi]$ (if ψ never holds : true!)
 $\bar{X}\varphi := \neg \bar{X}\neg\varphi$ (at $t0$: weak true. strong false)
Negation Normal Form
 $\neg(\varphi \wedge \psi) = \neg\varphi \vee \neg\psi$ $\neg(\varphi \vee \psi) = \neg\varphi \wedge \neg\psi$
 $\neg\neg\varphi = \varphi$ $\neg X\varphi = X\neg\varphi$
 $\neg G\varphi = F\neg\varphi$ $\neg F\varphi = G\neg\varphi$
 $\neg[\varphi \ U \ \psi] = [(\neg\varphi) \ \bar{B} \ \psi]$ $\neg[\varphi \ U \ \psi] = [(\neg\varphi) \ \bar{B} \ \psi]$
 $\neg[\varphi \ \bar{B} \ \psi] = [(\neg\varphi) \ \underline{U} \ \psi]$ $\neg[\varphi \ \bar{B} \ \psi] = [(\neg\varphi) \ \underline{U} \ \psi]$
 $\neg A\varphi = E\neg\varphi$ $\neg E\varphi = A\neg\varphi$
 $\neg X\varphi = \bar{X}\neg\varphi$ $\neg \bar{X}\varphi = \bar{X}\neg\varphi$
 $\neg G\varphi = F\neg\varphi$ $\neg F\varphi = G\neg\varphi$
 $\neg[\varphi \ \underline{U} \ \psi] = [(\neg\varphi) \ \bar{B} \ \psi]$ $\neg[\varphi \ \underline{U} \ \psi] = [(\neg\varphi) \ \bar{B} \ \psi]$
 $\neg[\varphi \ \bar{B} \ \psi] = [(\neg\varphi) \ \underline{U} \ \psi]$ $\neg[\varphi \ \bar{B} \ \psi] = [(\neg\varphi) \ \underline{U} \ \psi]$
Equivalences and Tips
 $[\varphi \underline{U} \psi] \equiv \varphi$ don't matter when ψ hold
 $[\varphi \bar{B} \psi] \equiv \psi$ can't hold when φ hold
 $[\varphi W \psi] \equiv \neg\psi$ hold until $\varphi \wedge \psi$
 $[\varphi U \psi] \equiv [\varphi \underline{U} \psi] \vee G\varphi$
 $[a U Fb] \equiv Fb$
 $F\psi \equiv [1 U \psi]$
 $F[a U b] \equiv Fb \equiv [Fa U Fb]$
 $[\varphi B \psi] \equiv [\varphi \bar{B} \psi] \vee G\neg\psi$
 $F[a B b] \equiv F[a \wedge \neg b]$
 $[\varphi W \psi] \equiv \neg[\neg\varphi \underline{W} \psi]$
 $FF\varphi \equiv F\varphi$ $\bullet GG\varphi \equiv G\varphi$
 $GF\varphi \equiv XGF\varphi \equiv FGF\varphi \equiv GGF\varphi \equiv GFGF\varphi \equiv$
 $FGGF\varphi$
 $F\bar{G}\varphi \equiv XFG\varphi \equiv FFG\varphi \equiv GFG\varphi \equiv GFFG\varphi \equiv$
 $FGFG\varphi$
 $GF(x \vee y) \equiv GFx \vee GFy$
 $E(\varphi \wedge \psi) \equiv E\varphi \wedge E\psi$ (Careful! Only sometimes!)
 $E(\varphi \vee \psi) \equiv E\varphi \vee E\psi$ (Careful! Only sometimes!)
 $E[(a \underline{U} b) \wedge (c \underline{U} d)] \equiv$

$E[(a \wedge c) \underline{U} (b \wedge E(c \underline{U} d) \vee d \wedge E(a \underline{U} b))]$
 $AEA \equiv A$ $\bullet G\bar{F}X \equiv G\bar{X}F$ $\bullet AG\bar{X}F \equiv AXGF$
 $AG(\varphi \wedge \psi) \equiv A(G\varphi \wedge G\psi) \equiv AG\varphi \wedge AG\psi$
 $A(G[a \underline{U} b]) \equiv G(a \vee b)$
 $A(G[a \bar{B} b]) \equiv G(\neg b)$
 $A(G[a \bar{W} b]) \equiv G(b \rightarrow a)$
 $A(G[a \underline{U} b]) \equiv G(a \vee b) \wedge G\bar{F}b$
 $A(G[a \bar{B} b]) \equiv G(\neg b) \wedge G\bar{F}a$
► The following are initially but not generally valid
 $A(G\bar{X}a \equiv Ga)$ $\bullet A(G\bar{X}a \equiv false)$
 $A(G\bar{G}a \equiv Ga)$ $\bullet A(G\bar{F}a \equiv a)$
 $A(G[a \underline{U} b]) \equiv G(a \vee b)$ $\bullet A(G[a \underline{U} b]) \equiv b \wedge G(a \vee b)$
 $A(G[a \bar{B} b]) \equiv G(\neg b)$ $\bullet A(G[a \bar{B} b]) \equiv a \wedge G(\neg b)$
 $A(G[a \bar{W} b]) \equiv G(b \rightarrow a)$ $\bullet A(G[a \bar{W} b]) \equiv b \wedge G(b \rightarrow a)$
 $A(F\bar{X}a \equiv true)$ $\bullet A(F\bar{X}a \equiv Fa)$
 $A(F\bar{G}a \equiv a)$ $\bullet A(F\bar{F}a \equiv Fa)$
 $A(F[a \underline{U} b]) \equiv Fb \vee F\bar{G}a$ $\bullet A(F[a \underline{U} b]) \equiv Fb$
 $A(F[a \bar{B} b]) \equiv F(a \wedge \neg b) \vee F\bar{G}(\neg a \wedge \neg b)$
 $A(F[a \bar{B} b]) \equiv F(a \wedge \neg b)$
 $A(F[a \bar{W} b]) \equiv F(a \wedge b) \vee FPG\neg b$
 $A(F[a \bar{W} b]) \equiv F(a \wedge b)$
Eliminate boolean op. after path quantify
 $[\varphi_1 \ \underline{U} \ \psi_1] \wedge [\varphi_2 \ \underline{U} \ \psi_2] =$

$$\left[(\varphi_1 \wedge \varphi_2) \ \underline{U} \ \left(\begin{matrix} \psi_1 \wedge [\varphi_2 \ \underline{U} \ \psi_2]^\vee \\ \psi_2 \wedge [\varphi_1 \ \underline{U} \ \psi_1] \end{matrix} \right) \right]$$

 $[\varphi_1 \ \underline{U} \ \psi_1] \wedge [\varphi_2 \ U \ \psi_2] =$

$$\left[(\varphi_1 \wedge \varphi_2) \ \underline{U} \ \left(\begin{matrix} \psi_1 \wedge [\varphi_2 \ U \ \psi_2]^\vee \\ \psi_2 \wedge [\varphi_1 \ \underline{U} \ \psi_1] \end{matrix} \right) \right]$$

 $[\varphi_1 \ U \ \psi_1] \wedge [\varphi_2 \ U \ \psi_2] =$

$$\left[(\varphi_1 \wedge \varphi_2) \ \underline{U} \ \left(\begin{matrix} \psi_1 \wedge [\varphi_2 \ U \ \psi_2]^\vee \\ \psi_2 \wedge [\varphi_1 \ U \ \psi_1] \end{matrix} \right) \right]$$

##MONADIC PREDICATE
LO2
first order terms are defined as:
 $-t \in V_\Sigma [typ_\Sigma(t) = \mathbb{N} \subseteq Term_\Sigma^{LO2}]$
formulas LO2 are defined as:
 $-t1 < t2 \in L_{LO2}$
 $-p^{(t)} \in L_{LO2}$
 $-\neg\varphi, \varphi \wedge \psi \in L_{LO2}$
 $-\exists t. \varphi \in L_{LO2}$
 $-\exists p. \varphi \in L_{LO2}$
where:
 $-t, t1, t2 \in V_\Sigma$ with $typ_\Sigma(t) = typ_\Sigma(t1) =$
 $typ_\Sigma(t2) = \mathbb{N}$
 $-\varphi, \psi \in \zeta_{LO2}$
 $-t \in V_\Sigma$ with $typ_\Sigma(t) = \mathbb{N}$
 $-p \in V_\Sigma$ with $typ_\Sigma(p) = \mathbb{N} \rightarrow \mathbb{B}$
##TRANSLATIONS
CTL* Modelchecking to LTL model checking
Let's φ_i be a pure path formula (without path quantifiers), Ψ be a propositional formula, abbreviate subformulas $E\varphi$ and $A\psi$ working bottom-up the syntax tree to obtain the following
normal form: $\phi = \text{let } \begin{bmatrix} x_1 = A\varphi_1 \\ \vdots \\ x_n = A\varphi_n \end{bmatrix} \text{ in } \Psi \text{ end}$
Use LTL model checking to compute
 $Q_i := \llbracket A\varphi_i \rrbracket \mathcal{K}_{i-1}$, where $\mathcal{K}_0 := \mathcal{K}$ and \mathcal{K}_{i+1} is obtained from \mathcal{K}_i by labelling the states Q_i with x_i .
Finally compute $\llbracket \Psi \rrbracket \mathcal{K}_n$
function LO2_S1S(Φ)

case Φ of
 $t1 < t2$: **return** $\exists p. [\forall t. p^{(t)} \rightarrow$
 $p^{(SUC(t))}] \wedge \neg p^{(t1)} \wedge p^{(t2)}$:
 $p^{(t)} : \text{return } p^{(t)}$;
 $\neg\varphi$: **return** $\neg LO2_S1S(\varphi)$;
 $\varphi \wedge \psi$: **return** $LO2_S1S(\varphi) \wedge LO2_S1S(\psi)$;
 $\exists t. \varphi$: **return** $\exists t. LO2_S1S(\varphi)$;
 $\exists p. \varphi$: **return** $\exists p. LO2_S1S(\varphi)$;
end
end
function S1S_LO2(Φ)
case Φ of
 $p^{(n)}$:
return $\exists t0...tn. p^{(tn)} \wedge zero(t0) \wedge \bigwedge_{i=0}^{n-1} succ(ti, ti+1)$;
 $p^{(t0+n)}$:
return $\exists t1...tn. p^{(tn)} \wedge \bigwedge_{i=0}^{n-1} succ(ti, ti+1)$;
 $\neg\varphi$: **return** $\neg S1S_LO2(\varphi)$;
 $\varphi \wedge \psi$: **return** $S1S_LO2(\varphi) \wedge S1S_LO2(\psi)$;
 $\exists t. \varphi$: **return** $\exists t. S1S_LO2(\varphi)$;
 $\exists p. \varphi$: **return** $\exists p. S1S_LO2(\varphi)$;
end
end
function Tp2Od($t0, \Phi$) temporal to LO1
case Φ of
 $is_var(\Phi) : \Psi(t0)$;
 $\neg\varphi$: **return** $\neg Tp2Od(\varphi)$;
 $\varphi \wedge \psi$: **return** $Tp2Od(\varphi) \wedge Tp2Od(\psi)$;
 $\varphi \vee \psi$: **return** $Tp2Od(\varphi) \vee Tp2Od(\psi)$;
 $X\varphi : \Psi$: $\Psi := \exists t1. (t0 < t1) \wedge$
 $\forall t2. t0 < t2 \rightarrow t1 \leq t2) \wedge Tp2Od(t1, \varphi)$;
 $[\varphi \underline{U} \psi] : \Psi$: $\Psi := \exists t1. t0 \leq t1 \wedge Tp2Od(t1, \psi) \wedge$
 $interval((t0, 1, t1, 0), \varphi)$;
 $[\varphi \bar{B} \psi] : \Psi$: $\Psi := \forall t1. t0 \leq t1 \wedge$
 $interval((t0, 1, t1, 0), \neg\varphi) \rightarrow Tp2Od(t1, \neg\psi)$;
 $\bar{X}\varphi : \Psi$: $\Psi := \forall t1. (t1 < t0) \wedge$
 $(\forall t2. t2 < t0 \rightarrow t2 \leq t1) \rightarrow Tp2Od(t1, \varphi)$;
 $\bar{X}\varphi : \Psi$: $\Psi := \exists t1. (t1 < t0) \wedge$
 $(\forall t2. t2 < t0 \rightarrow t2 \leq t1) \wedge Tp2Od(t1, \varphi)$;
 $[\varphi \underline{U} \psi] : \Psi$: $\Psi := \exists t1. t1 \leq t0 \wedge Tp2Od(t1, \psi) \wedge$
 $interval((t1, 0, t0, 1), \varphi)$;
 $[\varphi \bar{B} \psi] : \Psi$: $\Psi := \forall t1. t1 \leq t0 \wedge$
 $interval((t1, 0, t0, 1), \neg\varphi) \rightarrow Tp2Od(t1, \neg\psi)$;
end
return Ψ
end
function interval(l, φ)
case Φ of
 $(t0, 0, t1, 0)$:
return $\forall t2. t0 < t2 \wedge t2 < t1 \rightarrow Tp2Od(t2, \varphi)$;
 $(t0, 0, t1, 1)$:
return $\forall t2. t0 < t2 \wedge t2 \leq t1 \rightarrow Tp2Od(t2, \varphi)$;
 $(t0, 1, t1, 0)$:
return $\forall t2. t0 \leq t2 \wedge t2 < t1 \rightarrow Tp2Od(t2, \varphi)$;
 $(t0, 1, t1, 1)$:
return $\forall t2. t0 \leq t2 \wedge t2 \leq 3t1 \rightarrow Tp2Od(t2, \varphi)$;
end
end
 ω -Automaton to LO2
 $A_\exists(q1, ..., qn, \psi1, \psi R, \psi F)$ (input automaton)
 $\exists q1..qn. \Theta LO2(0, \psi I) \wedge (\forall t. \Theta LO2(t, \psi R)) \wedge$
 $(\forall t1 \exists t2. t1 < t2 \wedge \Theta LO2(t2, \psi F))$
Where $\Theta LO2(t, \Phi)$ is:
 $\neg \Theta LO2(t, p) := p(t)$ for variable p
 $\neg \Theta LO2(t, X\psi) := \Theta LO2(t+1, \psi)$
 $\neg \Theta LO2(t, \neg\psi) := \neg \Theta LO2(t, \psi)$
 $\neg \Theta LO2(t, \varphi \wedge \psi) := \Theta LO2(t, \varphi) \wedge \Theta LO2(t, \psi)$

$\neg \Theta LO2(t, \varphi \vee \psi) := \Theta LO2(t, \varphi) \vee \Theta LO2(t, \psi)$
LTL to ω -automata (from inside out the tree)
 $\Phi(X\varphi)_x \Leftrightarrow A_\exists(\{q\}, 1, q \leftrightarrow X\varphi, \Phi(q)_x)$
 $\Phi(X\varphi)_x \Leftrightarrow$
 $A_\exists(\{q0, q1\}, 1, (q0 \leftrightarrow \varphi) \wedge (q1 \leftrightarrow Xq0), \Phi(q1)_x)$
 $\Phi(G\varphi)_x \Leftrightarrow$
 $A_\exists(\{q\}, 1, q \leftrightarrow \varphi \wedge Xq, \Phi(q)_x \wedge GF[\varphi \rightarrow q])$
 $\Phi(F\varphi)_x \Leftrightarrow$
 $A_\exists(\{q\}, 1, q \leftrightarrow \varphi \vee Xq, \Phi(q)_x \wedge GF[q \rightarrow \varphi])$
 $\Phi([\varphi \ U \ \psi])_x \Leftrightarrow$
 $A_\exists(\{q\}, 1, q \leftrightarrow \psi \vee \varphi \wedge Xq, \Phi(q)_x \wedge GF[\varphi \rightarrow q])$
 $\Phi([\varphi \ \underline{U} \ \psi])_x \Leftrightarrow$
 $A_\exists(\{q\}, 1, q \leftrightarrow \psi \vee \varphi \wedge Xq, \Phi(q)_x \wedge GF[q \rightarrow \psi])$
 $\Phi([\varphi \ \bar{B} \ \psi])_x \Leftrightarrow$
 $A_\exists(\{q\}, 1, q \leftrightarrow \neg\psi \wedge (\varphi \vee Xq), \Phi(q)_x \wedge GF[q \vee \psi])$
 $\Phi([\varphi \ \bar{B} \ \psi])_x \Leftrightarrow$
 $A_\exists(\{q\}, 1, q \leftrightarrow \neg\psi \wedge (\varphi \vee Xq), \Phi(q)_x \wedge GF[q \rightarrow \varphi])$
 $\Phi(\bar{X}\varphi)_x \Leftrightarrow A_\exists(\{q\}, q, Xq \leftrightarrow \varphi, \Phi(q)_x)$
 $\Phi(\bar{X}\varphi)_x \Leftrightarrow A_\exists(\{q\}, \neg q, Xq \leftrightarrow \varphi, \Phi(q)_x)$
 $\Phi(\bar{G}\varphi)_x \Leftrightarrow A_\exists(\{q\}, q, Xq \leftrightarrow \varphi \wedge q, \Phi(\varphi \wedge q)_x)$
 $\Phi(\bar{F}\varphi)_x \Leftrightarrow A_\exists(\{q\}, \neg q, Xq \leftrightarrow \varphi \vee q, \Phi(\varphi \vee q)_x)$
 $\Phi([\varphi \ \underline{U} \ \psi])_x \Leftrightarrow$
 $A_\exists(\{q\}, q, Xq \leftrightarrow \psi \vee \varphi \wedge q, \Phi(\psi \vee \varphi \wedge q)_x)$
 $\Phi([\varphi \ \underline{U} \ \psi])_x \Leftrightarrow$
 $A_\exists(\{q\}, \neg q, Xq \leftrightarrow \psi \vee \varphi \wedge q, \Phi(\psi \vee \varphi \wedge q)_x)$
 $\Phi([\varphi \ \bar{B} \ \psi])_x \Leftrightarrow$
 $A_\exists(\{q\}, q, Xq \leftrightarrow \neg\psi \wedge (\varphi \vee q), \Phi(\neg\psi \wedge (\varphi \vee q))_x)$
 $\Phi([\varphi \ \bar{B} \ \psi])_x \Leftrightarrow$
 $A_\exists(\{q\}, \neg q, Xq \leftrightarrow \neg\psi \wedge (\varphi \vee q), \Phi(\neg\psi \wedge (\varphi \vee q))_x)$
CTL to μ -Calculus ($\Phi_{inf} = \nu y. \Diamond y$)
 $EX\varphi = \Diamond(\Phi_{inf} \wedge \varphi)$
 $EG\varphi = \nu x. \varphi \wedge \Diamond x$
 $EF\varphi = \mu x. \Phi_{inf} \wedge \varphi \vee \Diamond x$
 $E[\varphi \underline{U} \psi] = \mu x. (\Phi_{inf} \wedge \psi) \vee \varphi \wedge \Diamond x$
 $E[\varphi U \psi] = \nu x. (\Phi_{inf} \wedge \psi) \vee \varphi \wedge \Diamond x$
 $E[\varphi \bar{B} \psi] = \mu x. \neg\psi \wedge (\Phi_{inf} \wedge \varphi \vee \Diamond x)$
 $E[\varphi B \psi] = \nu x. \neg\psi \wedge (\Phi_{inf} \wedge \varphi \vee \Diamond x)$
 $AX\varphi = \Box(\Phi_{inf} \rightarrow \varphi)$
 $AG\varphi = \nu x. (\Phi_{inf} \rightarrow \varphi) \wedge \Box x$
 $AF\varphi = \mu x. \varphi \vee \Box x$
 $A[\varphi \underline{U} \psi] = \mu x. \psi \vee (\Phi_{inf} \rightarrow \varphi) \wedge \Box x$
 $A[\varphi U \psi] = \nu x. \psi \vee (\Phi_{inf} \rightarrow \varphi) \wedge \Box x$
 $A[\varphi \bar{B} \psi] = \mu x. (\Phi_{inf} \rightarrow \neg\psi) \wedge (\varphi \vee \Box x)$
 $A[\varphi B \psi] = \nu x. (\Phi_{inf} \rightarrow \neg\psi) \wedge (\varphi \vee \Box x)$
G and μ -calculus (safety property)
 $[\nu x. \varphi \wedge \Diamond x]_K$
-Contains states s where an infinite path π starts with $\forall t. \pi^{(t)} \in [\varphi]_K$
 $\neg\varphi$ holds always on π
F and μ -calculus (liveness property)
 $[\mu x. \varphi \vee \Diamond x]_K$
-Contains states s where a (possibly finite) path π starts with $\exists t. \pi^{(t)} \in [\varphi]_K$
 $\neg\varphi$ holds at least once on π
FG and μ -calculus (persistence property)
 $[\mu y. [\nu x. \varphi \wedge \Diamond x] \vee \Diamond y]_K$
-Contains states s where an infinite path π starts with $\exists t1. \forall t2. \pi^{(t1+t2)} \in [\varphi]_K$
 $\neg\varphi$ holds after some point on π
GF and μ -calculus (fairness property)
 $[\nu y. [\mu x. (y \wedge \varphi) \vee \Diamond x]_K]$
-Contains states s where an infinite path π starts with $\forall t1. \exists t2. \pi^{(t1+t2)} \in [\varphi]_K$?????? $t1 + t2$ or $t1 + t0$?????
 $\neg\varphi$ holds infinitely often on π