Bounded Model Checking

H. Peter Gumm

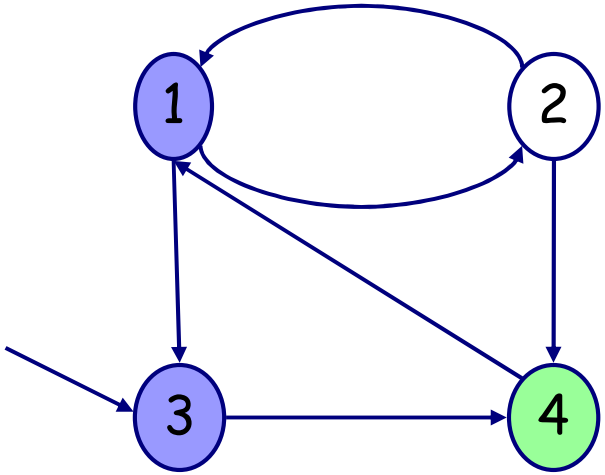
Philipps-Universität Marburg

Sommersemester 2007

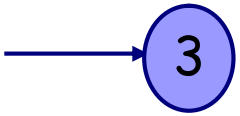
Idee

- Suche kleine Gegenbeispiele zu einer LTL-Formel φ in Kripke-Struktur M
  - Probiere Pfade der Länge k
  - Vergrößere k
  - Konstruiere jedesmal Formel, $[M,\varphi]_k$, die genau dann erfüllbar ist, wenn ein Gegenbeispiel der Länge k existiert
  - Größe der Formel $\leq k*|S|$
- Vermeidung der State explosion
  - in der Praxis: Formeln mit mehreren hunderttausend Variablen
  - möglich durch immensen Fortschrit in SAT-Solvern
- Sehr gut zum debugging
  - Gegenbeispiele werden automatisch gefunden
  - minimale Gegenbeispiele

G (blau⇒ F gruen) ?

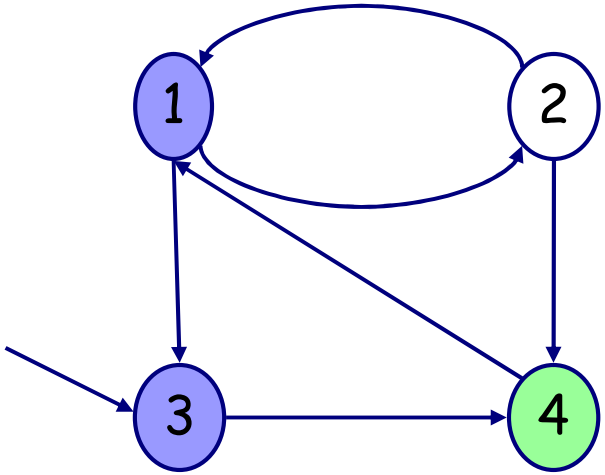


- $\varphi = G(b \Rightarrow F\, g)$
- k = 0 :

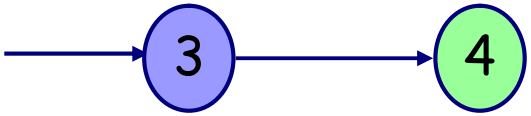


kein Gegenbeispiel

G blau⇒ F gruen ?

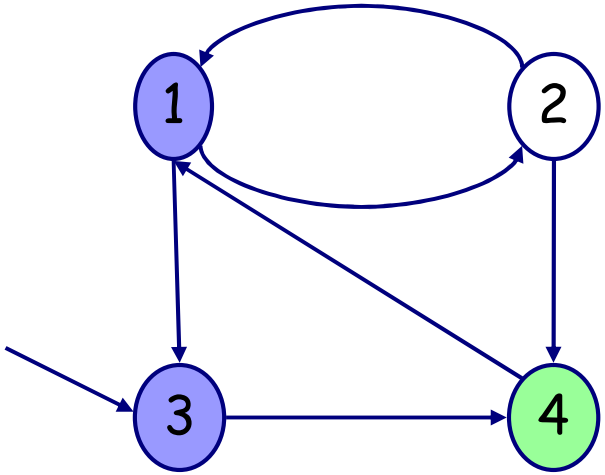


- $\varphi = G(b \Rightarrow F\, g)$
- k = 1 :

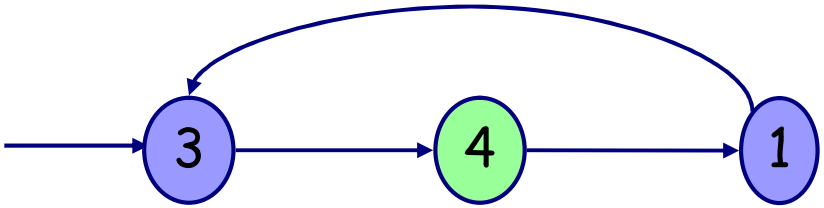


kein Gegenbeispiel

G blau⇒ F gruen ?

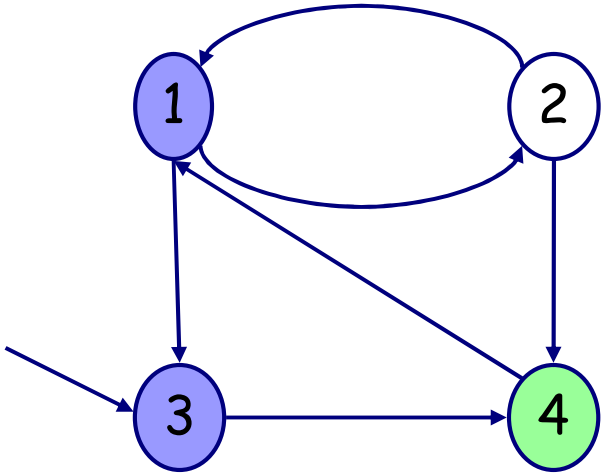


- $\varphi = G(b \Rightarrow F\, g)$
- k = 2 :

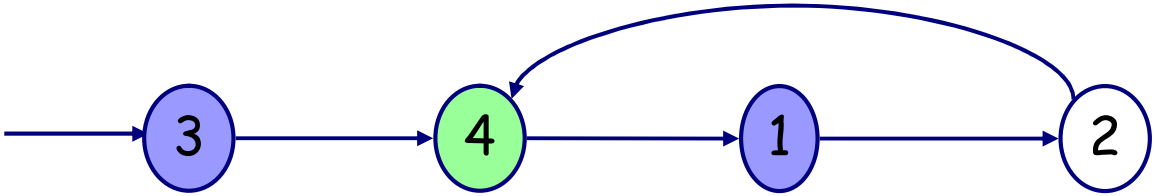


kein Gegenbeispiel

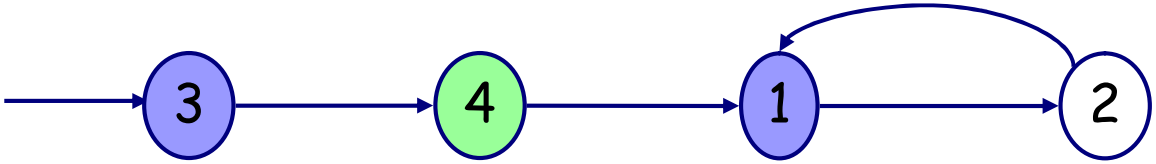
G blau⇒ F gruen ?



- $\varphi = G(b \Rightarrow F\, g)$
- k = 3 :



kein Gegenbeispiel

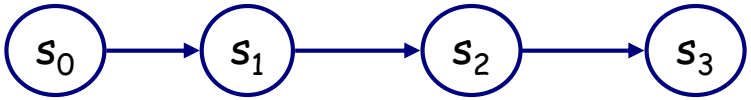


Gegenbeispiel !!!

Gegenbeispiel zu AG ¬p

- Seien s_0,s_1,s_2,s_3 Zustände. Wenn in der Kripke-Struktur M die Formel
  - $[EFp]_3 := I(s_0) \wedge R(s_0,s_1) \wedge R(s_1,s_2) \wedge R(s_2,s_3) \wedge (p(s_0) \vee p(s_1) \vee p(s_2) \vee p(s_3))$

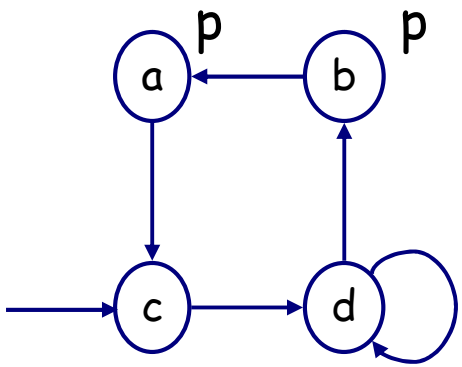
  erfüllbar ist, dann liefert das Pfadstück



  ein Gegenbeispiel zur Behauptung $M \models G\ \neg p$

- In Kripke-Struktur M gilt EFp genau dann wenn für mindestens ein k die folgende Formel erfüllbar ist:
  - $[EFp]_k(s_0,\ldots,s_k) = I(s_0) \wedge R(s_0,s_1) \wedge R(s_1,s_2) \wedge R(s_{k-1},s_k) \wedge (p(s_0) \vee \ldots \vee p(s_k))$

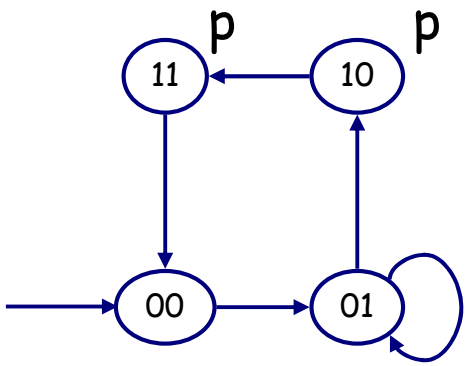
In konkreter Kripke-Struktur



- In der gezeigten Kripke-Struktur ist
  - $[EFp]_3 := I(s_0) \wedge R(s_0,s_1) \wedge R(s_1,s_2) \wedge R(s_2,s_3) \wedge (p(s_0) \vee p(s_1) \vee p(s_2) \vee p(s_3))$

  erfüllbar. Es gibt sogar zwei Erfüllungen:
  - s_0=c, s_1=d, s_2=d, s_3=b
  - s_0=c, s_1=d, s_2=b, s_3=a
- Jede davon ist ein Gegenbeispiel der Länge 4 zu
  - $M \models G\ \neg p$
- Es gibt sogar schon ein Gegenbeispiel der Länge 3
  - Keine Erfüllung von $[EFp]_2$

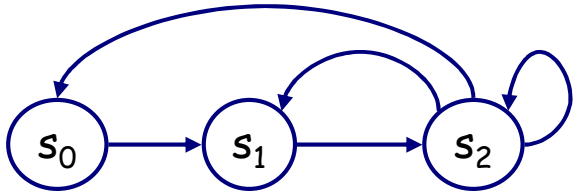
Symbolische Codierung



- Codierung mit booleschen Variablen v_0, v_1, v'_0, v'_1
- $s = (v_1, v_0)$, $s' = (v'_1, v'_0)$
  - $a = v_0 \wedge v_1$, $b = \neg v_0 \wedge v_1$, $c = \neg v_0 \wedge \neg v_1$, $d = v_0 \wedge \neg v_1$
  - $I(v_0, v_1) = \neg v_0 \wedge \neg v_1$
  - $p(v_0, v_1) = v_1$
  - $R(v_0, v_1, v'_0, v'_1) = (v'_0 \Leftrightarrow \neg v_0) \wedge (v'_1 \Leftrightarrow v_0 \oplus v_1) \vee (\neg v_1 \wedge v_0 \wedge \neg v'_1 \wedge v'_0)$
- Aus dieser Codierung von M und der Formel
  - $[EFp]_3 = I(s_0) \wedge R(s_0,s_1) \wedge R(s_1,s_2) \wedge R(s_1,s_2) \wedge (p(s_0) \vee p(s_1) \vee p(s_2) \vee p(s_3))$
- wird vermöge $s_0=(x_1,x_0)$, $s_1=(x'_1,x'_0)$, $s_2=(x''_1,x''_0)$, $s_3=(x'''_1,x'''_0)$ eine aussagenlogische Formel
  - $[M,EFp]_3 := \neg x_0 \wedge \neg x_1 \wedge (x'_0 \Leftrightarrow \neg x_0) \wedge (x'_1 \Leftrightarrow x_0 \oplus x_1) \vee (\neg x_1 \wedge x_0 \wedge \neg x'_1 \wedge x'_0)$
    $\wedge (x''_0 \Leftrightarrow \neg x'_0) \wedge (x''_1 \Leftrightarrow x'_0 \oplus x'_1) \vee (\neg x'_1 \wedge x'_0 \wedge \neg x''_1 \wedge x''_0)$
    $\wedge (x'''_0 \Leftrightarrow \neg x''_0) \wedge (x'''_1 \Leftrightarrow x''_0 \oplus x''_1) \vee (\neg x''_1 \wedge x''_0 \wedge \neg x'''_1 \wedge x'''_0)$
    $\wedge (x_1 \vee x'_1 \vee x''_1 \vee x'''_1)$
- [M,EFp] ist erfüllbar gdw. $[EFp]_3$ ist in der Kripke-Struktur erfüllbar gdw. $M \not\models G\neg p$

Gegenbeispiel zu AF ¬p

- Gesucht Zustände s_0,s_1,s_2,s_3, die beweisen, dass $M \models E\,Gp$
  - Aussage über alle Zustände eines unendliche langen Pfades
  - Zum Glück liefert jeder Loop einen unendlich langen Pfad



$[EGp]_3 := I(s_0) \wedge R(s_0,s_1) \wedge R(s_1,s_2) \wedge R(s_2,s_3) \wedge ((s_3=s_0) \vee (s_3 = s_1) \vee (s_3 = s_2)) \wedge p(s_0) \wedge p(s_1) \wedge p(s_2))$

Wenn diese Formel erfüllbar ist, dann liefern (s_0,s_1,s_2) einen Zeugen für $M \not\models AF\neg p$

- In einer endlichen Kripke-Struktur M gilt:
  - Jedes Gegenbeispiel zu AF¬p ist von der Bauart $[EGp]_k$ (mit k statt 3)

Bounded Model Checking

- Idee :
  - Suche nach Gegenbeispiel zu einer LTL-Formel φ, ohne komplette Kripke-Struktur zu erzeugen
    - Gegenbeispiel ist ein Pfad σ,
      - der in einem Anfangszustand beginnt
      - der nicht φ erfüllt
  - Erkunde Kripke Struktur M bis zur Tiefe k
    - Betrachte nur Pfade der Länge $\leq$ k
    - Erhöhe k nach Bedarf
- Verbindung mit symbolischen Model Checking
  - Existenz von Gegenbeispiel der Länge $\leq$ k ist als aussagenlogische Formel $\Psi_k = [M,\varphi]_k$ ausdrückbar.
    - Ψ_k erfüllbar $\Leftrightarrow$ Es gibt ein Gegenbeispiel zu $M \models \varphi$ der Länge $\leq$ k
  - Gegeben M, φ
    - Wähle ein k
    - Konstruiere $\Psi_k=[M_k,\varphi]_k$
    - Prüfe ob Ψ_k erfüllbar ist
- SAT-Solver
  - SAT-Problem
    - Gegeben aussagenlogische Formel Φ
    - Ist Φ erfüllbar

Kripke-Strukturen und LTL-Semantik

- M=(S,I,R,L) Kripke-Struktur (Modell) über AP
  - $I \subseteq S$, $R \subseteq S \times S$, $L: S \to \mathbb{P}(AP)$
    - statt sRt schreiben wir auch $s \to t$
  - In diesem Kapitel soll S endlich und R total sein
- Folgen
  - $S^\omega := \{ \pi : \mathbb{N} \to S \}$
  - $\pi(i)$: i-tes Folgenelement
  - π^k : Restfolge
    - $\pi^k = \lambda\, n{:}\mathbb{N}.\ \pi(n+k)$
- Pfade
  - $Paths(R) := \{ \pi \in S^\omega \mid \forall\, n \in \mathbb{N}.\ \pi(n)\ R\ \pi(n+1) \}$
  - $Paths(M) := \{ \pi \in Paths(R) \mid \pi(0) \in I \}$
- Pfadsemantik:
  - $\pi \models \varphi :\Leftrightarrow$ ... induktiv über den Aufbau von φ
- Zustandssemantik:
  - Gültigkeit für alle Pfade, die von einem Zustand s ausgehen
  - $s \models A\ \phi \ :\Leftrightarrow\ \forall\, \pi \in Paths(R).\ \pi(0)=s \Rightarrow \pi \models \varphi$
  - Entspricht der CTL*-Semantik
- Kripke-Semantik
  - φ gilt auf allen Pfaden, die in einem Anfangszustand entspringen
  - $M \models A\ \varphi :\Leftrightarrow \forall\, \pi \in Paths(M).\ \pi \models \varphi$

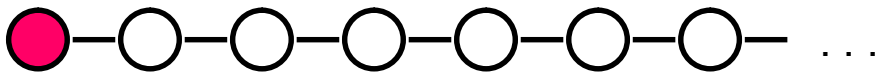
Gegenbeispiele

- Falls LTL-Formel φ nicht in M gilt, muss es ein Gegenbeispiel geben
  - $M \not\models \varphi \quad \Leftrightarrow \exists s \in I.\ s \not\models A\,\varphi$
    $\Leftrightarrow \exists\, s \in I.\ s \models E\neg\varphi$
- Gegenbeispiel zu $A\varphi$
  - ein Pfad π, mit $\pi(0) \in I$ und $\pi \models \neg\varphi$
  - oft schon nach k Schritten klar, dass es sich um ein Gegenbeispiel handelt
    - dass $\pi \not\models \varphi$
  - hängt aber von φ ab
- die ersten k Elemente eines Pfades π können zeigen, dass $\pi \models \varphi$
  - egal wie der Pfad zu einer unendlichen Folge fortgesetzt wird)*
  - $\pi \models_k \varphi :\Leftrightarrow \forall\, \sigma \in S^\omega.\ (\pi(0),\ldots,\pi(k\text{-}1)) = (\sigma(0),\ldots,\sigma(k\text{-}1)) \Rightarrow \sigma \models \varphi$
  - $\pi \models_0 \phi \Leftrightarrow \varphi \cong \text{true}$

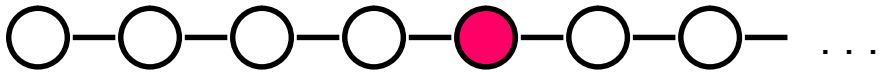
)* genau genommen setzen wir voraus, dass für jedes $p \in AP$ ein $s \in S$ existiert mit $p \notin L(s)$

Beispiele für $\pi \models_k \varphi$

$\pi \models_7$ rot

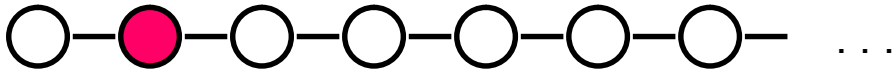


$\pi \models_7$ F rot, aber $\pi \not\models_4$ rot

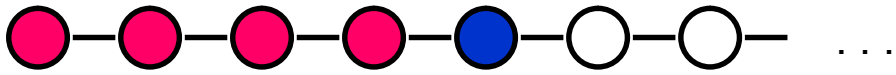


für jedes k: $\pi \not\models_k$ rot, wohl aber $\pi \models_k G(\neg rot \vee rot)$

Für jedes k>1 gilt: $\pi \models_k X$ rot



Für jedes k>4 gilt $\pi \models_k$ rot **U** blau



Zustand, in dem p gilt

Zustand, in dem q gilt

Rekursive Definition von $\models_k$

- benötigt Hilfsparameter $i \geq 0$:
  - $\pi \models_{i,k} \varphi :\Leftrightarrow \pi^i \models_{k-i}$

  Damit definiere:
  - $\pi \models_k \varphi :\Leftrightarrow \pi \models_{0,k} \varphi$

- Definiere $\models_{i,k}$ induktiv über den Aufbau von φ
- Wir setzen voraus, dass ϕ in positiver Normalform ist !!
  - Rekursionsanfang: Falls $i>k$, dann
    - $\pi \models_{i,k} \phi \;:\Leftrightarrow\; (\phi \cong \text{true})$ - das heißt normalerweise false.

Ansonsten:

wenn φ die Form … hat,	dann gilt $\pi \models_{i,k} \varphi$ falls …
$p \in Ap$:	$p \in L(\pi(i))$
$\neg p$	$p \notin L(\pi(i))$
$\phi_1 \wedge \phi_2$:	$\pi \models_{i,k} \phi_1$ und $\pi \models_{i,k} \phi_2$
$X\phi$:	$\pi \models_{i+1,k} \phi$
$\phi \, U \, \psi$:	$\pi \models_{i,k} \psi$ oder $\pi \models_{i,k} \phi$ und $\pi \models_{i+1,k} (\phi U \psi)$

für die abgeleiteten Operatoren

$\phi \, R \psi$:	$\pi \models_{i,k} \psi$ und $\pi \models_{i,k} \phi$ oder $\pi \models_{i+1,k} \phi \, R \, \psi$
$G\psi$:	$\pi \models_{i,k} \psi$ und $\pi \models_{i+1,k} G\psi$ $\Leftrightarrow$ ψ ist Tautologie !!!
$F\psi$:	$\pi \models_{i,k} \psi$ oder $\pi \models_{i+1,k} F\phi$

$\pi \models_k \varphi \Rightarrow \pi \models \varphi$

- Gleichzeitige Induktion über den Aufbau von φ und über k

- $\pi \models_k p \quad \Rightarrow p \in L(\pi(0)) \quad \Rightarrow \pi \models p$
- $\pi \models_k \neg p \quad \Rightarrow p \notin L(\pi(0)) \quad \Rightarrow \pi \models \neg p$

- $\pi \models_k \phi_1 \wedge \phi_2 \quad \Rightarrow \pi \models_k \phi_1$ und $\pi \models_k \phi_2$
  $\Rightarrow \pi \models \phi_1$ und $\pi \models \phi_2$ (hyp)
  $\Rightarrow \pi \models \phi_1 \wedge \phi_2$

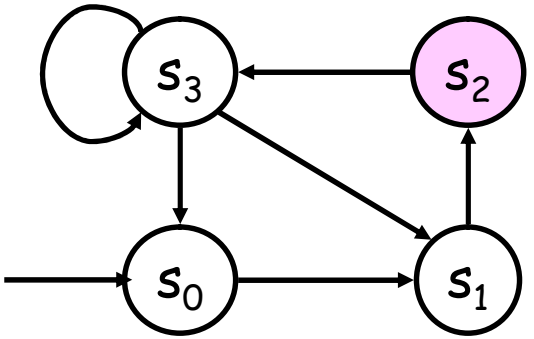
- $\pi \models_k X\varphi$
  $\Rightarrow \pi \models_{1,k} \varphi$
  $\Rightarrow \pi^1 \models_{(k-1)} \varphi$ (leichtes Lemma)
  $\Rightarrow \pi^1 \models \varphi$ (hyp k)
  $\Rightarrow \pi \models X\varphi$

- $\pi \models_k \phi U \psi$
  - 1. Fall: $\pi \models_k \psi \quad \Rightarrow \pi \models \psi$ (hyp)
    $\Rightarrow \pi \models \phi \; U \; \psi$
  - 2. Fall: $\pi \models_k \phi \wedge X(\phi \; U \; \psi)$
    $\Rightarrow \pi \models \phi$ und $\pi^1 \models_{k-1} \phi \; U \; \psi$ (hyp ϕ)
    $\Rightarrow \pi \models \phi$ und $\pi^1 \models \phi \; U \; \psi$ (hyp k,$\phi \; U \; \psi$)
    $\Rightarrow \pi \models \phi$ und $\pi \models X(\phi \; U \; \psi)$
    $\Rightarrow \pi \models \phi \, U \psi$

Frühe Gegenbeispiele

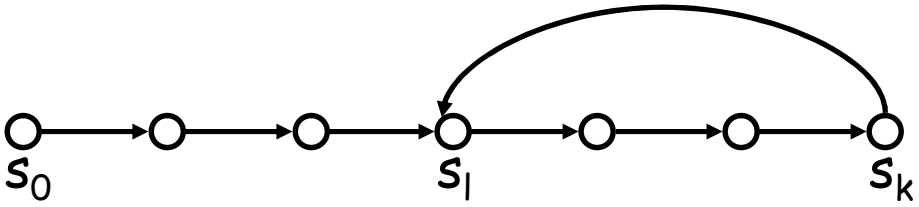
- Gegeben
  - Kripke-Struktur M=(S,I,R,L)
  - temporale Spezifikation $\varphi \in$ LTL
- Gesucht:
  - Gegenbeispiel: $s \in I$ mit $s \models E\neg\varphi$
- Hoffnung
  - kurzes Anfangsstück eines Pfades π mit $\pi(0) \in I$ beweist schon, dass ein solches Gegenbeispiel existiert
  - In der Tat: $\pi \models_k \varphi \Rightarrow \pi \models \phi$
- Problematisch
  - falls G in $\neg\varphi$ vorkommt
  - $\pi \models_k G\phi$ ist nur wahr, falls ϕ trivial ist.

Beispiel

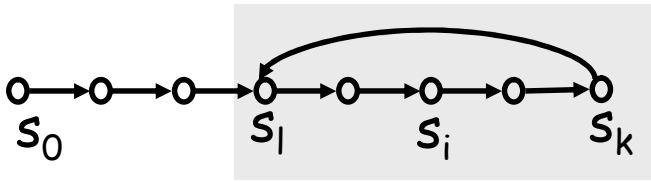


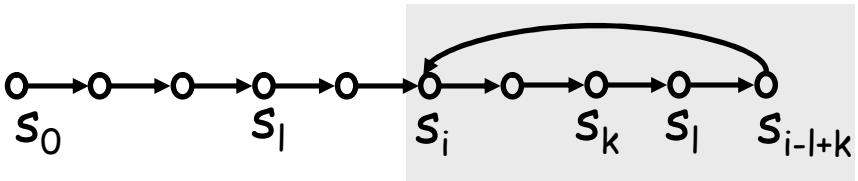
- Behauptung
  - $M \models E\ F\ rot$
    - $(s_0,s_1,s_2) \models_k rot$, also $s_0 \models E\ F\ rot$, also $M \models EF\ rot$ ☺
  - $M \models E\ GF\ rot$
    - $(s_0, s_1, s_2) \not\models_3 GF\ rot$
    - $(s_0, s_1, s_2, s_3) \not\models_4 GF\ rot$
    - $(s_0, s_1, s_2, s_3, s_1) \not\models_5 GF\ rot$
    - $(s_0, s_1, s_2, s_3, s_1, s_2) \not\models_6 GF\ rot$
    - ... so geht's nicht ☹
  - ... aber man müsste es doch folgern können aus
    - $s_4=s_1$ und
    - $(s_0, s_1, s_2, s_3, s_1) \models_{1,4} rot$
  - Schließlich muss in einem endlichen System ja jede Folge eine Schleife haben.

l-k-Loops

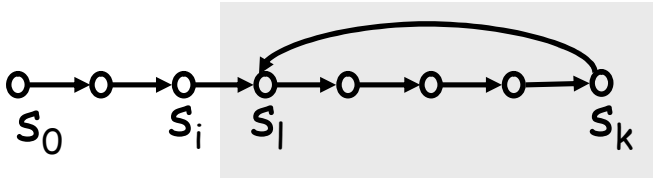


- l-k-Loop
  - Pfad, der in einer Schleife endet
  - $(s_0,\ldots,s_l,\ldots,s_k)$ mit s_l R s_k
  - Unendlicher Pfad ist: $(s_0,\ldots, (s_l,\ldots,s_k)^\omega)$
- Für solche Pfade definieren wir Relation $_l\models_k$ mit der intendierten Semantik
  $(s_0,\ldots,s_l,\ldots,s_k)\ {}_l\models_k \varphi \Leftrightarrow (s_0,\ldots, (s_l,\ldots,s_k)^\omega) \models \varphi$
- Hilfsfunktion succ:
  - succ(i)= if (i=k) then l else i+1
- Hilfsprädikat $_l\models_{i,k}$ mit
  - $_l\models_k \Leftrightarrow {}_l\models_{0,k}$
- Abrollen von Loops
  - zwei Darstellungen des gleichen Loops
  - i wurde vor die Schleife bewegt





Gültigkeit für l-k-Loops



- Definiere ${}_l\models_{i,k}$ induktiv über den Aufbau von φ
  - $i, l \leq k$

wenn φ die Form … hat, dann gilt mit $\pi := (s_0,\ldots,s_l,\ldots,s_k)$ dass $\pi \,{}_l\models_{i,k} \varphi$ falls …

- $p \in AP$: $p \in L(\pi(i))$
- $\neg p$ $p \notin L(\pi(i))$
- $\phi_1 \wedge \phi_2$: $\pi \,{}_l\models_{i,k} \phi_1$ und $\pi \models_{i,k} \phi_2$
- $X\,\phi$: $\pi \,{}_l\models_{succ(i),\,k} \phi$

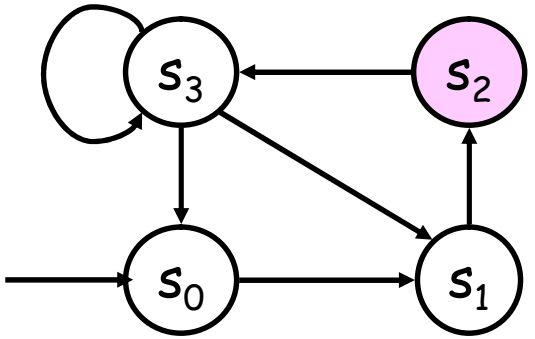
für die folgenden Fälle nehmen wir $i \leq l$ an, ansonsten: Schleife abrollen

- $\phi\, U\, \psi$: $\pi \,{}_l\models_{i,k} \psi \vee (\pi \,{}_l\models_{i,k} \phi \wedge \pi \,{}_l\models_{i+1,k} \psi) \vee \ldots \vee (\pi \,{}_l\models_{i,k} \phi \wedge \ldots \wedge \pi \,{}_l\models_{k-1,k} \phi \wedge \pi \,{}_l\models_{k,k} \psi)$

für die abgeleiteten Operatoren

- $\phi\, R\, \psi$: $(\pi \,{}_l\models_{i,k} \psi \wedge \phi) \vee (\pi \,{}_l\models_{i,k} \psi \wedge \pi \,{}_l\models_{i+1,k} \psi \wedge \phi) \vee \ldots \vee (\pi \,{}_l\models_{i,k} \psi \wedge \ldots \wedge \pi \,{}_l\models_{k-1,k} \psi \wedge \pi \,{}_l\models_{k,k} \psi \wedge \phi)$
- $G\psi$: $(\pi \,{}_l\models_{i,k} \psi \wedge \ldots \wedge \pi \,{}_l\models_{k-1,k} \psi \wedge \pi \,{}_l\models_{k,k} \psi)$
- $F\psi$: $(\pi \,{}_l\models_{i,k} \psi \vee \ldots \vee \pi \,{}_l\models_{k-1,k} \psi \vee \pi \,{}_l\models_{k,k} \psi)$

Beispiel



- Behauptung:

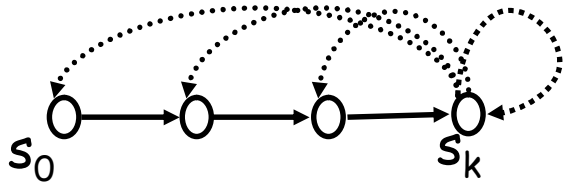
  $M \models E\,GF\,rot$

  - $(s_0, s_1, s_2, s_3)\ {}_1\models_3 GF\,rot$
    $\Leftrightarrow$ $(s_0, s_1, s_2, s_3)\ {}_1\models_{0,3} F\,rot$
    $\wedge\ (s_0, s_1, s_2, s_3)\ {}_1\models_{1,3} F\,rot$
    $\wedge\ (s_0, s_1, s_2, s_3)\ {}_1\models_{2,3} F\,rot$
    $\wedge\ (s_0, s_1, s_2, s_3)\ {}_1\models_{3,3} F\,rot$

  Für den ersten Konjunkt
  - $(s_0, s_1, s_2, s_3)\ {}_1\models_{0,3} F\,rot$
    $\Leftrightarrow$ $(s_0, s_1, s_2, s_3)\ {}_1\models_{0,3} rot$
    $\vee\ (s_0, s_1, s_2, s_3)\ {}_1\models_{1,3} rot$
    $\vee\ (s_0, s_1, s_2, s_3)\ {}_1\models_{2,3} rot$
    $\vee\ (s_0, s_1, s_2, s_3)\ {}_1\models_{3,3} rot$
    $\Leftrightarrow$ true

  - Analog:
  - $(s_0, s_1, s_2, s_3)\ {}_1\models_{1,3} F\,rot$
  - $(s_0, s_1, s_2, s_3)\ {}_1\models_{2,3} F\,rot$
  - $(s_0, s_1, s_2, s_3)\ {}_1\models_{3,3} F\,rot$

Pfad-Formeln



- Sei M=(S,I,R,L,AP) Kripke Struktur
- Wir übersetzen die Existenz eines k-Pfades in die Sprache des Kripke Modells. Zur Erinnerung:
  - I(x) : x ist Anfangszustand
  - p(x) : $p \in L(x)$
  - R(x,y) : $(x,y) \in R$
- Eine Folge von Zuständen bildet einen Pfad, falls
  - $\pi(s_0,\ldots,s_{k-1}) := I(s_0) \wedge R(s_0,s_1) \wedge \ldots \wedge R(s_{k-1},s_k)$
- Der k-Pfad ist ein l-k-Loop, falls zusätzlich gilt
  - $R(s_k,.s_l)$
- Der k-Pfad hat einen loop, wenn gilt
  - $L_k(s_0,\ldots,s_k) := R(s_k,s_0) \vee \ldots \vee R(s_k,s_k)$

Formeln für k-Pfade

- Wir unterscheiden, ob ein Pfad eine Schleife hat, oder nicht:
  - $\pi_k(s_0,\ldots,s_k) = \pi_k(s_0,\ldots,s_k) \wedge (\bigwedge_{i=0,k} \neg R(s_k,s_i) \quad \vee \quad \bigvee_{l=0,k} R(s_k,s_0) \vee \ldots \vee R(s_k,s_k))$

- Für jeden der Fälle drücken wir aus, dass der Pfad $(s_0,\ldots,s_k)$ die Formel φ wahr machen soll
  - $(s_0,\ldots,s_k) \models_k \varphi$, bzw.
  - $(s_0,\ldots,s_k) \; {}_l\models_k \varphi$

- Dazu übersetzen wir die Formel φ in eine Formel
  - $[\varphi]_k(s_0,\ldots,s_k)$ für die Pfade ohne loop
  - ${}_l[\varphi]_k(s_0,\ldots,s_k)$ für die Pfade mit l-k-Loop

- Die fertige Formel ist dann

- $[M,\varphi]_k(s_0,\ldots,s_k) := \underbrace{\pi_k(s_0,\ldots,s_k)}_{(s_0,\ldots,s_k) \text{ ist ein k-Pfad}} \wedge ((\underbrace{\bigwedge_{i=0,k} \neg R(s_k,s_i) \wedge [\varphi]_k(s_0,\ldots,s_k)}_{\text{wenn } \pi \text{ keine Schleife hat, soll } [\varphi]_\kappa \text{ gelten}}) \vee (\underbrace{\bigvee_{l = 0\ldots k} R(s_k,s_l) \wedge {}_l[\varphi]_k(s_0,\ldots,s_k)}_{\text{falls } \pi \text{ eine l-k-Schleife hat, soll } {}_l[\varphi]_k \text{ gelten}}))$

Übersetzung von φ für schleifenfreie Folgen

$[M,\varphi]_k(s_0,\ldots,s_k) := \pi_k(s_0,\ldots,s_k) \wedge ((\bigwedge_{i=0,k} \neg R(s_k,s_i) \wedge [\varphi]_k(s_0,\ldots,s_k)) \vee (\bigvee_{l=0\ldots k} R(s_k,s_l) \wedge {}_l[\varphi]_k(s_0,\ldots,s_k)))$

- Gleichzeitige Rekursion über den Aufbau von φ und über k

- $[p]_k(s_0,\ldots,s_k) = p(s_0)$
- $[\neg p]_k(s_0,\ldots,s_k) = \neg p(s_0)$

- $[\phi \wedge \psi]_k(s_0,\ldots,s_k) = [\phi]_k(s_0,\ldots,s_k) \wedge [\psi]_k(s_0,\ldots,s_k)$

- $[X\varphi]_0(s_0) = \text{false}$
- $[X\varphi]_k(s_0,\ldots,s_k) = [\varphi]_{k-1}(s_1,\ldots,s_k)$, falls k>0

- $[\phi U \psi]_0(s_0) = [\psi]_0(s_0)$
- $[\phi U \psi]_k(s_0,\ldots,s_k) = [\psi]_k(s_0,\ldots,s_k) \vee [\phi]_k(s_0,\ldots,s_k) \wedge [(\varphi U \psi)]_{k-1}(s_1,\ldots,s_k)$, falls k>0

Es folgt

- $[\phi R \psi]_0(s_0) = [\phi \wedge \psi]_0(s_0)$
- $[\phi R \psi]_k(s_0,\ldots,s_k) = [\phi \wedge \psi]_k(s_0,\ldots,s_k) \vee [\psi]_k(s_0,\ldots,s_k) \wedge [(\varphi R \psi)]_{k-1}(s_1,\ldots,s_k)$, falls k>0

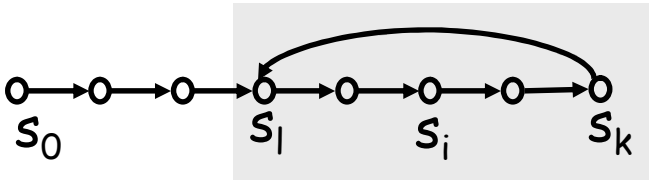
- $[F\phi]_0(s_0) = [\phi]_0(s_0)$
- $[F\phi]_k(s_0,\ldots,s_k) = [\phi]_k(s_0,\ldots,s_k) \vee [F\phi)]_{k-1}(s_1,\ldots,s_k)$, falls k>1

- $[G\phi]_0(s_0) = [\phi]_0(s_0)$
- $[G\phi]_k(s_0,\ldots,s_k) = [\phi]_k(s_0,\ldots,s_k) \wedge [G\phi)]_{k-1}(s_1,\ldots,s_k)$, falls k>1

${}_l[\varphi]_k$ für l-k-Loops

$[M,\varphi]_k(s_0,\ldots,s_k) := \pi_k(s_0,\ldots,s_k) \wedge ((\bigwedge_{i=0,k} \neg R(s_k,s_i) \wedge [\varphi]_k(s_0,\ldots,s_k)) \vee (\bigvee_{l=0\ldots k} R(s_k,s_l) \wedge {}_l[\varphi]_k(s_0,\ldots,s_k)))$

- Gleichzeitige Rekursion über den Aufbau von φ und über k



- ${}_l[p]_k(s_0,\ldots,s_k) = p(s_0)$
- ${}_l[\neg p]_k(s_0,\ldots,s_k) = \neg p(s_0)$

- ${}_l[\phi\wedge\psi]_k(s_0,\ldots,s_k) = {}_l[\phi]_k(s_0,\ldots,s_k) \wedge {}_l[\psi]_k(s_0,\ldots,s_k)$

- ${}_l[\,X\varphi]_k(s_0,\ldots,s_k) = {}_l[\varphi]_{k-1}(s_1,\ldots,s_k)$, falls l>0
- ${}_l[\,X\varphi]_k(s_0,\ldots,s_k) = {}_l[\varphi]_k(s_1,\ldots,s_k,s_0)$, falls l=0

- ${}_l[\,\phi U\psi]_k(s_0,\ldots,s_k) = {}_l[\psi]_k(s_0,\ldots,s_k) \vee {}_l[\phi]_k(s_0,\ldots,s_k) \wedge {}_l[(\phi\ U\ \psi)]_{k-1}(s_1,\ldots,s_k)$, falls l>0
- ${}_0[\,\phi U\psi]_k(s_0,\ldots,s_k) = {}_0[\psi]_k(s_0,\ldots,s_k) \vee {}_0[\phi]_k(s_0,\ldots,s_k) \wedge {}_1[(\phi\ U\ \psi)]_k(s_1,\ldots,s_k,s_0)$, falls l=0

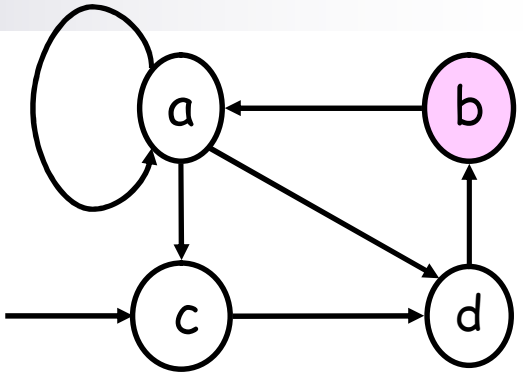
Es folgt insbesondere

- ${}_l[F\phi]_k(s_0,\ldots,s_k) = [\phi]_k(s_0,\ldots,s_k) \vee {}_l[F\phi]_{k-1}(s_1,\ldots,s_k)$, falls l>0
- ${}_l[F\phi]_k(s_0,\ldots,s_k) = [\phi]_k(s_0,\ldots,s_k) \vee [F\phi]_{k-1}(s_1,\ldots,s_k)$, falls l=0

- ${}_l[G\phi]_k(s_0,\ldots,s_k) = [\phi]_k(s_0,\ldots,s_k) \wedge [G\phi]_{k-1}(s_1,\ldots,s_k)$, falls l>0
- ${}_0[G\phi]_k(s_0,\ldots,s_k) = [\phi]_k(s_0,\ldots,s_k) \wedge [G\phi]_{k-1}(s_1,\ldots,s_k)$, falls l>0

genau genommen muss man diese Rekursion nach k Schritten abbrechen

Beispiel



- Behauptung
  - $M \models E\,GF\,rot$
    - $[M,FG\,\neg rot]_3(s_0, s_1, s_2, s_3) = I(s_0) \wedge R(s_0,s_1,) \wedge R(s_1,s_2) \wedge R(s_2,s_3) \wedge$
      $((\neg R(s_3,s_0) \wedge \neg R(s_3,s_1) \wedge \neg R(s_3,s_2) \wedge \neg R(s_3,s_3)) \wedge$
      $[G\neg rot]_3(s_0, s_1, s_2, s_3) \vee [G\neg rot]_2(s_0, s_1, s_2) \vee [G\neg rot]_1(s_0, s_1) \vee [G\neg rot]_0(s_0))$
      $\vee$
      $(R(s_3,s_0) \wedge_0 [G\neg rot]_3(s_0, s_1, s_2, s_3) \vee (R(s_3,s_1) \wedge_1 [G\neg rot]_3(s_1, s_2, s_3)$
      $\vee$
      $(R(s_3,s_2) \wedge_2 [G\neg rot]_3(s_2, s_3) \vee (R(s_3,s_3) \wedge_3 [G\neg rot]_3(s_3)$
    - $=$
    - $I(s_0) \wedge R(s_0,s_1,) \wedge R(s_1,s_2) \wedge R(s_2,s_3) \wedge$
      (false)
      $\vee$
      $(R(s_3,s_0) \wedge_0 [G\neg rot]_3(s_0, s_1, s_2, s_3) \vee (R(s_3,s_1) \wedge_1 [G\neg rot]_3(s_1, s_2, s_3)$
      $\vee$
      $(R(s_3,s_2) \wedge_2 [G\neg rot]_3(s_2, s_3) \vee (R(s_3,s_3) \wedge_3 [G\neg rot]_3(s_3)$
    - $=$
    - $s_0=c \wedge s_1=d \wedge s_2=b \wedge s_3=a$
      $\wedge {}_0[G\neg rot]_3(s_0, s_1, s_2, s_3) \vee {}_1[G\neg rot]_3(s_1, s_2, s_3) \vee {}_3[G\neg rot]_3(s_3)$
    - $= {}_0[G\neg rot]_3(c, d, b, a) \vee {}_1[G\neg rot]_3(d, b, a) \vee {}_3[G\neg rot]_3(a)$
    - $= {}_0[G\neg rot]_3(c, d, b, a) \vee {}_1[G\neg rot]_3(d, b, a) \vee {}_3[G\neg rot]_3(a)$

Korrektheit und Vollständigkeit

- Korrektheit:
  - $[M,\varphi]_k \Rightarrow M \vDash \varphi$
- Vollständigkeit, falls M endlich :
  - $M \vDash \varphi \Rightarrow \exists k.[M,\varphi]_k$
- Strategie also:
  - Gegeben LTLSPEC φ
  - Prüfe für wachsende k:
    - $[M,\neg\varphi]_1$, $[M,\neg\varphi]_2$, $[M,\neg\varphi]_3$,
  - Falls eine der Formeln erfüllbar ist, ist ein minimales Gegenbeispiel zu φ gefunden
- Erinnerung: $\Psi(s_0,\ldots,s_k)$ ist erfüllbar, wenn man für $s_0,\ldots,s_k$ geeignete Zustände einsetzen kann, die die Formel wahr machen

Rückführung auf SAT-Problem

Erinnerung

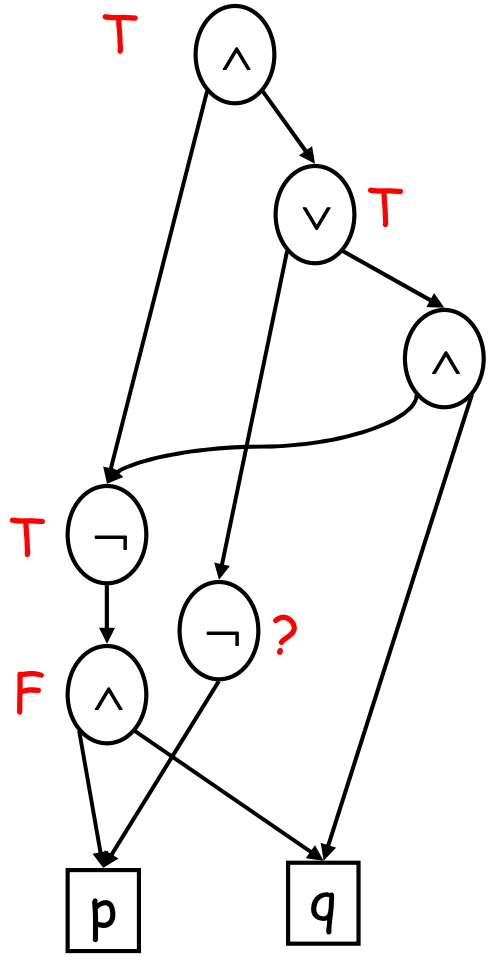
- $[M,\varphi]_k(s_0,\ldots,s_k) = \pi_k(s_0,\ldots,s_k) \wedge ((\bigwedge_{i=0,k} \neg R(s_k,s_i) \wedge [\varphi]_k(s_0,\ldots,s_k)) \vee (\bigvee_{l=0\ldots k} R(s_k,s_l) \wedge {}_l[\varphi]_k(s_0,\ldots,s_k)))$
  wobei
  $\pi(s_0,\ldots,s_{k-1}) = I(s_0) \wedge R(s_0,s_1) \wedge \ldots \wedge R(s_{k-1},s_k)$
- Beim symbolischen Model Checking ist die Kripke-Struktur symbolisch repräsentiert
  - $V = \{ x_1,\ldots,x_n,x'_1,\ldots x'_n \}$ aussagenlogische Variablen
  - $S, I \subseteq 2^n$
    - z.B. $s = (1,0,1,1)$ durch $x_1 \wedge \neg x_2 \wedge x_3 \wedge x_4$, $t=(0,1,1,0)$ durch $\neg x_1 \wedge x_2 \wedge x_3 \wedge \neg x_4$
  - $R \subseteq 2^n \times 2^n$
    - z.B. $(s,t) \in R$ durch $x_1 \wedge \neg x_2 \wedge x_3 \wedge x_4 \neg x'_1 \wedge x'_2 \wedge x'_3 \wedge \neg x'_4$
  - jedes $p \in AP$ als $[p] \subseteq 2^n$: $L(s) = (s \in p)$
- $[M,\varphi]_k(s_0,\ldots,s_k)$ ist eine Formel, in der
  - das zweistellige Prädikat R(-,-)
  - die einstelligen Prädikate I(-), sowie p(-) für jedes $p \in AP$
  - die logischen Konnektoren $\wedge\neg,\vee$ vorkommen, aber keine Quantoren
- Ersetzen wir in $[M,\varphi]_k(s_0,\ldots,s_k)$ alle Vorkommen von $s_0,\ldots,s_n$, $R(s_k,s_l)$, $[\varphi]_k$, ${}_l[\varphi]_k$ etc. entsprechend durch Literale $\ldots \wedge x_i \wedge \ldots \wedge x'_i \wedge \ldots$ so entsteht eine aussagenlogische Formel
  $[M,\varphi]_k(x_1,\ldots,x_n,x'_1,\ldots,x'_n)$
- $[M,\varphi]_k(x_1,\ldots,x_n,x'_1,\ldots,x'_n)$ ist erfüllbar $\Leftrightarrow$ Es gibt eine Folge der Länge k, die belegt dass φ gilt.

SAT-Probleme

- Sat-Problem ist
  - Gegeben aussabgenlogische Formel Ψ
  - Ist Ψ erfüllbar ?
- Sat Problem ist NP-vollständig
  - das gilt für den allgemeinen Fall
  - im Einzelfall kann es auch mal schneller gehen
  - in der Praxis geht e shäufig schneller
- in den letzten Jahren gab es enorme Fortschritte
  - SAT-Probleme mit mehreren tausend Variablen oft kein Problem mehr
  - immer bessere Algorithmen und Implementierungen
    - Stålmark's Algorithmus (patentiert in Schweden und USA)
    - Constraint Propagation
    - GRASP, CHAFF, MiniSAT, SATplan, …

SAT-Algorithmen

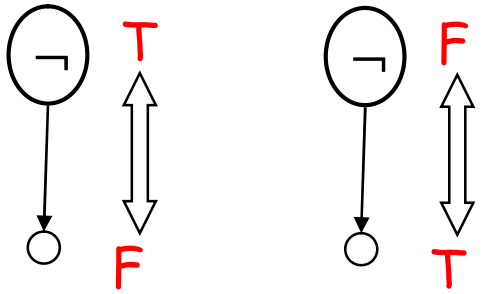
- Gegeben aussagenlogische Formel $\Psi(x_1,\ldots,x_n)$
  - Umwandlung der Formel in DNF zu aufwendig
    - exponentiell
- Repräsentiere Ψ als DAG (directed acyclic graph)
  - mehrfach vorkommende Teilformeln werden dadurch nur einmal repräsentiert
- Schreibe Bedingung T (für true) an die Wurzel
  - propagiere es zu den Kindern
  - dort entstehen Einschränkungen (constraints)
  - Constraints können auch von den Kindern auf die Eltern rückwirken
  - Wenn alle Constraints erfüllbar: Erfüllung gefunden
  - Wenn Constraints sich widersprechen: nicht erfüllbar

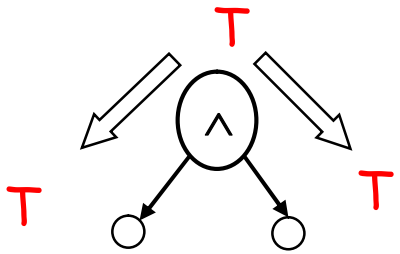


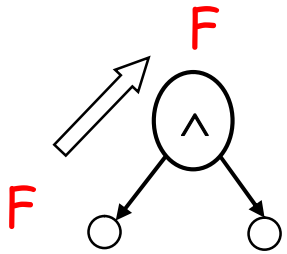
ein DAG, der die Formel $\neg(p\wedge q)\wedge(\neg p\vee\neg(p\wedge q)\wedge q)$ repräsentiert

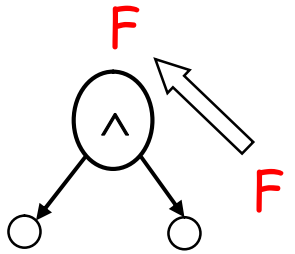
Constraint propagation für ¬ und ∧

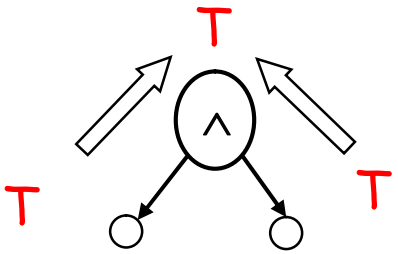
- Constraint propagation möglich, je nach Konnektor:
  - vom Vater zu den Kindern
  - von einem oder mehreren Kindern zum Vater
  - von Vater und Kind zum Bruder
- Für ∨, ⇒ und ⇔ analoge constraint propagation

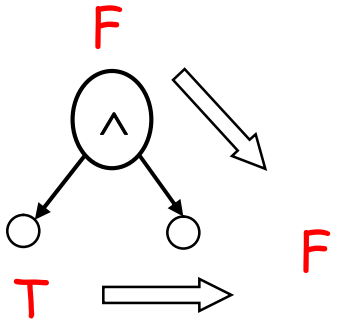


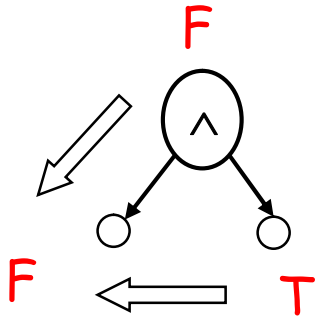






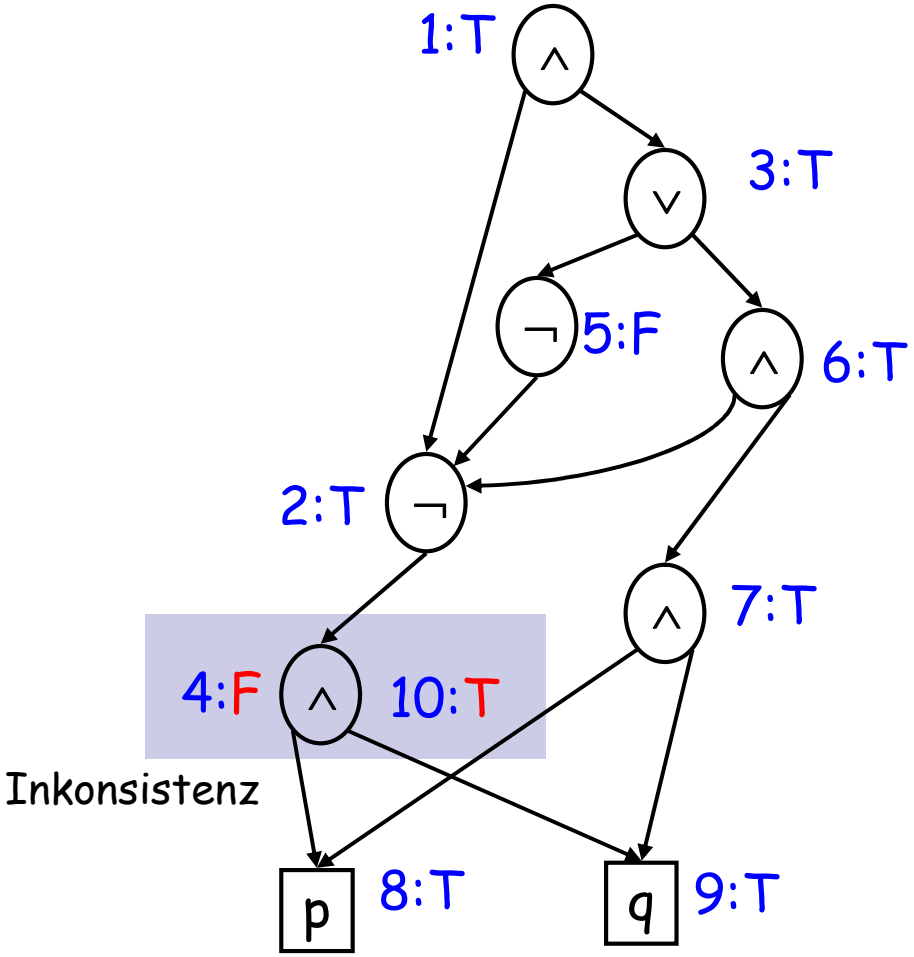






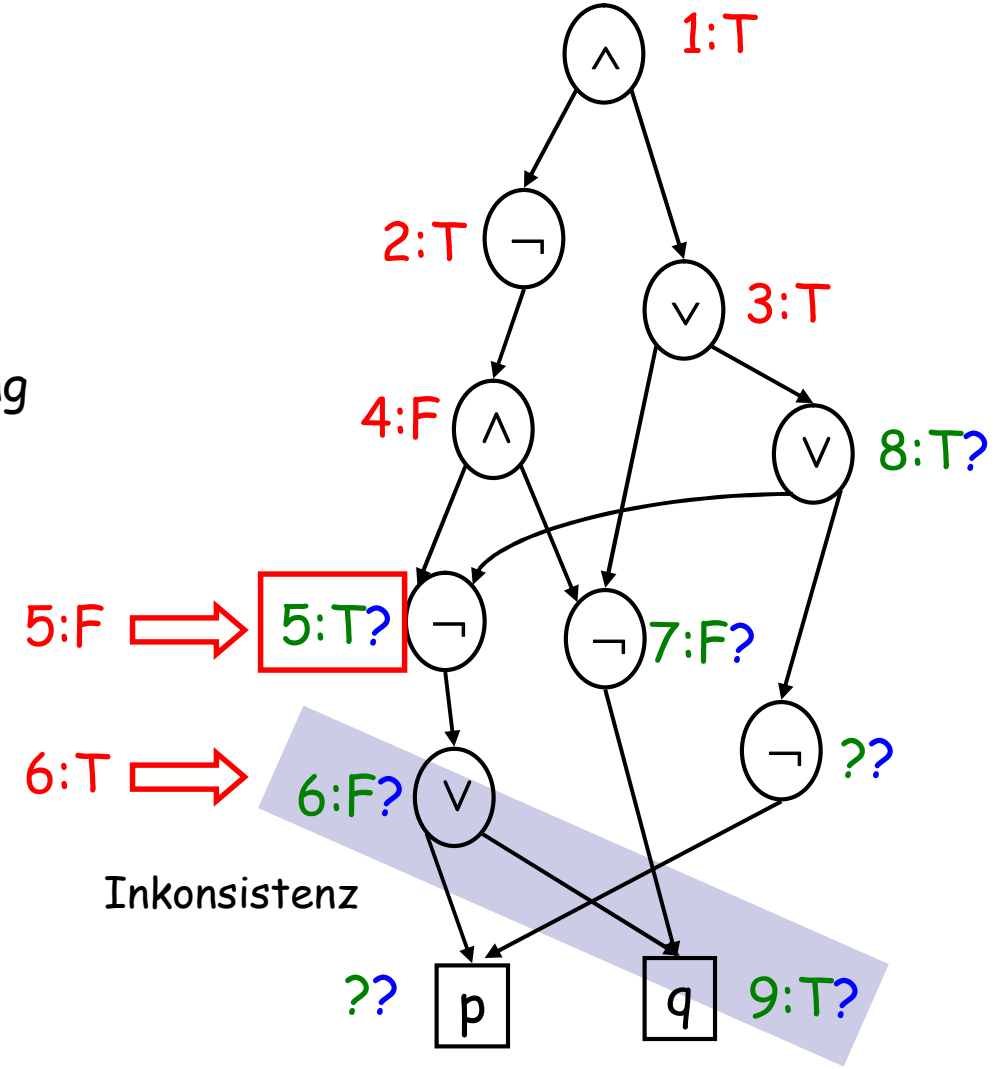
Möglichkeiten

- Propagierung der Regeln führt zu einer Inkonsistenz
  - Formel nicht erfüllbar
- Propagierung der Regeln weist jeder Position einen Wahrheitswert zu
  - Prüfen ob die Belegung konsistent ist
    - lineare Komplexität
  - Wenn ja: erfüllbar
  - Wenn nein: Inkonsistenz entdeckt
- Propagierung der Regeln weist nicht jeder Position einen Wahrheitswert zu
  - nächste Folie



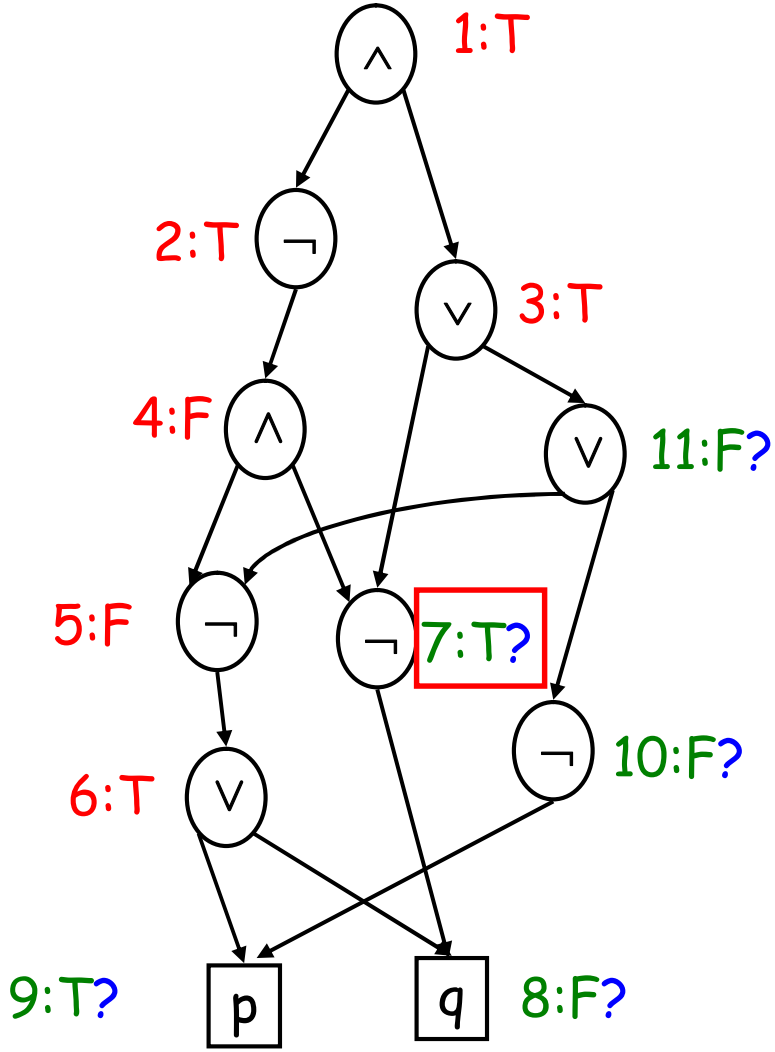
Ausschließen eines Falles

- Wähle eine Position
  - Teste einen Wert
  - teste anderen Wert
- Falls ein Wert zu konsistenter Belegung führt, dann ist die Formel erfüllbar.
- Falls ein Wert zu Inkonsistenz führt, behalte den anderen Wert
- Im Beispiel:
  - 5:T führt zu Inkonsistenz,
  - also setze Position auf 5:F
  - Propagieren und weiter



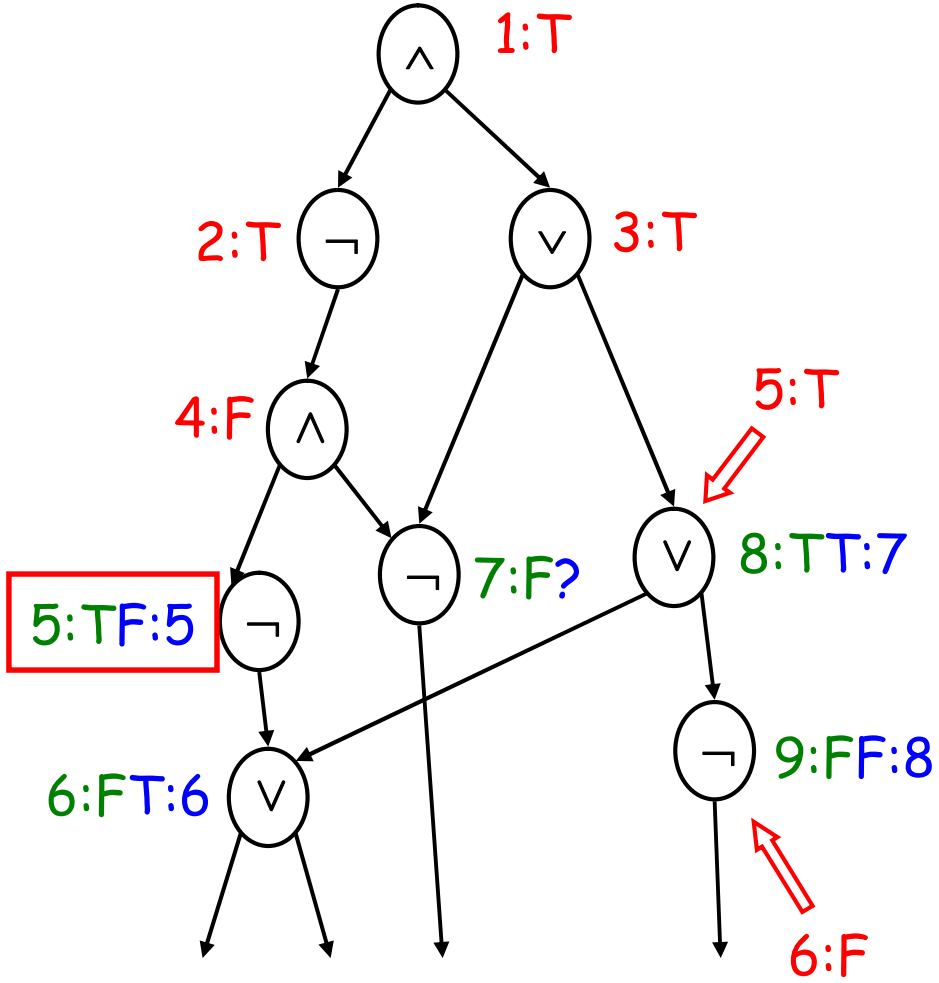
Fortsetzung

- Nächste Fallunterscheidung
  - 7:T führt zu konsistenter Belegung
  - $\Rightarrow$ erfüllbar



Fallunterscheidung erweitert Constraints

- Fallunterscheidung führt
  - weder zu vollständiger Belegung
  - noch zu Widerspruch
- Falls gemeinsamer Wert an einem anderen Knoten entsteht,
  - halte diesen fest
- Im Beispiel:
  - Weder 5:T noch 5:F führen zu Widerspruch oder konsistenter Belegung
  - aber: beide Male zum gleichen Wert für einige Knoten:
    - 8:TT:7 sowie 9:FF:8
  - Lege Werte für diese Knoten fest
    - 5:T und 6:F
  - und weiter …



Zusammenfassung

- Constraint Propagation sehr effektiv
  - kann Belegung liefern
    - Formel erfüllbar
  - kann Inkonsistenz zeigen
    - Formel nicht erfüllbar
  - Ansonsten
    - While not entschieden
      - Wähle Knoten für Fallunterscheidung
      - Probiere True und False
        - Inkonsistent: Lege Komplement als Wert fest; Continue
        - Konsistente Belegung: return erfüllbar
        - Sonst: halte Knoten mit gleicher Bewertung fest
      - Wähle nächsten Knoten

wff	vars	clauses	sato 1997	satz 1997	zChaff 2001	jerusat 2003	siege 2003	MiniSat 2005
p05	3,656	31,089	13.23	0.61	0.01	0.01	0.01	0.02
p15	10,671	143,838	x	4.85	0.05	0.13	0.03	0.09
p18	34,325	750,269	x	x	13.92	6.59	4.85	2.55
p20	40,304	894,643	x	x	14.75	10.35	8.68	10.03
p28	249,738	13,849,105	x	x	846.72	79.59	12.74	27.80

- Trotzdem bleibt SAT NP-vollständig
  - aber
    - bei Formeln der Praxis sind SAT-Prüfer mittlerweile sehr gut
    - Tausende von Variablen !!
- Kombination mit Model Checking
  - Bounded Model Checking industriell sehr erfolgreich
  - Viele Erfolgsstories
  - Erkennt minimale Fehlersituationen
  - Gut für debugging von Protokollen