Bernd Finkbeiner Date: February 05, 2013

Automata, Games, and Verification: Lecture 15

---

20 Alternating-time Temporal Logic

Syntax

- *ATL state formulas*:
  - $a \in AP$ — atomic proposition
  - $\neg\,\Phi$ and $\Phi \,\wedge\, \Psi$ — negation and conjunction
  - $\langle\langle A \rangle\rangle \varphi$ — agents in A have strategy to enforce φ
- *ATL path formulas* as for CTL.

$A \subseteq \{1, \ldots, k\}$ is a set of players.

Semantics

Definition 1 *A* **concurrent game structure** $(k, AP, S, s_0, d, \delta, L)$ *consists of*

- $k \in \mathbb{N}$*: number of players*
- *AP: atomic propositions*
- *S: finite set of states,* $s_0 \in S$*: initial state*
- $d : \{1, \ldots, k\} \times S \to \mathbb{N}$*: number of moves available to player*
- $\delta : S \times \{1, \ldots, d(1)\} \times \ldots \times \{1, \ldots, d(k)\} \to S$*: transition function*
- $L : S \to 2^{AP}$*: labeling function*

- A *strategy* for player a is a function $f_a : S^+ \to \mathbb{N}$
  such that $f_a(\sigma \cdot q) \leq d_a(q)$.
- Given a set $F_A = \{f_a \mid a \in A\}$ of strategies for a set of players A,
  the *outcomes* $\mathit{Outcomes}(F_A, s)$ of F_A from state s are the paths $s_0 s_1 s_2 \ldots$ such that $s_0 = s$ and for all $i \geq 0$ there is a vector $(j_1, \ldots, j_k) \in \mathbb{N}^k$ such that
  - $j_a = f_a(s_0 \ldots s_i)$ for all players $a \in A$, and
  - $\delta(s_i, j_1, \ldots, j_k) = s_{i+1}$
- $s \models \langle\langle A \rangle\rangle \varphi$ iff there exists a set of strategies F_A for the players in A,
  such that $\pi \models \varphi$ for all $\pi \in \mathit{Outcomes}(F_A, s)$.

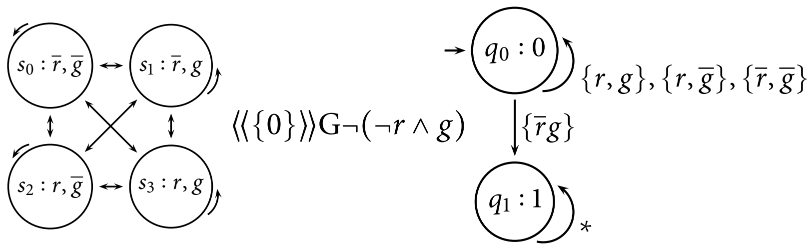
Synthesis games

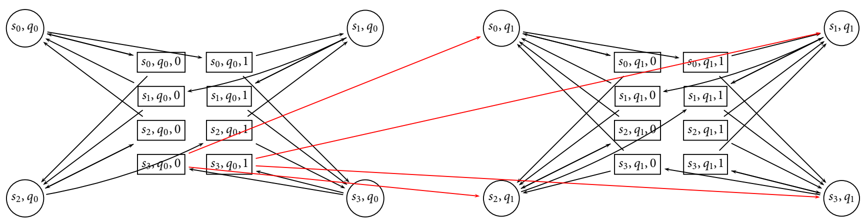
- Let $\mathcal{T} = (k, AP, S, s_0, d, \delta)$ be a concurrent game structure.
- Let $\langle\langle A\rangle\rangle\varphi$ be an ATL* formula where φ is an LTL formula.
- Let $\mathcal{A}_\varphi = (\Sigma, Q_\varphi, q_0, \delta_\varphi, c_\varphi)$ be a complete deterministic parity automaton with $\mathcal{L}(\mathcal{A}_\varphi) = \mathcal{L}(\varphi)$.

The *synthesis game* $\mathcal{G}(\mathcal{T}, \varphi, A) = ((V_0, V_1, E), c)$:

- $V_0 = S \times Q_\varphi$
- $V_1 = S \times Q_\varphi \times \{j_a : \{1, \dots, d_a(s)\}\}_{a \in A}$
- $E = \{(s,q), (s,q,\{j_a\}_{a\in A})) \mid j_a \in \{1,\dots,d_a(s)\}\}$
  $\cup \{((s,q,\{j_a\}_{a\in A}), (s',q')) \mid q' = \delta_\varphi(q, L(s))$
  and there is a vector $(j'_1, \dots, j'_k) \in \mathbb{N}^k$ such that
  $j'_a = j_a$ for all players $a \in A$ and $\delta(s, j'_1, \dots, j'_k) = s'\}$
- $c = c_\varphi(s)$ for $s \in S$ and 0 otherwise.

Example: Consider the following resource manager, where player 0 (the system) controls g (the grant) and player 1 (the environment) controls r (the request).





∎

ATL* model checking

for all $i \leq |\Phi|$
for all $\Psi \in \mathit{Subformulas}(\Phi)$ with $|\Psi| = i$
switch(Ψ):

$$\begin{array}{lcl} \mathit{true} & : & \mathit{Sat}(\Psi) := S; \\ a & : & \mathit{Sat}(\Psi) := \{q \in S \mid a \in L(q)\}; \\ a_1 \wedge a_2 & : & \mathit{Sat}(\Psi) := \mathit{Sat}(a_1) \cap \mathit{Sat}(a_2); \\ \neg a & : & \mathit{Sat}(\Psi) := S \setminus \mathit{Sat}(a); \\ \langle\langle A \rangle\rangle \varphi & : & \mathit{Sat} \text{ is winning set in synthesis game} \end{array}$$

end switch
$AP := AP \cup \{a_\Psi\}$; % *introduce fresh atomic proposition*
replace Ψ with a_Ψ
forall $q \in \mathit{Sat}(\Psi)$ **do** $L(q) := L(q) \cup \{a_\Psi\}$; **od**
return $\mathit{Sat}(\Phi)$

21 Strategy Logic

Variables:

- $x_1, x_2, \ldots$: strategies of Player 1
- $y_1, y_2, \ldots$: strategies of Player 2

SL state formulas:

$$\Phi ::= \mathit{true} \;\Big|\; a \;\Big|\; \Phi_1 \wedge \Phi_2 \;\Big|\; \neg\Phi \;\Big|\; \Psi$$

SL path formulas:

$$\varphi ::= \Phi \;\Big|\; \varphi_1 \wedge \varphi_2 \;\Big|\; \neg\varphi \;\Big|\; \mathrm{X}\varphi \;\Big|\; \varphi_1 \mathcal{U} \varphi_2$$

SL strategy formulas:

$$\Gamma ::= \varphi(x, y) \;\Big|\; \Gamma_1 \wedge \Gamma_2 \;\Big|\; \neg\Gamma \;\Big|\; \exists x.\Gamma \;\Big|\; \exists y.\Gamma \;\Big|\; \forall x.\Gamma \;\Big|\; \forall y.\Gamma$$

where $a \in AP$, Φ is a state formula, φ, φ_1 and φ_2 are path formulas, and Ψ is a closed strategy formula.
A formula is closed if all strategy variables are quantified.

Embedding of ATL* in SL

- Every ATL* formula can be expressed in SL:

$$\begin{array}{rcl} \langle\langle\{1\}\rangle\rangle \mathrm{F}p & = & \exists x.\forall y.\ (\mathrm{F}p)(x, y) \\ \langle\langle\{2\}\rangle\rangle \mathrm{F}p & = & \exists y.\forall x.\ (\mathrm{F}p)(x, y) \\ \langle\langle\{1, 2\}\rangle\rangle \mathrm{F}p & = & \exists x.\exists y.\ (\mathrm{F}p)(x, y) \end{array}$$

- Restricted strategies can be expressed in SL, but not in ATL*:

$$\exists x_1.\forall y_1.((\forall x_2.\varphi(x_2, y_1)) \Rightarrow \psi(z_1, y_1))$$

22 Summary

Automata

1. *Branching Mode*
   deterministic – nondeterministic – universal – alternating

2. *Acceptance Mode*
   Büchi – co-Büchi – parity – Streett – Rabin – Muller

3. *Input*
   words – trees

Expressive Power

Word automata:

	Büchi	co-Büchi	parity	Muller
deterministic	−	−	+	+
nondeterministic	+	−	+	+
universal	−	+	+	+
alternating	+	+	+	+

Tree automata:

	Büchi	co-Büchi	parity	Muller
deterministic	−	−	−	−
nondeterministic	−	−	+	+
universal	−	−	+	+
alternating	−	−	+	+

Characterization Theorems

Definition 2 *An ω-*regular *language is a finite union of ω-languages of the form $U \cdot V^\omega$ where $U, V \subseteq \Sigma^*$ are regular languages.*

Theorem 1 (Büchi's Characterization Theorem (1962)) *An ω-language is* Büchi recognizable *iff it is ω-regular.*

Theorem 2 *An ω-language $L \subseteq \Sigma^\omega$ is recognizable by a* deterministic Büchi automaton *iff there is a regular language $W \subseteq \Sigma^*$ s.t. $L = \overrightarrow{W}$.*

Theorem 3 *A language $\mathcal{L}$ is recognizable by a* deterministic Muller *automaton iff $\mathcal{L}$ is a boolean combination of languages $\overrightarrow{W}$ where $W \subseteq \Sigma^*$ is regular.*

Translating Branching Modes

- McNaughton: *nondeterministic Büchi word* automaton → *deterministic Muller*
- Miyano and Hayashi: *alternating Büchi word* → *nondeterministic Büchi*
- not covered: Muller and Schupp *alternating Rabin tree* automaton → *nondeterministic Rabin tree* automaton

Translating Acceptance Modes

- Büchi, co-Büchi, parity → *parity, Rabin, Streett* (easy: special cases);
- Büchi, co-Büchi, Rabin, Streett, parity → *Muller* (easy but expensive);
- Muller → *parity*: latest appearence record.

Automata and Games

1. *Acceptance* game of nondeterministic/alternating *word/tree* automata,
2. *Emptiness* game of nondeterministic *word/tree* automata

Over 1-letter alphabet: emptiness game = acceptance game

Applications:

- language emptiness test
- complementation of alternating automata, tree automata

Determinacy

1. Reachability, Büchi, co-Büchi, parity games are *memoryless determined.*
2. Muller, Streett, Rabin games are *determined*, but not memoryless determined.

Corollary: memoryless runs suffice for alternating Büchi, co-Büchi, parity automata.

Logics

$$\text{LTL} \subsetneq \text{QPTL} \approx \text{S1S}$$

$$\text{CTL} \subsetneq \text{CTL}^* \subsetneq \text{S2S}$$

Theorem 4 *LTL, QPTL, S1S, CTL, CTL*, S2S are* decidable *logics.*

Formula satisfiable? → translate formula to automaton → check emptiness.

CTL model checking

Does a given transition system M satisfy an CTL formula Φ?

CTL formula Φ

↓

alternating Büchi tree automaton $\mathcal{A}_\varphi$

↓

Emptiness game for M and $\mathcal{A}'_\varphi$

↙ ↘

Player 0 wins: $M \vDash \varphi$ | Player 1 wins: $M \nvDash \varphi$

LTL model checking

Does a given transition system M satisfy an LTL formula φ?

LTL formula φ

↓

universal co-Büchi word automaton $\mathcal{A}_\varphi$
construct via
nondeterministic Büchi automaton for $\neg\varphi$

↓

Universal *tree* automaton $\mathcal{A}'_\varphi$

↓

Emptiness game for M and $\mathcal{A}'_\varphi$

↙ ↘

Player 0 wins: $M \vDash \varphi$ | Player 1 wins: $M \nvDash \varphi$

Alternative view on LTL model checking

Program P → Safety automaton $\mathcal{A}_P$

LTL specification φ → Negation $\neg\varphi$ → Alternating Büchi automaton $\mathcal{A}_{\neg\varphi}$ → Nondeterministic Büchi automaton $\mathcal{A}'_{\neg\varphi}$

↘ ↙

Intersection: nondeterministic Büchi automaton $\mathcal{A}_{P,\neg\varphi}$

↓

Empty?

↙ ↘

Yes: *P satisfies* φ | No: *P violates* φ