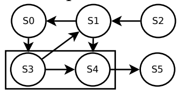
Propositional Logic Syntactic Sugar
$\varphi\Leftrightarrow\psi := (\neg\varphi\vee\psi)\wedge(\neg\psi\vee\varphi)$ $\quad\varphi\to\psi := \neg\varphi\vee\psi$
$\varphi\oplus\psi := (\varphi\wedge\neg\psi)\vee(\psi\wedge\neg\varphi)$ $\quad\varphi\,\bar{\wedge}\,\psi := \neg(\varphi\wedge\psi)$
$(\alpha\Rightarrow\beta|\gamma) := (\neg\alpha\vee\beta)\wedge(\alpha\vee\gamma)$ $\quad\varphi\bar{\vee}\psi := \neg(\varphi\vee\psi)$
Distributivity: $a\wedge(b\vee c)=(a\wedge b)\vee(a\wedge c)$
$a\vee(b\wedge c)=(a\vee b)\wedge(a\vee c)$
CNF: from truth table, take minterms that are 0. Each minterm is built as an OR of the negated variables. E.g., $(0,0,1)\to(x\vee y\vee\neg z)$.
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	$\frac{\phi\wedge\psi,\Gamma\vdash\Delta}{\phi,\psi,\Gamma\vdash\Delta}$	$\frac{\Gamma\vdash\phi\wedge\psi,\Delta}{\Gamma\vdash\phi,\Delta\qquad\Gamma\vdash\psi,\Delta}$
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.
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., $\neg a\vee b$ becomes $x_1\leftrightarrow\neg a; x_2\leftrightarrow x_1\vee b$. Use rules below to find CNF. Create last clause {Xn}
$x\leftrightarrow\neg y\Leftrightarrow(\neg x\vee\neg y)\wedge(x\vee y)$
$x\leftrightarrow y_1\wedge y_2\Leftrightarrow$
$(\neg x\vee y_1)\wedge(\neg x\vee y_2)\wedge(x\vee\neg y_1\vee\neg y_2)$
$x\leftrightarrow y_1\vee y_2\Leftrightarrow$
$(\neg x\vee y_1\vee y_2)\wedge(x\vee\neg y_1)\wedge(x\vee\neg y_2)$
$x\leftrightarrow y_1\to y_2\Leftrightarrow$
$(x\vee y_1)\wedge(x\vee\neg y_1\vee\neg y_2)\wedge(\neg x\vee\neg y_1\vee y_2)$
$x\leftrightarrow(y_1\leftrightarrow y_2)\Leftrightarrow(x\vee y_1\vee y_2)\wedge(x\vee\neg y_1\vee\neg y_2)\wedge$
$(\neg x\vee y_1\vee\neg y_2)\wedge(\neg x\vee\neg y_1\vee y_2)$
$x\leftrightarrow y_1\oplus y_2\Leftrightarrow(x\vee\neg y_1\vee y_2)\wedge(x\vee y_1\vee\neg y_2)\wedge$
$(\neg x\vee y_1\vee y_2)\wedge(\neg x\vee\neg y_1\vee\neg y_2)$
Quotient: Bisimulation of the kripke with itself
Product – $\mathcal{K}_1\times\mathcal{K}_2=(\mathcal{V}_1\cup\mathcal{V}_2,\varphi_{\mathcal{I}}\wedge\psi_{\mathcal{I}},\varphi_{\mathcal{R}}\wedge\psi_{\mathcal{R}})$
Quantif. $\exists x.\varphi := [\varphi]^1_x\vee[\varphi]^0_x$ ♣ $\forall x.\varphi := [\varphi]^1_x\wedge[\varphi]^0_x$
Tarski-Knaster Theorem: μ := starts $\perp\to$ least fixpoint ♠ ν := starts $\top\to$ greatest fixpoint



Example: $\Box/\overleftarrow{\Box}$
$pre^{\mathcal{R}}_\forall(\{S3,S4\})=\{S0,S5\}$
$suc^{\mathcal{R}}_\forall(\{S3,S4\})=\{S2,S5\}$

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

Local Model Checking

$\frac{s\vdash_\Phi\varphi\wedge\psi}{\{s\vdash_\Phi\varphi\}\quad\{s\vdash_\Phi\psi\}}\wedge$		$\frac{s\vdash_\Phi\varphi\vee\psi}{\{s\vdash_\Phi\varphi\}\quad\{s\vdash_\Phi\psi\}}\vee$	
$\frac{s\vdash_\Phi\Box\varphi}{\{s_1\vdash_\Phi\varphi\}\ldots\{s_n\vdash_\Phi\varphi\}}\wedge$		$\frac{s\vdash_\Phi\Diamond\varphi}{\{s_1\vdash_\Phi\varphi\}\ldots\{s_n\vdash_\Phi\varphi\}}\vee$	
$\frac{s\vdash_\Phi\overleftarrow{\Box}\varphi}{\{s'_1\vdash_\Phi\varphi\}\ldots\{s'_n\vdash_\Phi\varphi\}}\wedge$		$\frac{s\vdash_\Phi\overleftarrow{\Diamond}\varphi}{\{s'_1\vdash_\Phi\varphi\}\ldots\{s'_n\vdash_\Phi\varphi\}}\vee$	
$\frac{s\vdash_\Phi\mu x.\varphi}{s\vdash_\Phi\varphi}$	$\frac{s\vdash_\Phi\nu x.\varphi}{s\vdash_\Phi\varphi}$	$\frac{s\vdash_\Phi x}{s\vdash_\Phi\mathfrak{D}_\Phi(x)}$	$\frac{\mathfrak{D}_\Phi\text{(replace w.}}{\text{initial form.)}}$
$\{s_1\ldots s_n\}=suc^{\mathcal{R}}_\exists(s)$ and $\{s'_1\ldots s'_n\}=pre^{\mathcal{R}}_\exists(s)$			

If (s,$\mu x.\varphi$) repeats$\to$1 ♡ If (s,$\nu x.\varphi$) repeats$\to$0
Predecessor and Successor
$\Diamond := pre^{\mathcal{R}}_\exists(Q) := \exists x'_1,...,x'_n.\varphi_{\mathcal{R}}\wedge[\varphi_Q]^{x'_1,...,x'_n}_{x_1,...,x_n}$
$\overleftarrow{\Diamond} := suc^{\mathcal{R}}_\exists(Q) := [\exists x_1,...,x_n.\varphi_{\mathcal{R}}\wedge\varphi_Q]^{x_1,...,x_n}_{x'_1,...,x'_n}$
$\Box = pre^{\mathcal{R}}_\forall(Q) := \forall x'_1,...,x'_n.\varphi_{\mathcal{R}}\to[\varphi_Q]^{x'_1,...,x'_n}_{x_1,...,x_n}$
$\overleftarrow{\Box} := suc^{\mathcal{R}}_\forall(Q) := [\forall x_1,...,x_n.\varphi_{\mathcal{R}}\to\varphi_Q]^{x_1,...,x_n}_{x'_1,...,x'_n}$
Automata types: G$\to$Safety; F$\to$Liveness; FG$\to$Persistence/Co-Buchi; GF$\to$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
**** 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:
$$\begin{cases}\text{If } Q_f=\{\} & (suc^{\mathcal{R}_a}_\exists(Q),(suc^{\mathcal{R}_a}_\exists(Q)\cap\mathcal{F})\\ \text{Otherwise} & (suc^{\mathcal{R}_a}_\exists(Q),(suc^{\mathcal{R}_a}_\exists(Q_f)\cap\mathcal{F})\end{cases}$$
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
$$\mathcal{A}_\exists(Q^1,\mathcal{I}^1_1,\mathcal{R}^1_1,\mathcal{A}_\exists(Q^2,\mathcal{I}^2_1,\mathcal{R}^2_1,\mathcal{F}_1))$$
$$=\mathcal{A}_\exists(Q^1\cup Q^2,\mathcal{I}^1_1\wedge\mathcal{I}^2_1,\mathcal{R}^1_1\wedge\mathcal{R}^2_1,\mathcal{F}_1))$$
Boolean Operations of G
(1)$\neg G\varphi=F\neg\varphi$ $\quad$(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 $NDet_G$
$FG\varphi=\mathcal{A}_\exists(\{q\},\neg q,q\to\varphi\wedge q',Fq)$
$$FG\varphi=\mathcal{A}_\exists\left(\begin{array}{c}\{p,q\},\qquad\neg p\wedge\neg q,\\ \left[\begin{array}{c}(p\to p')\wedge(p'\to p\vee\neg q)\wedge\\(q'\leftrightarrow(p\wedge\neg q\vee\neg\varphi)\vee(p\wedge q))\end{array}\right],\\ G\neg q\wedge Fp\end{array}\right)$$
$$FG\varphi=\mathcal{A}_\exists\left(\begin{array}{c}\{p,q\},\qquad\neg p\wedge\neg q,\\ \left[\begin{array}{c}(p\to p')\wedge(p'\to p\vee\neg q)\wedge\\(q'\leftrightarrow(p\wedge\neg q\vee\neg\varphi)\vee(p\wedge q))\end{array}\right],\\ GF[p\wedge\neg q]\end{array}\right)$$
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.
$F\varphi=\varphi\vee XF\varphi$ $\qquad G\varphi=\varphi\wedge XG\varphi$
$[\varphi\ U\ \psi]=\psi\vee(\varphi\wedge X[\varphi\ U\ \psi])$
$[\varphi\ B\ \psi]=\neg\psi\wedge(\varphi\vee X[\varphi\ B\ \psi])$
$[\varphi\ W\ \psi]=(\psi\wedge\varphi)\vee(\neg\psi\wedge X[\varphi\ W\ \psi])$
Negation Normal Form

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

LTL Syntactic Sugar: analog for past operators
$G\varphi=\neg[1\ \underline{U}\ (\neg\varphi)]$ $\qquad F\varphi=[1\ \underline{U}\ \varphi]$
$[\varphi\ W\ \psi]=\neg[(\neg\varphi\vee\neg\psi)\ \underline{U}\ (\neg\varphi\wedge\psi)]$
$[\varphi\ \underline{W}\ \psi]=[(\neg\psi)\ \underline{U}\ (\varphi\wedge\psi)]$ *($\neg\psi$ 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)*
$[\varphi\ U\ \psi]=\neg[(\neg\psi)\ \underline{U}\ (\neg\varphi\wedge\neg\psi)]$
$[\varphi\ U\ \psi]=[\varphi\ \underline{U}\ \psi]\vee G\varphi$
$[\varphi\ \underline{U}\ \psi]=\neg[(\neg\psi)\ U\ (\neg\varphi\wedge\neg\psi)]$
$[\varphi\ \underline{U}\ \psi]=\neg[(\neg\psi)\ W\ (\varphi\to\psi)]$
$[\varphi\ \underline{U}\ \psi]=[\psi\ \underline{W}\ (\varphi\to\psi)]$
$[\varphi\ \underline{U}\ \psi]=\neg[(\neg\varphi)\ B\ \psi]$ *(φ doesn't matter when ψ holds)*
$[\varphi\ \underline{U}\ \psi]=[\psi\ \underline{B}\ (\neg\varphi\wedge\neg\psi)]$
CTL* Modelchecking to LTL model checking
Let's φ_i be a pure path formula (without path quantifiers), Ψ be a propositional formula, abbreviate subformulas $E\psi$ 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

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

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

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

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

obs. $EGF\varphi \neq EGEF\varphi \to$ 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

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

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 .

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

Equivalences and Tips

$[\varphi\underline{U}\psi] \equiv \varphi\ don't\ matter\ when\ \psi\ hold$
$[\varphi\underline{B}\psi] \equiv \psi\ can't\ hold\ when\ \varphi\ hold$
$[\varphi\underline{W}\psi] \equiv \neg\psi\ hold\ until\ \varphi\ \wedge\ \psi$
$F[a\underline{U}b] \equiv Fb \equiv [Fa\underline{U}Fb] \equiv [a\underline{U}Fb]$
$F[a\underline{B}b] \equiv F[a \wedge \neg b]$
$AEA \equiv A$ $\qquad GFX \equiv GXF$
$FF\varphi \equiv F\varphi$ $\qquad GG\varphi \equiv G\varphi$
$GF\varphi \equiv XGF\varphi \equiv FGF\varphi \equiv GGF\varphi \equiv GFGF\varphi \equiv FGGF\varphi$
$FG\varphi \equiv XFG\varphi \equiv FFG\varphi \equiv GFG\varphi \equiv GFFG\varphi \equiv FGFG\varphi$
$E[(a\underline{U}b) \wedge (c\underline{U}d)] \equiv$
$\quad E[(a \wedge c)\underline{U}(b \wedge E(c\underline{U}d) \vee d \wedge E(a\underline{U}b))]$
$\Rightarrow$ 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
$\forall t1.\exists t2.\pi^{(t1+t2)} \in [\varphi]_K ?????t1 + t2ort1 + t0?????$
-φ holds infinitely often on π