

Tree Automata

Kumar Neeraj Verma

Institut für Informatik, TU München

Summer Semester 2004

- Lectures: Monday 15-17
- Exercises: Tuesday 16-17
- Course materials: lecture slides, literature

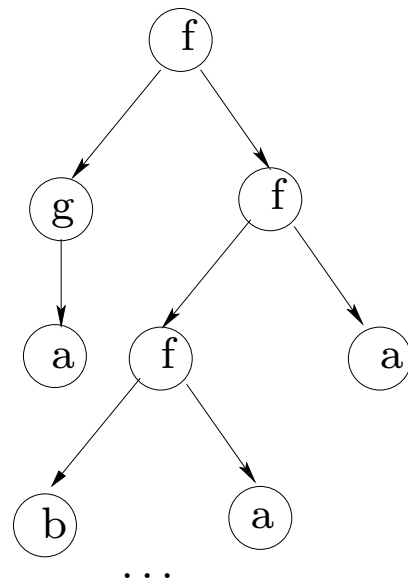
An Automaton is a means for describing a language of tree-like structures.

Words

Finite: **abbacaaab**

Infinite: **abbabbabbabb...**

Trees



A language is a set of such words/trees.

Examples

- **lexical analyzers**: work with finite words

Input program is a sequence of characters

```
begin print "hello world" end
```

Returns a sequence of tokens:

```
BEGIN PRINT STRING (...) END
```

- Pattern matching programs like **grep**

```
grep "a(bc)*ab" file
```

Search for patterns of the form `abcbcbc...bcab` in `file`.

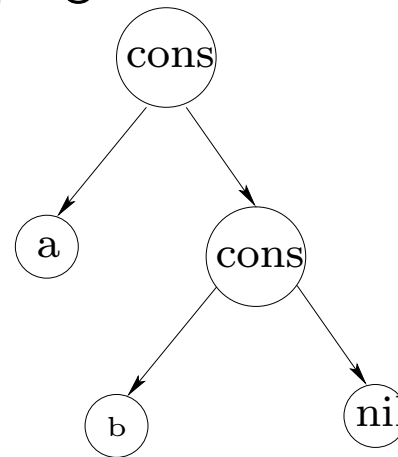
Both based on automata theoretic techniques.

Program analysis

Compute partial information about run-time behavior of programs

Use trees to represent values occurring in programs.

List `cons(a, cons(b, nil)) =`



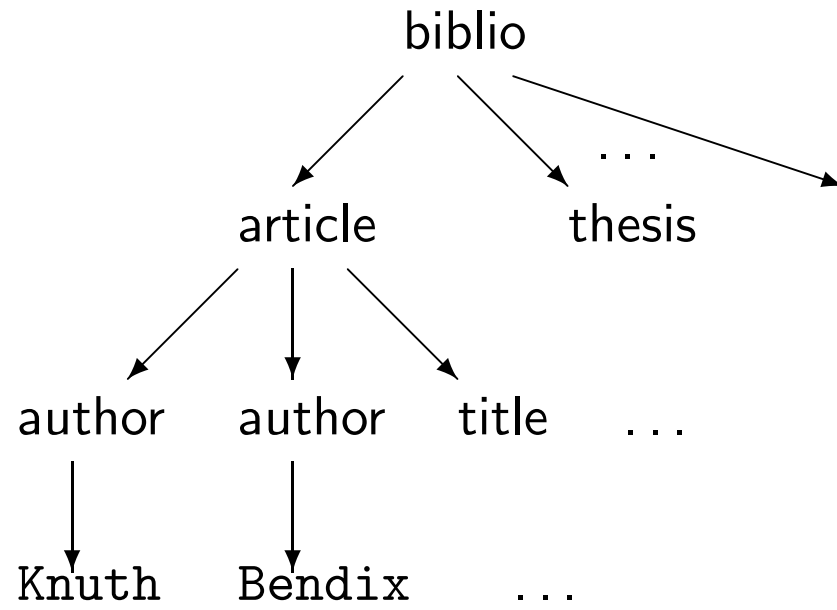
Example query: does variable `X` always take the same value at program point `p` ?

Document processing

A sample XML document:

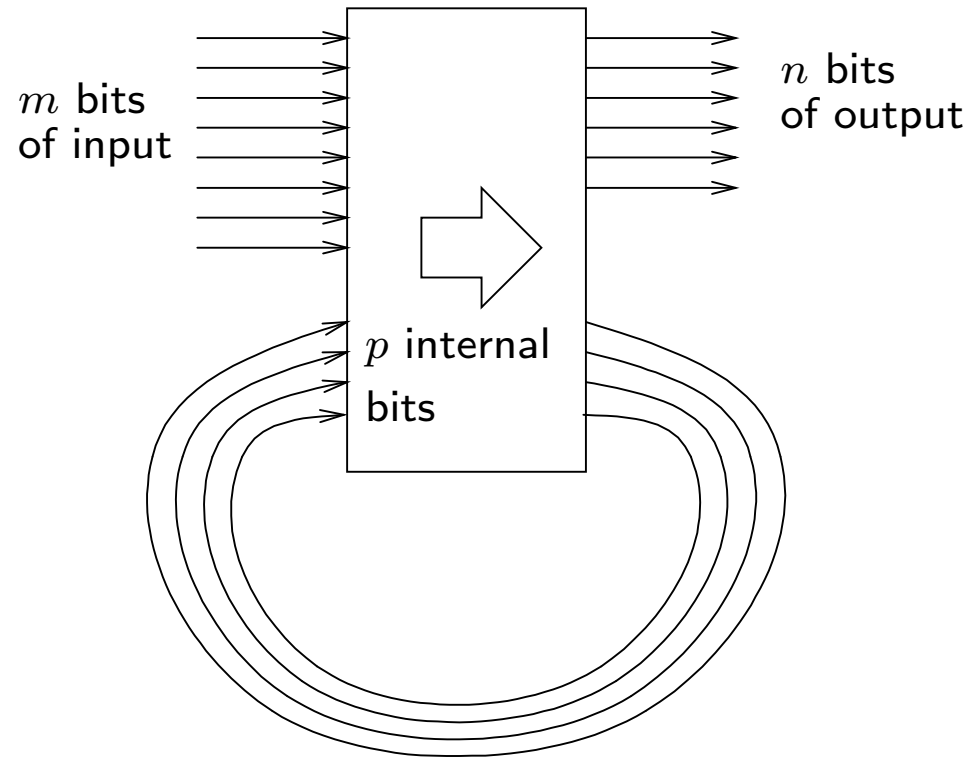
```
<biblio>
<article>
<author>Knuth</author>
<author>Benedix</author>
<title> ... </title>
</article>
<thesis>
</thesis>
...
</biblio>
```

As a tree:



Automata on infinite words and trees for verification

Finite state program:



Examples: hardware, communication protocols

If there are a total of N bits, the machine has 2^N possible states/configurations.

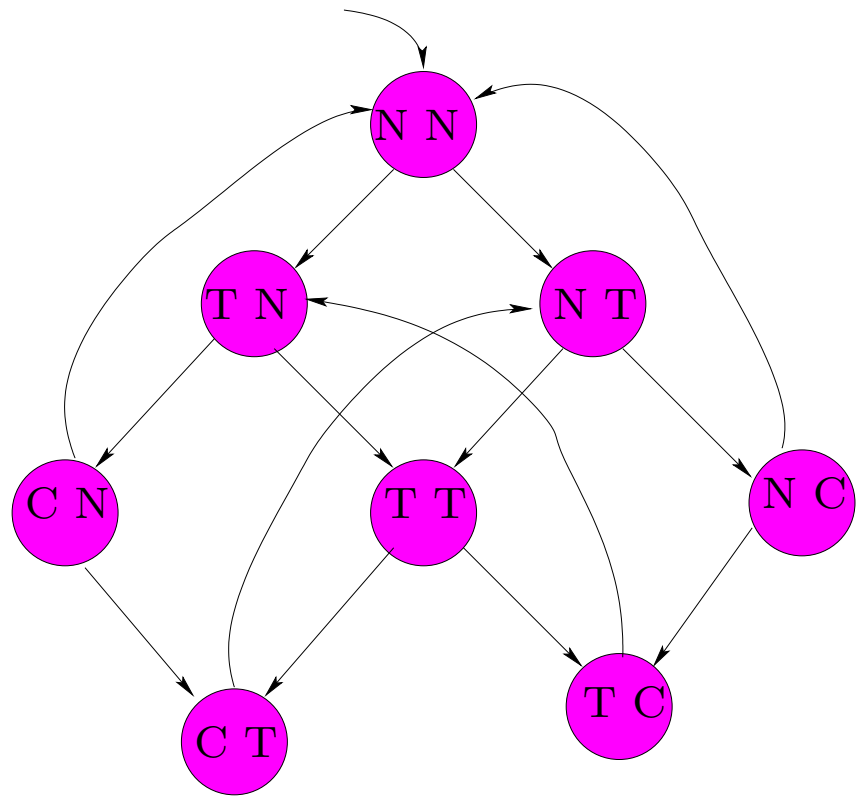
Executions are infinite.

We use temporal logics to specify desired properties of these sequences.

e.g.

- property ϕ eventually holds
- a bad state is never reached
- event x is eventually followed by event y

Mutual exclusion algorithm



N : non-critical

T : trying

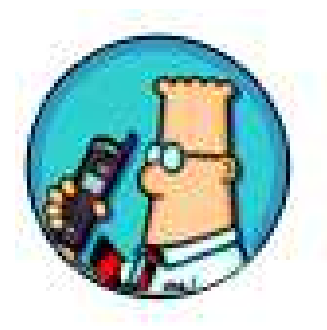
C : critical

- Both processes are never in the critical state together ?
- No process keeps trying infinitely long ?

Security protocols



Alice



Bob

Security protocols



Alice

Secret



Bob

Security protocols

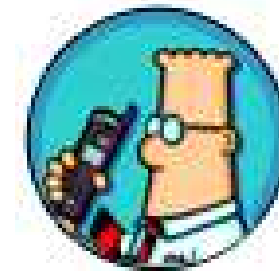


Alice

Secret



Charlie



Bob

Charlie can spy on the conversation!

Security protocols

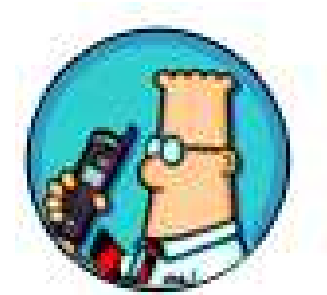


Alice

Secret



Charlie



Bob

Charlie can spy on the conversation!

Messages = terms built from data and cryptographic primitives

Use tree automata to represent the knowledge of intruder

Planned course contents

1. Automata on finite words and trees
2. Verification for temporal logics
 - ⇒ Automata on infinite words and linear temporal logics
 - ⇒ Automata on infinite trees and branching temporal logics
3. Monadic second order logics

4. Applications in compiler construction

⇒ Set constraints

⇒ Program analysis

⇒ Code selection

5. Application to verification of security protocols

6. Tree transducers and application to document processing