Büchi Store: An Open Repository of Büchi Automata

Yih-Kuen Tsay, Ming-Hsien Tsai, Jinn-Shu Chang, and Yi-Wen Chang

Department of Information Management, National Taiwan University, Taiwan

Abstract. We introduce the Büchi Store, an open repository of Büchi automata for model checking practice, research, and education. The repository contains Büchi automata and their complements for common specification patterns and numerous temporal formulae. These automata are made as small as possible by various construction techniques, in view that smaller automata are easier to understand and often help in speeding up the model checking process. The repository is open, allowing the user to add new automata or smaller ones that are equivalent to some existing automaton. Such a collection of Büchi automata is also useful as benchmark cases for evaluating translation or complementation algorithms and as examples for teaching or learning Büchi automata and temporal logic.

1 Introduction

Büchi automata [1] are finite automata operating on infinite words. They play a fundamental role in the automata-theoretic approach to linear-time model checking [20]. In the approach, model checking boils down to testing the emptiness of an intersection automaton $A \cap B_{\neg\varphi}$, where A is a Büchi automaton modeling the system and $B_{\neg\varphi}$ is another Büchi automaton representing all behaviors not permitted by a temporal specification formula φ. In general, for a given system, the smaller $B_{\neg\varphi}$ is, the faster the model checking task may be completed.

To apply the automata-theoretic approach, an algorithm for translating a temporal formula into an equivalent Büchi automaton is essential. There has been a long line of research on such algorithms, aiming to produce smaller automata. According to our experiments, none of the proposed algorithms outperforms the others for every temporal formula tested. The table below shows a comparison of some of the algorithms for three selected temporal formulae.

Formula	LTL2AUT[4]		Couvreur[3]		LTL2BA[5]		LTL2Buchi[6]		SPIN[7]	
	state	tran.	state	tran.	state	tran.	state	tran.	state	tran.
$\neg p \, \mathcal{W} \, q$	4	16	3	12	3	12	3	12	4	16
$\Box(p \rightarrow \Diamond q)$	4	30	3	20	6	41	3	20	4	28
$\Diamond\Box p \vee \Diamond\Box q$	8	38	5	28	5	28	5	28	3	12

Given that smaller automata usually expedite the model checking process, it is certainly desirable that one is always guaranteed to get the smallest possible automaton for (the negation of) a specification formula. One way to provide the guarantee is to try all algorithms or even manual creation and take the best result. This simple-minded technique turns out to be feasible, as most specifications use formulae of the same patterns and the tedious work of trying all alternatives needs only be done once for a particular pattern.

To give the specification as a temporal formula sometimes may not be practical, if not impossible (using quantification over propositions). When the specification is given directly as an automaton, taking the complement of the specification automaton becomes necessary. Consequently, in parallel with the research on translation algorithms, there has also been substantial research on algorithms for Büchi complementation. The aim again is to produce smaller automata.

Several Büchi complementation algorithms have been proposed that achieve the lower bound of $2^{\Omega(n \log n)}$ [11]. However, the performances of these "optimal" algorithms differ from case to case, sometimes quite dramatically. The table below shows a comparison of some of the complementation algorithms for four selected Büchi automata (identified in the table by equivalent temporal formulae). In the literature, evaluations of these algorithms usually stop at a theoretical-analysis level, partly due to the lack of or inaccessibility to actual implementations. This may be remedied if a suitable set of benchmark cases becomes available.

Formula	Safra[13]		Piterman[12]		Rank-Based[9, 14]		Slice-Based[8]	
	state	tran.	state	tran.	state	tran.	state	tran.
$\Box(p \to \Diamond(q \wedge \Diamond r))$	76	662	90	777	96	917	219	2836
$\Diamond\Box(\diamondminus p \to q)$	35	188	13	62	13	72	24	119
$\Box(p \to p\,\mathcal{U}\,(q\,\mathcal{U}\,r))$	17	192	8	76	7	54	7	49
$p\,\mathcal{U}\,q \vee p\,\mathcal{U}\,r$	5	34	5	34	8	23	3	12

The Büchi Store was motivated by the above considerations. It is implemented as a website, accessible at `http://buchi.im.ntu.edu.tw`. One advantage for the Store to be on the Web is that the user always gets the most recent collection of automata. Another, even more important, advantage is that it is easily made open for the user to contribute better (smaller) automata. The initial collection at the time of writing contains over six hundred Büchi automata. In the following sections we describe its implementation and main features, suggest several use cases, and then conclude by highlighting directions for improvement.

2 Implementation and Main Features

The basic client-server interactions in accessing the Büchi Store are realized by customizing the CodeIgniter [2], which is an open source Web application framework. To perform automata and temporal formulae related operations, such as equivalence checking and formula to automaton translation, the Store relies on the GOAL tool [19] and its recent extensions. One particularly important (and highly nontrivial) task is the classification of temporal formulae that identify the Büchi automata in the Store into the Temporal Hierarchy of Manna and

Pnueli [10]. To carry out the task automatically, we implemented the classification algorithm described in the same paper, which is based on characterization of a Streett automaton equivalent to the temporal formula being classified.

The main features of the current Büchi Store include:

- **Search**: Every automaton in the Store is identified by a temporal formula (more specifically in QPTL [15, 16]) that specifies the language of the automaton. The user may find the automata that accept a particular language by posing a query with an equivalent temporal formula. Propositions are automatically renamed to increase matches. This is like asking for a translation from the temporal formula into an equivalent Büchi automaton. A big difference is that the answer automata, if any, are the best among the results obtained from a large number of translation algorithms, enhanced with various optimization techniques such as simplification by simulation [17] or even having been manually optimized (and machine-checked for correctness).
- **Browse**: The user may browse the entire collection of Büchi automata by having the collection sorted according to temporal formula length, number of states, class in the Temporal Hierarchy, or class in the Specification Patterns [18]. While classification in the Temporal Hierarchy has been automated, the classification for the last sorting option has not. Rather, the Store relies on the user to provide suggestions, based on which a final classification could be made. This may be useful for educational purposes.
- **Upload**: The user may upload an automaton with a temporal logic specification. The automaton is checked for correctness, i.e., if it is indeed equivalent to the accompanying temporal formula. If it is correct and smaller than the automata for the formula in the Store, the repository is updated accordingly, keeping only the three smallest automata.

3 Use Cases

We describe three cases that we expect to represent typical usages of the Store.

- **Linear-time model checking**: The user may shop in the Store for the automata that are equivalent (with probable propositions renaming) to the negations of the temporal formulae which he wants to verify. The automata may be downloaded in the PROMELA format, for model checking using SPIN.
- **Benchmark cases for complementation or translation algorithms**: Every Büchi automaton in the initial collection has a complement, which is reasonably well optimized. A subset of the collection could serve as a set of benchmark cases for evaluating Büchi complementation algorithms. This use case can certainly be adapted for evaluating algorithms that translate temporal formulae into Büchi automata.
- **Classification of temporal formulae**: The look of a temporal formula may not tell immediately to which class it belongs in the Temporal Hierarchy. It should be educational to practice on the cases that do not involve the sophistication of going through Streett automata. For example, $\Box p \lor \Box q$ is

a safety formula because it is equivalent to $\Box(\boxminus p \vee \boxminus q)$ (where $\boxminus$ means "so-far" or "always in the past") in the canonical safety form.

Concluding Remarks To further improve the Store, first of all, we as the developers will continue to expand the collection, besides hoping for the user to do the same. Explanatory descriptions other than temporal formulae should be helpful additions for searching and understanding. Automatic classification of temporal formulae into the various specification patterns should also be useful.

References

1. J.R. Büchi. On a decision method in restricted second-order arithmetic. In *Int'l Congress on Logic, Methodology and Philosophy of Science*, pages 1–11, 1962.
2. CodeIgniter. http://codeigniter.com/.
3. J.M. Couvreur. On-the-fly verification of linear temporal logic. In *World Congress on Formal Methods, LNCS 1708*, pages 253–271. Springer, 1999.
4. M. Daniele, F. Giunchiglia, and M.Y. Vardi. Improved automata generation for linear temporal logic. In *CAV, LNCS 1633*, pages 249–260. Springer, 1999.
5. P. Gastin and D. Oddoux. Fast LTL to Büchi automata translation. In *CAV, LNCS 2102*, pages 53–65. Springer, 2001.
6. D. Giannakopoulou and F. Lerda. From states to transitions: Improving translation of LTL formulae to Büchi automata. In *FORTE, LNCS 2529*, pages 308–326. Springer, 2002.
7. G.J. Holzmann. *The SPIN Model Checker: Primer and Reference Manual*. Addison-Wesley, 2003.
8. D. Kähler and T. Wilke. Complementation, disambiguation, and determinization of Büchi automata unified. In *ICALP, LNCS 5125*, pages 724–735. Springer, 2008.
9. O. Kupferman and M.Y. Vardi. Weak alternating automata are not that weak. *ACM Transactions on Computational Logic*, 2(3):408–429, 2001.
10. Z. Manna and A. Pnueli. A hierarchy of temporal properties. In *PODC*, pages 377–408. ACM, 1990.
11. M. Michel. Complementation is more difficult with automata on infinite words. In *CNET, Paris*, 1988.
12. N. Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. In *LICS*, pages 255–264. IEEE, 2006.
13. S. Safra. On the complexity of ω-automta. In *FOCS*, pages 319–327. IEEE, 1988.
14. S. Schewe. Büchi complementation made tight. In *STACS*, pages 661–672, 2009.
15. A.P. Sistla. *Theoretical Issues in the Design and Verification of Distributed Systems*. PhD thesis, Harvard, 1983.
16. A.P. Sistla, M.Y. Vardi, and P. Wolper. The complementation problem for Büchi automata with applications to temporal logic. *TCS*, 49:217–237, 1987.
17. F. Somenzi and R. Bloem. Efficient Büchi automata from LTL formulae. In *CAV, LNCS 1855*, pages 248–263. Springer, 2000.
18. The Spec Patterns repository. http://patterns.projects.cis.ksu.edu/.
19. Y.-K. Tsay, Y.-F. Chen, M.-H. Tsai, W.-C. Chan, and C.-J. Luo. GOAL extended: Towards a research tool for omega-automata and temporal logic. In *TACAS, LNCS 4963*, pages 346–350. Springer, 2008.
20. M.Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *LICS*, pages 332–344. IEEE, 1986.

A Plan of Presentation

For the oral presentation, we plan to proceed as follows.

A.1 Background and Motivation

To set the stage, we will first recall the basics of Büchi automata and linear temporal logic and also the automata-theoretic approach to linear-time model checking. This will be done with the help of visualized examples. For instance, in Figure 1 is a Büchi automaton that is equivalent to $\Box(p \rightarrow \Diamond q)$.

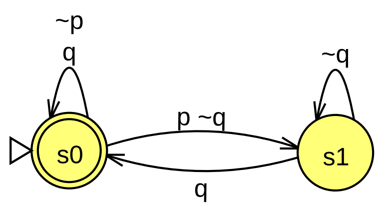


Fig. 1. A Büchi automaton for $\Box(p \rightarrow \Diamond q)$.

For a given system, performance of automata-theoretic model checking is influenced by the specification automaton. Table 1 shows how the sizes of the specification automata affect the performance. The statistics in the table do not include measures of time. However, for a system that satisfies a temporal property, the size of the product automaton is an indirect way to indicate the time spent (without considering optimization techniques such as partial order reduction or symmetry reduction).

	$\mathcal{A}_{\neg f}$		$\mathcal{M} \cap \mathcal{A}_{\neg f}$		Memory	
Desired property	st.	tran.	st.	tran.	usage (MB)	$\mathcal{A}_{\neg f}$ generated by
1. $\Box(p \rightarrow \Diamond q)$	2	7	75681	464848	5.626	Couvreur, LTL2Buchi+, Spin
	4	14	83653	501986	5.919	LTL2AUT+, LTL2BA, LTL2Buchi
	8	31	107558	648778	6.993	MoDeLLa
2. $\Box(c \rightarrow ((c\, \mathcal{U}\, d)\, \mathcal{U}\, e))$	4	32	108504	829956	6.993	Couvreur, LTL2BA, LTL2Buchi+
	5	39	135113	1040443	8.165	Spin
	10	78	143893	1096969	8.458	LTL2AUT+, LTL2Buchi
	26	215	141741	1043380	8.360	MoDeLLa

Table 1. Model checking a token ring of size 6. There are 76665 states and 460929 transitions in the Büchi automaton $\mathcal{M}$ representing the token ring. The two temporal formulae are satisfied by the token ring; meanings of the propositions in the formulae are irrelevant for our purpose here.

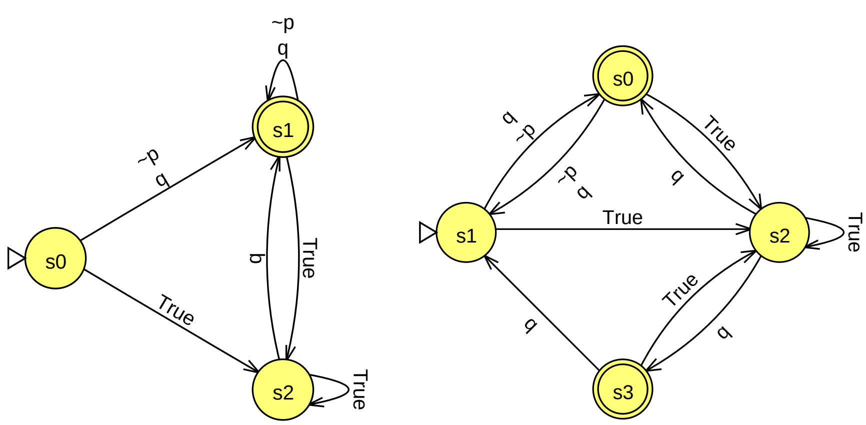
Note: for the textual representation of boolean connectives and temporal operators, Büchi Store follows the format as used in the GOAL tool [19], shown in Table 2.

Operator	$\neg$	$\vee$	$\wedge$	$\rightarrow$	$\leftrightarrow$	$\bigcirc$	$\Box$	$\Diamond$	$\mathcal{U}$	$\mathcal{W}$	$\ominus$	$\tilde{\ominus}$	$\boxminus$	$\diamondminus$	$\mathcal{S}$	$\mathcal{B}$	$\exists$	$\forall$
Format 1	`~`	`\/`	`/\`	`-->`	`<-->`	`()`	`[]`	`<>`	`U`	`W`	`(-)`	`(~)`	`[-]`	`<->`	`S`	`B`	`E`	`A`
Format 2	`~`	`\/`	`/\`	`-->`	`<-->`	`X`	`G`	`F`	`U`	`W`	`Y`	`Z`	`H`	`O`	`S`	`B`	`E`	`A`

Table 2. Syntax for temporal formulae.

A.2 Performance of Translation and Complementation Algorithms

We next give examples showing that translation and complementation algorithms perform differently, but none is dominant. For example, in Figure 2 are two Büchi automata generated using Couvreur's algorithm and Spin respectively for the formula $\Box(p \rightarrow \Diamond q)$. Here the result from Couvreur's algorithm is slightly better. However, for the case $\Diamond\Box p \vee \Diamond\Box q$ (not shown here), Spin is better.



(a) Couvreur's (b) Spin

Fig. 2. Two algorithmically-generated Büchi automata for $\Box(p \rightarrow \Diamond q)$.

A.3 Demo: Main Functions of the Büchi Store

The demo should include one or two examples for each of the (1) search, (2) browse, and (3) upload functions. Figure 3 shows the main page of the Büchi Store.

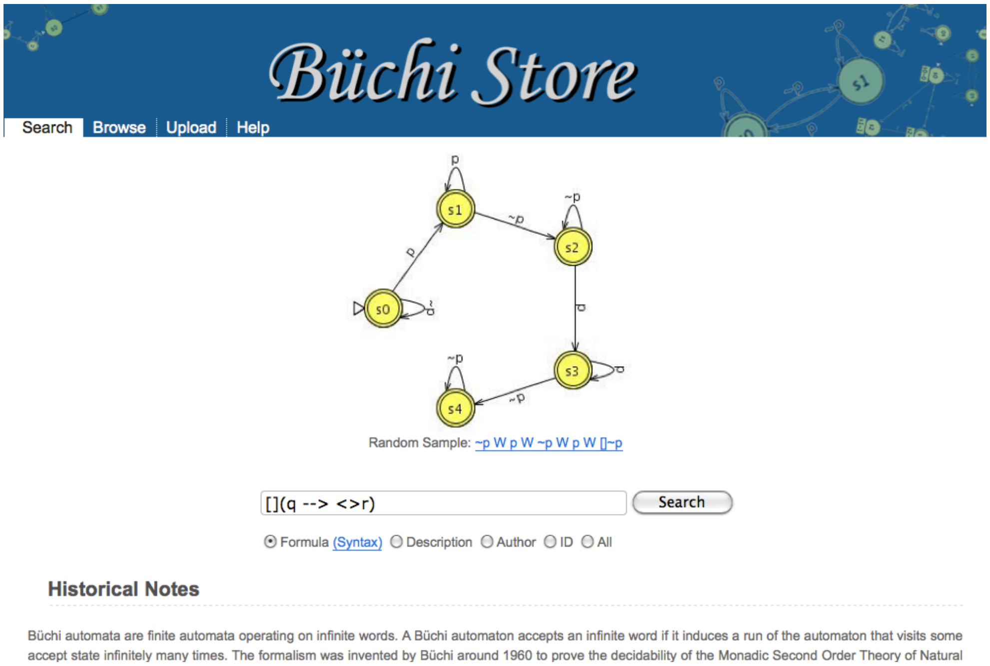


Fig. 3. Main page of the Büchi Store.

In Figure 4 is a snapshot showing the results from searching automata for $\square(q \rightarrow \diamond r)$. The search algorithm of the Store is able to rename the propositions so that automata for $\square(p \rightarrow \diamond q)$ or the like can be retrieved. Automata for formulae that contain $\square(p \rightarrow \diamond q)$ as a sub-formula (up to renaming of propositions) will also be returned. Following a particular entry in the returned results, the user can also find automata for formulae that are equivalent (but syntactically different) to the formula specifying the entry.

Automata in the Büchi Store may be grouped and browsed according to formula length, automaton size, class in the Temporal Hierarchy, or class in the Specification Patterns. Figure 5 shows what the user will see if he chooses to browse automata from the Response (Recurrence) class in the Temporal Hierarchy.

Figure 6 shows a failed attempt at uploading a Büchi automaton for $\square(p \rightarrow \diamond q)$ because the uploaded automaton is structurally identical to an existing one in the Store.

A.4 Classification of Automata

The Büchi Store automatically classifies automata (identified by temporal formulae) into the classes of Safety, Guarantee, Obligation, Response (Recurrence), Persistence and Reactivity in the Temporal Hierarchy of Manna and Pnueli. To

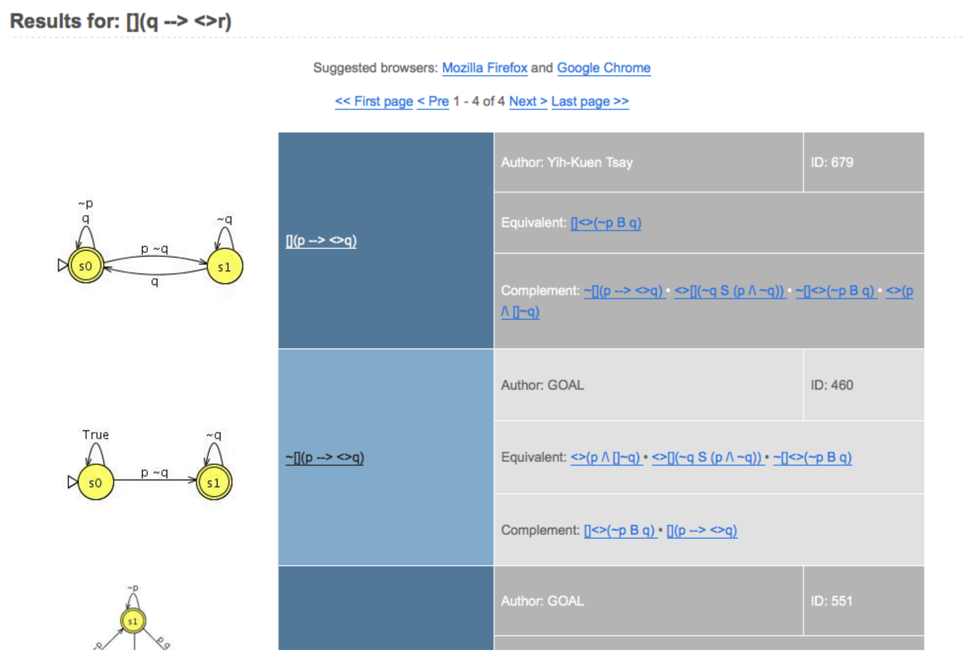


Fig. 4. Results from searching $\Box(q \to \Diamond r)$.

apply the classification algorithm of Manna and Pnueli which relies on the characteristics of a Streett automaton equivalent to the automaton being classified, we complement the Büchi automaton and then use Safra's complementation contruction to obtain the Streett automaton that is the complement of the complemented Büchi automaton (which is therefore equivalent to the original Büchi automaton).

In addition, the Büchi Store also automatically decides whether an automaton is semantically deterministic. By "semantically deterministic", we mean that a Büchi automaton (though nondeterministic syntactically) is equivalent to some deterministic Büchi automaton (syntactically).

A.5 Büchi Store for Benchmark Cases

For research, the Büchi Store is probably most useful as a source of benchmark cases for evaluating complementation or translation algorithms. Every Büchi automaton in the initial collection has a complement, which is reasonably well optimized. A subset of the collection could serve as a set of benchmark cases for evaluating Büchi complementation algorithms. Similarly, a selection of pairs of temporal formula and automaton can be used for evaluating algorithms that translate temporal formulae into Büchi automata. Tables 4 and 6 suggest how these evaluations may be carried out.

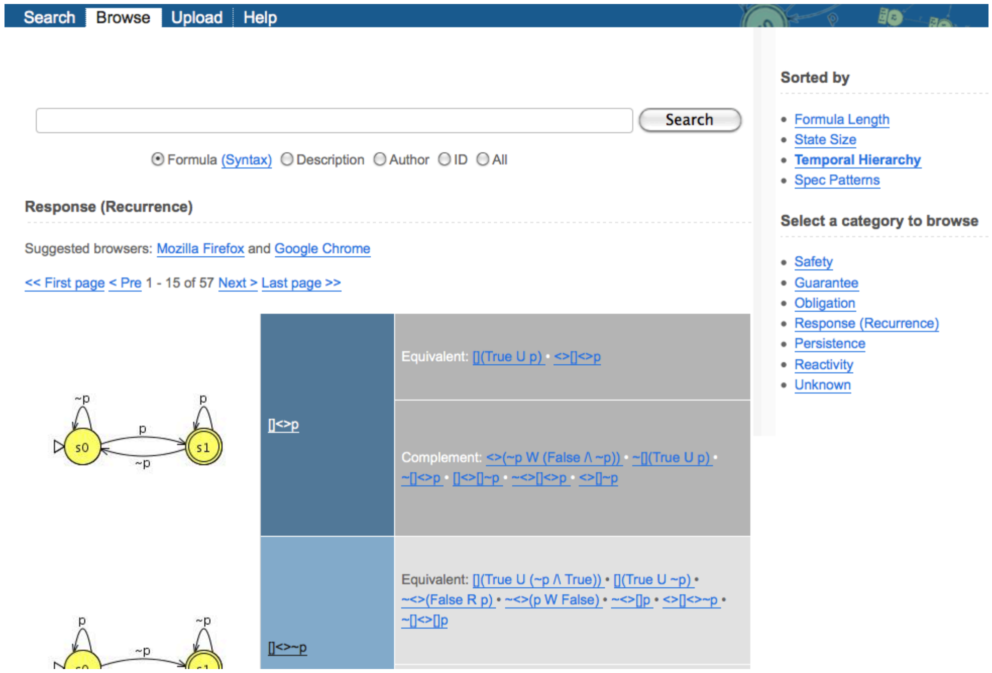


Fig. 5. Browsing Büchi automata from the Response (Recurrence) class in the Temporal Hierarchy.

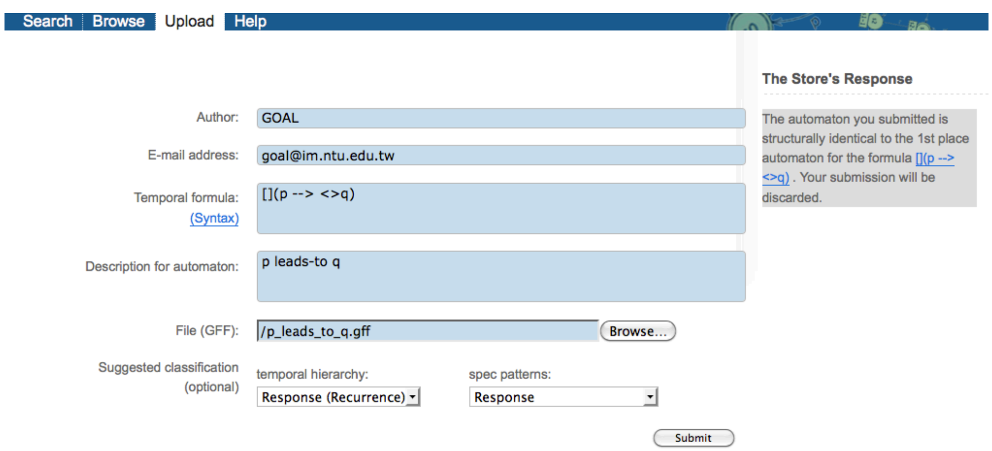


Fig. 6. A failed attempt at uploading a Büchi automaton.

Automaton (Formula)	Büchi Store state	Büchi Store tran.	Safra state	Safra tran.	Piterman state	Piterman tran.	Rank-Based state	Rank-Based tran.	Slice-Based state	Slice-Based tran.
$\Box p$	2	4	6	18	3	9	2	4	3	6
$\Diamond p$	1	1	2	3	3	5	2	3	2	2
$p\,\mathcal{U}\,q$	2	6	4	18	5	22	3	8	3	8
$\Diamond\Box p$	2	4	11	30	10	24	5	14	11	25
$\Diamond\Box p \vee \Diamond\Box q$	3	14	117	576	30	142	14	76	57	321
$\Box(\Box\Diamond p \to \Diamond q)$	3	9	34	136	21	81	20	91	115	601
$\Box(p \to \Diamond(q \wedge \Diamond r))$	3	21	76	662	90	777	96	917	219	2836
$\Box(p \to p\,\mathcal{U}\,q\,\mathcal{U}\,r)$	4	26	17	192	8	76	7	54	7	49
$\Box\Diamond p \to \Box\Diamond q$	3	9	87	349	35	137	19	81	56	234
$p\,\mathcal{U}\,q \vee p\,\mathcal{U}\,r$	2	10	5	34	5	34	8	23	3	12
$\Box(p \to \Box(q \to \Diamond r))$	3	23	8	69	5	44	5	44	5	40
$\Diamond p \to \neg q\,\mathcal{U}\,(r \vee p)$	3	18	16	118	6	42	4	20	8	41
$\Box(p \to \Diamond q)$	2	7	6	27	4	18	4	18	4	16
$\Diamond(p \wedge \bigcirc\Diamond q) \to \neg p\,\mathcal{U}\,r$	5	32	16	140	6	48	4	24	8	50
$\Box(p \to \Box(q \to r \wedge \bigcirc\Diamond s))$	4	62	23	528	8	166	7	114	7	108

Table 4. Comparing results of complementation algorithms against the Büchi Store.

	Büchi Store		LTL2AUT		LTL2BA		Couvreur		LTL2Buchi		Spin	
Formula	state	tran.	state	tran.	state	tran.	state	tran.	state	tran.	state	tran.
$\neg\Box p$	2	4	4	10	2	5	2	5	3	8	2	5
$\neg\Diamond p$	1	1	2	2	2	2	2	2	2	2	1	1
$\neg(p\,\mathcal{U}\,q)$	2	6	4	14	3	10	3	10	3	10	2	7
$\neg\Diamond\Box p$	2	4	3	9	5	13	2	4	3	9	2	5
$\neg(\Diamond\Box p \vee \Diamond\Box q)$	3	14	9	81	9	60	3	12	7	63	3	17
$\neg\Box(\Box\Diamond p \rightarrow \Diamond q)$	3	9	9	43	7	27	4	16	4	20	3	12
$\neg\Box(p \rightarrow \Diamond(q \wedge \Diamond r))$	3	21	6	48	3	24	3	24	4	36	4	32
$\neg\Box(p \rightarrow p\,\mathcal{U}\,q\,\mathcal{U}\,r)$	4	26	11	94	7	57	4	31	6	54	5	38
$\neg(\Box\Diamond p \rightarrow \Box\Diamond q)$	3	9	13	97	10	55	3	10	7	45	3	12
$\neg(p\,\mathcal{U}\,q \vee p\,\mathcal{U}\,r)$	2	10	10	66	7	42	5	30	5	30	2	11
$\neg\Box(p \rightarrow \Box(q \rightarrow \Diamond r))$	3	23	8	62	5	50	3	27	4	40	3	27
$\neg(\Diamond p \rightarrow \neg q\,\mathcal{U}\,(r \vee p))$	3	18	6	46	6	50	3	23	4	26	3	23
$\neg\Box(p \rightarrow \Diamond q)$	2	7	4	14	2	7	2	7	3	12	2	7
$\neg(\Diamond(p \wedge \bigcirc\Diamond q) \rightarrow \neg p\,\mathcal{U}\,r)$	5	42	22	207	12	124	8	78	9	88	8	78
$\neg\Box(p \rightarrow \Box(q \rightarrow r \wedge \bigcirc\Diamond s))$	4	62	12	212	6	140	4	82	5	112	4	82

Table 6. Comparing results of translation algorithms against the Büchi Store.