Propositional Logic - Syntactic Sugar

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

$\varphi \oplus \psi := (\varphi \wedge \neg\psi) \vee (\psi \wedge \neg\varphi)$ $\qquad$ $\varphi \barwedge \psi := \neg(\varphi \wedge \psi)$

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

Satisfiability, Validity and Equivalence

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

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

Conjunctive Normal Form: from truth table, take minterms that are 0. Each minterm is built as an OR of the negated variables. E.g.,
$(0,0,1) \rightarrow (x \vee y \vee \neg z)$.

Distributivity: $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$
$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$

Sequent Calculus:

1. Prove validity of ϕ: start with $\{\} \vdash \phi$; ϕ is valid iff $\Gamma \cap \Delta \neq \{\}$ for all leaves; else, counterexample: var is true, if $x \in \Gamma$; false otherwise; "don't care", if variable doesn't appear.

2. Prove satisfiability of ϕ: start with $\{\phi\} \vdash \{\}$; ϕ is satisfiable iff $\Gamma \cap \Delta = \{\}$ for at least one leaf. Satisfying interpretation: same as counterexample.

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

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

Linear Clause Forms (Computes CNF) - Bottom up 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.

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

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

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

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

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

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

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 $\underline{1}$ (tru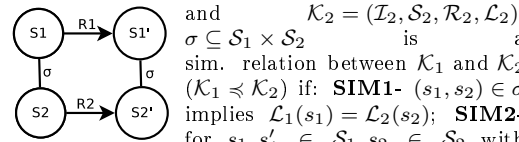e) $\rightarrow$ find implied variables. **(3)** For remaining variables: assume values and compute newly implied variables. **(4)** If contradiction reached: backtrack.

```
Apply(⊙, Bddnode a, b)
  int m; BddNode h, l;
  if isLeaf(a)&isLeaf(b)
    then
    return Eval(⊙,label(a),
                label(b));
  else
    m=max{label(a),label(b)}
    (a0,a1):=Ops(a,m);
    (b0,b1):=Ops(b,m);
    h:=Apply(⊙,a1,b1);
    l:=Apply(⊙,a0,b0);
    return CreateNode(m,h,l)
  end;
end
```

```
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)}
    (α0,α1):=Ops(α, m);
    (ψ0,ψ1):=Ops(ψ, m);
    h:=Compose(x,ψ1,α1);
    l:=Compose(x,ψ0,α0);
    return CreateNode(m,h,l)
  endif;end
```

```
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)}
    (i0,i1):=Ops(i,m);
    (j0,j1):=Ops(j,m);
    (k0,k1):=Ops(k,m);
    l:=ITE(i0,j0,k0);
    h:=ITE(i1, j1, k1);
    return CreateNode(m,h,l)
  end;end
```

```
Constrain(Φ, β)
  if β=0 then
    ret 0
  elseif Φ ∈ {0,1}(β = 1)
    ret Φ
  else
    m=max{label(β),label(Φ)}
    (Φ0, Φ1) :=Ops(Φ,m);
    (β0, β1) :=Ops(β,m);
    if β0=0
      ret Constrain(Φ1, β1)
    elseif β1=0 then
      ret Constrain(Φ0, β0)
    else
      l:=Constrain(Φ0, β0);
      h:=Constrain(Φ1, β1);
      ret CreateNode(m,h,l)
  endif;endif;end
```

```
Restrict(Φ, β)
  if β=0
    return 0
  elseif
    Φ ∈ {0,1} ∨ (β = 1)
    return Φ
  else
    m=max{label(β),label(Φ)}
    (Φ0, Φ1):=Ops(Φ,m);
    (β0, β1):=Ops(β,m)
    if β0=0
      return Restrict(Φ1, β1)
    elseif β1=0
      return Restrict(Φ0, β0)
    elseif m=label(Φ)
      return CreateNode(m,
        Restrict (Φ1, β1),
        Restrict (Φ0, β0))
    else
      return Restrict(Φ,
      Apply(∨, β0, β1))
  endif;endif;end
```

```
Ops(v,m)
  x:=label(v);
  if m=degree(x)
    return (low(v),high(v))
  else return(v, v)
end;end
--------------------
Other Diagrams:
TODO ZDD FDD
-----
```

$s_1, s_1' \in \mathcal{S}_1, s_2 \in \mathcal{S}_2, (s_1, s_2) \in \sigma, (s_1, s_1') \in \mathcal{R}_1$, imply that there is $s_2' \in \mathcal{S}_2$ with $(s_1', s_2') \in \sigma$ and $(s_2, s_2') \in \mathcal{R}_2$; **BISIM2b-** $s_2, s_2' \in \mathcal{S}_2, s_1 \in \mathcal{S}_1$, $(s_1, s_2) \in \sigma, (s_2, s_2') \in \mathcal{R}_2$, imply that there is $s_1' \in \mathcal{S}_1$ with $(s_1', s_2') \in \sigma$ and $(s_1, s_1') \in \mathcal{R}_1$; **BISIM3a-** for all $s_1 \in \mathcal{I}_1$, there is a $s_2 \in \mathcal{I}_2$ with $(s_1, s_2) \in \sigma$; **BISIM3b-** for all $s_1 \in \mathcal{I}_2$, there is a $s_2 \in \mathcal{I}_2$ with $(s_1, s_2) \in \sigma$.

Greatest Bisimulation Relation (Equivalence)

$(s_1, s_2) \in \mathcal{B}_0 \Leftrightarrow \mathcal{L}_1(s_1) = \mathcal{L}_2(s_2)$

$(s_1, s_2) \in \mathcal{B}_{i+1} \Leftrightarrow$

$\left(\begin{array}{c} (s_1, s_2) \in \mathcal{B}_i \wedge \\ \forall s_1' \in \mathcal{S}_1. \exists s_2' \in \mathcal{S}_2. \\ (s_1, s_1') \in \mathcal{R}_1 \rightarrow (s_2, s_2') \in \mathcal{R}_2 \wedge (s_1', s_2') \in \mathcal{B}_i \\ \forall s_2' \in \mathcal{S}_2. \exists s_1' \in \mathcal{S}_1. \\ (s_2, s_2') \in \mathcal{R}_2 \rightarrow (s_1, s_1') \in \mathcal{R}_1 \wedge (s_1', s_2') \in \mathcal{B}_i \end{array} \right)$

$\mathcal{B}_*$ is the greatest simulation relation if
$\mathcal{I}_1 \subseteq \{s_1 \in \mathcal{S}_1 | \exists s_2 \in \mathcal{I}_2.(s_1, s_2) \in \mathcal{B}_*\}$
$\mathcal{I}_2 \subseteq \{s_2 \in \mathcal{S}_2 | \exists s_1 \in \mathcal{I}_1.(s_1, s_2) \in \mathcal{B}_*\}$

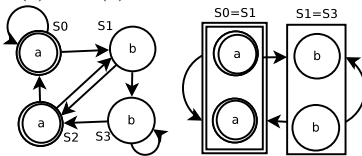**Quotient**: given $\mathcal{K} = (\mathcal{I}, \mathcal{S}, \mathcal{R}, \mathcal{L})$ and the equivalence relation $\sigma \subseteq \mathcal{S} \times \mathcal{S}$; Quotient structure $\mathcal{K}_{/\sigma} := (\widetilde{\mathcal{I}}, \widetilde{\mathcal{S}}, \widetilde{\mathcal{R}}, \widetilde{\mathcal{L}})$: $\widetilde{\mathcal{I}} := \{\{s' \in \mathcal{S} | (s, s') \in \sigma\} | s \in \mathcal{I}\}$

$\widetilde{\mathcal{S}} := \{\{s' \in \mathcal{S} | (s, s') \in \sigma\} | s \in \mathcal{S}\}$
$(\widetilde{s_1}, \widetilde{s_2}) \in \widetilde{\mathcal{R}} :\Leftrightarrow \exists s_1' \in \widetilde{s_1}. \exists s_2' \in \widetilde{s_2}.(s_1', s_2') \in \mathcal{R}$
$\widetilde{\mathcal{L}}(\widetilde{s}) := \mathcal{L}(s)$

Symbolic Product Computation - given $\mathcal{K}_1 = (\mathcal{V}_1, \varphi_\mathcal{I}, \varphi_\mathcal{R})$ and $\mathcal{K}_2 = (\mathcal{V}_2, \psi_\mathcal{I}, \psi_\mathcal{R})$, the product is: $\mathcal{K}_1 \times \mathcal{K}_2 = (\mathcal{V}_1 \cup \mathcal{V}_2, \varphi_\mathcal{I} \wedge \psi_\mathcal{I}, \varphi_\mathcal{R} \wedge \psi_\mathcal{R})$
Quantif. $\exists x.\varphi := [\varphi]_x^1 \vee [\varphi]_x^0 \clubsuit \forall x.\varphi := [\varphi]_x^1 \wedge [\varphi]_x^0$
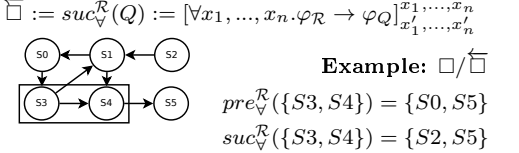**Predecessor and Successor**

$\diamondsuit := pre_\exists^\mathcal{R}(Q) := \exists x_1', ..., x_n'.\varphi_\mathcal{R} \wedge [\varphi_Q]_{x_1, ..., x_n}^{x_1', ..., x_n'}$

$\overleftarrow{\diamondsuit} := suc_\exists^\mathcal{R}(Q) := [\exists x_1, ..., x_n.\varphi_\mathcal{R} \wedge \varphi_Q]_{x_1', ..., x_n'}^{x_1, ..., x_n}$

$\square := pre_\forall^\mathcal{R}(Q) := \forall x_1', ..., x_n'.\varphi_\mathcal{R} \rightarrow [\varphi_Q]_{x_1, ..., x_n}^{x_1', ..., x_n'}$

$\overleftarrow{\square} := suc_\forall^\mathcal{R}(Q) := [\forall x_1, ..., x_n.\varphi_\mathcal{R} \rightarrow \varphi_Q]_{x_1', ..., x_n'}^{x_1, ..., x_n}$

Example: $\square / \overleftarrow{\square}$

$pre_\forall^\mathcal{R}(\{S3, S4\}) = \{S0, S5\}$
$suc_\forall^\mathcal{R}(\{S3, S4\}) = \{S2, S5\}$

```
pre_∀^R(Q = {S1, ..., Sn})
for each node n in K:
  if(n points to a node
       that is not in Q)
    n ∉ pre_∀^R(Q)
  else
    n ∈ pre_∀^R(Q)
```

```
suc_∀^R(Q = {S1, ..., Sn})
for each node n in K:
  if(n is pointed by a node
       that is not in Q)
    n ∉ suc_∀^R(Q)
  else
    n ∈ suc_∀^R(Q)
```

Tarski-Knaster Theorem: $\mu :=$ starts $\bot \rightarrow$ least fixpoint $\spadesuit$ $\nu :=$ starts $\top \rightarrow$ greatest fixpoint *

Rabin-Scott Subset Construction **1.** Initial state is a set of states containing all the initial states. **2.** For all transitions of a set of states, compute the successors and create a set of states containing all the possible reachable states when performing that transition. **3.** Acceptance condition are set of states containing acceptance states.

Local Model Checking

(1)	$\dfrac{s\vdash_\Phi \varphi\wedge\psi}{\{s\vdash_\Phi\varphi\} \quad \{s\vdash_\Phi\psi\}}\wedge$	(2)	$\dfrac{s\vdash_\Phi \varphi\vee\psi}{\{s\vdash_\Phi\varphi\} \quad \{s\vdash_\Phi\psi\}}\vee$
(3)	$\dfrac{s\vdash_\Phi \square\varphi}{\{s_1\vdash_\Phi\varphi\}...\{s_n\vdash_\Phi\varphi\}}\wedge$	(4)	$\dfrac{s\vdash_\Phi \diamondsuit\varphi}{\{s_1\vdash_\Phi\varphi\}...\{s_n\vdash_\Phi\varphi\}}$
(5)	$\dfrac{s\vdash_\Phi \square\varphi}{\{s_1'\vdash_\Phi\varphi\}...\{s_n'\vdash_\Phi\varphi\}}\wedge$	(6)	$\dfrac{s\vdash_\Phi \diamondsuit\varphi}{\{s_1'\vdash_\Phi\varphi\}...\{s_n'\vdash_\Phi\varphi\}}\vee$
$\dfrac{s\vdash_\Phi \mu x.\varphi}{s\vdash_\Phi\varphi}$	$\dfrac{s\vdash_\Phi \nu x.\varphi}{s\vdash_\Phi\varphi}$	$\dfrac{s\vdash_\Phi x}{s\vdash_\Phi\mathfrak{D}_\Phi(x)}$	$\mathfrak{D}_\Phi$(replace w. initial form.)

$\{s_1 ... s_n\} = suc_\exists^\mathcal{R}(s)$ and $\{s_1' ... s_n'\} = pre_\exists^\mathcal{R}(s)$

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	$apx_0(\nu x.\varphi) := 1$
$apx_{n+1}(\mu x.\varphi) := [\varphi]_x^{apxn(\mu x.\varphi)}$	
$apx_{n+1}(\nu x.\varphi) := [\varphi]_x^{apxn(\nu x.\varphi)}$	

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.
*** Breakpoint Construction 1.** Each state is composed by two components **2.** Initial state first component is a set of all initial states, and second component is the empty set. Ex.: $(\mathcal{I}, \{\})$. **3.** a successor for a state (Q,Q_f) is generated as follows:

$\begin{cases} \text{If } Q_f = \{\} & (suc_\exists^{\mathcal{R}_a}(Q), (suc_\exists^{\mathcal{R}_a}(Q) \cap \mathcal{F}) \\ \text{Otherwise} & (suc_\exists^{\mathcal{R}_a}(Q), (suc_\exists^{\mathcal{R}_a}(Q_f) \cap \mathcal{F}) \end{cases}$

4. Acceptance states are states where $Q_f \neq \{\}$.
Boolean Operations on ω-Automata

Complement

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

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

Conjunction

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

Disjunction

$(\mathcal{A}_\exists(Q_1, \mathcal{I}_1, \mathcal{R}_1, \mathcal{F}_1) \vee \mathcal{A}_\exists(Q_2, \mathcal{I}_2, \mathcal{R}_2, \mathcal{F}_2)) = $

$\mathcal{A}_\exists \left(\begin{array}{c} Q_1 \cup Q_2 \cup \{q\}, \\ (\neg q \wedge \mathcal{I}_1) \vee (q \wedge \mathcal{I}_2), \\ (\neg q \wedge \mathcal{R}_1 \wedge \neg q') \vee (q \wedge \mathcal{R}_2 \wedge q'), \\ (\neg q \wedge \mathcal{F}_1) \vee (q \wedge \mathcal{F}_2) \end{array} \right)$

If both automata are totally defined,

$(\mathcal{A}_\exists(Q_1, \mathcal{I}_1, \mathcal{R}_1, \mathcal{F}_1) \vee \mathcal{A}_\exists(Q_2, \mathcal{I}_2, \mathcal{R}_2, \mathcal{F}_2)) = $

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

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

$\mathcal{A}_\exists(Q^1, \mathcal{I}_1^1, \mathcal{R}_1^1, \mathcal{A}_\exists(Q^2, \mathcal{I}_1^2, \mathcal{R}_1^2, \mathcal{F}_1))$

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

Boolean Operations of G

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

Simulation: given $\mathcal{K}_1 = (\mathcal{I}_1, \mathcal{S}_1, \mathcal{R}_1, \mathcal{L}_1)$ and $\mathcal{K}_2 = (\mathcal{I}_2, \mathcal{S}_2, \mathcal{R}_2, \mathcal{L}_2)$; $\sigma \subseteq \mathcal{S}_1 \times \mathcal{S}_2$ is a sim. relation between $\mathcal{K}_1$ and $\mathcal{K}_2$ $(\mathcal{K}_1 \preceq \mathcal{K}_2)$ if: **SIM1-** $(s_1, s_2) \in \sigma$ implies $\mathcal{L}_1(s_1) = \mathcal{L}_2(s_2)$; **SIM2-** for $s_1, s_1' \in \mathcal{S}_1, s_2 \in \mathcal{S}_2$ with $(s_1, s_2) \in \sigma$ and $(s_1, s_1') \in \mathcal{R}_1$, there must be $s_2' \in \mathcal{R}_2$ with $(s_1', s_2') \in \sigma$ $(s_2, s_2') \in \mathcal{R}_2$; **SIM3-** for all $s_1 \in \mathcal{I}_1$, there is a $s_2 \in \mathcal{I}_2$ with $(s_1, s_2) \in \sigma$.

Greatest Simulation Relation

$(s_1, s_2) \in \mathcal{H}_0 \Leftrightarrow \mathcal{L}_1(s_1) = \mathcal{L}_2(s_2)$
$(s_1, s_2) \in \mathcal{H}_{i+1} \Leftrightarrow$

$\left(\begin{array}{c} (s_1, s_2) \in \mathcal{H}_i \wedge \\ \forall s_1' \in \mathcal{S}_1. \exists s_2' \in \mathcal{S}_2. \\ (s_1, s_1') \in \mathcal{R}_1 \rightarrow (s_2, s_2') \in \mathcal{R}_2 \wedge (s_1', s_2') \in \mathcal{H}_i \end{array} \right)$

$\mathcal{H}_*$ is the greatest simulation relation if **SIM3**:
$\mathcal{I}_1 \subseteq \{s_1 \in \mathcal{S}_1 | \exists s_2 \in \mathcal{I}_2.(s_1, s_2) \in \mathcal{H}_*\}$
Bisimulation: $\sigma \subseteq \mathcal{S}_1 \times \mathcal{S}_2$ is a bisim. relation between $\mathcal{K}_1$ and $\mathcal{K}_2$ $(\mathcal{K}_1 \approx \mathcal{K}_2)$ if: **BISIM1-** $(s_1, s_2) \in \sigma$ implies $\mathcal{L}_1(s_1) = \mathcal{L}_2(s_2)$; **BISIM2a-**

Boolean Operations of F
$(1)\neg F\varphi = G\neg\varphi$ $(2)F\varphi \vee F\psi = F[\varphi \vee \psi]$
$(3)F\varphi \wedge F\psi = \mathcal{A}_\exists(\{p,q\},\neg p \wedge \neg q,$
$\qquad [p' \leftrightarrow p \vee \varphi] \wedge [q' \leftrightarrow q \vee \psi], F[p \wedge q])$

Boolean Operations of FG
$(1)\neg FG\varphi = GF\neg\varphi$ $(2)FG\varphi \wedge FG\psi = FG[\varphi \wedge \psi]$
$(3)FG\varphi \vee FG\psi = \mathcal{A}_\exists(\{q\},\neg q, q' \leftrightarrow (q \Rightarrow \psi|\neg\varphi),$
$\qquad FG[\neg q \vee \psi])$

Boolean Operations of GF
$(1)\neg GF\varphi = FG\neg\varphi$ $(2)GF\varphi \vee GF\psi = GF[\varphi \vee \psi]$
$(3)GF\varphi \wedge GF\psi = \mathcal{A}_\exists(\{q\},\neg q, q' \leftrightarrow (q \Rightarrow \neg\psi|\varphi),$
$\qquad GF[q \wedge \psi])$

Transformation of Acceptance Conditions
Reduction of G
$G\varphi = \mathcal{A}_\exists(\{q\},q,\varphi \wedge q \wedge q', Fq))$
$G\varphi = \mathcal{A}_\exists(\{q\},q,q' \leftrightarrow q \wedge \varphi, FGq)$
$G\varphi = \mathcal{A}_\exists(\{q\},q,q' \leftrightarrow q \wedge \varphi, GFq)$
Reduction of F
$F\varphi$ can **not** be expressed by $NDet_G$
$F\varphi = \mathcal{A}_\exists(\{q\},\neg q,q' \leftrightarrow q \vee \varphi, FGq)$
$F\varphi = \mathcal{A}_\exists(\{q\},\neg q,q' \leftrightarrow q \vee \varphi, GFq)$
Reduction of FG
$FG\varphi$ can **not** be expressed by $NDet_G$
$FG\varphi = \mathcal{A}_\exists(\{q\},\neg q, q \to \varphi \wedge q', Fq)$

$FG\varphi = \mathcal{A}_\exists \begin{pmatrix} \{p,q\}, \qquad \neg p \wedge \neg q, \\ \begin{bmatrix} (p \to p') \wedge (p' \to p \vee \neg q)\wedge \\ (q' \leftrightarrow (p \wedge \neg q \vee \neg\varphi) \vee (p \wedge q)) \end{bmatrix}, \\ G\neg q \wedge Fp \end{pmatrix}$

$FG\varphi = \mathcal{A}_\exists \begin{pmatrix} \{p,q\}, \qquad \neg p \wedge \neg q, \\ \begin{bmatrix} (p \to p') \wedge (p' \to p \vee \neg q)\wedge \\ (q' \leftrightarrow (p \wedge \neg q \vee \neg\varphi) \vee (p \wedge q)) \end{bmatrix}, \\ GF[p \wedge \neg q] \end{pmatrix}$

Temporal Logics *Beware of Finite Paths*
E and A quantify over infinite paths.
$A\varphi$ holds on every state that has no infinite path;
$E\varphi$ is false on every state that has no infinite path;
$A0$ holds on states with only finite paths;
$E1$ is false on state with only finite paths;
$\square 0$ holds on states with no successor states;
$\diamond 1$ holds on states with successor states.
$F\varphi = \varphi \vee XF\varphi$ $G\varphi = \varphi \wedge XG\varphi$
$[\varphi U \psi] = \psi \vee (\varphi \wedge X[\varphi U \psi])$
$[\varphi B \psi] = \neg\psi \wedge (\varphi \vee X[\varphi B \psi])$
$[\varphi W \psi] = (\psi \wedge \varphi) \vee (\neg\psi \wedge X[\varphi W \psi])$

Negation Normal Form
$\neg(\varphi \wedge \psi) = \neg\varphi \vee \neg\psi$ $\neg(\varphi \vee \psi) = \neg\varphi \wedge \neg\psi$
$\neg\neg\varphi = \varphi$ $\neg X\varphi = X\neg\varphi$
$\neg G\varphi = F\neg\varphi$ $\neg F\varphi = G\neg\varphi$
$\neg[\varphi U \psi] = [(\neg\varphi) \underline{B} \psi]$ $\neg[\varphi \underline{U} \psi] = [(\neg\varphi) B \psi]$
$\neg[\varphi B \psi] = [(\neg\varphi) \underline{U} \psi]$ $\neg[\varphi \underline{B} \psi] = [(\neg\varphi) U \psi]$
$\neg A\varphi = E\neg\varphi$ $\neg E\varphi = A\neg\varphi$
$\neg \overleftarrow{X}\varphi = \overleftarrow{X}\neg\varphi$ $\neg \overleftarrow{\underline{X}}\varphi = \overleftarrow{\underline{X}}\neg\varphi$
$\neg \overleftarrow{G}\varphi = \overleftarrow{F}\neg\varphi$ $\neg \overleftarrow{F}\varphi = \overleftarrow{G}\neg\varphi$
$\neg[\varphi \overleftarrow{U} \psi] = [(\neg\varphi) \overleftarrow{\underline{B}} \psi]$ $\neg[\varphi \overleftarrow{\underline{U}} \psi] = [(\neg\varphi) \overleftarrow{B} \psi]$
$\neg[\varphi \overleftarrow{B} \psi] = [(\neg\varphi) \overleftarrow{\underline{U}} \psi]$ $\neg[\varphi \overleftarrow{\underline{B}} \psi] = [(\neg\varphi) \overleftarrow{U} \psi]$

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

$[\varphi U \psi] = \neg[(\neg\psi) U (\neg\varphi \wedge \neg\psi)]$
$[\varphi \underline{U} \psi] = \neg[(\neg\psi) W (\varphi \to \psi)]$
$[\varphi \underline{U} \psi] = [\psi \underline{W} (\varphi \to \psi)]$
$[\varphi \underline{U} \psi] = \neg[(\neg\varphi) B \psi]$*(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
$EF\varphi = E[1 \underline{U} \varphi]$
$EG\varphi = E[\varphi U 0]$
$E[\varphi U \psi] = E[\varphi \underline{U} \psi] \vee EG\varphi$
$E[\varphi B \psi] = E[(\neg\psi) \underline{U} (\varphi \wedge \neg\psi)] \vee EG\neg\psi$
$E[\varphi \underline{B} \psi] = E[(\neg\psi) U (\varphi \wedge \neg\psi)]$
$E[\varphi \underline{B} \psi] = E[(\neg\psi) \underline{U} (\varphi \wedge \neg\psi)]$
$E[\varphi W \psi] = E[(\neg\psi) \underline{U} (\varphi \wedge \psi)] \vee EG\neg\psi$
$E[\varphi W \psi] = E[(\neg\psi) U (\varphi \wedge \psi)]$
$E[\varphi \underline{W} \psi] = E[(\neg\psi) \underline{U} (\varphi \wedge \psi)]$

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

CTL to $\mu - Calculus(\Phi_{inf} = \nu y.\diamond y)$
$EX\varphi = \diamond(\Phi_{inf} \wedge \varphi)$
$EG\varphi = \nu x.\varphi \wedge \diamond x$
$EF\varphi = \mu x.\Phi_{inf} \wedge \varphi \vee \diamond x$
$E[\varphi \underline{U} \psi] = \mu x.(\Phi_{inf} \wedge \psi) \vee \varphi \wedge \diamond x$
$E[\varphi U \psi] = \nu x.(\Phi_{inf} \wedge \psi) \vee \varphi \wedge \diamond x$
$E[\varphi \underline{B} \psi] = \mu x.\neg\psi \wedge (\Phi_{inf} \wedge \varphi \vee \diamond x)$
$E[\varphi B \psi] = \nu x.\neg\psi \wedge (\Phi_{inf} \wedge \varphi \vee \diamond x)$
$AX\varphi = \square(\Phi_{inf} \to \varphi)$
$AG\varphi = \nu x.(\Phi_{inf} \to \varphi) \wedge \square x$
$AF\varphi = \mu x.\varphi \vee \square x$
$A[\varphi \underline{U} \psi] = \mu x.\psi \vee (\Phi_{inf} \to \varphi) \wedge \square x$
$A[\varphi U \psi] = \nu x.\psi \vee (\Phi_{inf} \to \varphi) \wedge \square x$
$A[\varphi \underline{B} \psi] = \mu x.(\Phi_{inf} \to \neg\psi) \wedge (\varphi \vee \square x)$
$A[\varphi B \psi] = \nu x.(\Phi_{inf} \to \neg\psi) \wedge (\varphi \vee \square x)$

CTL* to CTL - Existential Operators
$EX\varphi = EXE\varphi$
$EF\varphi = EFE\varphi$ $EFG\varphi \equiv EFEG\varphi$
$E[\varphi W \psi] = E[(E\varphi) W \psi]$
$E[\varphi \underline{W} \psi] = E[(E\varphi) \underline{W} \psi]$
$E[\psi U \varphi] = E[\psi U E(\varphi)]$
$E[\psi \underline{U} \varphi] = E[\psi \underline{U} E(\varphi)]$
$E[\varphi B \psi] = E[(E\varphi) B \psi]$
$E[\varphi \underline{B} \psi] = E[(E\varphi) \underline{B} \psi]$
obs. $EGF\varphi \neq EGEF\varphi \to$ can't be converted

CTL* to CTL - Universal Operators
$AX\varphi = AXA\varphi$
$AG\varphi = AGA\varphi$
$A[\varphi W \psi] = A[(A\varphi) W \psi]$
$A[\varphi \underline{W} \psi] = A[(A\varphi) \underline{W} \psi]$
$A[\varphi U \psi] = A[A(\varphi) U \psi]$
$A[\varphi \underline{U} \psi] = A[A(\varphi) \underline{U} \psi]$
$A[\psi B \varphi] = A[\psi B (E(\varphi))]$

$A[\psi \underline{B} \varphi] = A[\psi \underline{B} (E(\varphi))]$
Eliminate boolean op. after path quantify
$[\varphi_1 \underline{U} \psi_1] \wedge [\varphi_2 \underline{U} \psi_2] =$
$(\varphi_1 \wedge \varphi_2) \underline{U} \begin{pmatrix} \psi_1 \wedge [\varphi_2 \underline{U}\psi_2]\vee \\ \psi_2 \wedge [\varphi_1 \underline{U}\psi_1] \end{pmatrix}$
$[\varphi_1 \underline{U} \psi_1] \wedge [\varphi_2 U \psi_2] =$
$(\varphi_1 \wedge \varphi_2) \underline{U} \begin{pmatrix} \psi_1 \wedge [\varphi_2 U\psi_2]\vee \\ \psi_2 \wedge [\varphi_1 \underline{U}\psi_1] \end{pmatrix}$
$[\varphi_1 U \psi_1] \wedge [\varphi_2 U \psi_2] =$
$(\varphi_1 \wedge \varphi_2) \underline{U} \begin{pmatrix} \psi_1 \wedge [\varphi_2 U\psi_2]\vee \\ \psi_2 \wedge [\varphi_1 U\psi_1] \end{pmatrix}$

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 = $ let $\begin{bmatrix} x_1 = A\varphi_1 \\ \vdots \\ x_n = A\varphi_n \end{bmatrix}$ in Ψ end
Use LTL model checking to compute
$Q_i := [\![A\varphi_i]\!]_{\mathcal{K}_{i-1}}$, 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 $[\![\Psi]\!]_{\mathcal{K}_n}$

LTL to ω-automata
$\phi\langle X\varphi\rangle_x \Leftrightarrow \mathcal{A}_\exists(\{q\},1,q \leftrightarrow X\varphi, \phi\langle q\rangle_x)$
$\phi\langle X\varphi\rangle_x \Leftrightarrow$
$\quad \mathcal{A}_\exists(\{q_0,q_1\},1,(q_0 \leftrightarrow \varphi) \wedge (q_1 \leftrightarrow Xq_0), \phi\langle q_1\rangle_x)$
$\phi\langle G\varphi\rangle_x \Leftrightarrow$
$\quad \mathcal{A}_\exists(\{q\},1,q \leftrightarrow \varphi \wedge Xq, \phi\langle q\rangle_x \wedge GF[\varphi \to q])$
$\phi\langle F\varphi\rangle_x \Leftrightarrow$
$\quad \mathcal{A}_\exists(\{q\},1,q \leftrightarrow \varphi \vee Xq, \phi\langle q\rangle_x \wedge GF[q \to \varphi])$
$\phi\langle[\varphi U \psi]\rangle_x \Leftrightarrow$
$\quad \mathcal{A}_\exists(\{q\},1,q \leftrightarrow \psi \vee \varphi \wedge Xq, \phi\langle q\rangle_x \wedge GF[\varphi \to q])$
$\phi\langle[\varphi \underline{U} \psi]\rangle_x \Leftrightarrow$
$\quad \mathcal{A}_\exists(\{q\},1,q \leftrightarrow \psi \vee \varphi \wedge Xq, \phi\langle q\rangle_x \wedge GF[q \to \psi])$
$\phi\langle[\varphi B \psi]\rangle_x \Leftrightarrow$
$\quad \mathcal{A}_\exists(\{q\},1,q \leftrightarrow \neg\psi \wedge (\varphi \vee Xq), \phi\langle q\rangle_x \wedge GF[q \vee \psi])$
$\phi\langle[\varphi \underline{B} \psi]\rangle_x \Leftrightarrow$
$\quad \mathcal{A}_\exists(\{q\},1,q \leftrightarrow \neg\psi \wedge (\varphi \vee Xq), \phi\langle q\rangle_x \wedge GF[q \to \varphi])$
$\phi\langle \overleftarrow{X}\varphi\rangle_x \Leftrightarrow \mathcal{A}_\exists(\{q\},q,Xq \leftrightarrow \varphi, \phi\langle q\rangle_x)$
$\phi\langle \overleftarrow{\underline{X}}\varphi\rangle_x \Leftrightarrow \mathcal{A}_\exists(\{q\},\neg q,Xq \leftrightarrow \varphi, \phi\langle q\rangle_x)$
$\phi\langle \overleftarrow{G}\varphi\rangle_x \Leftrightarrow \mathcal{A}_\exists(\{q\},q,Xq \leftrightarrow \varphi \wedge q, \phi\langle \varphi \wedge q\rangle_x)$
$\phi\langle \overleftarrow{F}\varphi\rangle_x \Leftrightarrow \mathcal{A}_\exists(\{q\},\neg q,Xq \leftrightarrow \varphi \vee q, \phi\langle \varphi \vee q\rangle_x)$
$\phi\langle[\varphi \overleftarrow{U} \psi]\rangle_x \Leftrightarrow$
$\quad \mathcal{A}_\exists(\{q\},q,Xq \leftrightarrow \psi \vee \varphi \wedge q, \phi\langle \psi \vee \varphi \wedge q\rangle_x)$
$\phi\langle[\varphi \overleftarrow{\underline{U}} \psi]\rangle_x \Leftrightarrow$
$\quad \mathcal{A}_\exists(\{q\},\neg q,Xq \leftrightarrow \psi \vee \varphi \wedge q, \phi\langle \psi \vee \varphi \wedge q\rangle_x)$
$\phi\langle[\varphi \overleftarrow{B} \psi]\rangle_x \Leftrightarrow$
$\quad \mathcal{A}_\exists(\{q\},q,Xq \leftrightarrow \neg\psi \wedge (\varphi \vee q), \phi\langle \neg\psi \wedge (\varphi \vee q)\rangle_x)$
$\phi\langle[\varphi \overleftarrow{\underline{B}} \psi]\rangle_x \Leftrightarrow$
$\quad \mathcal{A}_\exists(\{q\},\neg q,Xq \leftrightarrow \neg\psi \wedge (\varphi \vee q), \phi\langle \neg\psi \wedge (\varphi \vee q)\rangle_x)$

S1S
First order terms are defined as follows:
$-0 \in Term_\Sigma^{S1S}$
$-t \in V_\Sigma | typ_\Sigma(t) = \mathbb{N} \subseteq Term_\Sigma^{S1S}$
$-SUC(\tau) \in Term_\Sigma^{S1S} if \tau \in Term_\Sigma^{S1S}$
Formulas ζ_{S1S} are defined as:
$-p^{(t)} \in L_{S1S}$ (predicate p at time t)
$-\neg\varphi,\varphi \wedge \psi \in L_{S1S}$
$-\exists t.\varphi \in L_{S1S}$

$-\exists p.\varphi \in L_{S1S}$
where:
$-\tau \in Term_\Sigma^{S1S}$
$-\varphi,\psi \in \zeta_{S1S}$
$-t \in V_\Sigma$ with $typ_\Sigma(t) = \mathbb{N}$
$-p \in V_\Sigma$ with $typ_\Sigma(p) = \mathbb{N} \to \mathbb{B}$

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:
$-t,t_1,t_2\tau \in V_\Sigma$ with $typ_\Sigma(t) = typ_\Sigma(t_1) = typ_\Sigma(t_2) = \mathbb{N}$
$-\varphi,\psi \in \zeta_{LO2}$
$-t \in V_\Sigma$ with $typ_\Sigma(t) = \mathbb{N}$
$-p \in V_\Sigma$ with $typ_\Sigma(p) = \mathbb{N} \to \mathbb{B}$

function LO2_SIS(Φ)
case Φ of
$t1 < t2 :$ return $\exists p.[\forall t.p^{(t)} \to p^{(SUC(t))}] \wedge \neg p^{(t1)} \wedge p^{(t2)} :$
$\quad p^{(t)} :$ return $p^{(t)};$
$\quad \neg\varphi :$ return $\neg LO2\_S1S(\varphi);$
$\quad \varphi \wedge \psi :$ return $LO2\_S1S(\varphi) \wedge LO2\_S1S(\psi);$
$\quad \exists t.\varphi :$ return $\exists t.LO2\_S1S(\varphi);$
$\quad \exists p.\varphi :$ return $\exists p.LO2\_S1S(\varphi);$
end
end

function S1S_LO2(Φ)
case Φ of
$\quad p^{(n)} :$
return $\exists t0...tn.p^{(tn)} \wedge zero(t0) \wedge \bigwedge_{i=0}^{n-1} succ(ti,ti+1);$
$\quad p^{(t0+n)} :$
return $\exists t1...tn.p^{(tn)} \wedge \bigwedge_{i=0}^{n-1} succ(ti,ti + 1);$
$\quad \neg\varphi :$ return $\neg S1S\_LO2(\varphi);$
$\quad \varphi \wedge \psi :$ return $S1S\_LO2(\varphi) \wedge S1S\_LO2(\psi);$
$\quad \exists t.\varphi :$ return $\exists t.S1S\_LO2(\varphi);$
$\quad \exists p.\varphi :$ return $\exists p.S1S\_LO2(\varphi);$
end
end

LO2' Consider the following set $\zeta_{LO2'}$ of formulas:
$-Subset(p,q), Sing(p), and PSUC(p,q) belong to \zeta_{LO2'}$
$-\neg\varphi,\varphi \wedge \psi$
$-\exists p.\varphi$
where $-\varphi,\psi \in \zeta_{LO2'}$
$-p \in V_\Sigma$ with $typ_\Sigma(p) = \mathbb{N} \to \mathbb{B}$
$\zeta_{LO2'}$ has non numeric variables
numeric variable t is replaced by a singleton set p_t
$\zeta_{LO2'}$ is as expressive as LO2 and S1S

function ElimFO(Φ) (LO2 TO LO2')
case Φ of
$\quad t1 = t2 :$ return $Subset(q_{t1},q_{t2}) \wedge Subset(q_{t2},q_{t1})$
$\quad t1 < t2 : \Psi :\equiv \forall q1.\forall q2.PSUC(q1,q2) \to [Subset(q1,p) \to Subset(q2,p)];$
$\qquad$ return $\exists p.\Psi \wedge \neg Subset(qt1,p) \wedge Subset(qt2,p);$
$\quad p^{(t)} :$ return $Subset(qt,p)$
$\quad \neg\varphi :$ return $\neg ElimFO(\varphi);$
$\quad \varphi \wedge \psi :$ return $ElimFO(\varphi) \wedge ElimFO(\psi);$
$\quad \varphi \vee \psi :$ return $ElimFO(\varphi) \vee ElimFO(\psi);$

$\exists t.\varphi : \textbf{return } \exists qt.Sing(qt) \wedge ElimFO(\varphi);$

$\exists p.\varphi : \textbf{return } \exists p.ElimFO(\varphi);$

end

end

function $Tp2Od(t0, \Phi)$ *temporal to LO1*

  case Φ **of**

    $is\_var(\Phi) : \Psi^{(t0)};$

    $\neg\varphi : \textbf{return } \neg Tp2Od(\varphi);$

    $\varphi \wedge \psi : \textbf{return } Tp2Od(\varphi) \wedge Tp2Od(\psi);$

    $\varphi \vee \psi : \textbf{return } Tp2Od(\varphi) \vee Tp2Od(\psi);$

    $X\varphi : \Psi := \exists t1.(t0 < t1) \wedge (\forall t2.t0 < t2 \to t1 \leq t2) \wedge Tp2Od(t1, \varphi);$

    $[\varphi \underline{U} \psi] : \Psi := \exists t1.t0 \leq t1 \wedge Tp2Od(t1, \psi) \wedge interval((t0,1,t1,0), \varphi);$

    $[\varphi \underline{B} \psi] : \Psi := \forall t1.t0 \leq t1 \wedge interval((t0,1,t1,0), \neg\varphi) \to Tp2Od(t1, \neg\psi);$

    $\overleftarrow{X}\varphi : \Psi := \forall t1.(t1 < t0) \wedge (\forall t2.t2 < t0 \to t2 \leq t1) \to Tp2Od(t1, \varphi);$

    $\overleftarrow{X}\varphi : \Psi := \exists t1.(t1 < t0) \wedge (\forall t2.t2 < t0 \to t2 \leq t1) \wedge Tp2Od(t1, \varphi);$

    $[\varphi \overleftarrow{\underline{U}} \psi] : \Psi := \exists t1.t1 \leq t0 \wedge Tp2Od(t1, \psi) \wedge interval((t1,0,t0,1), \varphi);$

    $[\varphi \overleftarrow{B} \psi] : \Psi := \forall t1.t1 \leq t0 \wedge interval((t1,0,t0,1), \neg\varphi) \to Tp2Od(t1, \neg\psi);$

  end

  return Ψ

end

function $interval(l, \varphi)$

  case Φ **of**

    $(t0,0,t1,0) :$

      $\textbf{return } \forall t2.t0 < t2 \wedge t2 < t1 \to Tp2Od(t2, \varphi);$

    $(t0,0,t1,1) :$

      $\textbf{return } \forall t2.t0 < t2 \wedge t2 \leq t1 \to Tp2Od(t2, \varphi);$

    $(t0,1,t1,0) :$

      $\textbf{return } \forall t2.t0 \leq t2 \wedge t2 < t1 \to Tp2Od(t2, \varphi);$

    $(t0,1,t1,1) :$

      $\textbf{return } \forall t2.t0 \leq t2 \wedge t2 \leq 3t1 \to Tp2Od(t2, \varphi);$

  end

end

Temporal Logic Equivalences and Tips

$[\varphi \underline{U} \psi] \equiv \varphi \; don't \; matter \; when \; \psi \; hold$

$[\varphi \underline{B} \psi] \equiv \psi \; can't \; hold \; when \; \varphi \; hold$

$[\varphi \underline{W} \psi] \equiv \neg\psi \; hold \; until \; \varphi \wedge \psi$

$[\varphi U \psi] \equiv [\varphi \underline{U} \psi] \vee G\varphi$

$[aUFb] \equiv Fb$

$F[aUb] \equiv Fb \equiv [FaUFb]$

$[\varphi B \psi] \equiv [\varphi \underline{B} \psi] \vee G\neg\psi$

$F[a\underline{B}b] \equiv F[a \wedge \neg b]$

$[\varphi W \psi] \equiv \neg[\neg\varphi \underline{W} \psi]$

$E(\varphi \wedge \psi) \equiv E\varphi \wedge E\psi(in \; general)$

$AEA \equiv A$

$GF(x \vee y) \equiv GFx \vee GFy$

$FF\varphi \equiv F\varphi$

$GG\varphi \equiv G\varphi$

$GF\varphi \equiv XGF\varphi \equiv FGF\varphi \equiv GGF\varphi \equiv GFGF\varphi \equiv FGGF\varphi$

$FG\varphi \equiv XFG\varphi \equiv FFG\varphi \equiv GFG\varphi \equiv GFFG\varphi \equiv FGFG\varphi$

G and μ-calculus (safety property)

- $[\nu x.\varphi \wedge \Diamond x]_K$
- Contains states s where an infinite path π starts with $\forall t.\pi^{(t)} \in [\varphi]_K$
- φ 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 $\exists t.\pi^{(t)} \in [\varphi]_K$
- φ holds at least once on π

FG and μ-calculus (persistence property)

- $[\mu y.[\nu x.\varphi \vee \Diamond x] \vee \Diamond y]_K$
- Contains states s where an infinite path π starts with $\exists t1.\forall t2.\pi^{(t1+t2)} \in [\varphi]_K$
- φ 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
- $\forall t1.\exists t2.\pi^{(t1+t2)} \in [\varphi]_K$?????$t1 + t2ort1 + t0$?????
- φ holds infinitely often on π

ω-Automaton to LO2

$A_{\exists}(q1,...,qn, \psi R, \psi I, \psi F)(input \; automaton)$

$\exists q1..qn, \Theta LO2(0, \psi I) \wedge (\forall t.\Theta LO2(t, \psi R)) \wedge$

$(\forall.t1 \exists t2.t1 < t2 \wedge \Theta LO2(t2, \psi F))$

Where $\Theta LO2(t, \Phi)$ is:

- $\Theta LO2(t,p) := p(t) \; for \; variable \; p$
- $\Theta LO2(t, X\psi) := \Theta LO2(t + 1, \psi)$
- $\Theta LO2(t, \neg\psi) := \neg\Theta LO2(t, \psi)$
- $\Theta LO2(t, \varphi \wedge \psi) := \Theta LO2(t, \varphi) \wedge \Theta LO2(t, \psi)$
- $\Theta LO2(t, \varphi \vee \psi) := \Theta LO2(t, \varphi) \vee \Theta LO2(t, \psi)$

Temporal logic set examples

- Pure LTL: AFGa
- Pure CTL: AGEFa
- LTL + CTL: AFa
- CTL*: AFGa $\vee$ AGEFa

Extra Equations G

$AG[\varphi U \psi] = AG(\varphi \vee \psi)$

$AG[\varphi B \psi] = AG(\neg\psi)$

$AG[\varphi W \psi] = AG(\psi \to \varphi)$

$AG[\varphi \underline{U} \psi] = A(G(\varphi \vee \psi) \wedge GF\psi)$

$AG[\varphi \underline{B}\psi] = A(G(\neg\psi) \wedge GF\varphi)$

$AG[\varphi \underline{W}\psi] = A(G(\psi \to \varphi) \wedge GF\psi)$

// note that the following are only initially, but not

generally valid

$AG\overleftarrow{X}\varphi = AG\varphi$

$AG\underleftarrow{X}\varphi = A(\text{false})$

$AG\overleftarrow{G}\varphi = AG\varphi$

$AG\overleftarrow{F}\varphi = A\varphi$

$AG[\varphi \overleftarrow{U} \psi] = AG(\varphi \vee \psi)$

$AG[\varphi \overleftarrow{B} \psi] = AG(\neg\psi)$

$AG[\varphi \overleftarrow{W} \psi] = AG(\psi \to \varphi)$

$AG[\varphi \overleftarrow{\underline{U}} \psi] = A(\psi \wedge G(\varphi \vee \psi))$

$AG[\varphi \overleftarrow{\underline{B}} \psi] = A(\varphi \wedge G(\neg\psi))$

$AG[\varphi \overleftarrow{\underline{W}} \psi] = A(\psi \wedge G(\psi \to \varphi))$

Extra Equations F

$AFF\psi = AF\psi$

$AF[\varphi \underline{U} \psi] = AF\psi$

$AF[\varphi \underline{B} \psi] = AF(\varphi \wedge \neg\psi)$

$AF[\varphi \underline{W} \psi] = AF(\varphi \vee \psi)$

$AF[\varphi U \psi] = A(F(\psi) \vee FG\varphi)$

$AF[\varphi B \psi] = A(F(\varphi \wedge \neg\psi) \vee FG(\neg\varphi \wedge \neg\psi))$

$AF[\varphi W \psi] = A(F(\varphi \wedge \psi) \vee FG\neg\psi)$

// note that the following are only initially, but not generally valid

$AF\overleftarrow{X}\varphi = A(\text{true})$

$AF\underleftarrow{X}\varphi = AF\varphi$

$AF\overleftarrow{G}\varphi = A\varphi$

$AF\overleftarrow{F}\varphi = AF\varphi$

$AF[\varphi \overleftarrow{\underline{U}} \psi] = AF\psi$

$AF[\varphi \overleftarrow{\underline{B}} \psi] = AF(\varphi \wedge \neg\psi)$

$AF[\varphi \overleftarrow{\underline{W}} \psi] = AF(\varphi \wedge \psi)$

$AF[\varphi \overleftarrow{U} \psi] = A(F\psi \vee F\overleftarrow{G}\varphi)$

$AF[\varphi \overleftarrow{B} \psi] = A(F(\varphi \wedge \neg\psi) \vee F\overleftarrow{G}(\neg\varphi \wedge \neg\psi))$

$AF[\varphi \overleftarrow{W} \psi] = A(F(\varphi \wedge \psi) \vee F\overleftarrow{G}\neg\psi)$