SLR 204: Basics of verification of distributed systems

Vadim Malvone
vadim.malvone@telecom-paris.fr
Formal verification of LTL properties
Decision Problems Using Automata: Model Checking

- Given an automaton $A_M$ for a system model $M$
- Given an automaton $A_{\neg \phi}$ accepting all models of the complement of a specification $\phi$

$M$ satisfies $\phi$ if and only if

$$L(A_M) \cap L(A_{\neg \phi}) = \emptyset$$
Decision Problems Using Automata: Satisfiability

- Given an automaton $A_\phi$ accepting all models of $\phi$

$\phi$ is satisfiable if and only if

$L(A_\phi) \neq \emptyset$
Automata-Theoretic approach

In order to use an automata-theoretic approach, we need to discuss:

- **Which kind of automata**
  - Branching mode: deterministic – nondeterministic – universal – alternating
  - Acceptance mode: Büchi – co-Büchi – parity – Streett – Rabin – Muller
  - Input: words – trees

- **How to check the (non-)emptiness problem**

- **How to implement model and specification translations**
Deterministic Finite Automata (DFA)

An DFA is a tuple $D = < Q, \Sigma, \delta, q_0, F >$

- $Q$ is the set of states;
- $\Sigma$ is the alphabet;
- $\delta : Q \times \Sigma \rightarrow Q$ is the transition function;
- $q_0 \in Q$ is the initial state;
- $F \subseteq Q$ is a set of accepting states.
Acceptance condition for DFA

- Given an automaton $D$ and a word $w = w_1, \ldots, w_n$, where $w_i \in \Sigma$.
- $w$ is accepted by $D$ if there exists a run $r = r_0, \ldots, r_n$ such that:
  - $r_0 = q_0$;
  - $r_{i+1} = \delta(r_i, w_{i+1})$, $\forall \ 0 \leq i < n$;
  - $r_n \in F$.

- For each word accepted, we have exactly one run.
- $r$ is called an accepting run.
- The language $L$ of $D$, denoted $L(D)$, is the set of all words accepted by $D$. 
Non-deterministic Finite Automata (NFA)

An NFA is a tuple $N = < Q, \Sigma, \delta, Q_0, F >$

- $Q$ is the set of states;
- $\Sigma$ is the alphabet;
- $\delta : Q \times \Sigma \rightarrow 2^Q$ is the transition function;
- $Q_0 \subseteq Q$ is the set of initial states;
- $F \subseteq Q$ is a set of accepting states.
Acceptance condition for NFA

- Given an automaton $N$ and a word $w = w_1, \ldots, w_n$, where $w_i \in \Sigma$.
- $w$ is accepted by $N$ if there exists a run $r = r_0, \ldots, r_n$ such that:
  - $r_0 \in Q_0$;
  - $r_{i+1} = \delta(r_i, w_{i+1}), \ \forall \ 0 \leq i < n$;
  - $r_n \in F$.

- $r$ is called an accepting run.
- The language $L$ of $N$, denoted $L(N)$, is the set of all words accepted by $N$. 
NFA Properties

NFA is closed under:

- union;
- intersection;
- complementation.
NFA Properties: Union

Given $N_1 = < Q_1, \Sigma, \delta_1, Q_{01}, F_1 >$ and $N_2 = < Q_2, \Sigma, \delta_2, Q_{02}, F_2 >$, there is a non-deterministic automaton $N$ accepting $L(N) = L(N_1) \cup L(N_2)$.

- $N = < Q, \Sigma, \delta, Q_0, F >$ runs non-deterministically either $N_1$ or $N_2$ on the input word:
  - $Q = Q_1 \cup Q_2$;
  - $Q_0 = Q_{01} \cup Q_{02}$;
  - $F = F_1 \cup F_2$;
  - $\delta(q,a) = \begin{cases} \delta_1(q,a) & \text{if } q \in Q_1 \\ \delta_2(q,a) & \text{if } q \in Q_2 \end{cases}$
**NFA Properties: Intersection**

- Given $N_1 = < Q_1, \Sigma, \delta_1, Q_{01}, F_1 >$ and $N_2 = < Q_2, \Sigma, \delta_2, Q_{02}, F_2 >$, there is a non-deterministic automaton $N$ accepting $L(N) = L(N_1) \cap L(N_2)$.

- $N = < Q, \Sigma, \delta, Q_0, F >$ runs simultaneously both automata $N_1$ and $N_2$ on the input word:
  - $Q = Q_1 \times Q_2$;
  - $Q_0 = Q_{01} \times Q_{02}$;
  - $F = F_1 \times F_2$;
  - $\delta((q_1,q_2),a) = \delta_1(q_1,a) \times \delta_2(q_2,a)$, where $q_1 \in Q_1$ and $q_2 \in Q_2$. 

NFA Properties: Complementation (I)

Solution:
1. Determinize the automaton using the subset construction.
2. Complement the resulting deterministic automaton.

Given an NFA $N = < Q, \Sigma, \delta, Q_0, F >$, we can construct an equivalent DFA $D = < Q', \Sigma, \delta', q_0, F' >$, where:
   - $Q' = 2^Q$;
   - $q_0 = Q_0$;
   - $F' = \{ q' \subseteq Q' \mid q' \cap F \neq \emptyset \}$;
   - $\delta'(q', a) = \bigcup_{q \in q'} \delta(q, a)$. 
NFA Properties: Complementation (II)

- Given a deterministic automaton $D = < Q, \Sigma, \delta, Q_0, F >$, the complement of $D$ is $D^c = < Q, \Sigma, \delta, Q_0, Q \setminus F >$.

- The complexity of the whole procedure is exponential in the size of the original automaton.

- In the worst case, the number of states of the final automaton is $2^{|Q|}$. 
Non-deterministic Büchi Automata (NBA)

- NBA extends classical finite automata in order to accept infinite words (also called ω-words).

- An NBA is a tuple $B = \langle Q, \Sigma, \delta, Q_0, F \rangle$
  - $Q$ is the set of states;
  - $\Sigma$ is the alphabet;
  - $\delta : Q \times \Sigma \to 2^Q$ is the transition function;
  - $Q_0 \subseteq Q$ is the set of initial states;
  - $F$ is an acceptance condition for infinite words.
Acceptance condition for NBA

- Given a run $r$, we define $\text{inf}(r) = \{s \mid s \text{ appears infinitely often on } r\}$.
- Given a Büchi automaton $B$ and an infinite word $w = w_1, w_2, w_3, \ldots$.
- $w$ is accepted by $B$ if there exists a run $r = r_0, r_1, r_2, \ldots$ such that:
  • $r_0 \in Q_0$;
  • $r_{i+1} = \delta(r_i, w_{i+1}), \ \forall \ i \geq 0$;
  • $\text{inf}(r) \cap F \neq \emptyset$.
- In other words, $w$ is accepted by $B$ if and only if there is a run $r$ of $B$ on $w$ visiting a final state $q \in F$ infinitely often.
- $r$ is called an accepting run.
- The language $L$ of $B$, denoted $L(B)$, is the set of all words accepted by $B$. 


Example

- \( L := \{ \alpha \in \{a, b\}^\omega | \alpha \text{ ends with } a^\omega \text{ or with } (ab)^\omega \} \)
NBA Properties

- NBA is closed under union, intersection, and complementation.

- NBA is not determinizable:
  - An example: \( \Sigma = \{a, b\} \) and \( L \) contains infinite words with a finite number of \( a \), i.e. \( L = (a+b)^* b^\omega \)
NBA Properties: Union

- Given Büchi automata $B_1 = < Q_1, \Sigma, \delta_1, Q_{01}, F_1 >$ and $B_2 = < Q_2, \Sigma, \delta_2, Q_{02}, F_2 >$, there is a Büchi automaton $B$ accepting $L(B) = L(B_1) \cup L(B_2)$.

- The construction is the same as for ordinary automata.

- $B = < Q, \Sigma, \delta, Q_0, F >$ is defined as follows:
  - $Q = Q_1 \cup Q_2$;
  - $Q_0 = Q_{01} \cup Q_{02}$;
  - $F = F_1 \cup F_2$;
  - $\delta(q,a) = \begin{cases} 
    \delta_1(q,a) & \text{if } q \in Q_1 \\
    \delta_2(q,a) & \text{if } q \in Q_2
  \end{cases}$
NBA Properties: Intersection (I)

- Given Büchi automata $B_1 = \langle Q_1, \Sigma, \delta_1, Q_{01}, F_1 \rangle$ and $B_2 = \langle Q_2, \Sigma, \delta_2, Q_{02}, F_2 \rangle$, there is a Büchi automaton $B$ accepting $L(B) = L(B_1) \cap L(B_2)$.
- The intersection construction for ordinary automata does not work for Büchi automata.
- $B = \langle Q, \Sigma, \delta, Q_0, F \rangle$ is defined as follows:
  - $Q = Q_1 \times Q_2 \times \{1,2\}$;
  - $Q_0 = Q_{01} \times Q_{02} \times \{1\}$;
  - $F = F_1 \times Q_2 \times \{1\}$;
  - $\delta((q_1, q_2, i), a) = \{(q, q', j) \mid q \in \delta_1(q_1, a), q' \in \delta_2(q_2, a)\}$,
    where $j = \begin{cases} 
      2 & \text{if } q_1 \in F_1 \text{ and } i = 1 \\
      1 & \text{if } q_2 \in F_2 \text{ and } i = 2 \\
      i & \text{otherwise}
    \end{cases}$
The automaton remembers two tracks, track1 and track2, one for each automaton, and at each step it points to one of the tracks.

As soon as it goes through an accepting state on the current track, it changes track.

The accepting condition and the transition function ensure that this change of track must happen infinitely often.

As soon as it visits an accepting state in track1, it switches to track2 and then to track1 again but only after visiting an accepting state in track2.

Therefore, to visit infinitely often a state in $F$ ($F_1$), the automaton must also visit infinitely often at least one state of $F_2$. 
NBA Properties: Complementation

- It is a complicated construction.
- The standard subset construction for determinizing automata does not work.
- Since non-deterministic automata are more powerful than deterministic ones (e.g. \( L = (a+b)^* b^\omega \)).

**Solution (use of another kind of automata):**
- Transform the non-deterministic Büchi automaton into a non-deterministic Rabin automaton (a more general automaton).
- Determinize and then complement the Rabin automaton.
- Transform the Rabin automaton into a Büchi automaton.

Therefore, Büchi automata are also closed under complementation.
Rabin automata

- A Rabin automaton is like a Büchi automaton, except that the accepting condition is defined differently.
- \( R = < Q, \Sigma, \delta, Q_0, F > \), where \( F = \{(G_1,B_1), \ldots, (G_m,B_m)\} \).
- The acceptance condition for a run \( r = r_0, r_1, r_2, \ldots \) is as follows:

\[
\inf(r) \cap G_i \neq \emptyset \text{ and } \inf(r) \cap B_i = \emptyset
\]

for some \( i \in \{1,\ldots,m\} \)

- In other words, there is a pair \( (G_i,B_i) \) such that the “good” set \( G_i \) is visited infinitely often, while the “bad” set \( B_i \) is visited only finitely many times.
Rabin versus Büchi automata

The Büchi automaton for $L = (a+b)^* b^\omega$

The Rabin automaton for $L = (a+b)^* b^\omega$

where $F = \{(G_1, B_1)\}$ with $G_1 = \{q_1\}$ and $B_1 = \{q_0\}$

the Rabin automaton is deterministic!
Emptiness Problem

- In finite automata, the non-emptiness check is reachability:
  - A final state is reachable from an initial state.
  - Complexity: linear time.

- The check for non-emptiness in NBA requires double state reachability:
  - A final state is reachable:
    - from an initial state, and
    - from itself (lasso).
  - Complexity: linear time.
Generalized Non-deterministic Büchi Automata (GNBA)

- Often, it is more convenient to consider a generalized Büchi condition.
- The set $F$ becomes a set of subset, i.e. $F = \{F_1, \ldots, F_k\}$.
- We require that for each $F_i \in F$ the Büchi condition is satisfied.

- A GNBA can be polynomially translated into a classical NBA.
  - Hint: in the translation, make the intersection of $k$ copies of the input automaton where each copy $i$ has $F_i$ as accepting states, for all $1 \leq i \leq k$.

- For LTL specification, model checking and satisfiability concern checking for NBA non-emptiness.
From Kripke structure to NBA

- Given a Kripke structure
  \[ M = (AP, S, S_0, R, \text{Lab}) \]

- We can build an equivalent NBA
  \[ B_M = < Q, \Sigma, \delta, Q_0, F > \]

where:
- \( Q = S; \)
- \( \Sigma = 2^{AP}; \)
- \( (s, a, t) \in \delta \) if and only if \( (s, t) \in R \) and \( a = \text{Lab}(s); \)
- \( Q_0 = S_0; \)
- \( F = S. \)
From LTL to NBA: basic concepts (I)

- First, we put $\varphi$ in normal form:
  - we only use boolean operators $\neg$, $\lor$, and $\land$;
  - but negations are “pushed” in front of the atomic propositions;
  - we only use temporal operators $X$, $U$, and $R$.

- For example:

  $$(p \lor q) \rightarrow F r$$

  $$\neg(p \lor q) \lor F r$$

  $$\neg(p \lor q) \lor (true \lor r)$$

  $$(\neg p \lor \neg q) \lor (true \lor r)$$

  $$(false \rightarrow (F p)) \rightarrow (true \lor r)$$

  $$(false \rightarrow (true \lor p)) \rightarrow (true \lor r)$$

  $$\neg(false \rightarrow (true \lor p)) \lor (true \lor r)$$

  $$(true \lor (false \rightarrow \neg p)) \lor (true \lor r)$$

  $$(true \lor (false \rightarrow \neg p)) \lor (true \lor r)$$

  $$(true \lor (false \rightarrow \neg p)) \lor (true \lor r)$$
From LTL to NBA: basic concepts (II)

- States of $A_\varphi$ will be “consistent” sets of sub-formulas of $\varphi$.
  - Given $\varphi = p \cup q$, a state is a subset of $\text{sub}(\varphi) = \{p, q, p \cup q\}$.

- Given a formula $\varphi$, $\text{sub}(\varphi)$ is the least set of formulas satisfying:

  - $\varphi \in \text{sub}(\varphi)$;
  - if $\neg \psi \in \text{sub}(\varphi)$ then $\psi \in \text{sub}(\varphi)$;
  - if $\psi_1 \lor \psi_2 \in \text{sub}(\varphi)$ then $\psi_1 \in \text{sub}(\varphi)$ and $\psi_2 \in \text{sub}(\varphi)$;
  - if $\psi_1 \land \psi_2 \in \text{sub}(\varphi)$ then $\psi_1 \in \text{sub}(\varphi)$ and $\psi_2 \in \text{sub}(\varphi)$;
  - if $X \psi \in \text{sub}(\varphi)$ then $\psi \in \text{sub}(\varphi)$;
  - if $\psi_1 \cup \psi_2 \in \text{sub}(\varphi)$ then $\psi_1 \in \text{sub}(\varphi)$ and $\psi_2 \in \text{sub}(\varphi)$;
  - if $\psi_1 R \psi_2 \in \text{sub}(\varphi)$ then $\psi_1 \in \text{sub}(\varphi)$ and $\psi_2 \in \text{sub}(\varphi)$. 
From LTL to NBA: basic concepts (III)

- Let $w = w_1, w_2, w_3, \ldots$ be a word modelling $\varphi$, i.e. $w \in L(\varphi)$.

- By means of a run, we associate to each $w_i$ a state $H_i$ whose subformulas are true:
  - $H_0$ contains $\varphi$;
  - if $\varphi = X\psi$, then $H_0$ contains $X\psi$ and $H_1$ contains $\psi$;
  - if $\varphi = \psi_1 U \psi_2$, then either $H_0$ contains $\psi_1$ and $H_1$ contains $\varphi$ or $H_0$ contains $\psi_2$;
  - if $\varphi = \psi_1 R \psi_2$ then $H_0$ contains $\psi_2$ and either $H_0$ contains $\psi_1$ or $H_1$ contains $\varphi$.

- $A_\varphi$ is built upon this consistency.
From LTL to NBA: details (I)

- We define $A_\varphi = (Q, \Sigma, \delta, Q_0, F)$ as follows.

- $Q = \{H \subseteq \text{sub}(\varphi) | H \text{ is locally consistent}\}$, i.e. $Q$ is the set of locally consistent sub-formulas of $\varphi$.

- A set $H$ is locally consistent when:
  - if $p \in H$ then $\neg p \notin H$;
  - if $\neg p \in H$ then $p \notin H$;
  - if $\psi_1 \lor \psi_2 \in H$ then $\psi_1 \in H$ or $\psi_2 \in H$;
  - if $\psi_1 \land \psi_2 \in H$ then $\psi_1 \in H$ and $\psi_2 \in H$;
  - if $\psi_1 U \psi_2 \in H$ then $\psi_2 \in H$ or $\psi_1 \in H$;
  - if $\psi_1 R \psi_2 \in H$ then $\psi_2 \in H$. 
From LTL to NBA: details (II)

- $\Sigma = 2^{AP}$.

- $Q_0 = \{ q \in Q \mid \varphi \in q \}$, i.e. we set as initial states those containing $\varphi$.

- Given $(s, \alpha, t) \in \delta$, the transition relation is defined as follows:
  - for all $p \in AP$, $p \in s$ if and only if $p \in \alpha$;
  - if $\varphi \in s$ then $\varphi \in t$;
  - if $\psi_1 U \psi_2 \in s$ then $\psi_2 \in s$ or ($\psi_1 \in s$ and $\psi_1 U \psi_2 \in t$);
  - if $\psi_1 R \psi_2 \in s$ then $\psi_2 \in s$ and ($\psi_1 \in s$ or $\psi_1 R \psi_2 \in t$).
From LTL to NBA: details (III)

- For each sub-formula $\psi_1 U \psi_2$, the transition relation ensures that $\psi_1 U \psi_2$ will appear again and again until the first occurrence of $\psi_2$.

- To avoid Until sub-formulas to remain unsatisfied, it is sufficient to ensure that, for each $\psi_1 U \psi_2$, the automaton goes infinitely often:
  - either through a state in which this does not appear, or
  - through a state in which both $\psi_1 U \psi_2$ and $\psi_2$ appear.

- Notice that if there are no $\psi_1 U \psi_2 \in \text{sub}(\varphi)$, then the acceptance condition is the trivial one: all states are accepting.
From LTL to NBA: details (IV)

- For each $\psi_1 U \psi_2 \in \text{sub}(\varphi)$, there is a set $F_i \in F$, such that:

  $$F_i = \{s \in Q \mid \psi_1 U \psi_2 \in s \text{ and } \psi_2 \in s \text{ or } \psi_1 U \psi_2 \not\in s\}$$

- The acceptance condition is a generalized Büchi one.
Given an LTL formula $\varphi$ we build an NBA $A_\varphi$ that accepts all words models of $\varphi$.

$\varphi = Xp$

- $Q = \{\emptyset, \{p\}, \{Xp\}, \{p, Xp\}\}$;
- $\Sigma = \{\emptyset, \{p\}\}$;
- $Q_0 = \{Xp\}, \{p, Xp\}\}$;
- $F = Q$. 
From LTL to NBA: Eventually operator

Given an LTL formula $\varphi$ we build an NBA $A_\varphi$ that accepts all words models of $\varphi$.

$\varphi = F \ p = true \ U \ p$

- $Q = \{\emptyset, \{p\}, \{true \ U \ p\}, \{p, true \ U \ p\}\}$;
- $\Sigma = \{\emptyset, \{p\}\}$;
- $Q_0 = \{\{true \ U \ p\}, \{p, true \ U \ p\}\}$;
- $F = \{\emptyset, \{p\}, \{p, true \ U \ p\}\}$. 

![State diagram](image_url)
LTL Model Checking (I)

- Theorem: given a model $M$ and an LTL formula $\varphi$
  \[ M \models \varphi \text{ iff } L(A_M) \cap L(A_{\neg \varphi}) = \emptyset \]

- The size of $A_M$ is linear in the size of $M$.

- Due to the subset operation over formulas, the size of $A_\varphi$ is exponential in the size of the formula (actually we build $A_{\neg \varphi}$).

- $A_\varphi$ is a GNBA that can be translated into an NBA in PTIME.
LTL Model Checking (II)

- The intersecting language $L(A_M) \cap L(A_{\neg \varphi})$ is an NBA and it can be performed in polynomial time.

- The non-emptiness of an NBA can be checked in linear time.

- The size of the NBA is polynomial in the size of $M$ and exponential in the size of $\varphi$. 
LTL Satisfiability

- Theorem: given an LTL formula $\phi$, there exists a model $M$ such that $M \models \phi$ iff $L(A_\phi) \neq \emptyset$.

- Due to the subset operation over formulas, the size of $A_\phi$ is exponential in the size of the formula.

- The non-emptiness of an NBA can be checked in linear time.

- LTL satisfiability is exponential in the size of $\phi$. 