From an LTL formula to a Büchi automaton

Step 1 of LTL model checking needs a Büchi automaton that accepts the words satisfying a formula φ. In the lecture's exercise sheet, we do not explain an algorithm to obtain such an automaton from φ. In this sheet, we address that limitation using Huth and Ryan's construction [1]. Each step is exemplified on $\mathrm{F}\, p$, and the exercises ask you to do it for $a \,\mathrm{U}\, b$. The construction is designed only for the until (U) and next (X) operators. So φ must first be rewritten so that its only temporal operators are those [1].

The idea. Let $w = \ell_0\ell_1\ell_2\ldots$ be a word. At each position i, each subformula of φ is either true or false on the suffix $\ell_i\ell_{i+1}\ldots$ A state of the automaton records a "guess" of these truth values at the current position. A run moves from guess to guess, and the construction only allows the guesses and moves that agree with the meaning of the operators.

1 The closure

The *closure* $C(\varphi)$ is the set of subformulas of φ and their negations.[1] For example,

$$C(\mathrm{F}\, p) = C(\top \,\mathrm{U}\, p) = \{\top,\ p,\ \mathrm{F}\, p,\ \neg\top,\ \neg p,\ \neg\,\mathrm{F}\, p\}.$$

Exercise 1 (The closure). Complete the closure of $a \,\mathrm{U}\, b$:

$$C(a \,\mathrm{U}\, b) = _____$$

2 The states

The states of the Büchi automaton are sets $\mathcal{S} \subseteq C(\varphi)$ that satisfy the following conditions [1]:

(C1) For each non-negated formula $\psi \in C(\varphi)$, exactly one of ψ and $\neg\psi$ is in $\mathcal{S}$.

(C2) $\mathcal{S}$ is *consistent with respect to propositional logic* for all formulas in $C(\varphi)$, i.e.

- $\top \in \mathcal{S}$,
- $\bot \notin \mathcal{S}$,
- $\psi_1 \wedge \psi_2 \in \mathcal{S}$ if and only if $\psi_1 \in \mathcal{S}$ and $\psi_2 \in \mathcal{S}$,
- $\psi_1 \vee \psi_2 \in \mathcal{S}$ if and only if $\psi_1 \in \mathcal{S}$ or $\psi_2 \in \mathcal{S}$.

(C3) If $\psi_1 \,\mathrm{U}\, \psi_2 \in \mathcal{S}$, then $\psi_1 \in \mathcal{S}$ or $\psi_2 \in \mathcal{S}$.

(C4) If $\neg(\psi_1 \,\mathrm{U}\, \psi_2) \in \mathcal{S}$, then $\neg\psi_2 \in \mathcal{S}$.

Conditions (C3) and (C4) for until, are inspired by the expansion laws:

$$\psi_1 \,\mathrm{U}\, \psi_2 \equiv \psi_2 \vee \big(\psi_1 \wedge \mathrm{X}(\psi_1 \,\mathrm{U}\, \psi_2)\big), \quad \text{and} \quad \neg(\psi_1 \,\mathrm{U}\, \psi_2) \equiv \neg\psi_2 \wedge \big(\neg\psi_1 \vee \mathrm{X}\,\neg(\psi_1 \,\mathrm{U}\, \psi_2)\big),$$

and speak about the current position only. There is no condition for X, because $\mathrm{X}\,\psi$ says nothing about the current position.

For the formula $\mathrm{F}\, p$, every state contains $\top$ by (C2), and by (C1) it contains one of $p, \neg p$ and one of $\mathrm{F}\, p, \neg\,\mathrm{F}\, p$. This gives four candidates:

[1] The double negation $\neg\neg\psi$ is identified with ψ [1]

Candidate state name	Candidate state content	Violated condition	Is a state
u_1	$\{\top,\ p,\ \mathrm{F}\,p\}$	none	(✓)
u_2	$\{\top,\ p,\ \neg\,\mathrm{F}\,p\}$	(C4): needs $\neg p$	( )
u_3	$\{\top,\ \neg p,\ \mathrm{F}\,p\}$	none	(✓)
u_4	$\{\top,\ \neg p,\ \neg\,\mathrm{F}\,p\}$	none	(✓)

Exercise 2 (The states).

Part 1 — *Checking the conditions.* For $a\ \mathrm{U}\ b$, a set that satisfies (C1) is fixed by its choices for a, b and $a\ \mathrm{U}\ b$. Complete the table: give the violated condition, if any, and tick the candidates that are states.

Candidate state name	Candidate state content	Violated condition	Is a state
c_1	$\{a,\ b,\ a\ \mathrm{U}\ b\}$	_____	( )
c_2	$\{a,\ b,\ \neg(a\ \mathrm{U}\ b)\}$	_____	( )
c_3	$\{a,\ \neg b,\ a\ \mathrm{U}\ b\}$	_____	( )
c_4	$\{a,\ \neg b,\ \neg(a\ \mathrm{U}\ b)\}$	_____	( )
c_5	$\{\neg a,\ b,\ a\ \mathrm{U}\ b\}$	_____	( )
c_6	$\{\neg a,\ b,\ \neg(a\ \mathrm{U}\ b)\}$	_____	( )
c_7	$\{\neg a,\ \neg b,\ a\ \mathrm{U}\ b\}$	_____	( )
c_8	$\{\neg a,\ \neg b,\ \neg(a\ \mathrm{U}\ b)\}$	_____	( )

3 The transitions

The transitions of the Büchi automaton follow the rules below. Each rule has a premise about a state $\mathcal{S}$ and a requirement on a state $\mathcal{S}'$. There is a transition from $\mathcal{S}$ to $\mathcal{S}'$ exactly when $\mathcal{S}'$ meets the requirement of every rule whose premise holds in $\mathcal{S}$ [1]. If no premise holds in $\mathcal{S}$, there is a transition from $\mathcal{S}$ to every state.

(T1) If $\mathrm{X}\,\psi \in \mathcal{S}$, then $\psi \in \mathcal{S}'$, and if $\neg\,\mathrm{X}\,\psi \in \mathcal{S}$, then $\neg\psi \in \mathcal{S}'$.

(T2) If $\psi_1\ \mathrm{U}\ \psi_2 \in \mathcal{S}$ and $\psi_2 \notin \mathcal{S}$, then $\psi_1\ \mathrm{U}\ \psi_2 \in \mathcal{S}'$.

(T3) If $\neg(\psi_1\ \mathrm{U}\ \psi_2) \in \mathcal{S}$ and $\psi_1 \in \mathcal{S}$, then $\neg(\psi_1\ \mathrm{U}\ \psi_2) \in \mathcal{S}'$.

Rule (T1) is for the next operator, while (T2) and (T3) are for the until operator: they are parts of the expansion laws above.

For $\mathrm{F}\,p = \top\ \mathrm{U}\ p$, rule (T1) never applies, because there is no X:

State	Premise of (T2)	Premise of (T3)	Successors
$u_1 = \{\top,\ p,\ \mathrm{F}\,p\}$	no: $p \in u_1$	no: $\neg\,\mathrm{F}\,p \notin u_1$	all: $u_1,\ u_3,\ u_4$
$u_3 = \{\top,\ \neg p,\ \mathrm{F}\,p\}$	yes	no: $\neg\,\mathrm{F}\,p \notin u_3$	those with $\mathrm{F}\,p$: $u_1,\ u_3$
$u_4 = \{\top,\ \neg p,\ \neg\,\mathrm{F}\,p\}$	no: $\mathrm{F}\,p \notin u_4$	yes, since $\top \in u_4$	those with $\neg\,\mathrm{F}\,p$: u_4

Exercise 3 (The transitions). Complete the table for the states of $a \,\mathrm{U}\, b$ from Exercise 2: for each state, decide whether the premises of (T2) and (T3) hold, and list its successors.

State	Premise of (T2)	Premise of (T3)	Successors
$c_1 = \{a,\ b,\ a \,\mathrm{U}\, b\}$			
$c_3 = \{a,\ \neg b,\ a \,\mathrm{U}\, b\}$			
$c_4 = \{a,\ \neg b,\ \neg(a \,\mathrm{U}\, b)\}$			
$c_5 = \{\neg a,\ b,\ a \,\mathrm{U}\, b\}$			
$c_8 = \{\neg a,\ \neg b,\ \neg(a \,\mathrm{U}\, b)\}$			

4 Initial and accept states

- The initial states are the states that contain φ.
- If $\psi_1 \,\mathrm{U}\, \psi_2$ is the only until formula in $C(\varphi)$, the accept states are the states that contain $\neg(\psi_1 \,\mathrm{U}\, \psi_2)$ or ψ_2 [1]:

$$F = \{\mathcal{S} \mid \neg(\psi_1 \,\mathrm{U}\, \psi_2) \in \mathcal{S} \text{ or } \psi_2 \in \mathcal{S}\}.$$

  The reason is that rule (T2) enables runs where $\psi_1 \,\mathrm{U}\, \psi_2$ occurs infinitely often but not ψ_2. Such a run should not be accepting.

Several U formulas. When $C(\varphi)$ contains k formulas $\chi_1 \,\mathrm{U}\, \psi_1, \ldots, \chi_k \,\mathrm{U}\, \psi_k$, there is one such set of states for each of them, and a run must visit each of the k sets infinitely often. These are fairness constraints $C = \{\neg(\chi_i \,\mathrm{U}\, \psi_i) \vee \psi_i \mid 1 \le i \le k\}$, so Step 4 of the model-checking sheet still applies: check $\exists_C \Box \top$ with the fair-SCC method. This is how NuSMV implements LTL model checking [1].

For $\mathrm{F}\, p$, the initial states are u_1 and u_3, and the accept states are u_1, which contains p, and u_4, which contains $\neg \mathrm{F}\, p$. Figure 1 (top) shows the result.

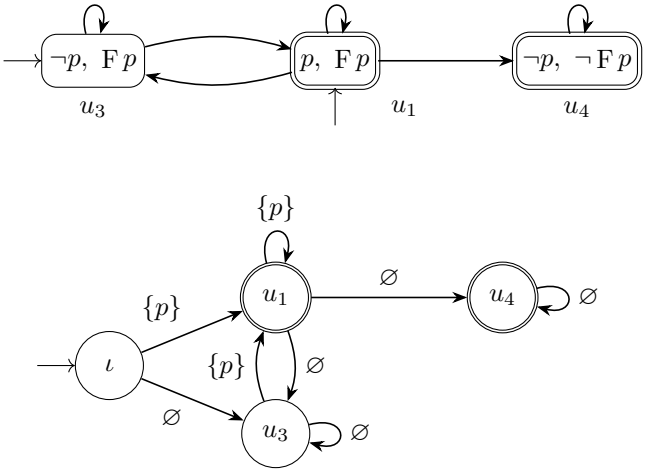


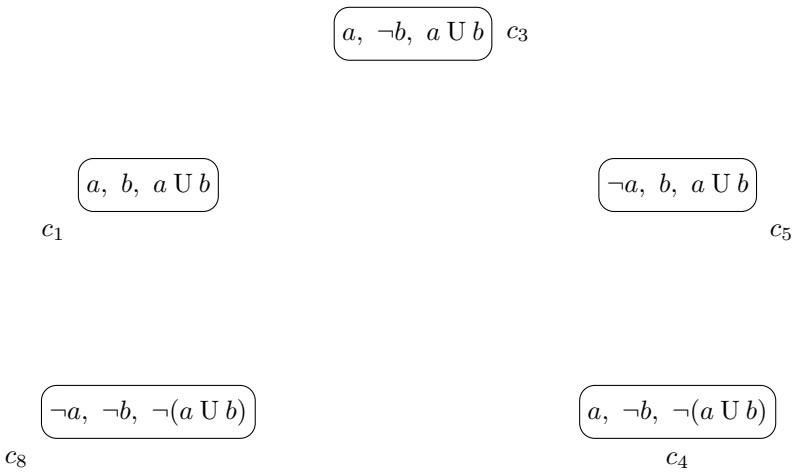
Figure 1: The automaton for $\mathrm{F}\, p$ (omitting $\top$). At the top, the result of the algorithm so far. At the bottom, the same automaton over $P = \{p\}$ with the letters on the edges (Section 6).

Exercise 4 (The automaton for $a \,\mathrm{U}\, b$).

Part 1 — *Initial and accept states.* Complete:

$$Q_0 = _____, \qquad F = _____.$$

Part 2 — *Drawing.* Complete the drawing below with the transitions of Exercise 3, arrows for the initial states and double frames for the accept states. You may draw one arrow with two heads for two transitions in opposite directions.



5 Words and runs

The edges of the constructed automaton have no letters. Instead, a run $\mathcal{S}_0\mathcal{S}_1\mathcal{S}_2\ldots$ reads the word $\ell_0\ell_1\ell_2\ldots$ if, for every i, ℓ_i is the set of atomic propositions in $\mathcal{S}_i$. As usual, the run must start in an initial state, and it is accepting if it visits accept states infinitely often.

For $\mathrm{F}\,p$ (Figure 1, top), the word $\varnothing\,\{p\}\,\varnothing^\omega$ has the accepting run $u_3\,u_1\,u_4^\omega$. The word $\varnothing^\omega$ is rejected: its run must start in u_3 and stay there, and u_3 is not an accept state.

Exercise 5 (Runs). For each word, give an accepting run of the automaton for $a\,\mathrm{U}\,b$ drawn in Exercise 4, or explain why there is none.

Word	Accepting run, or reason for rejection
$\{a\}\,\{a\}\,\{b\}\,\varnothing^\omega$	
$(\{a\}\,\{a,b\})^\omega$	
$\{b\}^\omega$	
$\{a\}^\omega$	
$\varnothing\,\{b\}^\omega$	

Which of these words satisfy $a\,\mathrm{U}\,b$? _____

6 Letters on the edges

The result of the algorithm so far resembles more an LTS (e.g. top of Figure 1) since the letters appear inside the states. Our model-checking sheet uses Büchi automata with the letters on the edges. To convert the LTS into a Büchi automaton, follow Step 2 of that sheet:

add a fresh initial state ι with an edge to each initial state, and assign to each edge the set of atomic propositions of where it enters. For $\mathrm{F}\, p$, this is the automaton (bottom) of Figure 1.

Exercise 6 (Letters on the edges). Write the sets read by the edges into each state of the automaton for $a \,\mathrm{U}\, b$, and the edges from ι.

Edges into	c_1	c_3	c_4	c_5	c_8
read	_____	_____	_____	_____	_____

The state ι has edges to _____.

7 Comparison and cost

Exercise 7 (Comparing with the hand-drawn automaton). Step 1 of the model-checking sheet uses the automaton for $a \,\mathrm{U}\, b$ in Figure 2.

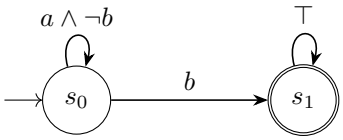


Figure 2: The hand-drawn automaton for $a \,\mathrm{U}\, b$.

Compare it with the automaton for $a \,\mathrm{U}\, b$ that you drew in Exercise 4. Mark the true statements with a checkmark (✓):

( ) Both automata accept exactly the words that satisfy $a \,\mathrm{U}\, b$.

( ) The state c_3 plays the role of s_0: it waits for b while reading $\{a\}$.

( ) The construction always gives an automaton with the fewest possible states.

( ) The constructed automaton is larger because it keeps guessing the truth of $a \,\mathrm{U}\, b$ at every position, also after the first b.

The cost of the construction. By (C1), a candidate state is fixed by choosing, for each formula of $C(\varphi)$ that is not a negation, either that formula or its negation. If there are n such formulas, there are 2^n candidates.

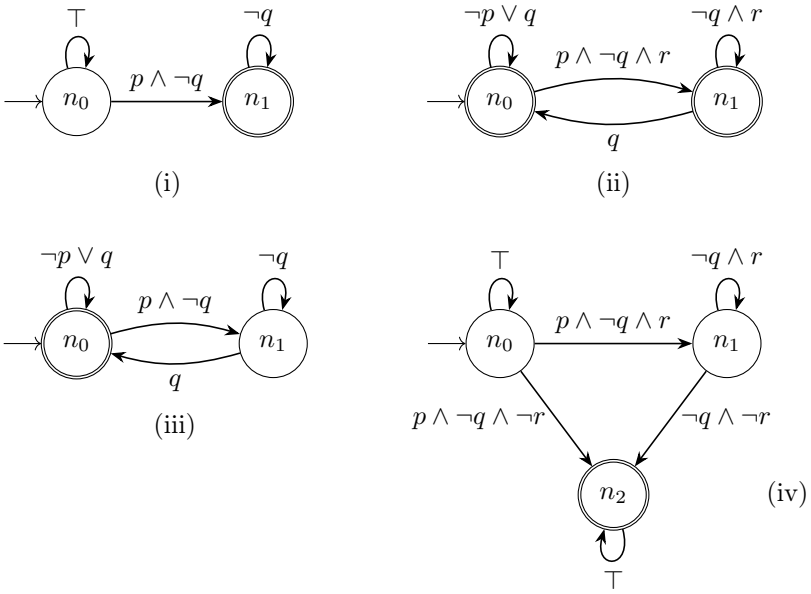
- For $a \,\mathrm{U}\, b$, the formulas that are not negations are a, b and $a \,\mathrm{U}\, b$. So $n = 3$, which gives $2^3 = 8$ candidates (c_1–c_8), of which 5 are states.
- For $\mathrm{F}\, p = \top \,\mathrm{U}\, p$, they are $\top$, p and $\mathrm{F}\, p$. So $n = 3$ as well, but (C2) forces $\top$, which leaves 4 candidates (u_1–u_4), of which 3 are states.

The number n grows with the length of φ, so the automaton can be exponentially larger than the formula. LTL model checking with this construction takes time linear in the size of the LTS and exponential in the length of the formula [2].

8 Extra exercises

A Büchi automaton *captures* an LTL formula φ if it accepts exactly the words that satisfy φ [2]. To keep drawings small, an edge labelled with a propositional formula ψ stands for one edge per letter ℓ that satisfies ψ.

Exercise 8 (Matching specification patterns with automata). The automata (i)–(iv) below read words over the atomic propositions $P = \{p, q, r\}$. Two formulas use the *weak until* $r \,\mathrm{W}\, q \equiv (r \,\mathrm{U}\, q) \vee \mathrm{G}\, r$ [2].



Write next to each formula the automaton (i)–(iv) that captures it. Each formula on the right is the negation of the formula on its left.

( )	$\mathrm{G}(p \to \mathrm{F}\, q)$	( )	$\mathrm{F}(p \wedge \mathrm{G}\, \neg q)$
( )	$\mathrm{G}(p \to r \,\mathrm{W}\, q)$	( )	$\mathrm{F}\big(p \wedge \neg(r \,\mathrm{W}\, q)\big)$

Exercise 9 (LTS or Büchi automaton?). Tick the column of each statement that holds for LTSs and for Büchi automata with the letters on the edges. Some statements hold for both. The first row is the example.

Statement	LTS	Büchi automaton
Each state is labelled with a set of atomic propositions.	(✓)	( )
Each transition is labelled with a letter of the alphabet.	( )	( )
A state can have several successors.	( )	( )
Every path or run can always continue.	( )	( )
Infinite runs are classified by the states they visit infinitely often.	( )	( )
A word needs only one accepting run.	( )	( )

References

[1] Michael Huth and Mark Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems*, 2nd edition. Cambridge University Press, 2004.

[2] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008.