Erreichbarkeitsanalyse:

Ziel:
- Prüfe automatisch, ob eine gewisse (abstrakte) endliche Konfiguration (c, σ) von der initialen (abstrakten) Konfiguration (c_0, σ_0) erreichbar ist
- Prüfe also: $(c_0, \sigma_0) \Rightarrow^* (c, \sigma)$?
  - ↳ Hierbei handelt es sich um eine Safety-Eigenschaft.
  - ↳ Neben Safety existieren Liveness als weitere Klasse von Eigenschaften. (enthält Safety)

(den Erreichbarkeitsgraph)

Ansatz:
- Fasse die abstrakte Semantik als Transitionssystem auf.
- Prüfe, ob alle erreichbaren Zustände eine gegebene Eigenschaft erfüllen.
- Kodiere Transitionssystem und Eigenschaft symbolisch, also als Formeln
- Benutze SAT-Solver.

Definition:

Ein symbolisches Transitionssystem S ist ein 3-Tupel $S = (V, Q, T)$ bestehend aus:
- Menge an boolschen Variablen V
- Startzuständen Q, und
- Transitionsrelation T.

S heißt symbolisch, da wir Q und T als Formeln darstellen. Dabei ist:
- Q eine Formel über V, und
- T eine Formel über $V \cup V'$.

Die Menge V' enthält die gestrichenen Variablen, $V' = \{x' \mid x \in V\}$.

Wir nehmen $V \cap V' = \emptyset$ an.

Gestrichene Variablen werden benutzt, um Nachfolgezustände zu kodieren.
Eine Transition von Zustand p nach q, zum Beispiel, kodiere wir als

$$p \wedge q'$$

(p: Ausgangszustand, q': Folgezustand)

Beachte, dass p und q hier logische Formeln über V sind, die (Mengen von) Zustände kodieren. Ferner ist q' eine Formel über V', die aus q hervorgeht, indem alle Variablen "gestrichen" werden, $q' = q[V \mapsto V']$.

Definition

Die Nachfolger eines Zustands p in $S = (V, Q, T)$ sind gegeben durch:

$$post(p) := \{ q \mid q' \models \exists V.\ p \wedge T \}$$

Intuition:

1. Bilde $p \wedge T$.
2. Lösche alle ungestrichenen Variablen (Effekt von "$\exists V.$").
3. Benenne die gestrichenen Variablen in die entsprechenden ungestrichenen Variablen um.

Bemerkung:

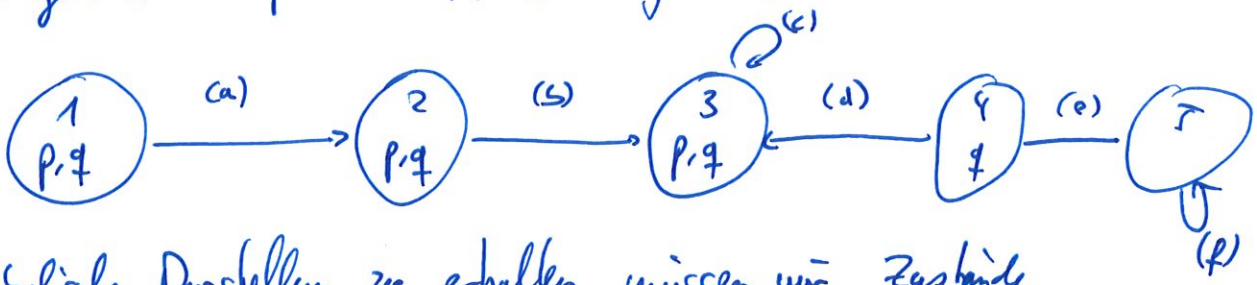
Mit einer SAT-Anfrage für die Formel

$$p \wedge T \wedge q'$$

lässt sich prüfen, ob q' ein Nachfolger von p ist.
Modelle für die Formel $p \wedge T$ liefern Nachfolger von p.

Beispiel:

Betrachte folgendes "explizites" Transitionssystem:



Um eine symbolische Darstellung zu erhalten, müssen wir Zustände und Transitionen als Formeln kodieren.

Wähle als Menge der Variablen: $V = \{p, q, s_1, \dots, s_5\}$

Definiere $enc(i) = s_i \wedge \bigwedge_{j \neq i} \neg s_j$ // Kodierung für Zustände

Dann ergibt sich: $S = (V, Q, T)$ mit

$$Q \equiv enc(1) \wedge p \wedge q$$

$$T \equiv (a) \vee (b) \vee (c) \vee (d) \vee (e) \vee (f)$$

$$(a) \equiv enc(1) \wedge p \wedge q \wedge enc(2)' \wedge p' \wedge q'$$

$$(b) \equiv enc(2) \wedge p \wedge q \wedge enc(3)' \wedge p' \wedge q'$$

$$(c) \equiv enc(3) \wedge p \wedge q \wedge enc(3)' \wedge p' \wedge q'$$

$$(d) \equiv enc(4) \wedge \neg p \wedge q \wedge enc(3)' \wedge p' \wedge q'$$

$$(e) \equiv enc(4) \wedge \neg p \wedge q \wedge enc(5)' \wedge \neg p' \wedge \neg q'$$

$$(f) \equiv enc(5) \wedge \neg p \wedge \neg q \wedge enc(5)' \wedge \neg p' \wedge \neg q'$$

Beachte:

- Man hätte Zustände effizienter (mit weniger Variablen) kodieren können.
- Das symbolische Transitionssystem enthält Zustände, die das explizite Transitionssystem nicht enthält.
  Zum Beispiel: $enc(1) \wedge \neg p \wedge \neg q$

Die erreichbaren Zustände in einem symbolischen Transitionssystem $S = (V, Q, T)$ lassen sich mit folgender Fixpunktgleichung beschreiben:

$$f(X) = X \vee post(X)$$

Beachte, dass wir die Menge $post(X)$ hier als Disjunktion ihrer Elemente interpretieren.

Sind die erreichbaren Zustände berechnet, lässt sich prüfen, ob diese eine gewünschte Eigenschaft aufweisen.

Definition:

Eine Safety-Eigenschaft f für ein symbolisches Transitionssystem $S = (V, Q, T)$ ist eine Formel über V.

Wir sagen S ist safe, falls alle erreichbaren Zustände f erfüllen, also falls gilt:

$$LFP(f) \Rightarrow f$$

Beispiel

Im Beispiel oben erhält man für $LFP(f)$ die Zustände 1, 2, und 3.
Formal sind dies: $(enc(1) \wedge p \wedge q) \vee (enc(2) \wedge p \wedge q) \vee (enc(3) \wedge p \wedge q)$.
Wählen wir als Eigenschaft $f \equiv p$ oder $f \equiv q$, so ist S safe bzgl. f.

Problem: Wie berechnen wir post?

↳ Erinnerung: wir können für einzelne Zustände prüfen, ob sie Nachfolger sind, bzw. einzelne Nachfolger berechnen. Alle Nachfolger können mittels SAT Anfrage nicht direkt berechnet werden.

Mehrere Möglichkeiten:

① Bounded Model Checking (BMC)

② Binary Decision Diagrams (BDDs)

③ Model Checking mit Craig Interpolanten

④ Property Directed Reachability (PDR, oder auch IC3)

↳ state-of-the-art Ansatz

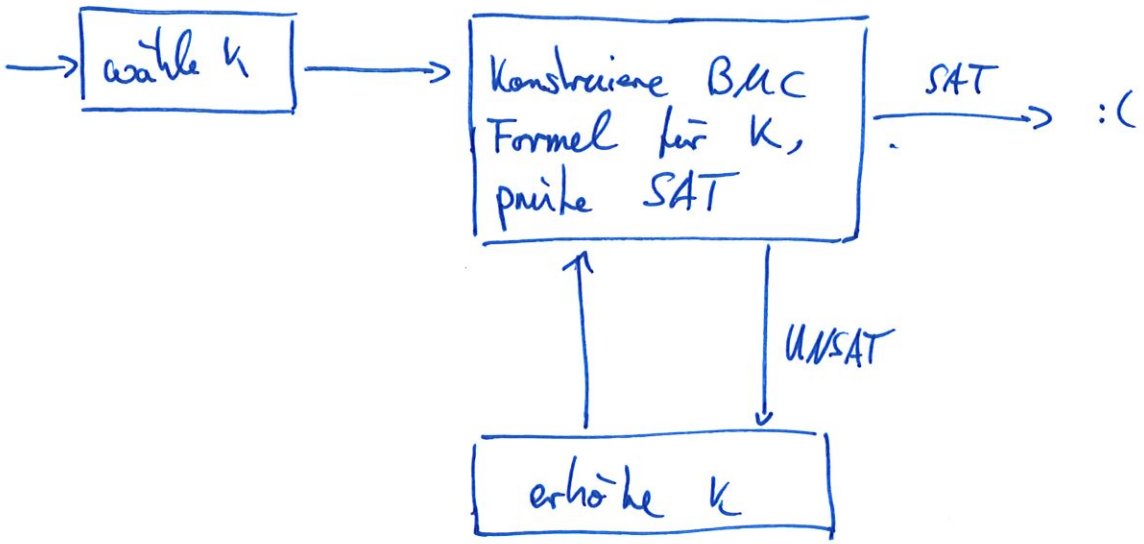
(Weiteres Problem: wie überführen wir ein Program bzw. eine Prädikatenabstraktion eines Programms effizient in ein symbolisches Transitionssystem, ohne den Konfigurationsgraph (vollständig) zu berechnen?)

Bounded Model Checking (BMC)

Ziel: • Bughunting, also schnelles Finden von Fehlern

Idee: • Betrachte Programmausführungen beschränkter Länge
• Darin enthalte Transitionssysteme k mal
↳ finde so Fehler/Gegenbeispiele, die innerhalb von k Programmschritten auftauchen.

Ansatz:



Konstruieren nun eine Formel $path_k$, deren Erfüllbarkeit einen Pfad der Länge k liefert.

Dazu betrachte Transitionssystem $S = (V, Q, T)$.

Um $path_k$ zu konstruieren, benötigen wir $k+1$ Kopien der Variablen:

$$V^{(0)}, \ldots, V^{(k)}$$

~~Hierbei handelt es sich~~

Bei $V = V^{(0)}$ handelt es sich um die ungestrichenen, bei $V^{(1)}$ um die (einfach) gestrichenen, und bei $V^{(i)}$ um die i-fach gestrichenen Variablen Kopien von V.

Dementsprechend ist Q eine Formel über $V^{(0)}$ und T eine Formel über $V^{(i)}$ und $V^{(i+1)}$.

Wir generalisieren T auf beliebig gestrichene Variablen mengen und schreiben $T^{(i)}$, ~~also $T^{(i)}$ über $V^{(i)}$ und $V^{(i+1)}$~~

~~$T(V^{(i)}, V^{(i+1)})$~~.

Damit meinen wir T, wobei alle Variablen i "zusätzliche Striche erhalten".

~~Intuitiv gilt: $Q^{(0)}$ stellt Q, wo die involvierten Variablen explizit übergeben~~

Nun lässt sich $path_k$ wie folgt definieren:

$$path_k :\Leftrightarrow \underbrace{Q^{(0)}}_{\text{wir starten in einem initialen Zustand.}} \wedge \bigwedge_{i=0}^{k-1} \underbrace{T^{(i)}(V^{(i)}, V^{(i+1)})}_{\text{jeder Schritt folgt der Transitionsrelation.}}$$

Beachte: $path_k$ ist eine Formel über $V^0, \ldots, V^k$; enthält also $k+1$ Kopien von $V^{(0)}$.

Um einen Fehler zu finden, müssen wir einen Pfad finden der einen Fehler enthält.

Betrachte eine Safety Eigenschaft f. Wir erweitern $path_k$ wie folgt:

$$BMC_k :\Leftrightarrow path_k \wedge \bigvee_{i=0}^{k} \neg f^{(i)}$$

Ähnlich wie oben entspricht $f^{(i)}$ der Formel f, wobei alle Variablen i-fach gestrichen werden.

Beispiel 1

Betrachte folgendes Transitionssystem

Als symbolische Darstellung wähle:

$S = (V, Q, T)$ mit

$V = \{u, v\}$

$Q^{(0)} = \neg u^{(0)} \wedge \neg v^{(0)}$

$T^{(i)} = \neg u^{(i)} \wedge \neg v^{(i)} \wedge u^{(i+1)} \wedge \neg v^{(i+1)}$

$\vee \; u^{(i)} \wedge \neg v^{(i)} \wedge v^{(i+1)}$ // 1 -> 2 + 1 -> 3

~~$\vee \neg u^{(i)} \wedge v^{(i)} \wedge \neg u^{(i+1)} \wedge \neg v^{(i+1)}$~~

~~$\vee u^{(i)} \wedge v^{(i)} \wedge u^{(i+1)} \wedge v^{(i+1)}$~~

Sei $f^{(i)} = \neg u^{(i)}$ ~~$\wedge \neg v^{(i)}$~~ und $k = 2$. Wir bekommen:

$$BMC_2 :\Leftrightarrow Q^{(0)} \wedge T^{(0)} \wedge T^{(1)} \wedge \left(\neg f^{(0)} \vee \neg f^{(1)} \vee \neg f^{(2)}\right)$$

~~Eine erfüllende Belegung ist~~

$$\Leftrightarrow \neg u^{(0)} \wedge \neg v^{(0)}$$
$$\wedge \left(\neg u^{(0)} \wedge \neg v^{(0)} \wedge u^{(1)} \wedge \neg v^{(1)} \vee u^{(0)} \wedge \neg v^{(0)} \wedge v^{(1)}\right)$$
$$\wedge \left(\neg u^{(1)} \wedge \neg v^{(1)} \wedge u^{(2)} \wedge \neg v^{(2)} \vee u^{(1)} \wedge \neg v^{(1)} \wedge v^{(2)}\right)$$
$$\wedge \left(u^{(0)} \vee u^{(1)} \vee u^{(2)}\right)$$

Eine erfüllende Belegung ist: $u^{(0)} = v^{(0)} = 0$; $u^{(1)} = 1$, $v^{(1)} = 0$

$u^{(2)} = *$, $v^{(2)} = 1$

Beobachtung:

Die Formel BMC$_k$ geht davon aus, dass jeder Zustand einen Nachfolger hat. Im obigen Beispiel ist dies nicht der Fall. Dementsprechend ist $path_3$ nicht erfüllbar und Fehler ~~stec~~ die bei $k=2$ gefunden werden, werden bei $k=3$ übersehen.

Um dies zu lösen, kann man jedem Zustand ein Selbloop hinzufügen. D.h. wir betrachten

$$\tilde{T}^{(i)} := T^{(i)} \vee \bigwedge_{v \in V} v^{(i)} \leftrightarrow v^{(i+1)}$$

Problem:

BMC kann nicht direkt die Korrektheit von Transitionssystemen nachweisen, da nur endliche Pfade aller Länge betrachtet werden.

Es existieren Schranken für k, die eine vollständige Exploration garantieren, die BMC also vollständig machen.

vollständig: alle Fehler werden gefunden.

Unter anderem garantieren folgende Wahlen von k Vollständigkeit:

- $k >$ Anzahl erreichbarer Zustände
- $k >$ Durchmesser (Diameter) des Transitionssystems
  ↳ längste kürzeste Pfad zwischen zwei Knoten im Transitionssystem

Nachzulesen in: "Model Checking" Clarke et al

Die Beweise verlassen sich darauf, dass Schleifen im Transitionssystem nicht betrachtet werden müssen.
Wenn das explizite Transitionssystem nicht bekannt ist, wird eine Berechnung der oberen Schranken allerdings schwer.
$\sum_{k=2}^{|V|}$ gibt allerdings Vollständigkeit.