The Büchi Complementation Saga

Moshe Y. Vardi

Rice University

Logical Implication

Logical Implication:

- *Given*: Assumption ψ, Assertion φ.
- *Implication*: $\psi \models \varphi$
- *Semantics*: $models(\psi) \subseteq models(\varphi)$
- *Unsatisfiability*: $\psi \wedge \neg\varphi$ is unsatisfiable.

Important: ability to complement assertions

Design Verification

Design Verification:

- *Given*: Implementation ψ, Specification φ.
- *Conformance*: $\psi \models \varphi$
- *Semantics*: $models(\psi) \subseteq models(\varphi)$
- *Emptiness*: $\psi \cap \overline{\varphi}$ is empty.

Important: ability to complement specifications

Complementation:

- Easy for logic
- Hard for automata

Warm-Up: Automata on Finite Words

NFA: $A = (\Sigma, S, S_0, \rho, F)$

- Σ: finite alphabet
- S: finite state set
- $S_0 \subseteq S$: initial state set
- $\rho : S \times \Sigma \rightarrow 2^S$: transition function
- $F \subseteq S$: accepting state set

Input: $w = a_0, \ldots, a_{n-1}$
Run: $r = s_0, \ldots, s_{n-1}$
- $s_0 \in S_0$
- $s_{i+1} \in \rho(s_i, a_i)$

Acceptance: $s_n \in F$

Idea:

- Automata describe sets of finite traces.
- Nondeterminism abstracts unknown or complex parts of system.

Complementation I

Run Forest of A on w:

- Roots: elements of S_0.
- Children of s at level i: elements of $\rho(s, a_i)$.
- Rejection: no leaf is accepting.

Key Observation: collapse forest into a DAG – at most one copy of a state at a level; width of DAG is $|S|$.

Subset Construction Rabin-Scott, 1959:

- $A^c = (\Sigma, 2^S, \{S_0\}, \rho^c, F^c)$
- $F^c = \{T : T \cap F = \emptyset\}$
- $\rho^c(T, a) = \bigcup_{t \in T} \rho(t, a)$
- $L(A^c) = \Sigma^* - L(A)$

Complementation Blow-Up

$A = (\Sigma, S, S_0, \rho, F)$, $|S| = n$
$A^c = (\Sigma, 2^S, \{S_0\}, \rho^c, F^c)$

Blow-Up: 2^n upper bound

Can we do better?

Lower Bound: 2^n
Sakoda-Sipser 1978, Birget 1993

$L_n = (0+1)^*1(0+1)^{n-1}0(0+1)^*$
- L_n is easy for NFA
- $\overline{L_n}$ is hard for NFA

From Finite to Infinite Computations

Historical Perspective

- 1960–1975: programs need to terminate
  - proving termination
- 1975–present: some programs need not terminate
  - Operating systems
  - Device drivers
  - Routers
  - Microprocessors

Example: a modern microprocessor runs on the average at least 10^{15} cycles.

Automata on Infinite Words I

Looping automaton: $A = (\Sigma, S, S_0, \rho)$

- Σ: finite alphabet
- S: finite state set
- $S_0 \subseteq S$: initial state set
- $\rho : S \times \Sigma \to 2^S$: transition function

Input: $w = a_0, a_1 \ldots$
Run: $r = s_0, s_1 \ldots$
- $s_0 \in S_0$
- $s_{i+1} \in \rho(s_i, a_i)$

Acceptance: $\exists$ infinite run

Motivation: simplest model for infinite computations.

Complementation II

Infinite Run Forest of A on w:

- Roots: elements of S_0.
- Children of s at level i: elements of $\rho(s, a_i)$.
- Rejection: no infinite path, i.e., forest is finite (by König's lemma)

Key Observation: collapse forest into a DAG – at most one copy of a state at a level; width of DAG is $|S|$.

Subset Construction

- $A^c = (\Sigma, 2^S, \{S_0\}, \rho^c, \emptyset)$
- $\rho^c(T, a) = \bigcup_{t \in T} \rho(t, a)$
- Note: A^c is an automaton on finite words!
- *Blow-up*: 2^n - tight!

Automata on Infinite Words II

Büchi Automaton: $A = (\Sigma, S, S_0, \rho, F)$

- Σ: finite alphabet
- S: finite state set
- $S_0 \subseteq S$: initial state set
- $\rho : S \times \Sigma \rightarrow 2^S$: transition function
- $F \subseteq S$: accepting state set

Input: $w = a_0, a_1 \ldots$
Run: $r = s_0, s_1 \ldots$
- $s_0 \in S_0$
- $s_{i+1} \in \rho(s_i, a_i)$

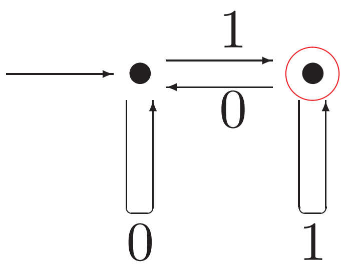
Acceptance: run visits F *infinitely often*.

Motivation:

- characterizes ω-*regular* languages
- equally expressive as S1S (Büchi 1962)
- more expressive than Linear Temporal Logic
- looping automaton with fairness

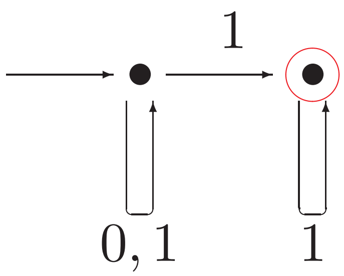
Examples

$((0+1)^*1)^\omega$:



– infinitely many 1's

$(0+1)^*1^\omega$:



– finitely many 0's

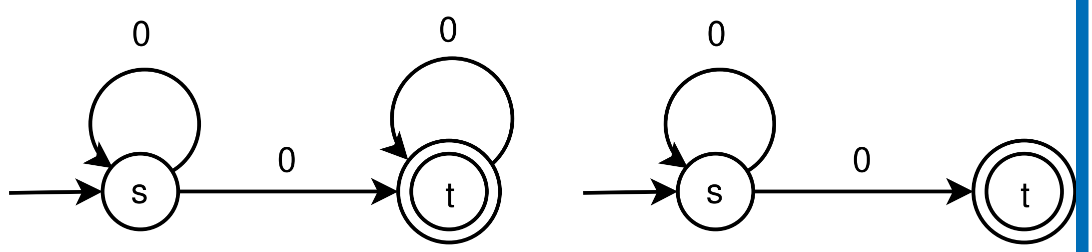
Complementation III

Infinite Run Forest of A on w:

- Roots: elements of S_0.
- Children of s at level i: elements of $\rho(s, a_i)$.
- Rejection: no path visits F infinitely often.

Key Observation: collapse forest into a DAG – at most one copy of a state at a level; width of DAG is $|S|$.

Problem: subset construction fails!



Büchi 1962

Motivation: decidability of S1S.
- automata for atomic formulas of S1S
- closure under intersection and projection
- *closure under complementation*
- decidability of nonemptiness problem

Büchi Complementation:
$A = (\Sigma, S, S_0, \rho, F)$, $w = a_0, a_1, \ldots$

- Characterize $w[i, j] = a_i, \ldots, a_{j-1}$ with

– A has a run from s to t on $w[i, j]$.

– A has a run from s to t, including a visit to F, on $w[i, j]$.

- Finite number of such characterizations – "*colors*"

- Apply Infinite Ramsey Theorem to the set $\{(i, j) : i < j\}$ – an infinite mono-chromatic set.

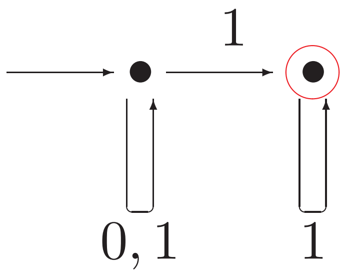
- Use above to construct a *nondeterministic* Büchi automaton that complement A.

Sistla-V.-Wolper 1985: Construction can be implemented with blow-up 16^{n^2}.

Determinism

Determinism: $|S_0| = 1$, $|\rho(s,a)| = 1$

$(0+1)^*1^\omega$:



– finitely many 0's

Fact: $(0{+}1)^*1^\omega$ cannot be defined by a deterministic Büchi automaton.

Muller 1963: Consider more general acceptance condition

- $\lim(r)$: states visited infinitely often in run r.
- *Muller condition*: $G \subseteq 2^S$ (generalizes Büchi condition)
- *Acceptance*: $\lim(r) \in G$

Example: Acceptance condition for $(0+1)^*1^\omega$

$$G = \{right\}$$

Determinization

Complementation: Deterministic Muller automata are closed under complementation – just complement acceptance condition

> McNaughton 1966: Every Muller automaton is equivalent to a deterministic Muller automaton.

Question: What is the blow-up?

Answer

- McNaughton 1985: "Lots and lots"
- Careful analysis: $2^{2^{O(n)}}$ (more than $2^{O(n^2)}$)

Safra 1989: An improved construction with $n^{O(n)}$ blow-up.

Corollary: $n^{O(n)}$ complementation construction.

Lower Bound

Easy: 2^n – just as on finite words

Question: Can we show a stronger lower bound?

The language of infinite paths:

- $\Sigma = \{1, \ldots, n\}$
- $w = a_0, a_1, \ldots$
- *Infinite path in* w: $a_{i_0}, a_{i_0+1}, a_{i_1}, a_{i_1+1}, \ldots$ such $a_{i_j+1} = a_{i_{j+1}}$ (i.e., a subword of matching pairs)
- $L_n = \{w\ :\ \text{there is an infinite path in } w\}$
- $\overline{L_n} = \Sigma^\omega - L_n$

Michel 1988:
- L_n can be defined using a Büchi automaton with n states.
- $\overline{L_n}$ cannot be defined using a Büchi automaton with less that $n! = n^{\Omega(n)}$ states.

Bottom Line: Büchi complementation is harder than NFA complementation.

Is the Problem Solved?

No!

- Safra's construction impractical
  - Subset construction does not blow up in practice.
  - Safra's construction does!

- Significant gap between $n^{O(n)}$ upper bound and $n!$ lower bound (contrast with tight 2^n in the finite case).

Goal

- Practical construction

- Tight Bounds

Klarlund, 1991

- Complementation without determinization, simpler than Safra
- Upper bound of $n^{O(n)}$
- Unclear practicality

Transition Systems

Fair Transition System: $M = (W, W_0, R, F)$

- W: state set (not necessarily finite)
- $W_0 \subseteq W$: initial state set
- $R \subseteq W^2$: transition relation
- $F \subseteq W$: fair state set

Fair path: infinite path in M that visits F infinitely often.

Fair termination: no fair path in M from W_0
- Checking livelock can be reduced to fair termination.
- Model checking LTL properties can be reduced to fair termination.

Fair-Termination Checking

$M = (W, W_0, R, F)$

Definition: Let $X, Y \subseteq W$. $Until(X, Y)$ is the set of states in X that can properly reach Y, while staying in X.

EC'80: characterization of fair termination

$Q \leftarrow W$
while change do
$\quad Q \leftarrow Q \cap Until(Q, Q \cap F)$
endwhile
return $(I \cap Q = \emptyset)$

Intuition: Repeatedly delete states that cannot be on a fair path because they cannot properly reach F even once.

Note: Fixpoint may require transfinite stages to converge, when W is infinite.

- *Rank of a state*: number of stage in which it was deleted from Q.

Complementation IV

Büchi Automaton: $A = (\Sigma, S, S_0, \rho, F)$

Infinite Run Forest of A on A:

- Roots: elements of S_0.
- Children of s at level i: elements of $\rho(s, a_i)$.
- Collapse forest into a DAG – at most one copy of a state at a level; width of DAG is $|S|$.
- Rejection: no path visits F infinitely often – *fair Termination*

Kupferman-V. 1997: run DAG is a fairly terminating transition system.

Termination and Complementation:
- Rank every node in the DAG forest.
- Every level is now a ranked subset of S.
- Apply *ranked* subset construction.

Problem: Ranks are transfinite.

Back to Termination

Looping system: $M = (W, W_0, R, W)$

Termination: No infinite path in M from W_0.

Definition: Let $X \subseteq W$. $Pre(X)$ is the set of states that can reach X in one step.

Fixpoint Characterization (Rubin, 1975): used by symbolic model checkers to eliminate finite paths.

$Q \leftarrow W$
while change do
    $Q \leftarrow Q \cap Pre(Q)$
endwhile
return $(I \cap Q = \emptyset)$

Intuition: Repeatedly delete dead-end states.

Fair Termination Revisited

$M = (W, W_0, R, F)$

Fair Termination: No infinite path in M from W_0 visits F infinitely often.

Fisler et al. 2001: OWCTY

```
Q ← W
while change do
        while change do
                Q ← Q ∩ Pre(Q)
        endwhile
        Q ← Q ∩ Until(Q, Q ∩ F)
endwhile
return (I ∩ Q = ∅)
```

Intuition: Dead-end states cannot lie on a fair path.

Complementation V

Büchi Automaton: $A = (\Sigma, S, S_0, \rho, F)$, $|S| = n$

Infinite Run Forest of A on A:

- Roots: elements of S_0.
- Children of s at level i: elements of $\rho(s, a_i)$.
- Collapse forest into a DAG – at most one copy of a state at a level; width of DAG is n.
- Rejection: no path visits F infinitely often – *fair Termination*

Kupferman-V. 1997: run DAG is a fairly terminating transition system.

- OWCTY is applicable to run DAG.
- Every node is deleted in one of the external iterations – use iteration number as rank.
- Only n external iterations – each external iteration reduces width by at least one.

Ranks

OWCTY:

> $Q \leftarrow W$
> while change do
>   while change do
>   (1) $Q \leftarrow Q \cap Pre(Q)$
>   endwhile
>   (2) $Q \leftarrow Q \cap Until(Q, Q \cap F)$
> endwhile
> return $(I \cap Q = \emptyset)$

Acceptance: $Q = \emptyset$

Ranks:

- $rank(x) = 2i$: x deleted at (1) in iteration i.
- $rank(x) = 2i + 1$: x deleted at (2) in iteration i.

Maximum Rank: $2n$ - *optimal*.

Termination and Complementation:

- Rank every node in the DAG forest.
- Every level is now a ranked subset of S.
- Apply *ranked* subset construction.

Ranks and Complementation

$A = (\Sigma, S, S_0, \rho, F)$, $|S| = n$

- *Rank Assignment*: mapping $r : S \rightarrow \{0, \ldots, 2n\}$
- $\alpha = \{0, \ldots, 2n\}^S$: all rank assignments.
- $A^c = (\Sigma, 2^S \times \alpha, \{S_0\} \times \alpha, \rho^c, F^c)$
- ρ^c:
  - State ranks cannot increase.
  - State ranks cannot stay even forever.
  - F-states cannot get odd ranks.
- *Consequence*: F cannot be visited infinitely often.
- **Complexity**: $n^{O(n)}$ states.

Practicality

2^n vs. n^n:

- $n = 5$: 32 vs. 3125
- $n = 10$: 1024 vs. $10,000,000,000$

Question: Is construction practical?

Gurumurthy et. al 2003: Yes, after many heuristic optimizations (tool: *Wring*)

Test:

- Generate random LTL formula φ
- compile φ and $\neg\varphi$ into complementary Büchi automata A_φ and $A_{\neg\varphi}$.
- Complement A_φ and compare its size to the size of $A_{\neg\varphi}$
- Complement A_φ twice and compare size to original size.

Results (across 1000 formulas):

- Ratio for single complementation: 1.09
- Ratio for double complementation: 1.13

Case Closed?

Not quite!

- *Upper Bound*: $(6n)^n$ blow-up
- *Lower Bound*: $(n/e)^n$ blow-up

Contrast: Tight 2^n for NFA vs. $(6/e)^n$ gap for Büchi automata.

Question: Have we really reached practicality?

Complementation of Random Automata

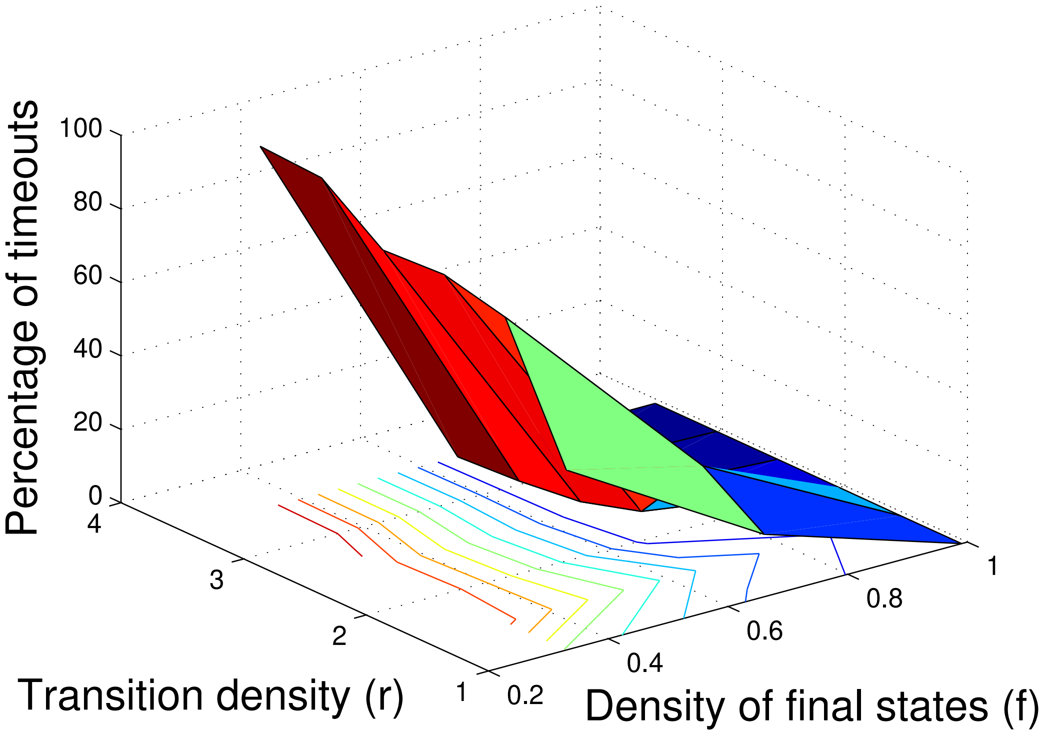


Figure 1: Wring timeouts, by D. Tabakov

Automata: Random automata with *six* states

Timeout: 3600 sec.

Tight Rank Assignments

Tight Rank Assignment: $r : S \to \{0, \ldots, 2n\}$
- $oddmax(r)$: maximal odd rank of r
- If $j < oddmax(r)$ is odd, then $j \in range(r)$

Lemma: Friedgut-Kupferman-V. 2004
It suffices to consider tight rank assignments in the construction.

- *Improve the construction*: Use tight rank assignments.
- *Improve the analysis*: Count tight rank assignments.

Combinatorics

$S(n, m)$: Stirling numbers of the second kind – number of ways of partitioning a set of n elements into m nonempty sets

Counting Functions: There are $m!S(n, m)$ functions from $\{1, \ldots, n\}$ *onto* $\{1, \ldots, m\}$.

Asymptotics:

- How many functions from $\{1, \ldots, n\}$ *into* $\{1, \ldots, \beta n\}$? $(\beta n)^n$
- How many functions from $\{1, \ldots, n\}$ *onto* $\{1, \ldots, \beta n\}$? $\left((\beta/(1-\beta))^{1-\beta} e^{-\beta} n\right)^n$ (Temme 1993)

Counting Tight Assignments: Choose $oddmax(r)$ to maximize possibilities.

- $tight(n) = (0.63n)^n$

Bottom Line: $(0.97n)^n$ blow-up
(Friedgut-Kupferman-V. 2004)

Numerical Analysis

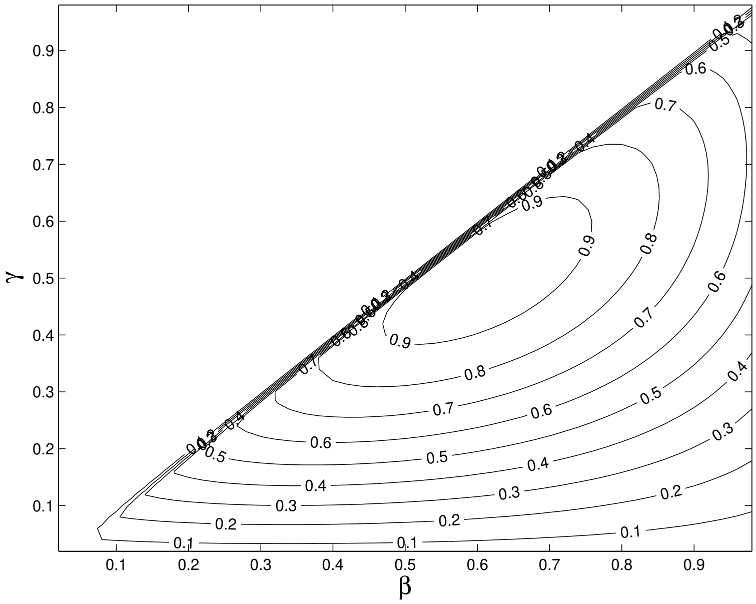


Figure 2: Contour lines of $h'(\beta, \gamma)$, for $0 \leq \gamma \leq \beta \leq 1$.

Discussion

- **Optimality**: $(cn)^n$

  - *Upper Bound*: $(0.97n)^n$
  - *Lower Bound*: $(0.76n)^n$ (Q. Yan 2005)

  Still: an exponential gap!

- **Practicality**

  – *Common perception*: Complementation is not practical.

  – *Challenge*: Make it practical, e.g., symbolic implementation.

Symbolic Complementation of Random Automata

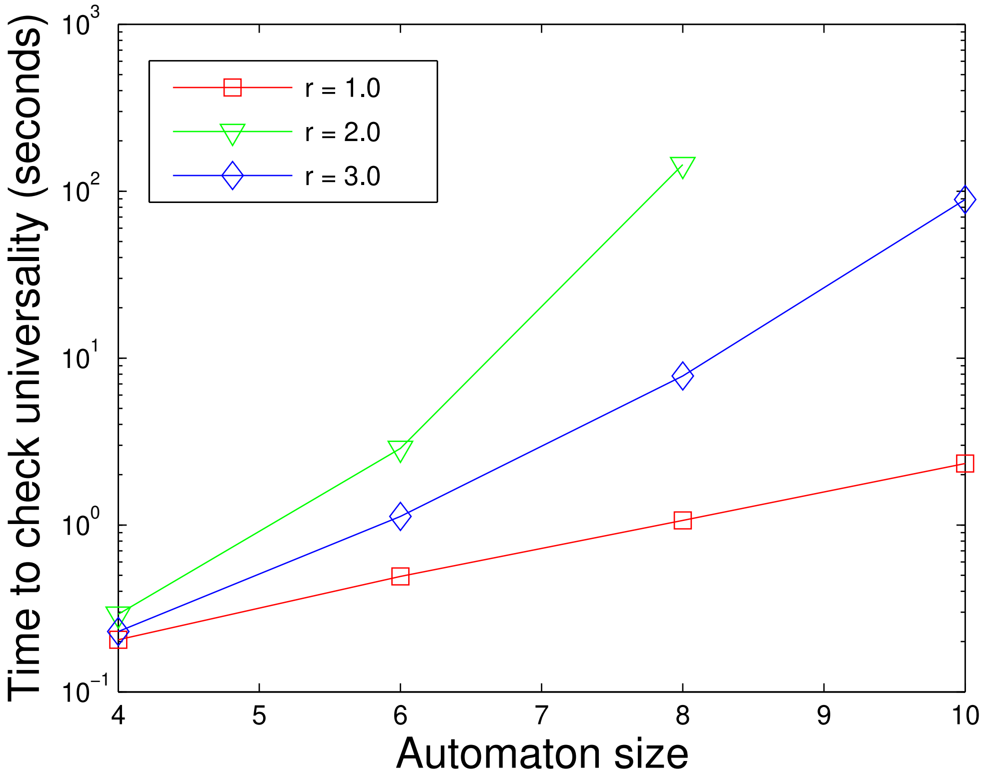


Figure 3: Symbolic complementation, by D. Tabakov

Also: L. Doyen and J.F. Raskin, *Improved Algorithms for the Automata-based Approach to Model Checking*. (TACAS'07): up to 30 states.

Contrast with NFA Complementation: 600+ states

Beyond Büchi Automata

Streett Automaton: $A = (\Sigma, S, S_0, \rho, \alpha)$

- $\alpha \subseteq 2^S \times 2^S$ – a set of pairs of sets of states
- *Index*: number of pairs
- *Acceptance*: For each pair $(L, R) \in \alpha$, if run visits L i.o., then run visits R i.o.
- *Intuition*: if a transition is enabled i.o., then it is scheduled i.o. – strong fairness.

Streett Complementation: $(nk)^{O(nk)}$
(Klarlund, 1991, Safra, 1992, Kupferman-V., 2005)

Open:

- Optimality (*lower bound* – $(kn)^{\Omega(n)}$ (Q. Yan, 2005))
- Practicality?

In Summary

Büchi Complementation:

- Classical problem
- 45 years of progress
- Tantalizing complexity gap
- Challenging practical problem