IC3

Sebastian Wolff

Motivation

- Korrektheit von Systemen nicht mehr überschaubar
- semi-formale Methoden (Testen, …)
  - aufwändig
  - Interaktion benötigt
  - oft nicht ausreichend

Motivation

- formale Methoden (hier: Model Checking)
  - automatisch, keine Interaktion
  - SAT als Subproblem
  - nicht immer möglich
    - ☞ Speicher/Laufzeit als limitierender Faktor
    - ☞ endliche Systeme

HWMCC

- Wettbewerb für Hardware Model Checker
- HWMCC 2010:
  - 9 Teilnehmer insgesamt
  - 1 IC3 Teilnehmer ☞ 3. Platz
- HWMCC 2013: nahezu alle Teilnehmer mit IC3

Spezifikation IC3

Eingabe:

- endliches Transitionssystem $S = (X, I, T)$
- Charakterisierung von „guten“ Zuständen P

Ausgabe:

- Gegenbeispiel oder „korrekt“

Idee

Generiere Sequenz $F_0, \ldots, F_k$ mit:

(1) F_0 enthält alle Startzustände

(2) F_i erfüllt Spezifikation P

(3) F_i enthält alle in i Schritten erreichbaren Zustände

Ziel: finde j mit $F_j = F_{j+1}$

Idee

Generiere Sequenz $F_0, \ldots, F_k$ mit:

(1) $I \Rightarrow F_0$

(2) F_i erfüllt Spezifikation P

(3) F_i enthält alle in i Schritten erreichbaren Zustände

Ziel: finde j mit $F_j = F_{j+1}$

Idee

Generiere Sequenz $F_0, \ldots, F_k$ mit:

(1) $I \Rightarrow F_0$

(2) $F_i \Rightarrow P$

(3) F_i enthält alle in i Schritten erreichbaren Zustände

Ziel: finde j mit $F_j = F_{j+1}$

Idee

Generiere Sequenz $F_0, \ldots, F_k$ mit:

(1) $I \Rightarrow F_0$

(2) $F_i \Rightarrow P$

(3) $F_i \Rightarrow F_{i+1}$ und $F_i \wedge T \Rightarrow F'_{i+1}$

Ziel: finde j mit $F_j = F_{j+1}$

Idee

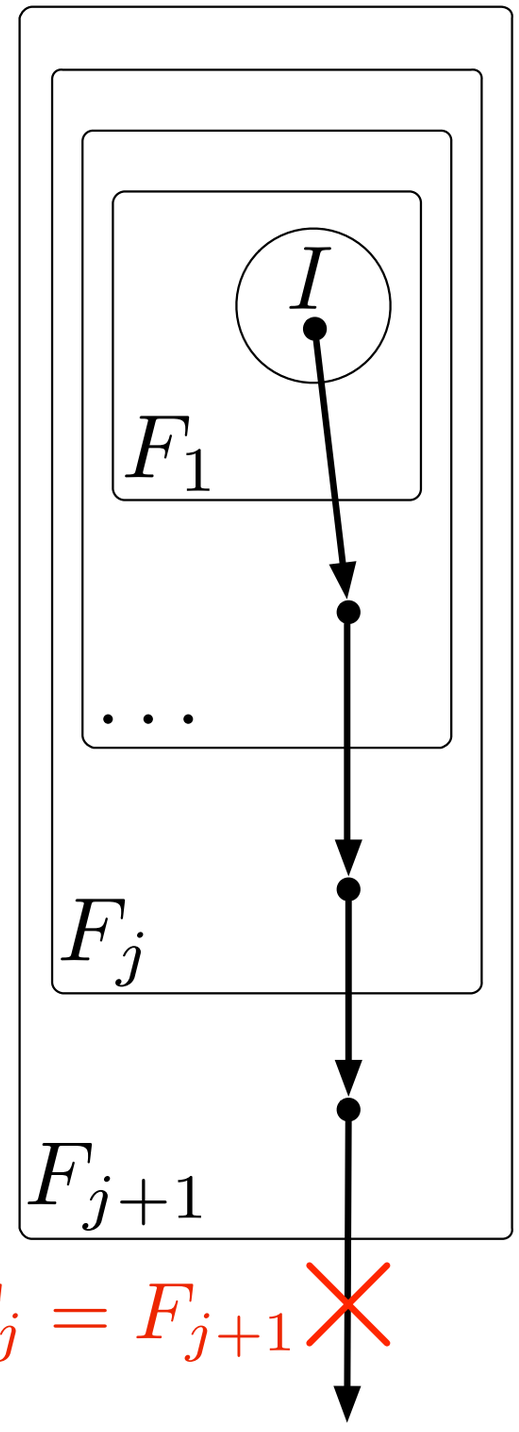
Generiere Sequenz $F_0, \ldots, F_k$ mit:

(1) $I \Rightarrow F_0$

(2) $F_i \Rightarrow P$

(3) $F_i \Rightarrow F_{i+1}$ und $F_i \wedge T \Rightarrow F'_{i+1}$

Ziel: finde j mit $F_j = F_{j+1}$



Sequenz Generieren

- Beginne mit: $F_0 = I$
- Erweitere $F_0, \ldots, F_k$ wie folgt:
- Setze $F_{k+1} = P$ $\rightsquigarrow$ (1) und (2)
- Prüfe (3) durch $sat(F_k \wedge T \wedge \neg P')$

  ☞ Zeuge oder unerfüllbar

Sequenz Generieren

- Beginne mit: $F_0 = I$
- Erweitere $F_0, \ldots, F_k$ wie folgt:
  - Setze $F_{k+1} = P$ $\rightsquigarrow$ (1) und (2)
  - Prüfe (3) durch $sat(F_k \wedge T \wedge \neg P')$

    ☞ Zeuge oder unerfüllbar

$I \Rightarrow F_0$

$F_i \Rightarrow P$

Sequenz Generieren

- Beginne mit: $F_0 = I$
- Erweitere $F_0, \ldots, F_k$ wie folgt:
  - Setze $F_{k+1} = P$ $\rightsquigarrow$ (1) und (2)
  - Prüfe (3) durch $sat(F_k \wedge T \wedge \neg P')$

    ☞ Zeuge oder unerfüllbar

$F_i \Rightarrow F_{i+1}$

$F_i \wedge T \Rightarrow F'_{i+1}$

Zeuge

- Zeuge liefert Zustand s der $\neg P$ erreicht
- Kann s von I erreicht werden?

  ☞ ja: *Bug!*

  ☞ nein: entferne s und seine Vorgänger

Erreichbarkeit

Kann s von I erreicht werden?

- Bisher: BMC

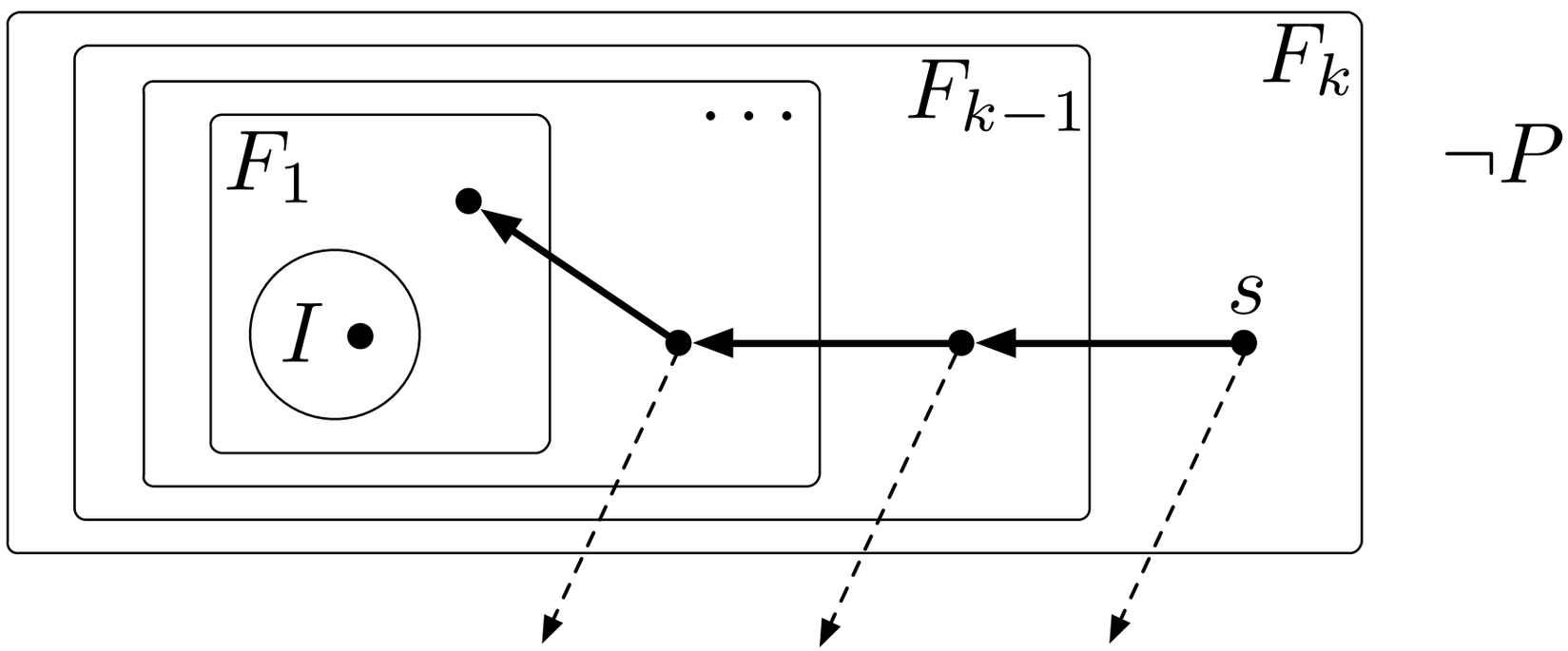
  ☞ Unrolling der Transitionsrelation

$$I(X^0) \wedge \left(\bigwedge_{0 \leq i < k} T(X^i, X^{i+1}) \right) \wedge s(X^k)$$

Erreichbarkeit

Kann s von I erreicht werden?

- Jetzt: IC3s rekursive Rückwärtssuche



Rückwärtssuche

Sei $\langle q, n \rangle$ gegeben

1. Wenn $sat(I \wedge T \wedge q')$: *Bug!*
2. Wähle $min\ j$ mit: $sat(F_j \wedge \neg q \wedge T \wedge q')$
3. Entferne q aus $F_0, ..., F_j$
4. Wenn $j \geq n$: fertig

Rückwärtssuche

Sei $\langle q, n \rangle$ gegeben

1. Wenn $sat(I \wedge T \wedge q')$: *Bug!*
2. Wähle $min\ j$ mit: $sat(F_j \wedge \neg q \wedge T \wedge q')$
3. Entferne q aus $F_0, ..., F_j$
4. Wenn $j \geq n$: fertig

$F_i \wedge T \Rightarrow F'_{i+1}$

Rückwärtssuche

5. Finde Vorgänger t von q aus SAT Anfrage

$$sat(F_j \wedge \neg q \wedge T \wedge q')$$

6. Fahre rekursiv fort mit $\langle t, j \rangle$

7. Wiederhole

Rückwärtssuche

5. Finde Vorgänger t von q aus SAT Anfrage

$$sat(F_j \wedge \neg q \wedge T \wedge q')$$

6. Fahre rekursiv fort mit $\langle t, j \rangle$

$F_i \wedge T \Rightarrow F'_{i+1}$

7. Wiederhole

Terminierung

IC3 terminiert wenn

1) ein j mit $F_j = F_{j+1}$ existiert

2) Rückwärtssuche für $\langle q, n \rangle$ terminiert

Terminierung

ad 1) es existiert ein j mit $F_j = F_{j+1}$

- $F_0, \ldots, F_k$ ist monoton da $F_i \Rightarrow F_{i+1}$
- Transitionssystem ist endlich

Terminierung

ad 2) Rückwärtssuche für $\langle q, n \rangle$

- Verfahren wird für alle Vorgänger wiederholt

  ☞ endlich viele

- Rekursiver Aufruf $\langle t, j \rangle$ mit $j < n$

  ☞ Absteigende Kette (primitiv rekursiv)

Korrektheit

IC3 ist korrekt gdw.

folgende Eigenschaften invariant sind:

(1) $I \Rightarrow F_0$

(2) $F_i \Rightarrow P$

(3) $F_i \Rightarrow F_{i+1}$ und $F_i \wedge T \Rightarrow F'_{i+1}$

Übersicht IC3

- Erreichbarkeitsanalyse bzgl. Fehlerzuständen
- Inkrementell
- Fokussiert einzelne Zustände und Transitionen
- benutzt SAT als Subproblem

  ☞ viele kleine Anfragen

Vielen Dank

Quellen

- Aaron R. Bradley. Sat-based model checking without unrolling. In Ranjit Jhala and David A. Schmidt, editors, Verification, Model Checking, and Abstract Interpretation - 12th International Conference, VMCAI 2011, Austin, TX, USA, January 23-25, 2011. Proceedings, volume 6538 of Lecture Notes in Computer Science, pages 70–87. Springer, 2011.

- Aaron R. Bradley. Understanding IC3. In Alessandro Cimatti and Roberto Sebastiani, editors, Theory and Applications of Satisfiability Testing - SAT 2012 - 15th International Conference, Trento, Italy, June 17-20, 2012. Proceedings, volume 7317 of Lecture Notes in Computer Science, pages 1–14. Springer, 2012.

- Yakir Vizel and Orna Grumberg. Interpolation-sequence based model checking. In Proceedings of 9th International Conference on Formal Methods in Computer-Aided Design, FMCAD 2009, 15-18 November 2009, Austin, Texas, USA, pages 1–8. IEEE, 2009.

- HWMCC'10 Homepage und Ergebnisfolien: http://fmv.jku.at/hwmcc10/results.html und http://fmv.jku.at/biere/talks/Biere-HWMCC10-talk.pdf

- HWMCC'13 Homepage und Ergebnisfolien: http://fmv.jku.at/hwmcc13/ und http://fmv.jku.at/hwmcc13/hwmcc13.pdf