

Jatatosk: A Lightweight Model-Checker for a Small Fragment of MSO

Max Bannach

April 23, 2018

We describe the usage of the lightweight monadic second-order model-checker **Jatatosk**. A model-checker is a tool that, given a *vocabulary* (or *signature*) τ , a *logical* τ -*structure* \mathcal{S} , and a τ -*formula* φ of some logic, decides whether or not $\mathcal{S} \models \varphi$ holds. In the case that \mathcal{S} is a model of φ , the model-checker is also required to output an assignment of the free and existential-bounded second-order variables of φ as a certificate. **Jatatosk** expects all parts as input, i. e., a file containing a signature, a structure, and a formula. This documents explains how these objects have to be presented to **Jatatosk**, and which restrictions we postulate for the signature, the structure, and the formula. It does, however, not serve as an introduction to logic and we refer the interested reader to the standard textbooks, for instance [2, 1].

1 Defining the Vocabulary

The first thing that has to be defined is the vocabulary $\tau = (R_1, R_2, \dots, R_r)$, which may only contain *relational symbols* of *arity* $\text{ar}(R_i)$. In particular, we do not allow function symbols or constants. Furthermore, τ *must* contain a *binary relation* E . To present the vocabulary to **Jatatosk**, the first input line must be of the following form:

```
vocabulary R1 ar(R1) R2 ar(R2) ... Rr ar(Rr)
```

The following examples show from left to right: a vocabulary for graphs; for graphs in which vertices may be colored red or green; and graphs in which we have additional (directed) red edges.

```
vocabulary E 2
```




```
vocabulary E 2 R 1 G 1
```

```
vocabulary E 2 R 2
```

2 Specifying the Structure

Once the vocabulary is defined, the next thing to do is to specify the actual structure on which we work. As mentioned in the previous section, the structure *must* contain the binary relation E , and we further stipulate that the structure *must* interpret this relation in a symmetric fashion. In other words, the structure *must* contain an *undirected graph* as substructure. To present the structure to **Jatatosk**, the second input-line must have the following form:

```
structure n m
```

This defines that the *universe* of the structure is $U = \{0, 1, \dots, n-1\}$ and that exactly m lines follow that define the interpretation of the R_i in the structure. Assume we wish to define the structures , , and , over the vocabularies of the previous section, then we may achieve this as follows:

```
structure 4 5
```

```
E 0 1  
E 0 2  
E 1 2  
E 1 3  
E 2 3
```

```
structure 4 9
```

```
E 0 1  
E 0 2  
E 1 2  
E 1 3  
E 2 3  
R 0  
R 3  
G 1  
G 2
```

```
structure 4 6
```

```
E 0 1  
E 0 2  
E 1 2  
E 1 3  
E 2 3  
R 0 3
```

Observe that E is *automatically* interpreted *symmetrically*, i. e., the graph will always be undirected. In contrast, the other relations that we have specified (as the binary relation R in the third example) are *not* symmetric.

3 Adding a Formula

The last ingredient we need is the formula. **Jatatosk** works on a small fragment of monadic second-order logic, which we shall define first. All formulas φ must be of the form:

$$\varphi = Q_1 X_1 Q_2 X_2 \dots Q_q X_q \bigwedge_{i=1}^{\ell} \psi_i,$$

where the Q_j are second-order quantifiers to be defined in a second, the X_j are monadic second-order variables, and the ψ_i are first-order formulas to be defined in a minute. There are three existential quantifiers available: $Q_j \in \{\exists^{\text{partition}}, \exists^{\text{connected}}, \exists^{\text{forest}}\}$. The *partition quantifier* $\exists^{\text{partition}} P_1, P_2, \dots, P_k$ quantifies over partitions of the universe (into k partitions). Note that we can express a “classical” existential quantifier simply as $\exists X \equiv \exists^{\text{partition}} X, \bar{X}$. The *connected quantifier* $\exists^{\text{connected}} C$ quantifies over subsets of the universe that are connected with respect to the relation E (in graph theoretic terms), i. e., it quantifies over connected subgraphs. Finally, the *forest quantifier* $\exists^{\text{forest}} F$ quantifies over subsets of the universe that are acyclic with respect to E (in graph theoretic terms), i. e., it quantifies over subgraphs that are forests. To specify these quantifiers in the input, we add one line per quantifier that starts with the name (**exists**, **connected**, **forest**), followed by a list of bounded variables. For instance, the following examples show: a partition of the universe into R, G, B , i. e., a 3-coloring; a quantifier that guesses a connected set C ; and a quantifier that guesses a forest F .

exists R G B

connected C

forest F

The first-order formulas ψ_i have to be of a very restricted form as well. In particular, only formulas of the form presented in the left side of the following table are allowed. Here, the χ_i are quantifier free first-order formulas in canonical normal form. The right side of the table describes the syntax that is used to present such a formula to **Jatatosk**, where the p is an integer that indicates the number of clauses of χ_i (which are defined in the following p lines).

Formula	Input
$\forall x \forall y E(x, y) \rightarrow \chi_i(x, y)$	axay p
$\forall x \exists y E(x, y) \wedge \chi_i(x, y)$	axey p
$\exists x \forall y E(x, y) \rightarrow \chi_i(x, y)$	exay p
$\exists x \exists y E(x, y) \wedge \chi_i(x, y)$	exey p
$\forall x \chi_i(x)$	ax p
$\exists x \chi_i(x)$	ex p

After a ψ_i was introduced as defined in the table, the following p lines each define one clause of χ_i . Every clause is presented by a space-separated line that contains a relation symbol R (either a quantified second-order variable, a free second-order variable, or a relation of the structure), followed by $\text{ar}(R)$ first-order variables (x and y) or elements of the universe. In particular, a clause may be defined as follows:

$R \ x_1 \ x_2 \ \dots \ x_{\text{ar}(R)} \ \tilde{R} \ \tilde{x}_1 \ \tilde{x}_2 \ \dots \ \tilde{x}_{\text{ar}(\tilde{R})} \ \dots$

Additionally, we may negate relational symbols by adding a minus-sign (“-”) in front of it ($-R$). Furthermore, we may have that R is “=”, which is a special symbol that is directly build into the logic. As one might expect, “=” is of arity 2 and does exactly evaluate to true for equal elements. We can express the following three formulas $\forall x \forall y E(x, y) \rightarrow (\neg R(x) \vee \neg R(y)) \wedge (\neg G(x) \vee \neg G(y)) \wedge (\neg B(x) \vee \neg B(y))$, $\forall x (R(x) \rightarrow C(x))$, and $\exists x \forall y E(x, y) \rightarrow R(x, y)$ over the three vocabularies of the previous sections as follows:

axay 3

-R x -R y
-G x -G y
-B x -B y

ax 1
-R x C x

exay 1
R x y

4 Free Variables and Optimization Problems

Jatatosk may also be used to solve *optimization problems*, which are usually not captured by classical model-checking. To express an optimization problem, we consider formulas $\varphi(X_1, \dots, X_k)$ with free monadic second-order variables X_1, \dots, X_k , and weight functions $\omega_1, \dots, \omega_k$ with $\omega_i: U \rightarrow \mathbb{Z}$. Jatatosk can now find sets $S_1, \dots, S_k \subseteq U$ that minimize $\sum_{i=1}^k \sum_{s \in S_i} \omega_i(s_i)$ with respect to $\mathcal{S} \models \varphi(S_1, \dots, S_k)$. To define a free variable, we present a line to Jatatosk that starts with `min`, followed by the name of the variable, followed by n integers that define the weights for the universe. For instance, the following examples add free variables X , Y , and Z with weight functions $x \mapsto 1$, $x \mapsto x$, and $x \mapsto x^2$, respectively.

```
min X 1 1 1 1
```

```
min Y 0 1 2 3
```

```
min Z 0 1 4 9
```

5 Comments and Restrictions

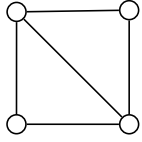
To increase the readability of structures and formulas that are stored in a file, we may add at any point a comment-line, which is indicated with a leading `c`.

```
c We may add a comment before we even start!
c
vocabulary E 2
c A comment between the vocabulary and the structure.
structure 4 5
E 0 1
E 0 2
c
c We may also comment parts of the structure.
c
E 1 2
E 1 3
E 2 3
axay 3
-R x -R y
c Or we may add a comment to the formulas.
-G x -G y
-B x -B y
```

To make the input format well defined, we *forbid* that any relational symbol of the vocabulary, any free variable, and any bounded variable has the name “c”, “=”, or a name that starts with “-”.

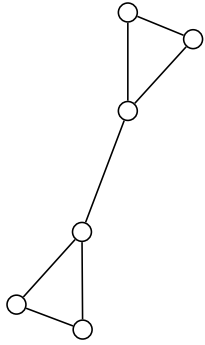
6 Examples

We illustrate the full input format with some complete examples. In the following overview, on the left we show a structure, in the center a formula, and at the right the corresponding input file. A comment-line in the input mentions the problem defined by the formula.



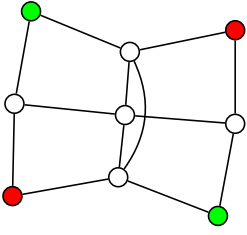
$$\exists^{\text{partition}} R, G, B \bigwedge_{C \in \{R, G, B\}} \forall x \forall y E(x, y) \rightarrow (\neg C(x) \vee \neg C(y))$$

```
c
c Solve 3-coloring.
c
vocabulary E 2
structure 4 5
E 0 1
E 0 2
E 1 2
E 1 3
E 2 3
exists R G B
axay 3
-R x -R y
-G x -G y
-B x -B y
```



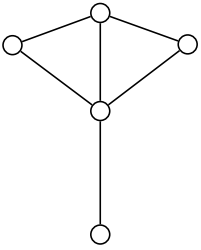
$$\varphi(S) = \forall x \forall y E(x, y) \rightarrow (S(x) \vee S(y))$$

```
c
c Compute a
c minimum
c vertex-cover.
c
vocabulary E 2
structure 6 7
E 0 1
E 0 2
E 1 2
E 1 3
E 3 4
E 3 5
E 4 5
min S 1 1 1 1 1 1 1
axay 1
S x S y
```



$$\exists^{\text{connected}} A \exists^{\text{connected}} B \forall x (R(x) \rightarrow A(x)) \wedge (G(x) \rightarrow B(x)) \wedge (\neg A(x) \vee \neg B(x))$$

c
c Find disjoint
c paths between
c red and green
c vertices.
c
vocabulary E 2 R 1 G 1
structure 9 17
E 0 1
E 0 3
E 1 4
E 1 2
E 2 5
E 3 4
E 3 6
E 4 5
E 4 7
E 5 8
E 6 7
E 7 8
E 3 5
R 2
R 6
G 0
G 8
connected A
connected B
ax 4
-R x A x
-G x B x
-A x -B x



$$\varphi(S) = \exists^{\text{forest}} F \forall x (S(x) \vee F(x))$$

c
c Compute a minimum
c feedback-vertex
c set.
c
vocabulary E 2
structure 5 6
E 0 1
E 0 2
E 0 3
E 0 4
E 2 3
E 3 4
min S 1 1 1 1 1
forest F
ax 1
S x F x

References

- [1] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.
- [2] N. Immerman. *Descriptive Complexity*. Springer, 1999.