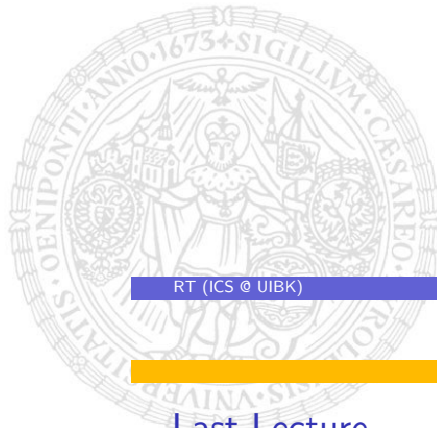


Introduction to Model Checking

René Thiemann

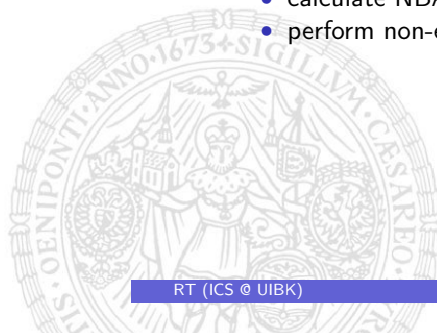
Institute of Computer Science
University of Innsbruck

WS 2007/2008



Last Lecture

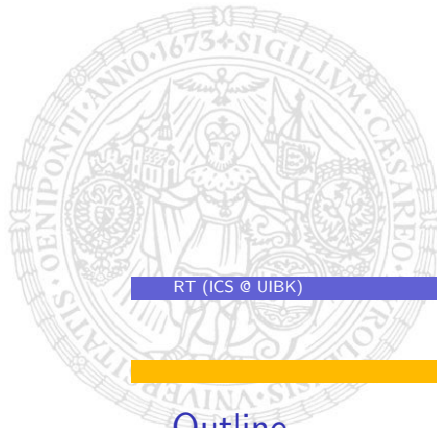
- System: $Traces(TS)$ for transition system $TS = (S, \rightarrow, I, AP, L)$
- Specification: $\mathcal{L}(\mathcal{A})$ for NBA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$
- Model checking: $TS \models \mathcal{A}$ iff $Traces(TS) \cap \mathcal{L}(\overline{\mathcal{A}}) = \emptyset$
 - calculate NBA $\overline{\mathcal{A}}$ with $\mathcal{L}(\overline{\mathcal{A}}) = \overline{\mathcal{L}(\mathcal{A})}$ (ignored)
 - calculate NBA \mathcal{A}' for intersection of $\overline{\mathcal{A}}$ and TS (todo)
 - perform non-emptiness test for $\mathcal{L}(\mathcal{A}')$ (done)



Language of an NBA

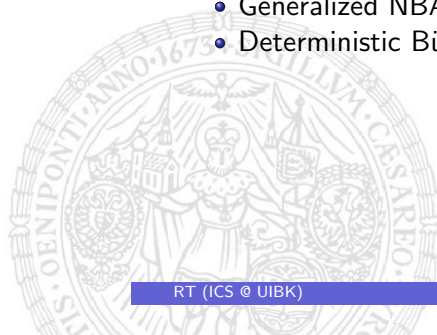
- NBA $\mathcal{A} = (\mathcal{Q}, \Sigma, \delta, q_0, F)$ and word $w = A_1 \dots A_n \dots \in \Sigma^\omega$
- Run $q_0 q_1 \dots q_n \dots$ is **accepting** if
for infinitely many indices $i: q_i \in F$
- The *accepted language* of \mathcal{A} :

$$\mathcal{L}(\mathcal{A}) = \{ w \in \Sigma^\omega \mid \text{there exists an accepting run for } w \text{ in } \mathcal{A} \}$$



Outline

- Intersection of TS with NBAs
- Variants of NBAs
 - Generalized NBAs
 - Deterministic Büchi Automata

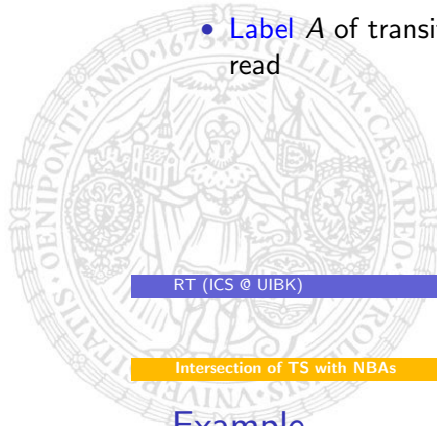


Intersecting *Traces*(TS) and $\mathcal{L}(\mathcal{A})$

Given: $TS = (S, \rightarrow, I, AP, L)$ and $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$.

Idea: Construct a new NBA accepting the intersection of both languages.

- Use **Cartesian product** of states of TS and of \mathcal{A} as new set of states
- **Additional initial state** needed, since TS may have several initial states, whereas NBA only allows one
- **Label** A of transition system state **corresponds to upcoming letter** to read



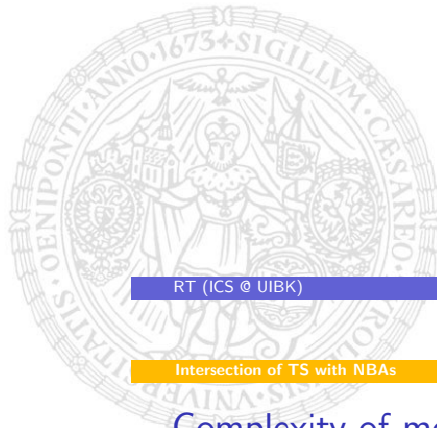
Example



Intersecting *Traces*(TS) and $\mathcal{L}(\mathcal{A})$

Theorem

$$\mathcal{L}(TS \otimes \mathcal{A}) = \text{Traces}(TS) \cap \mathcal{L}(\mathcal{A})$$



Complexity of model checking

$TS = (S, \rightarrow, I, AP, L)$ and $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$. Let $n := |S|$, $m := |Q|$.

Model checking: $\text{Traces}(TS) \cap \mathcal{L}(\overline{\mathcal{A}}) = \emptyset$

- calculate NBA $\overline{\mathcal{A}}$ with $\mathcal{L}(\overline{\mathcal{A}}) = \overline{\mathcal{L}(\mathcal{A})}$

$\Rightarrow \overline{\mathcal{A}}$ has $2^{\mathcal{O}(m \cdot \log(m))}$ states ($m!$, $m^m \in 2^{\mathcal{O}(m \cdot \log(m))}$)

- calculate NBA \mathcal{A}' for intersection of $\overline{\mathcal{A}}$ and TS

$\Rightarrow \mathcal{A}'$ has $n \cdot 2^{\mathcal{O}(m \cdot \log(m))} + 1$ states

- perform non-emptiness test for $\mathcal{L}(\mathcal{A}')$

\Rightarrow needs $\mathcal{O}(n \cdot 2^{\mathcal{O}(m \cdot \log(m))})$ time

\Rightarrow Model checking with NBA's is **linear in the system size** and **exponential in the specification size**

(Problematic factor: n due to state space explosion problem, specification often small)

\Rightarrow If NBA $\overline{\mathcal{A}}$ instead of \mathcal{A} is given then model checking is linear in both system and specification.

Extending NBAs to Generalized NBAs

A **generalized NBA (GNBA)** \mathcal{A} is a tuple $(Q, \Sigma, \delta, q_0, F_1, \dots, F_k)$ where

- Q, Σ, δ, q_0 are as before
- $F_1, \dots, F_k \subseteq Q$ are several sets of final states

A run $q_0 q_1 q_2 \dots$ is **accepting** iff every F_i is visited infinitely often

Example

$\mathcal{A} = (\{q_A, q_B\}, \{A, B\}, \delta, q_A, \{q_A\}, \{q_B\})$ with $\delta(q_L, L) = \{q_L\}$

$$\mathcal{L}(\mathcal{A}) = \{w \mid \text{both } A \text{ and } B \text{ occur infinitely often in } w\}$$

It turns out that NBAs are as expressive as GNBA

Transforming GNBA to NBA

Given: GNBA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F_1, \dots, F_k)$

Problem: Taking $\bigcap_{i=1}^k F_i$ as final states is too restrictive,
visit of different F_i at different times possible

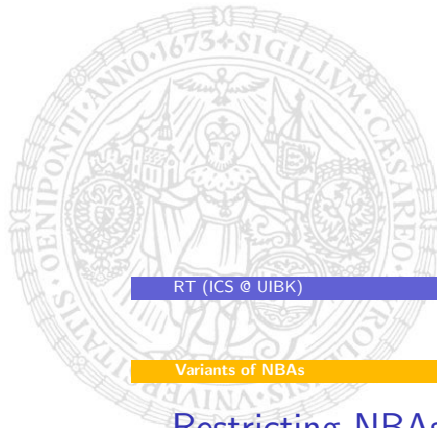
Idea for transformation:

- Wait for final states F_i **one after another**:
 $F_1, F_2, \dots, F_k, F_1, F_2, \dots, F_k, F_1, \dots$
- Use k **copies of \mathcal{A}** and store which F_i has to be visited next
- Only mark required visits to F_1 as “final”

Result: NBA $\mathcal{A}' = (Q \times \{1, \dots, k\}, \Sigma, \delta', (q_0, 1), F_1 \times \{1\})$ where

$$\delta'((q, i), A) = \begin{cases} \delta(q, A) \times \{i\} & \text{if } q \notin F_i \\ \delta(q, A) \times \{i+1\} & \text{if } q \in F_i, i < k \\ \delta(q, A) \times \{1\} & \text{if } q \in F_i, i = k \end{cases}$$

Example



Restricting NBAs to DBAs

Deterministic automata: $(Q, \Sigma, \delta, q_0, F)$ with $\delta : Q \times \Sigma \rightarrow Q$

For finite words non-determinism is not required:

Theorem (Powerset construction)

For every NFA there is an equivalent deterministic finite automaton (DFA).

Question: How about NBAs and DBAs?

Lemma

The language $\mathcal{L} = \{w \mid w \text{ only contains finitely many } A\text{'s}\}$ is recognized by some NBA, but not by some DBA.

Corollary

NBAs are strictly more expressive than DBAs.

Proof of Lemma

Suppose $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ would recognize \mathcal{L} . Since $w_0 = B^\omega \in \mathcal{L}$ there must be an accepting run $q_0 q_1 q_2 \dots$ on w_0 . Hence, there is some k_0 with $q_{k_0} \in F$. Then choose $w_1 = B^{k_0} A B^\omega \in \mathcal{L}$. Due to determinism, the accepting run starts in the same way, and hence looks like $q_0 \dots q_{k_0} q'_{k_0+1} q'_{k_0+2} \dots$ where again for some k_1 the state $q'_{k_0+1+k_1} \in F$.

Continuing in this way we will get a word $w_n = B^{k_0} A B^{k_1} A B^{k_2} A \dots B^{k_n} A B^\omega \in \mathcal{L}$ where the accepting run will visit a final state before every A . For $n = |F|$ this shows that in this run some final state $q \in F$ is visited more than once, say for indices i and j with $i < j$. Thus, starting from q with the word $w = B^{k_{i+1}} A \dots B^{k_j} A$ one ends again in q . Hence, iterating w ad infinitum after some finite prefix which will lead to q (so, the complete word is $B^{k_0} A \dots B^{k_i} A w^\omega$) yields an accepting run. This is a contradiction, since w contains at least one A and hence, w^ω has infinitely many A 's. ■

Summary

- Intersection of transition system and NBAs results in new NBA (linear complexity)
- $\text{NFA} \equiv \text{DFA}$, $\text{GNBA} \equiv \text{NBA} \sqsupset \text{DBA}$
- Closure properties:

	\cap	\cup	$\bar{}$
DFA	✓	✓	✓
NFA	✓	✓	✓
DBA	✓	✓	–
NBA	✓	✓	✓
GNBA	✓	✓	✓

Exercises

- Give a construction from two NBAs \mathcal{A}_1 and \mathcal{A}_2 to a new NBA which recognizes $\mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$.
Hint: Only give a construction which results in a GNBA. This suffices since every GNBA can then be transformed into an NBA.
- Apply your algorithm on the following two NBAs, and then use the non-emptiness check from the last lecture to see, whether there is a word which is accepted by both NBAs.

