Propositional Logic - Syntactic Sugar

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

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

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

Satisfiability, Validity and Equivalence

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

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

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

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	$\frac{\neg\phi,\Gamma\vdash\Delta}{\Gamma\vdash\phi,\Delta}$	$\frac{\Gamma\vdash\neg\phi,\Delta}{\phi,\Gamma\vdash\Delta}$
AND	$\frac{\phi\wedge\psi,\Gamma\vdash\Delta}{\phi,\psi,\Gamma\vdash\Delta}$	$\frac{\Gamma\vdash\phi\wedge\psi,\Delta}{\Gamma\vdash\phi,\Delta \qquad \Gamma\vdash\psi,\Delta}$
OR	$\frac{\phi\vee\psi,\Gamma\vdash\Delta}{\phi,\Gamma\vdash\Delta \qquad \psi,\Gamma\vdash\Delta}$	$\frac{\Gamma\vdash\phi\vee\psi,\Delta}{\Gamma\vdash\phi,\psi,\Delta}$

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

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: $\{a\} + \{\neg a, b\} \to \{b\}; \{b\} + \{\neg b\} \to \{\}$ (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.

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

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

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

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

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

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

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 $\underline{1}$ (true) → 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
```

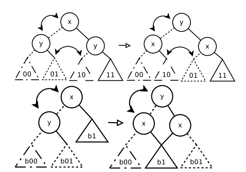
```
Compose(int x, BddNode ψ, α)
 int m; BddNode h, l;
 if x>label(ψ) then
  return ψ;
 elseif x=label(ψ) then
  return ITE(α,high(ψ),
        low(ψ));
 else
  m=max{label(ψ),label(α)}
  (α0,α1):=Ops(α, m);
  (ψ0,ψ1):=Ops(ψ, m);
  h:=Compose(x,ψ1,α1);
  l:=Compose(x,ψ0,α0);
  return CreateNode(m,h,l)
 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
```

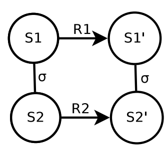
```
Constrain(Φ, β)
 if β=0 then
  ret 0
 elseif Φ ∈ {0,1}(β = 1)
     ret Φ
 else
  m=max{label(β),label(Φ)}
    (Φ0, Φ1):=Ops(Φ,m);
    (β0, β1):=Ops(β,m);
    if β0=0
  ret Constrain(Φ1, β1)
 elseif β1=0 then
  ret Constrain(Φ0, β0)
 else
  l:=Constrain(Φ0, β0);
  h:=Constrain(Φ1, β1);
  ret CreateNode(m,h,l)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

-----
```



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 $\mathcal{K}_2$ ($\mathcal{K}_1 \preccurlyeq \mathcal{K}_2$) 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

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

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

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

$\mathcal{H}_*$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-** $s_1, s_1' \in \mathcal{S}_1, s_2 \in \mathcal{S}_2$, $(s_1, s_2) \in \sigma$, $(s_1, s_1') \in \mathcal{R}_1$, 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)

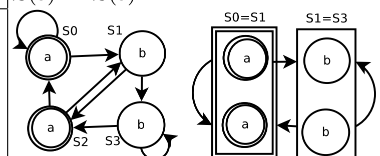
$(s_1, s_2) \in \mathcal{B}_0 \Leftrightarrow \mathcal{L}_1(s_1) = \mathcal{L}_2(s_2)$

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

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

$\mathcal{B}_*$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 $\mathcal{K}_{/\sigma} = (\widetilde{\mathcal{I}}, \widetilde{\mathcal{S}}, \widetilde{\mathcal{R}}, \widetilde{\mathcal{L}})$: $\widetilde{\mathcal{I}} := \{\{s' \in \mathcal{S} | (s, s') \in \sigma\} | s \in \mathcal{I}\}$
$\widetilde{\mathcal{S}} := \{\{s' \in \mathcal{S} | (s, s') \in \sigma\} | s \in \mathcal{S}\}$
$(\widetilde{s}_1, \widetilde{s}_2) \in \mathcal{R} :\Leftrightarrow \exists s_1' \in \widetilde{s}_1 . \exists s_2' \in \widetilde{s}_2 . (s_1', s_2') \in \mathcal{R}$
$\widetilde{\mathcal{L}}(\widetilde{s}) := \mathcal{L}(s)$



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

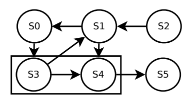
Predecessor and Successor

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

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

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

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



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

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

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

```
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 $\bot \to$ least fixpoint ♠ $\nu :=$ starts $\top \to$ 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) $\frac{s \vdash_{\Phi} \varphi \wedge \psi}{\{s \vdash_{\Phi} \varphi\} \quad \{s \vdash_{\Phi} \psi\}} \wedge$		(2) $\frac{s \vdash_{\Phi} \varphi \vee \psi}{\{s \vdash_{\Phi} \varphi\} \quad \{s \vdash_{\Phi} \psi\}} \vee$	
(3) $\frac{s \vdash_{\Phi} \Box \varphi}{\{s_1 \vdash_{\Phi} \varphi\} \ldots \{s_n \vdash_{\Phi} \varphi\}} \wedge$		(4) $\frac{s \vdash_{\Phi} \Diamond \varphi}{\{s_1 \vdash_{\Phi} \varphi\} \ldots \{s_n \vdash_{\Phi} \varphi\}} \vee$	
(5) $\frac{s \vdash_{\Phi} \overleftarrow{\Box} \varphi}{\{s_1' \vdash_{\Phi} \varphi\} \ldots \{s_n' \vdash_{\Phi} \varphi\}} \wedge$		(6) $\frac{s \vdash_{\Phi} \overleftarrow{\Diamond} \varphi}{\{s_1' \vdash_{\Phi} \varphi\} \ldots \{s_n' \vdash_{\Phi} \varphi\}} \vee$	
$\frac{s \vdash_{\Phi} \mu x . \varphi}{s \vdash_{\Phi} \varphi}$	$\frac{s \vdash_{\Phi} \nu x . \varphi}{s \vdash_{\Phi} \varphi}$	$\frac{s \vdash_{\Phi} x}{s \vdash_{\Phi} \mathfrak{D}_{\Phi}(x)}$	$\mathfrak{D}_{\Phi}$(replace w. initial form.)
$\{s_1 \ldots s_n\} = suc_{\exists}^{\mathcal{R}}(s)$ and $\{s_1' \ldots s_n'\} = pre_{\exists}^{\mathcal{R}}(s)$			

Approximations and Ranks

If (s,$\mu x . \varphi$) repeats→return 0	$apx_0(\mu x . \varphi) := 0$
If (s,$\nu x . \varphi$) repeats→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→Safety; F→Liveness; FG→Persistence/Co-Buchi; GF→Fairness/Buchi.

Automaton Determinization

$\mathbf{NDet_G \to Det_G}$: 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.
$\mathbf{\{NDet_F(partial)\ or\ NDet_{prefix}\} \to Det_{FG}}$: Breakpoint Construction.
$\mathbf{NDet_F\ (total) \to Det_F}$: Subset Construction.
$\mathbf{NDet_{FG} \to Det_{FG}}$: Breakpoint Construction.
$\mathbf{NDet_{GF} \to \{Det_{Rabin}\ or\ Det_{Streett}\}}$: 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,$\mathrm{Q_f}$) 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)$\neg G\varphi = F\neg\varphi$ (2)$G\varphi \wedge G\psi = G[\varphi \wedge \psi]$
(3)$G\varphi \vee G\psi = \mathcal{A}_{\exists}(\{p, q\}, p \wedge q, [p' \leftrightarrow p \wedge \varphi] \wedge [q' \leftrightarrow q \wedge \psi], G[p \vee q])$

Boolean Operations of F

$(1)\neg F\varphi = G\neg\varphi$ $\quad (2) F\varphi \vee F\psi = F[\varphi \vee \psi]$

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

Boolean Operations of FG

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

$(3) FG\varphi \vee FG\psi = \mathcal{A}_\exists(\{q\}, \neg q, q' \leftrightarrow (q \Rightarrow \psi | \neg\varphi),$
$FG[\neg q \vee \psi])$

Boolean Operations of GF

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

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

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 $NDet_G$
$FG\varphi = \mathcal{A}_\exists(\{q\}, \neg q, q \rightarrow \varphi \wedge q', Fq)$

$$FG\varphi = \mathcal{A}_\exists\left(\begin{array}{c}\{p,q\}, \qquad \neg p \wedge \neg q,\\ \left[\begin{array}{c}(p \rightarrow p') \wedge (p' \rightarrow p \vee \neg q) \wedge \\ (q' \leftrightarrow (p \wedge \neg q \vee \neg\varphi) \vee (p \wedge q))\end{array}\right],\\ G\neg q \wedge Fp\end{array}\right)$$

$$FG\varphi = \mathcal{A}_\exists\left(\begin{array}{c}\{p,q\}, \qquad \neg p \wedge \neg q,\\ \left[\begin{array}{c}(p \rightarrow p') \wedge (p' \rightarrow p \vee \neg q) \wedge \\ (q' \leftrightarrow (p \wedge \neg q \vee \neg\varphi) \vee (p \wedge q))\end{array}\right],\\ GF[p \wedge \neg q]\end{array}\right)$$

Temporal Logics *Beware of Finite Paths*

E and A quantify over infinite paths.
Aφ holds on every state that has no infinite path;
Eφ 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;
$\Box 0$ holds on states with no successor states;
$\Diamond 1$ holds on states with successor states.

$F\varphi = \varphi \vee XF\varphi$ $\quad G\varphi = \varphi \wedge XG\varphi$
$[\varphi\ U\ \psi] = \psi \vee (\varphi \wedge X[\varphi\ U\ \psi])$
$[\varphi\ B\ \psi] = \neg\psi \wedge (\varphi \vee X[\varphi\ B\ \psi])$
$[\varphi\ W\ \psi] = (\psi \wedge \varphi) \vee (\neg\psi \wedge X[\varphi\ W\ \psi])$

Negation Normal Form

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

LTL Syntactic Sugar: analog for past operators

$G\varphi = \neg[1\ \underline{U}\ (\neg\varphi)]$ $\quad F\varphi = [1\ \underline{U}\ \varphi]$
$[\varphi\ W\ \psi] = \neg[(\neg\varphi \vee \neg\psi)\ \underline{U}\ (\neg\varphi \wedge \psi)]$
$[\varphi\ \underline{W}\ \psi] = [(\neg\psi)\ \underline{U}\ (\varphi \wedge \psi)]$ *($\neg\psi$ holds until $\varphi \wedge \psi$)*
$[\varphi\ B\ \psi] = \neg[(\neg\varphi)\ \underline{U}\ \psi)]$
$[\varphi\ \underline{B}\ \psi] = [(\neg\psi)\ \underline{U}\ (\varphi \wedge \neg\psi)]$ *(ψ can't hold when φ holds)*
$[\varphi\ U\ \psi] = \neg[(\neg\psi)\ \underline{U}\ (\neg\varphi \wedge \neg\psi)]$
$[\varphi\ U\ \psi] = [\varphi\ \underline{U}\ \psi] \vee G\varphi$
$[\varphi\ \underline{U}\ \psi] = \neg[(\neg\psi)\ U\ (\neg\varphi \wedge \neg\psi)]$
$[\varphi\ \underline{U}\ \psi] = \neg[(\neg\psi)\ W\ (\varphi \rightarrow \psi)]$
$[\varphi\ \underline{U}\ \psi] = [\psi\ \underline{W}\ (\varphi \rightarrow \psi)]$
$[\varphi\ \underline{U}\ \psi] = \neg[(\neg\varphi)\ B\ \psi]$ *(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

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

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 $\mu - Calculus(\Phi_{inf} = \nu y.\Diamond y)$

$EX\varphi = \Diamond(\Phi_{inf} \wedge \varphi)$
$EG\varphi = \nu x.\varphi \wedge \Diamond x$
$EF\varphi = \mu x.\Phi_{inf} \wedge \varphi \vee \Diamond x$
$E[\varphi \underline{U} \psi] = \mu x.(\Phi_{inf} \wedge \psi) \vee \varphi \wedge \Diamond x$
$E[\varphi U \psi] = \nu x.(\Phi_{inf} \wedge \psi) \vee \varphi \wedge \Diamond x$
$E[\varphi \underline{B} \psi] = \mu x.\neg\psi \wedge (\Phi_{inf} \wedge \varphi \vee \Diamond x)$
$E[\varphi B \psi] = \nu x.\neg\psi \wedge (\Phi_{inf} \wedge \varphi \vee \Diamond x)$
$AX\varphi = \Box(\Phi_{inf} \rightarrow \varphi)$
$AG\varphi = \nu x.(\Phi_{inf} \rightarrow \varphi) \wedge \Box x$
$AF\varphi = \mu x.\varphi \vee \Box x$
$A[\varphi \underline{U} \psi] = \mu x.\psi \vee (\Phi_{inf} \rightarrow \varphi) \wedge \Box x$
$A[\varphi U \psi] = \nu x.\psi \vee (\Phi_{inf} \rightarrow \varphi) \wedge \Box x$
$A[\varphi \underline{B} \psi] = \mu x.(\Phi_{inf} \rightarrow \neg\psi) \wedge (\varphi \vee \Box x)$
$A[\varphi B \psi] = \nu x.(\Phi_{inf} \rightarrow \neg\psi) \wedge (\varphi \vee \Box x)$

CTL* to CTL - Existential Operators

$EX\varphi = EXE\varphi$
$EF\varphi = EFE\varphi$ $\quad EFG\varphi \equiv EFEG\varphi$
$E[\varphi\ W\ \psi] = E[(E\varphi)\ W\ \psi]$
$E[\varphi\ \underline{W}\ \psi] = E[(E\varphi)\ \underline{W}\ \psi]$
$E[\psi\ U\ \varphi] = E[\psi\ U\ E(\varphi)]$
$E[\psi\ \underline{U}\ \varphi] = E[\psi\ \underline{U}\ E(\varphi)]$
$E[\varphi\ B\ \psi] = E[(E\varphi)\ B\ \psi]$
$E[\varphi\ \underline{B}\ \psi] = E[(E\varphi)\ \underline{B}\ \psi]$

obs. $EGF\varphi \neq EGEF\varphi \rightarrow$ can't be converted

CTL* to CTL - Universal Operators

$AX\varphi = AXA\varphi$
$AG\varphi = AGA\varphi$
$A[\varphi\ W\ \psi] = A[(A\varphi)\ W\ \psi]$
$A[\varphi\ \underline{W}\ \psi] = A[(A\varphi)\ \underline{W}\ \psi]$
$A[\varphi\ U\ \psi] = A[A(\varphi)\ U\ \psi]$
$A[\varphi\ \underline{U}\ \psi] = A[A(\varphi)\ \underline{U}\ \psi]$
$A[\psi\ B\ \varphi] = A[\psi\ B\ (E(\varphi)]$
$A[\psi\ \underline{B}\ \varphi] = A[\psi\ \underline{B}\ (E(\varphi)]$

Eliminate boolean op. after path quantify

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

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

$$[\varphi_1\ U\ \psi_1] \wedge [\varphi_2\ U\ \psi_2] = \left[(\varphi_1 \wedge \varphi_2)\ U\ \begin{pmatrix}\psi_1 \wedge [\varphi_2\ U\psi_2] \vee \\ \psi_2 \wedge [\varphi_1\ U\psi_1]\end{pmatrix}\right]$$

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: $\phi = \text{let} \begin{bmatrix} x_1 = A\varphi_1 \\ \vdots \\ x_n = A\varphi_n \end{bmatrix}$ 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 Model Checking Given LTL formula $\Phi \equiv A\varphi$, translate $\neg\varphi$ to an ω-automaton $\mathfrak{A}_{\neg\varphi} = \mathcal{A}_\exists(Q, \varphi_\mathcal{I}, \varphi_\mathcal{R}, \varphi_\mathcal{F})$. Thus: $\mathcal{K} \vDash A\varphi \Leftrightarrow \mathcal{K} \vDash \neg E\neg\varphi \Leftrightarrow \mathcal{K} \vDash \mathfrak{A}_{\neg\varphi} \Leftrightarrow \mathcal{K} \times \mathcal{K}_\mathfrak{A} \vDash \neg E\varphi_\mathcal{F}$
Reduction to ω-automaton emptiness.

LTL to ω-automata

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

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* $typ_\Sigma(p) = \mathbb{N} \rightarrow \mathbb{B}$

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* $typ_\Sigma(_t) = typ_\Sigma(_{t1}) = typ_\Sigma(t_{t2}) = \mathbb{N}$
$-\varphi, \psi \in \zeta_{LO2}$
$-t \in V_\Sigma$ *with* $typ_\Sigma(t) = \mathbb{N}$
$-p \in V_\Sigma$ *with* $typ_\Sigma(p) = \mathbb{N} \rightarrow \mathbb{B}$

function LO2_S1S(Φ)
  case Φ **of**
    $t1 < t2$: **return** $\exists p.[\forall t.p^{(t)} \rightarrow p^{(SUC(t))}] \wedge \neg p^{(t1)} \wedge p^{(t2)}$:
    $p^{(t)}$: **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

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* $typ_\Sigma(p) = \mathbb{N} \rightarrow \mathbb{B}$
$\zeta_{LO2'}$ *hasnonumericvariables*
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**
    $t1 = t2$: **return** $Subset(q_{t1}, q_{t2}) \wedge Subset(q_{t2}, q_{t1})$

$t1 < t2 : \Psi :\equiv \forall q1.\forall q2.PSUC(q1, q2) \rightarrow [Subset(q1, p) \rightarrow Subset(q2, p)]$;
return $\exists p.\Psi \wedge \neg Subset(qt1, p) \wedge Subset(qt2, p)$;
$p^{(t)}$: **return** $Subset(qt, p)$
$\neg\varphi$: **return** $\neg ElimFO(\varphi)$;
$\varphi \wedge \psi$: **return** $ElimFO(\varphi) \wedge ElimFO(\psi)$;
$\varphi \vee \psi$: **return** $ElimFO(\varphi) \vee ElimFO(\psi)$;
$\exists t.\varphi$: **return** $\exists qt.Sing(qt) \wedge ElimFO(\varphi)$;
$\exists p.\varphi$: **return** $\exists p.ElimFO(\varphi)$;
end
end
function Tp2Od(t0, Φ) *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** $Tp2Od(\varphi) \vee Tp2Od(\psi)$;
$X\varphi : \Psi := \exists t1.(t0 < t1) \wedge (\forall t2.t0 < t2 \rightarrow t1 \leq t2) \wedge Tp2Od(t1, \varphi)$;
$[\varphi \underline{U} \psi] : \Psi := \exists t1.t0 \leq t1 \wedge Tp2Od(t1, \psi) \wedge interval((t0, 1, t1, 0), \varphi)$;
$[\varphi B \psi] : \Psi := \forall t1.t0 \leq t1 \wedge interval((t0, 1, t1, 0), \neg\varphi) \rightarrow Tp2Od(t1, \neg\psi)$;
$\overleftarrow{X}\varphi : \Psi := \forall t1.(t1 < t0) \wedge (\forall t2.t2 < t0 \rightarrow t2 \leq t1) \rightarrow Tp2Od(t1, \varphi)$;
$\overleftarrow{\underline{X}}\varphi : \Psi := \exists t1.(t1 < t0) \wedge (\forall t2.t2 < t0 \rightarrow t2 \leq t1) \wedge Tp2Od(t1, \varphi)$;
$[\varphi \overleftarrow{\underline{U}} \psi] : \Psi := \exists t1.t1 \leq t0 \wedge Tp2Od(t1, \psi) \wedge interval((t1, 0, t0, 1), \varphi)$;
$[\varphi \overleftarrow{B} \psi] : \Psi := \forall t1.t1 \leq t0 \wedge interval((t1, 0, t0, 1), \neg\varphi) \rightarrow Tp2Od(t1, \neg\psi)$;
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

Temporal Logic Equivalences and Tips

$[\varphi \underline{U} \psi] \equiv \varphi\ don't\ matter\ when\ \psi\ hold$
$[\varphi \underline{B} \psi] \equiv \psi\ can't\ hold\ when\ \varphi\ hold$
$[\varphi \underline{W} \psi] \equiv \neg\psi\ hold\ until\ \varphi\ \wedge\ \psi$
$[\varphi U \psi] \equiv [\varphi \underline{U} \psi] \vee G\varphi$
$[a \underline{U} Fb] \equiv Fb$
$F[a \underline{U} b] \equiv Fb \equiv [Fa \underline{U} Fb]$
$[\varphi B \psi] \equiv [\varphi \underline{B} \psi] \vee G\neg\psi$
$F[a \underline{B} b] \equiv F[a \wedge \neg b]$
$[\varphi W \psi] \equiv \neg[\neg\varphi \underline{W} \psi]$
$E(\varphi \wedge \psi) \equiv E\varphi \wedge E\psi (in\ general)$
$AEA \equiv A$
$GF(x \vee y) \equiv GFx \vee GFy$
$FF\varphi \equiv F\varphi$
$GG\varphi \equiv G\varphi$
$GF\varphi \equiv XGF\varphi \equiv FGF\varphi \equiv GGF\varphi \equiv GFGF\varphi \equiv FGGF\varphi$
$FG\varphi \equiv XFG\varphi \equiv FFG\varphi \equiv GFG\varphi \equiv GFFG\varphi \equiv FGFG\varphi$
$\Rightarrow$ Rules from F apply to E and rules from G to A.
LTL formulas that can't be translated for CTL
* E[([a1 SU b1] | [a2 SU b2]) SU c]
* E[(a | [b SU c]) SU d]
* E[[[a SU b] SU c] SU d]
* E[[a SU [b SU c]] SU d]
* EG[([a1 SU b1] | [a2 SU b2]) SU c]
* EG(a | X a)
* EG(a | [b SU c])
* EG[[a SU b] SU c]
* EG[a SU [b SU c]]

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)

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

ω-Automaton to LO2

$A_\exists(q1, ..., qn, \psi I, \psi R, \psi F) (input\ automaton)$
$\exists q1..qn, \Theta LO2(0, \psi I) \wedge (\forall t.\Theta LO2(t, \psi R)) \wedge (\forall.t1 \exists t2.t1 < t2 \wedge \Theta LO2(t2, \psi F))$

Where ΘLO2(t, Φ) 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

$AG[\varphi\ U\ \psi] = AG(\varphi \vee \psi)$
$AG[\varphi\ B\ \psi] = AG(\neg\psi)$
$AG[\varphi\ W\ \psi] = AG(\psi \rightarrow \varphi)$
$AG[\varphi\ \underline{U}\ \psi] = A(G(\varphi \vee \psi) \wedge GF\psi)$
$AG[\varphi\ \underline{B}\psi] = A(G(\neg\psi) \wedge GF\varphi)$
$AG[\varphi\ \underline{W}\psi] = A(G(\psi \rightarrow \varphi) \wedge GF\psi)$
// note that the following are only initially, but not generally valid
$AG\overleftarrow{X}\varphi = AG\varphi$
$AG\overleftarrow{\underline{X}}\varphi = A(\text{false})$
$AG\overleftarrow{G}\varphi = AG\varphi$
$AG\overleftarrow{F}\varphi = A\varphi$
$AG[\varphi\ \overleftarrow{U}\ \psi] = AG(\varphi \vee \psi)$
$AG[\varphi\ \overleftarrow{B}\ \psi] = AG(\neg\psi)$
$AG[\varphi\ \overleftarrow{W}\ \psi] = AG(\psi \rightarrow \varphi)$
$AG[\varphi\ \overleftarrow{\underline{U}}\ \psi] = A(\psi \wedge G(\varphi \vee \psi))$
$AG[\varphi\ \overleftarrow{\underline{B}}\ \psi] = A(\varphi \wedge G(\neg\psi))$
$AG[\varphi\ \overleftarrow{\underline{W}}\ \psi] = A(\psi \wedge G(\psi \rightarrow \varphi))$

Extra Equations F

$AFF\psi = AF\psi$
$AF[\varphi\ \underline{U}\ \psi] = AF\psi$
$AF[\varphi\ \underline{B}\ \psi] = AF(\varphi \wedge \neg\psi)$
$AF[\varphi\ \underline{W}\ \psi] = AF(\varphi \wedge \psi)$
$AF[\varphi\ U\ \psi] = A(F(\psi) \vee FG\varphi)$
$AF[\varphi\ B\ \psi] = A(F(\varphi \wedge \neg\psi) \vee FG(\neg\varphi \wedge \neg\psi))$
$AF[\varphi\ W\ \psi] = A(F(\varphi \wedge \psi) \vee FG\neg\psi)$
// note that the following are only initially, but not generally valid
$AF\overleftarrow{X}\varphi = A(\text{true})$
$AF\overleftarrow{\underline{X}}\varphi = AF\varphi$
$AF\overleftarrow{G}\varphi = A\varphi$
$AF\overleftarrow{F}\varphi = AF\varphi$
$AF[\varphi\ \overleftarrow{\underline{U}}\ \psi] = AF\psi$
$AF[\varphi\ \overleftarrow{\underline{B}}\ \psi] = AF(\varphi \wedge \neg\psi)$
$AF[\varphi\ \overleftarrow{\underline{W}}\ \psi] = AF(\varphi \wedge \psi)$
$AF[\varphi\ \overleftarrow{U}\ \psi] = A(F\psi \vee F\overleftarrow{G}\varphi)$
$AF[\varphi\ \overleftarrow{B}\ \psi] = A(F(\varphi \wedge \neg\psi) \vee F\overleftarrow{G}(\neg\varphi \wedge \neg\psi))$
$AF[\varphi\ \overleftarrow{W}\ \psi] = A(F(\varphi \wedge \psi) \vee F\overleftarrow{G}\neg\psi)$