Automata, Games & Verification

Summary

Automata

1. Branching Mode
   deterministic – nondeterministic – universal – alternating

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

3. Input
   words – trees

Word Automata and Languages

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

$+$	recognizes ω-regular languages;
$-$	recognizes subset of ω-regular languages.

Characterization Theorems

Definition 1. *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

Theorem 4. [McNaughton's Theorem (1966)] *For every nondeterministic Büchi automaton $\mathcal{A}$, there is a deterministic Muller automaton $\mathcal{A}'$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.*

Theorem 5. [Miyano and Hayashi, 1984] *For every alternating Büchi automaton $\mathcal{A}$, there exists a nondeterministic Büchi automaton $\mathcal{A}'$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.*

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;
- Rabin, Streett → parity: index appearence record.

Automata and Games

1. Acceptance game of alternating word automata

2. Acceptance game of nondeterministic tree automata

3. Emptiness game of nondeterministic tree automata

Over 1-letter alphabet: emptiness game = acceptance game

$$\begin{array}{rl} & \text{emptiness game of NTA} \\ = & \text{acceptance game of 1-letter NTA} \\ = & \text{acceptance game of 1-letter AWA} \end{array}$$

Determinacy

1. Reachability, Büchi, co-Büchi, parity games are memoryless determined.

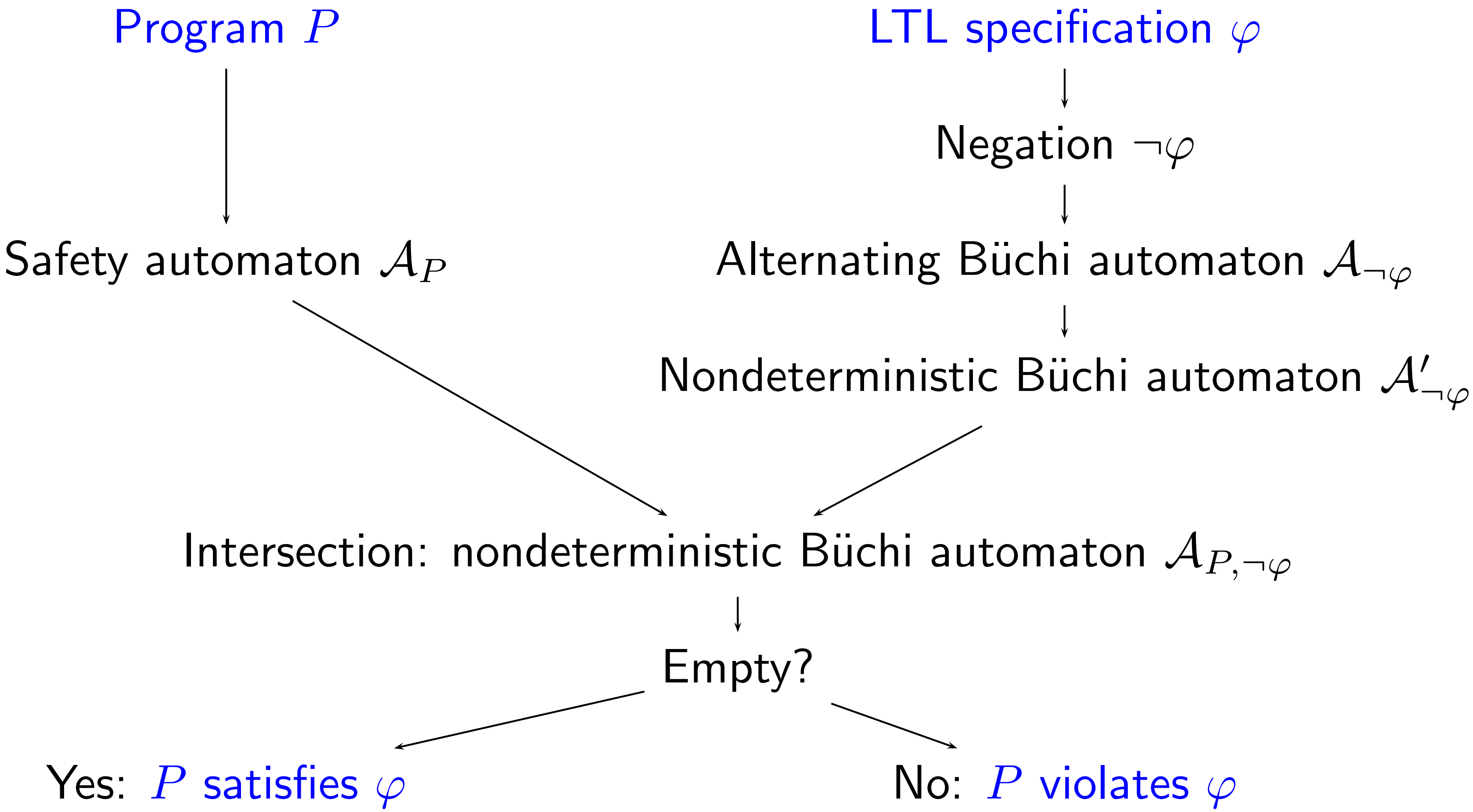
2. Muller, Streett, Rabin games are determined, but not memoryless determined.

Logics

$$\text{LTL} \subsetneq \text{QPTL} \approx \text{S1S} \approx \text{WS1S} \subsetneq \text{S2S}$$

Theorem 6. *LTL, QPTL, S1S, WS1S, QPTL, S2S are decidable logics.*

Verification



Program P

LTL specification φ

Negation $\neg\varphi$

Safety automaton $\mathcal{A}_P$

Alternating Büchi automaton $\mathcal{A}_{\neg\varphi}$

Nondeterministic Büchi automaton $\mathcal{A}'_{\neg\varphi}$

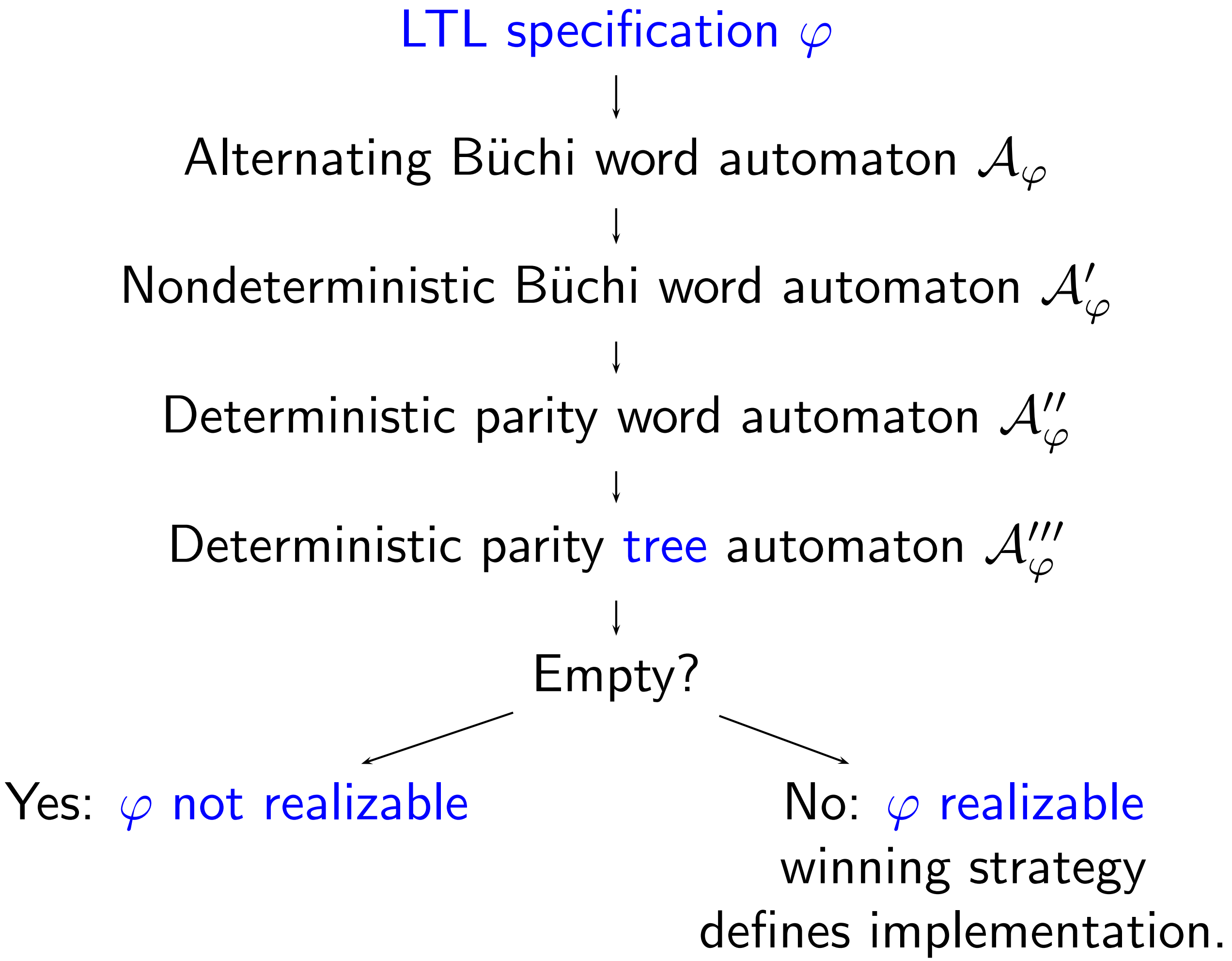
Intersection: nondeterministic Büchi automaton $\mathcal{A}_{P,\neg\varphi}$

Empty?

Yes: P satisfies φ

No: P violates φ

Synthesis



LTL specification φ

Alternating Büchi word automaton $\mathcal{A}_\varphi$

Nondeterministic Büchi word automaton $\mathcal{A}'_\varphi$

Deterministic parity word automaton $\mathcal{A}''_\varphi$

Deterministic parity tree automaton $\mathcal{A}'''_\varphi$

Empty?

Yes: φ not realizable

No: φ realizable
winning strategy
defines implementation.

Advertisement: Friday, July 25, 2008, 16:30
Sven Schewe, Ph.D. defense **Synthesis of Distributed Systems**

This thesis offers a comprehensive solution of the distributed synthesis problem. It starts with the problem of solving Parity games, which form an integral part of the automata-theoretic synthesis algorithms we use. We improve the known complexity bound for solving parity games with n positions and c colors approximately from $O(n^{\frac{1}{2}c})$ to $O(n^{\frac{1}{3}c})$, and introduce an accelerated strategy improvement technique that can consider all combinations of local improvements in every update step, selecting the globally optimal combination. We then demonstrate the decidability and finite model property of alternating-time specification languages, and determine the complexity of the satisfiability and synthesis problem for the alternating-time μ-calculus and the temporal logic ATL*. The impact of the architecture, that is, the set of system processes with known (white-box) and unknown (black-box) implementation, and the communication structure between them, is determined. We introduce information forks, a simple but comprehensive criterion that characterizes all architectures for which the synthesis problem is undecidable. The information fork criterion takes the impact of nondeterminism, the communication topology, and the specification language into account. For decidable architectures, we present an automata-based synthesis algorithm. We introduce bounded synthesis, which deviates from general synthesis by considering only implementations up to a predefined size, and thus avoids the expensive representation of all solutions. We develop a SAT based approach to bounded synthesis, which is nondeterministic quasilinear in the minimal implementation instead of nonelementary in the system specification. We determine the complexity of open synthesis under the assumption of probabilistic or reactive environments. Our automata based approach allows for a seamless integration of the new environment models into the uniform synthesis algorithm. Finally, we study the synthesis problem for asynchronous systems. We show that distributed synthesis remains only decidable for architectures with a single black-box process, and determine the complexity of the synthesis problem for different scheduler types. Furthermore, we combine the undecidability results and synthesis procedures for synchronous and asynchronous systems; systems that are globally asynchronous and locally synchronous are decidable if all black-box components are contained in a single fork-free synchronized component.