Propositional Logic Syntactic Sugar

$\varphi \Leftrightarrow \psi := (\neg\varphi \lor \psi) \land (\neg\psi \lor \varphi)$ $\quad \varphi \to \psi := \neg\varphi \lor \psi$

$\varphi \oplus \psi := (\varphi \land \neg\psi) \lor (\psi \land \neg\varphi)$ $\quad \varphi \bar{\land} \psi := \neg(\varphi \land \psi)$

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

Distributivity: $a \land (b \lor c) = (a \land b) \lor (a \land c)$
$a \lor (b \land c) = (a \lor b) \land (a \lor c)$

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

Satisfiability, Validity and Equivalence

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

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

Sequent Calculus:

— *Validity* of ϕ: start with $\{\} \vdash \phi$; ϕ is valid iff $\Gamma \cap \Delta \neq \{\}$ FOR ALL leaves.

— *Satisfiability* of ϕ: start with $\{\phi\} \vdash \{\}$; ϕ is satisfiable iff $\Gamma \cap \Delta = \{\}$ for AT LEAST ONE leaf.

— 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	$\frac{\phi\land\psi,\Gamma\vdash\Delta}{\phi,\psi,\Gamma\vdash\Delta}$	$\frac{\Gamma\vdash\phi\land\psi,\Delta}{\Gamma\vdash\phi,\Delta \qquad \Gamma\vdash\psi,\Delta}$
OR	$\frac{\phi\lor\psi,\Gamma\vdash\Delta}{\phi,\Gamma\vdash\Delta \qquad \psi,\Gamma\vdash\Delta}$	$\frac{\Gamma\vdash\phi\lor\psi,\Delta}{\Gamma\vdash\phi,\psi,\Delta}$

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.

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) $\to$ 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., $\neg a \lor b$ becomes $x_1 \leftrightarrow \neg a; x_2 \leftrightarrow x_1 \lor b$. Use rules below to find CNF. Create last clause {Xn}

$x \leftrightarrow \neg y \Leftrightarrow (\neg x \lor \neg y) \land (x \lor y)$

$x \leftrightarrow y_1 \land y_2 \Leftrightarrow$
$(\neg x \lor y_1) \land (\neg x \lor y_2) \land (x \lor \neg y_1 \lor \neg y_2)$

$x \leftrightarrow y_1 \lor y_2 \Leftrightarrow$
$(\neg x \lor y_1 \lor y_2) \land (x \lor \neg y_1) \land (x \lor \neg y_2)$

$x \leftrightarrow y_1 \to y_2 \Leftrightarrow$
$(x \lor y_1) \land (x \lor \neg y_1 \lor \neg y_2) \land (\neg x \lor \neg y_1 \lor y_2)$

$x \leftrightarrow (y_1 \leftrightarrow y_2) \Leftrightarrow (x \lor y_1 \lor y_2) \land (x \lor \neg y_1 \lor \neg y_2) \land$
$(\neg x \lor y_1 \lor \neg y_2) \land (\neg x \lor \neg y_1 \lor y_2)$

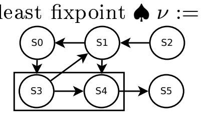
$x \leftrightarrow y_1 \oplus y_2 \Leftrightarrow (x \lor \neg y_1 \lor y_2) \land (x \lor y_1 \lor \neg y_2) \land$
$(\neg x \lor y_1 \lor y_2) \land (\neg x \lor \neg y_1 \lor \neg y_2)$

Quotient: Bisimulation of the kripke with itself

Product – $\mathcal{K}_1 \times \mathcal{K}_2 = (\mathcal{V}_1 \cup \mathcal{V}_2, \varphi_\mathcal{I} \land \psi_\mathcal{I}, \varphi_\mathcal{R} \land \psi_\mathcal{R})$

Quantif. $\exists x.\varphi := [\varphi]_x^1 \lor [\varphi]_x^0$ ♣ $\forall x.\varphi := [\varphi]_x^1 \land [\varphi]_x^0$

Tarski-Knaster Theorem: μ := starts $\bot$ ⇀ least fixpoint ♠ ν := starts $\top$ ⇀ greatest fixpoint



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

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

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

```
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
```

```
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(∨,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
      1 if happens!)
```

$\varphi = [\varphi]_x^0 \oplus x \land (\partial\varphi/\partial x)$
$(\partial\varphi/\partial x) := [\varphi]_x^0 \oplus [\varphi]_x^1$

Local Model Checking

$\frac{s\vdash_\Phi \varphi\land\psi}{\{s\vdash_\Phi\varphi\} \quad \{s\vdash_\Phi\psi\}}\land$		$\frac{s\vdash_\Phi \varphi\lor\psi}{\{s\vdash_\Phi\varphi\} \quad \{s\vdash_\Phi\psi\}}\lor$	
$\frac{s\vdash_\Phi \Box\varphi}{\{s_1\vdash_\Phi\varphi\}\ldots\{s_n\vdash_\Phi\varphi\}}\land$		$\frac{s\vdash_\Phi \Diamond\varphi}{\{s_1\vdash_\Phi\varphi\}\ldots\{s_n\vdash_\Phi\varphi\}}\lor$	
$\frac{s\vdash_\Phi \overleftarrow{\Box}\varphi}{\{s_1'\vdash_\Phi\varphi\}\ldots\{s_n'\vdash_\Phi\varphi\}}\land$		$\frac{s\vdash_\Phi \overleftarrow{\Diamond}\varphi}{\{s_1'\vdash_\Phi\varphi\}\ldots\{s_n'\vdash_\Phi\varphi\}}\lor$	
$\frac{s\vdash_\Phi \mu x.\varphi}{s\vdash_\Phi\varphi}$	$\frac{s\vdash_\Phi \nu x.\varphi}{s\vdash_\Phi\varphi}$	$\frac{s\vdash_\Phi x}{s\vdash_\Phi\mathfrak{D}_\Phi(x)}$	$\mathfrak{D}_\Phi$(replace w. initial form.)
$\{s_1\ldots s_n\} = suc_\exists^\mathcal{R}(s)$ and $\{s_1'\ldots s_n'\} = pre_\exists^\mathcal{R}(s)$			

If (s,$\mu x.\varphi$) repeats→0 ♡ If (s,$\nu x.\varphi$) repeats→1

Predecessor and Successor

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

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

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

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

Automata types: G→Safety; F→Liveness; FG→Persistence/Co-Buchi; GF→Fairness/Buchi.

Automaton Determinization

$\mathrm{NDet_G} \to \mathrm{Det_G}$: 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.

$\{\mathrm{NDet_F(partial)}$ or $\mathrm{NDet_{prefix}}\} \to \mathrm{Det_{FG}}$: Breakpoint Construction.

$\mathrm{NDet_F}$ (total)$\to\mathrm{Det_F}$: Subset Construction.

$\mathrm{NDet_{FG}} \to \mathrm{Det_{FG}}$: Breakpoint Construction.

$\mathrm{NDet_{GF}} \to \{\mathrm{Det_{Rabin}}$ or $\mathrm{Det_{Streett}}\}$: Safra

*** *Rabin-Scott Subset Construction*** Acceptance condition:set of states containing acceptance states.

*** *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

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

Disjunction

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

If both automata are totally defined,

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

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

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

Boolean Operations of G

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

(3)$G\varphi \lor G\psi = \mathcal{A}_\exists(\{p,q\}, p \land q,$
$[p' \leftrightarrow p \land \varphi] \land [q' \leftrightarrow q \land \psi], G[p \lor q])$

Boolean Operations of F

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

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

Boolean Operations of FG

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

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

Boolean Operations of GF

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

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

Transformation of Acceptance Conditions

Reduction of G

$G\varphi = \mathcal{A}_\exists(\{q\}, q, \varphi \land q \land q', Fq))$

$G\varphi = \mathcal{A}_\exists(\{q\}, q, q' \leftrightarrow q \land \varphi, FGq)$

$G\varphi = \mathcal{A}_\exists(\{q\}, q, q' \leftrightarrow q \land \varphi, GFq)$

Reduction of F

$F\varphi$ can **not** be expressed by $NDet_G$

$F\varphi = \mathcal{A}_\exists(\{q\}, \neg q, q' \leftrightarrow q \lor \varphi, FGq)$

$F\varphi = \mathcal{A}_\exists(\{q\}, \neg q, q' \leftrightarrow q \lor \varphi, GFq)$

Reduction of FG

$FG\varphi$ can **not** be expressed by $NDet_G$

$FG\varphi = \mathcal{A}_\exists(\{q\}, \neg q, q \to \varphi \land q', Fq)$

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

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

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.

$F\varphi = \varphi \lor XF\varphi$ $\qquad G\varphi = \varphi \land XG\varphi$

$[\varphi\ U\ \psi] = \psi \lor (\varphi \land X[\varphi\ U\ \psi])$

$[\varphi\ B\ \psi] = \neg\psi \land (\varphi \lor X[\varphi\ B\ \psi])$

$[\varphi\ W\ \psi] = (\psi \land \varphi) \lor (\neg\psi \land X[\varphi\ W\ \psi])$

Negation Normal Form

$\neg(\varphi \land \psi) = \neg\varphi \lor \neg\psi$ $\qquad \neg(\varphi \lor \psi) = \neg\varphi \land \neg\psi$

$\neg\neg\varphi = \varphi$ $\qquad \neg X\varphi = X\neg\varphi$

$\neg G\varphi = F\neg\varphi$ $\qquad \neg F\varphi = G\neg\varphi$

$\neg[\varphi\ U\ \psi] = [(\neg\varphi)\ \underline{B}\ \psi]$ $\qquad \neg[\varphi\ \underline{U}\ \psi] = [(\neg\varphi)\ B\ \psi]$

$\neg[\varphi\ B\ \psi] = [(\neg\varphi)\ \underline{U}\ \psi]$ $\qquad \neg[\varphi\ \underline{B}\ \psi] = [(\neg\varphi)\ U\ \psi]$

$\neg A\varphi = E\neg\varphi$ $\qquad \neg E\varphi = A\neg\varphi$

$\neg\overleftarrow{X}\varphi = \overleftarrow{\underline{X}}\neg\varphi$ $\qquad \neg\overleftarrow{\underline{X}}\varphi = \overleftarrow{X}\neg\varphi$

$\neg\overleftarrow{G}\varphi = \overleftarrow{F}\neg\varphi$ $\qquad \neg\overleftarrow{F}\varphi = \overleftarrow{G}\neg\varphi$

$\neg[\varphi\ \overleftarrow{U}\ \psi] = [(\neg\varphi)\ \overleftarrow{\underline{B}}\ \psi]$ $\qquad \neg[\varphi\ \overleftarrow{\underline{U}}\ \psi] = [(\neg\varphi)\ \overleftarrow{B}\ \psi]$

$\neg[\varphi\ \overleftarrow{B}\ \psi] = [(\neg\varphi)\ \overleftarrow{\underline{U}}\ \psi]$ $\qquad \neg[\varphi\ \overleftarrow{\underline{B}}\ \psi] = [(\neg\varphi)\ \overleftarrow{U}\ \psi]$

LTL Syntactic Sugar: analog for past operators

$G\varphi = \neg[1\ \underline{U}\ (\neg\varphi)]$ $\qquad F\varphi = [1\ \underline{U}\ \varphi]$

$[\varphi\ W\ \psi] = \neg[(\neg\varphi \lor \neg\psi)\ \underline{U}\ (\neg\varphi \land \psi)]$

$[\varphi\ \underline{W}\ \psi] = [(\neg\psi)\ \underline{U}\ (\varphi \land \psi)]$ *($\neg\psi$ holds until $\varphi \land \psi$)*

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

$[\varphi\ \underline{B}\ \psi] = [(\neg\psi)\ \underline{U}\ (\varphi \land \neg\psi)]$*($\psi$ can't hold when φ holds)*

$[\varphi\ U\ \psi] = \neg[(\neg\psi)\ \underline{U}\ (\neg\varphi \land \neg\psi)]$

$[\varphi\ U\ \psi] = [\varphi\ \underline{U}\ \psi] \lor G\varphi$

$[\varphi\ \underline{U}\ \psi] = \neg[(\neg\psi)\ U\ (\neg\varphi \land \neg\psi)]$

$[\varphi\ \underline{U}\ \psi] = \neg[(\neg\psi)\ W\ (\varphi \to \psi)]$

$[\varphi\ \underline{U}\ \psi] = [\psi\ \underline{W}\ (\varphi \to \psi)]$

$[\varphi\ \underline{U}\ \psi] = \neg[(\neg\varphi)\ B\ \psi]$*($\varphi$ doesn't matter when ψ holds)*

$[\varphi\ \underline{U}\ \psi] = [\psi\ \underline{B}\ (\neg\varphi \land \neg\psi)]$

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: ϕ = let $\left[\begin{array}{c} x_1 = A\varphi_1 \\ \vdots \\ x_n = A\varphi_n \end{array}\right]$ 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 $[\![\Psi]\!]_{\mathcal{K}_n}$

CTL Syntactic Sugar: analog for past operators

Existential Operators

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

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

STRONG: op=μ / WEAK: op=ν
$EX\varphi = \Diamond(\Phi_{inf} \wedge \varphi)$
$EG\varphi = \nu x.\varphi \wedge \Diamond x$
$EF\varphi = \mu x.\Phi_{inf} \wedge \varphi \vee \Diamond x$
$E[\varphi U \psi] = \text{op } x.(\Phi_{inf} \wedge \psi) \vee \varphi \wedge \Diamond x$
$E[\varphi B \psi] = \text{op } x.\neg\psi \wedge (\Phi_{inf} \wedge \varphi \vee \Diamond x)$
$AX\varphi = \Box(\Phi_{inf} \rightarrow \varphi)$
$AG\varphi = \nu x.(\Phi_{inf} \rightarrow \varphi) \wedge \Box x$
$AF\varphi = \mu x.\varphi \vee \Box x$
$A[\varphi U \psi] = \text{op } x.\psi \vee (\Phi_{inf} \rightarrow \varphi) \wedge \Box x$
$A[\varphi B \psi] = \text{op } x.(\Phi_{inf} \rightarrow \neg\psi) \wedge (\varphi \vee \Box x)$

CTL* to CTL - Existential Operators (weak = str)

$EX\varphi = EXE\varphi$
$EF\varphi = EFE\varphi$ $\quad EFG\varphi \equiv EFEG\varphi$
$E[\varphi\ W\ \psi] = E[(E\varphi)\ W\ \psi]$
$E[\psi\ U\ \varphi] = E[\psi\ U\ E(\varphi)]$
$E[\varphi\ B\ \psi] = E[(E\varphi)\ B\ \psi]$
obs. $EGF\varphi \neq EGEF\varphi \rightarrow$ can't be converted

CTL* to CTL - Universal Operators (weak = str)

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

Weak Equivalences

$*[\varphi U \psi] := [\varphi \underline{U} \psi] \vee G\varphi$ $\quad *[\varphi B \psi] := [\varphi \underline{B} \psi] \vee G\neg\psi$
*same to past version $\quad [\varphi W \psi] := \neg[(\neg\varphi)\underline{W}\psi]$
$\overleftarrow{X}\varphi := \neg\underline{\overleftarrow{X}}\neg\varphi$ $(at\ t0:\ weak\ true.\ strong\ false)$

Eliminate boolean op. after path quantify

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

LTL to ω-automata

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

Equivalences and Tips

$[\varphi\ \underline{U}\ \psi] \equiv \varphi\ don't\ matter\ when\ \psi\ hold$
$[\varphi\ \underline{B}\ \psi] \equiv \psi\ can't\ hold\ when\ \varphi\ hold$
$[\varphi\ \underline{W}\ \psi] \equiv \neg\psi\ hold\ until\ \varphi\ \wedge\ \psi$
$F[a\ \underline{U}\ b] \equiv Fb \equiv [Fa\underline{U}Fb] \equiv [a\underline{U}Fb]$
$F[a\ \underline{B}\ b] \equiv F[a \wedge \neg b]$
$AEA \equiv A$ $\quad GFX \equiv GXF$
$FF\varphi \equiv F\varphi$ $\quad GG\varphi \equiv G\varphi$
$GF\varphi \equiv XGF\varphi \equiv FGF\varphi \equiv GGF\varphi \equiv GFGF\varphi \equiv FGGF\varphi$
$FG\varphi \equiv XFG\varphi \equiv FFG\varphi \equiv GFG\varphi \equiv GFFG\varphi \equiv FGFG\varphi$
$E([a\ \underline{U}\ b] \wedge [c\ \underline{U}\ d]) \equiv$
$\quad E[(a \wedge c)\underline{U}(b \wedge E([c\ \underline{U}\ d]) \vee d \wedge E([a\ \underline{U}\ b]))]$
$\Rightarrow$ Rules from F apply to E and rules from G to A.

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