Encodings of Reactive Synthesis

QUANTIFY 2015, Berlin

Peter Faymonville
Bernd Finkbeiner
Markus Rabe
Leander Tentrup

Reactive Systems Group
Saarland University

August 3rd, 2015

Outline

Bounded Synthesis

Encodings

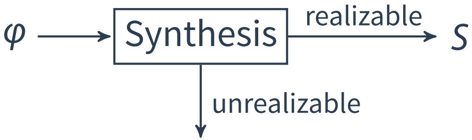
Experimental Results

Strategy Extraction

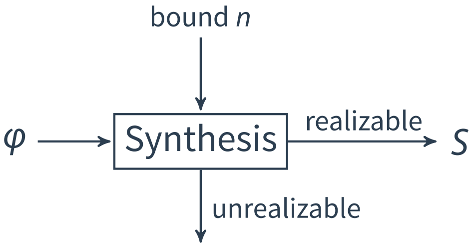
Synthesis of Reactive Systems

- Systems that react on external events
- Interest accelerated towards automatic construction (synthesis)
- Two iterations of SyntComp (restricted safety format)
- Full synthesis track in planning

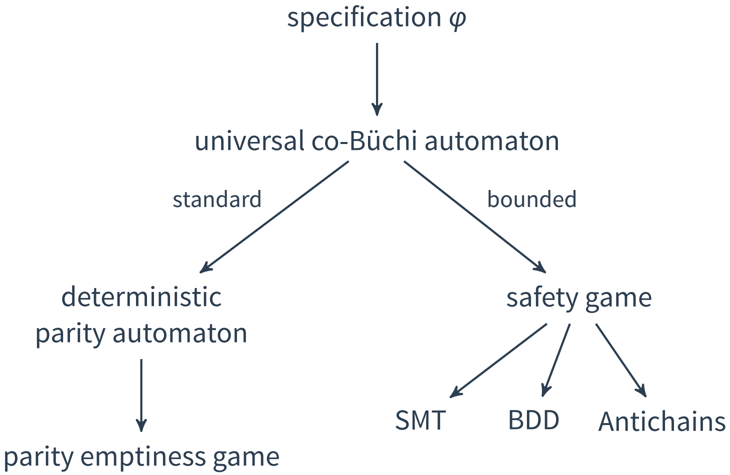
Reactive Synthesis



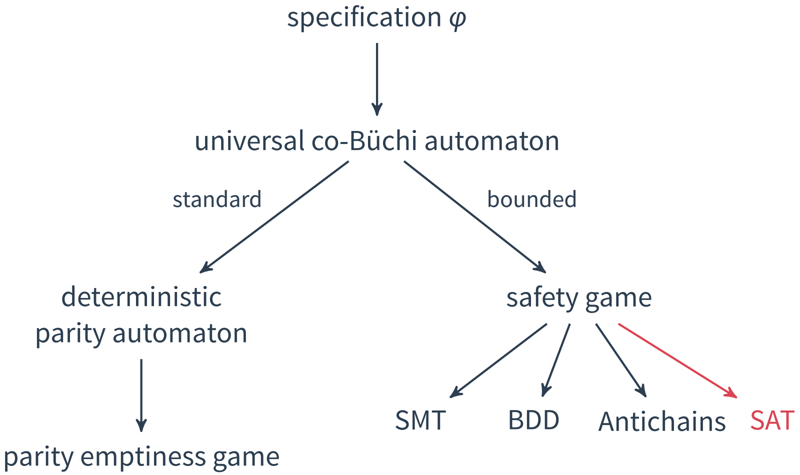
Bounded Synthesis



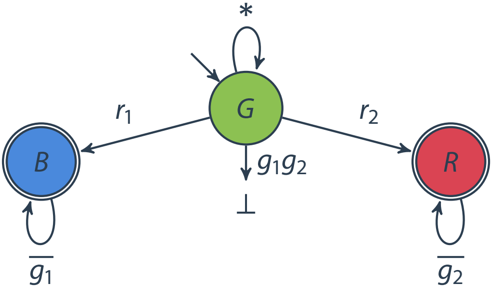
Single Process Synthesis



Single Process Synthesis



Universal Co-Büchi Automata



Example (Simplified Arbiter)

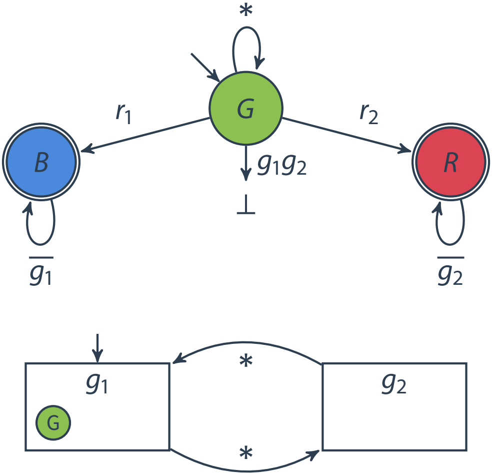
$$\varphi = \Box(r_1 \rightarrow \bigcirc\Diamond g_1) \land \Box(r_2 \rightarrow \bigcirc\Diamond g_2) \land \neg\Diamond(g_1 \land g_2)$$

Acceptance

A **transition system** is accepted by an universal co-Büchi automaton if **all paths** in the (unique) run graph contain only **finitely** many visits to rejecting states.

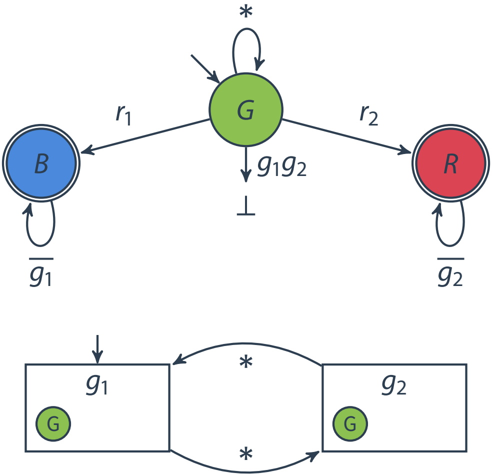
Acceptance of a Transition System

example



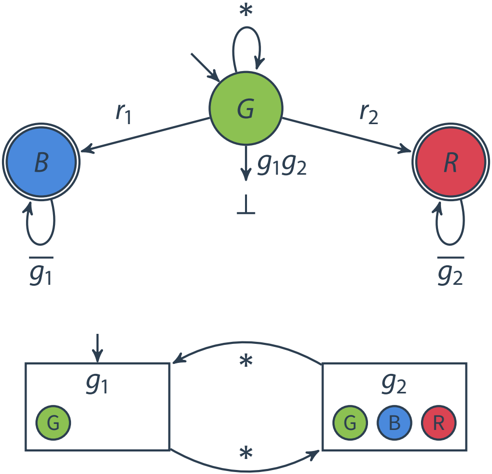
Acceptance of a Transition System

example



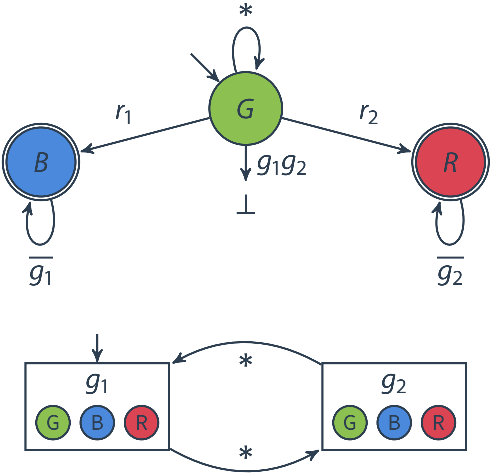
Acceptance of a Transition System

example



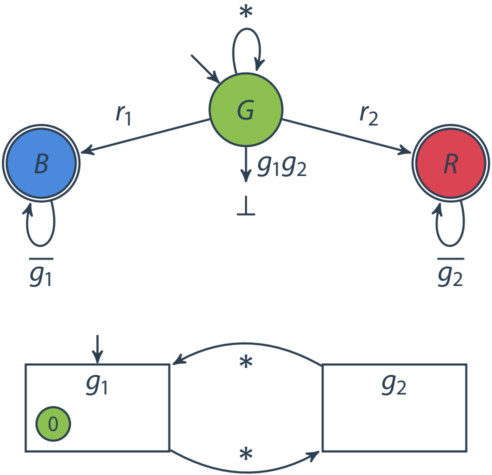
Acceptance of a Transition System

example



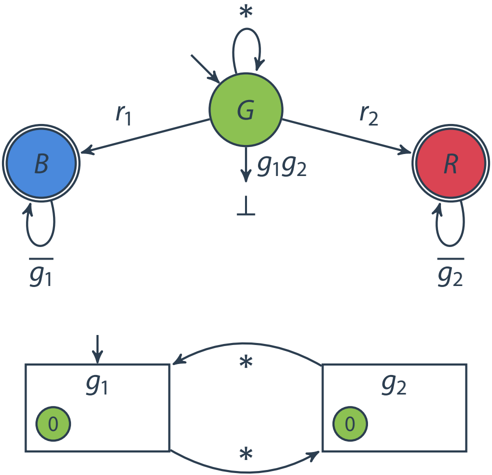
Acceptance of a Transition System

example

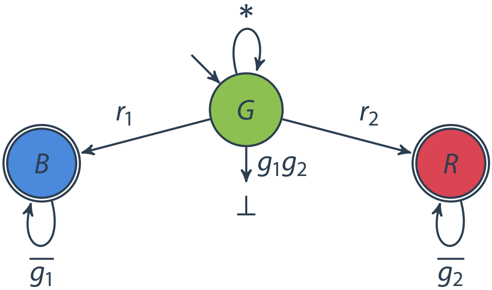


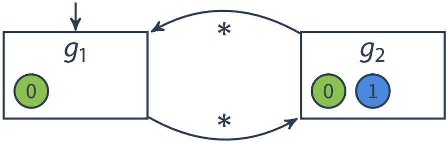
Acceptance of a Transition System

example



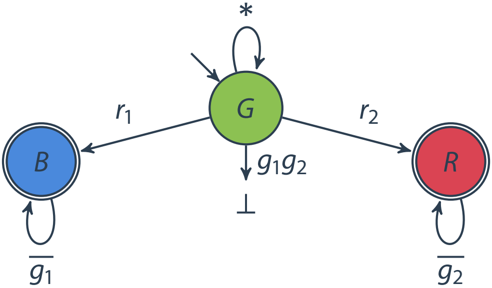
Acceptance of a Transition System
example

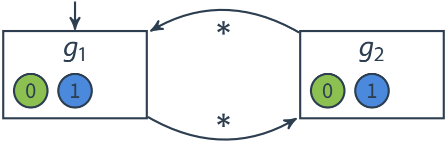




Acceptance of a Transition System

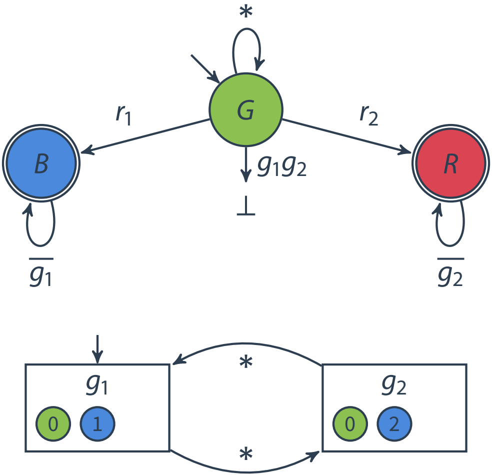
example





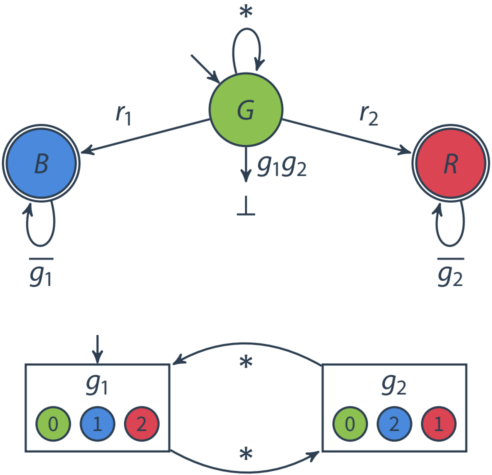
Acceptance of a Transition System

example



Acceptance of a Transition System

example

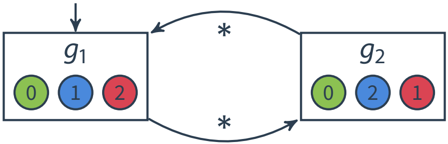


Annotated Transition System

- collects the paths of the run graph that lead to a state in the transition system
- for each automaton state, indicates whether state visited on some path, and if so, max number of visits to rejecting states

Theorem (Finkbeiner & Schewe'07)

A transition system is accepted by a universal co-Büchi automaton ⇔ it has a valid annotation



Encodings

Build a constraint system that specifies the existence of an annotated transition system

- Representation of transition system
  - states
  - labeling
  - transitions
- Representation of annotation
  - state occurrence
  - rejecting bound

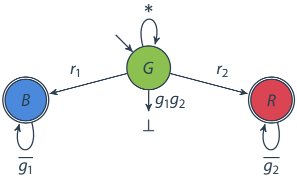
SMT Encoding

Inputs I, Outputs O, universal co-Büchi automaton $\langle Q, q_0, \delta, R \rangle$

- Representation of transition system
  - **states**: $\mathbb{N}_n = \{0, \ldots, n-1\}$
  - **labeling**: functions $o : \mathbb{N}_n \to \mathbb{B}$ for every $o \in O$
  - **transitions**: functions $\tau_I : \mathbb{N}_n \to \mathbb{N}_n$
- Representation of annotation
  - **state occurrence**: functions $\lambda_q^{\mathbb{B}} : \mathbb{N}_n \to \mathbb{B}$
  - **rejecting bound**: functions $\lambda_q^{\#} : \mathbb{N}_n \to \mathbb{N}$

SMT Encoding

example



- $\lambda^{\mathbb{B}}_G(0)$
- $$\begin{aligned}\forall s.\, \lambda^{\mathbb{B}}_G(s) \to\ & \lambda^{\mathbb{B}}_G(\tau_{\overline{r_1 r_2}}(s)) \wedge \lambda^{\#}_G(\tau_{\overline{r_1 r_2}}(s)) \geq \lambda^{\#}_G(s)\\ \wedge\ & \lambda^{\mathbb{B}}_G(\tau_{\overline{r_1} r_2}(s)) \wedge \lambda^{\#}_G(\tau_{\overline{r_1} r_2}(s)) \geq \lambda^{\#}_G(s)\\ \wedge\ & \lambda^{\mathbb{B}}_G(\tau_{r_1 \overline{r_2}}(s)) \wedge \lambda^{\#}_G(\tau_{r_1 \overline{r_2}}(s)) \geq \lambda^{\#}_G(s)\\ \wedge\ & \lambda^{\mathbb{B}}_G(\tau_{r_1 r_2}(s)) \wedge \lambda^{\#}_G(\tau_{r_1 r_2}(s)) \geq \lambda^{\#}_G(s)\end{aligned}$$
- $\forall s.\, \lambda^{\mathbb{B}}_G(s) \to \neg g_1(s) \vee \neg g_2(s)$
- $$\begin{aligned}\forall s.\, \lambda^{\mathbb{B}}_G(s) \wedge r_1(s) \to\ & \lambda^{\mathbb{B}}_B(\tau_{\overline{r_1 r_2}}(s)) \wedge \lambda^{\#}_B(\tau_{\overline{r_1 r_2}}(s)) > \lambda^{\#}_G(s)\\ \wedge\ & \lambda^{\mathbb{B}}_B(\tau_{\overline{r_1} r_2}(s)) \wedge \lambda^{\#}_B(\tau_{\overline{r_1} r_2}(s)) > \lambda^{\#}_G(s)\\ \wedge\ & \lambda^{\mathbb{B}}_B(\tau_{r_1 \overline{r_2}}(s)) \wedge \lambda^{\#}_B(\tau_{r_1 \overline{r_2}}(s)) > \lambda^{\#}_G(s)\\ \wedge\ & \lambda^{\mathbb{B}}_B(\tau_{r_1 r_2}(s)) \wedge \lambda^{\#}_B(\tau_{r_1 r_2}(s)) > \lambda^{\#}_G(s)\end{aligned}$$
- $$\begin{aligned}\forall s.\, \lambda^{\mathbb{B}}_B(s) \wedge \neg g_1(s) \to\ & \lambda^{\mathbb{B}}_B(\tau_{\overline{r_1 r_2}}(s)) \wedge \lambda^{\#}_B(\tau_{\overline{r_1 r_2}}(s)) > \lambda^{\#}_B(s)\\ \wedge\ & \lambda^{\mathbb{B}}_B(\tau_{\overline{r_1} r_2}(s)) \wedge \lambda^{\#}_B(\tau_{\overline{r_1} r_2}(s)) > \lambda^{\#}_B(s)\\ \wedge\ & \lambda^{\mathbb{B}}_B(\tau_{r_1 \overline{r_2}}(s)) \wedge \lambda^{\#}_B(\tau_{r_1 \overline{r_2}}(s)) > \lambda^{\#}_B(s)\\ \wedge\ & \lambda^{\mathbb{B}}_B(\tau_{r_1 r_2}(s)) \wedge \lambda^{\#}_B(\tau_{r_1 r_2}(s)) > \lambda^{\#}_B(s)\end{aligned}$$

Generic Propositional Encoding

Inputs I, Outputs O, universal co-Büchi automaton $\langle Q, q_0, \delta, R \rangle$

- Representation of transition system
  - **states**: $S = \{s_0, \ldots, s_{n-1}\}$
  - **labeling**: $o : S \to \mathbb{B}$ for every $o \in O$
  - **transitions**: $\tau : S \times \mathbb{B}^{|I|} \to S$
- Representation of annotation
  - **state occurrence**: $\lambda : S \times Q \to \mathbb{B}$
  - **rejecting bound**: $\lambda^{\#} : S \times Q \to \mathbb{B}^{b}$ (b-bit counter)

Generic Propositional Encoding

Inputs I, Outputs O, universal co-Büchi automaton $\langle Q, q_0, \delta, R \rangle$

- Representation of transition system
  - **states**: $S = \{s_0, \ldots, s_{n-1}\}$
  - **labeling**: $o : S \to \mathbb{B}$ for every $o \in O$
  - **transitions**: $\tau : S \times \mathbb{B}^{|I|} \to S$
- Representation of annotation
  - **state occurrence**: $\lambda : S \times Q \to \mathbb{B}$
  - **rejecting bound**: $\lambda^{\#} : S \times Q \to \mathbb{B}^{b}$ (b-bit counter)

$\exists \lambda(s, q), \lambda^{\#}(s, q), o(s), \tau(s, i).$

Generic Propositional Encoding

Inputs I, Outputs O, universal co-Büchi automaton $\langle Q, q_0, \delta, R \rangle$

- Representation of transition system
  - **states**: $S = \{s_0, \ldots, s_{n-1}\}$
  - **labeling**: $o : S \to \mathbb{B}$ for every $o \in O$
  - **transitions**: $\tau : S \times \mathbb{B}^{|I|} \to S$
- Representation of annotation
  - **state occurrence**: $\lambda : S \times Q \to \mathbb{B}$
  - **rejecting bound**: $\lambda^\# : S \times Q \to \mathbb{B}^b$ (b-bit counter)

$$\exists \lambda(s, q), \lambda^\#(s, q), o(s), \tau(s, i). \quad \forall s, s', q, q', i.$$

Generic Propositional Encoding

Inputs I, Outputs O, universal co-Büchi automaton $\langle Q, q_0, \delta, R \rangle$

- Representation of transition system
  - **states**: $S = \{s_0, \ldots, s_{n-1}\}$
  - **labeling**: $o : S \to \mathbb{B}$ for every $o \in O$
  - **transitions**: $\tau : S \times \mathbb{B}^{|I|} \to S$
- Representation of annotation
  - **state occurrence**: $\lambda : S \times Q \to \mathbb{B}$
  - **rejecting bound**: $\lambda^{\#} : S \times Q \to \mathbb{B}^{b}$ (b-bit counter)

$$\exists \lambda(s, q), \lambda^{\#}(s, q), o(s), \tau(s, i). \quad \forall s, s', q, q', i. \quad \lambda(s_0, q_0) \wedge$$

Generic Propositional Encoding

Inputs I, Outputs O, universal co-Büchi automaton $\langle Q, q_0, \delta, R \rangle$

- Representation of transition system
  - **states**: $S = \{s_0, \dots, s_{n-1}\}$
  - **labeling**: $o : S \to \mathbb{B}$ for every $o \in O$
  - **transitions**: $\tau : S \times \mathbb{B}^{|I|} \to S$
- Representation of annotation
  - **state occurrence**: $\lambda : S \times Q \to \mathbb{B}$
  - **rejecting bound**: $\lambda^{\#} : S \times Q \to \mathbb{B}^{b}$ (b-bit counter)

$$\exists \lambda(s,q), \lambda^{\#}(s,q), o(s), \tau(s,i). \quad \forall s, s', q, q', i. \quad \lambda(s_0, q_0) \wedge$$
$$\left(\lambda(s,q) \wedge \delta(q, o(s), i, q') \wedge (\tau(s,i) = s')\right)$$

Generic Propositional Encoding

Inputs I, Outputs O, universal co-Büchi automaton $\langle Q, q_0, \delta, R\rangle$

- Representation of transition system
  - **states**: $S = \{s_0, \dots, s_{n-1}\}$
  - **labeling**: $o : S \to \mathbb{B}$ for every $o \in O$
  - **transitions**: $\tau : S \times \mathbb{B}^{|I|} \to S$
- Representation of annotation
  - **state occurrence**: $\lambda : S \times Q \to \mathbb{B}$
  - **rejecting bound**: $\lambda^\# : S \times Q \to \mathbb{B}^b$ (b-bit counter)

$$\exists \lambda(s,q), \lambda^\#(s,q), o(s), \tau(s,i). \quad \forall s, s', q, q', i. \quad \lambda(s_0, q_0) \wedge$$
$$\big(\lambda(s,q) \wedge \delta(q, o(s), i, q') \wedge (\tau(s,i) = s')\big) \to \big(\lambda(s',q') \wedge \lambda^\#(s',q') \rhd \lambda^\#(s,q)\big)$$

$$\rhd := \begin{cases} > & \text{if } q \text{ rejecting} \\ \geq & \text{otherwise} \end{cases}$$

Derived Encodings

- SAT: complete unrolling

$$\exists \lambda_{s,q}, \lambda^{\#}_{s,q}, o_s, \tau_{s,i,s'}$$

- QBF: input symbolic encoding

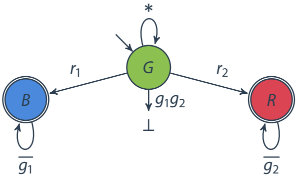
$$\exists \lambda_{s,q}, \lambda^{\#}_{s,q}, o_s . \forall i . \tau_{s,s'}$$

- DQBF: state and input symbolic encoding

$$\begin{array}{l} \forall s . \exists \lambda_q, \lambda^{\#}_q, o . \forall i . \tau \\ \forall s' . \exists \lambda'_q, \lambda'^{\#}_q . \end{array} \quad \bigwedge_q (s = s') \rightarrow (\lambda_q = \lambda'_q) \wedge (\lambda^{\#}_q = \lambda'^{\#}_q)$$

QBF Encoding

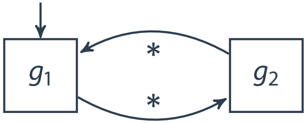
example



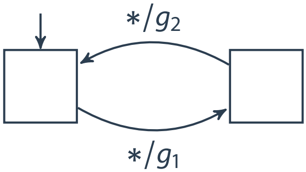
- $\lambda_{s_0,G}$
- $\bigwedge_{s\in S}\left(\lambda_{s,G} \rightarrow \bigwedge_{s'\in S}(\tau_{s,s'} \rightarrow \lambda_{s',G} \wedge \lambda^{\#}_{s',G} \geq \lambda^{\#}_{s,G})\right)$
- $\bigwedge_{s\in S}\left(\lambda_{s,G} \rightarrow \neg g_1^s \vee \neg g_2^s\right)$
- $\bigwedge_{s\in S}\left(\lambda_{s,G} \wedge r_1 \rightarrow \bigwedge_{s'\in S}(\tau_{s,s'} \rightarrow \lambda_{s',B} \wedge \lambda^{\#}_{s',B} > \lambda^{\#}_{s,G})\right)$
- $\bigwedge_{s\in S}\left(\lambda_{s,B} \wedge \neg g_1^s \rightarrow \bigwedge_{s'\in S}(\tau_{s,s'} \rightarrow \lambda_{s',B} \wedge \lambda^{\#}_{s',B} > \lambda^{\#}_{s,B})\right)$

Mealy and Moore Transition Systems

Moore: State-labeled transition systems

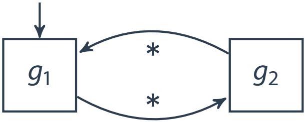


Mealy: Edge-labeled transition systems



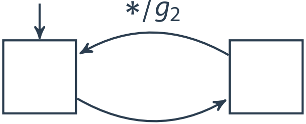
Mealy and Moore Transition Systems

Moore: State-labeled transition systems



$$\exists \lambda_{s,q}, \lambda^{\#}_{s,q}, o_s.\ \forall i.\ \exists \tau_{s,s'}$$

Mealy: Edge-labeled transition systems



$$\exists \lambda_{s,q}, \lambda^{\#}_{s,q}.\ \forall i.\ \exists o_s, \tau_{s,s'}$$

Implementation

Encoding	Solver	Strategy Extraction
SMT	Z3, CVC4	✓
SAT	MiniSat, PicoSAT	✓
QBF	RAReQS, DepQBF, Bloqqer	✓
DQBF	iDQ, eprover (EPR)	

Experiments

Arbiter

$$\bigwedge_i \Box(r_i \rightarrow \Diamond g_i) \qquad \text{(response)}$$

$$\bigwedge_{i \neq j} \Box\neg(g_i \wedge g_j) \qquad \text{(mutex)}$$

Arbiter without spurious grants

$$\bigwedge_i \neg((\neg r_i \wedge \neg g_i)\,\mathcal{U}\,(\neg r_i \wedge g_i)) \qquad \text{(no-spurious-start)}$$

$$\bigwedge_i \neg\Diamond\Big(g_i \wedge \bigcirc\Big((\neg r_i \wedge \neg g_i) \wedge \big((\neg r_i \wedge \neg g_i)\mathcal{U}(\neg r_i \wedge g_i)\big)\Big)\Big) \qquad \text{(no-spurious)}$$

$$\bigwedge_i \Box((\neg r_i \wedge g_i) \rightarrow \Diamond((r_i \wedge g_i) \vee \neg g_i)) \qquad \text{(lowered)}$$

Results

Quad-Core Intel Xeon @ 3.6 GHz, 32 GB RAM, 1*h* timeout

Instance	SMT Z3		SAT MiniSat		QBF RAReQS+B		DQBF iDQ	
	mealy	moore	mealy	moore	mealy	moore	mealy	moore
arbiter-2	0.22	0.21	0.21	0.21	0.19	0.19	0.23	0.23
arbiter-3	0.45	0.39	0.30	0.31	0.21	0.21	0.62	0.63
arbiter-4	1428	2234	0.73	0.76	0.25	0.25	1.71	1.83
arbiter-5	TO	TO	4.15	3.72	0.40	0.40	24.6	24.6
arbiter-6	TO	TO	21.0	21.0	0.71	0.71	76.8	79.9
arbiter-7	TO	TO	155.6	102.5	7.02	5.98	294.6	294.3
arbiter-8	TO	TO	2384	TO	397.6	406.6	TO	TO
full-arbiter-2	0.49	2.75	0.39	0.63	0.28	0.38	20.6	19.9
full-arbiter-3	TO	TO	16.6	34.9	9.28	17.6	TO	TO
full-arbiter-4	TO	TO	TO	TO	TO	TO	TO	TO

QBF Encoding

Quad-Core Intel Xeon @ 3.6 GHz, 32 GB RAM, 1h timeout

Instance	QBF Mealy				
	RAReQS+B	RAReQS	DepQBF+B	DepQBF	RAReQS-QCIR
arbiter-2	0.19	0.21	0.20	0.21	0.26
arbiter-3	0.21	0.21	0.22	0.21	0.33
arbiter-4	0.25	0.31	0.25	0.30	0.52
arbiter-5	0.40	0.34	0.41	0.45	1.2
arbiter-6	0.71	0.90	1.28	2.46	4.46
arbiter-7	7.02	30.3	128.9	TO	94.8
arbiter-8	397.6	TO	TO	TO	TO
full-arbiter-2	0.28	0.28	0.27	0.35	1.05
full-arbiter-3	9.28	867	TO	TO	968
full-arbiter-4	TO	TO	TO	TO	TO

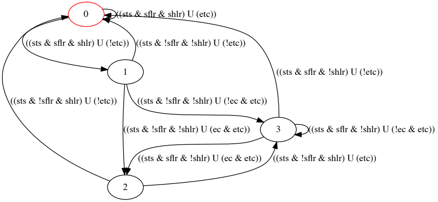
Strategy Extraction

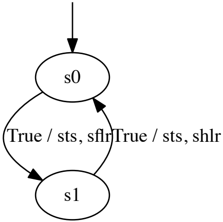
- Use certification feature of solvers to get witness for o and τ
  - Model from SMT solver
  - Assignments from SAT solver
  - Skolem functions from QBF solver
- Build transition system and encode it in SMV
- Model-check solution (NuSMV)

Strategy Extraction

example

Traffic Light Controller (Lily benchmark)





Acacia+ (optimal strategy option)

QBF Encoding

Conclusions
and future work

- Generic propositional encoding
- Encodings to SAT, QBF, DQBF
- All propositional encodings outperform SMT

- Optimizations similar to previous work (decompose specification into safety/liveness, conjunctions, etc.)
- Incremental solving