Bounded Model Checking using SMT Solvers for RKIP Inhibited ERK Pathway Model

1. Preliminaries

Satisfiability (SAT) solvers are software programs that determine if a given Boolean formula has a satisfying input combination. If a formula has a satisfying input, when the formula is evaluated with this input combination, it results in true. When this is the case, the formula is considered "sat." SAT solvers have a variety of applications, including in systems biology and in computational models. However, traditional SAT formulas are limited to containing only Boolean values, restricting what can be added to them. Satisfiability modulo theories (SMT) solvers build on the foundation created by SAT solvers to allow additional data types to be used.

In the simulation completed with SAT solvers, the midpoint of each interval was used to represent its chemical's concentration in reaction calculations, with the reaction result value being used to assign a new interval to each chemical. Because SAT formulas are limited to Boolean values, this was one approach to allow the representation of these intervals. However, with an SMT solver, these intervals can be directly integrated into the logic of the simulation. Specifically, one operation that can be used in SMT formulas but not SAT formulas is the inequality, allowing the intervals to be more easily expressed as part of the formula.

The addition of inequalities, the intervals used in the Erk pathway simulations can be directly expressed in SMT formulas. In this example, the first four reactions of the Erk pathway were modelled in Python using Microsoft Z3, a type of SMT solver. To begin, Real variables were created to represent the initial and final values of each chemical, in each reaction. Next, a function defines the steps to build the model based on the *k* bound set for Bounded Model Checking (BMC), which is defined by the user. Each *k* step represents a single reaction. For each reaction, constraints are added to the SMT formula to restrict each chemical to its intervals. The reaction logic is also defined, with the input chemical of minimum concentration multiplied by the reaction's rate being used to determine the product of the reaction. The product is evenly removed from the input chemicals and added to the output chemicals according to this logic, and constraints representing this are added to the SMT formula. The constraints for each reaction are added to the SMT formula in accordance with the BMC *k* bound.

When the queries are executed, the SMT solver checks for counterexamples disproving the property, up to the BMC *k* bound. If a counterexample is found, the solver returns "sat" along with the model showing the counterexample. If a counterexample is not identified within the bound, the solver returns "unsat," signifying that the property holds for the given bound.

2. Bounded Model Checking Queries

Here is a sample of queries that were evaluated using bounded model checking.

a) *raf = S500E700*
For all four *k* values, this query returns "unsat" when executed. This means that the property always holds for all potential paths in the simulation. The initial interval of *raf* is *S500E700* in this model, and the value of *raf* must not hit 500 or below at any state to satisfy this query. The system did not find any counterexamples where *raf* exits the *S500E700* interval.

b) *raf = S500E700 ∧ rkip = ZERO*
For this query, the solver returns "sat" for all four *k* values. In its initial state, *rkip* is within the interval *S8E16*, but because it only reduces concentration in reaction 1, this means that, for at least one state, the reaction was not enough to bring its concentration to *ZERO*. Because the solver returns "sat," there exists at least one state that does not satisfy this query.

c) *raf = S500E700 ∧ rkip = S2E4*
This query also returns "sat" for all four *k* steps. This example also starts with *rkip* in its maximum state of *S8E16*, and like Query 2, *rkip* must lose enough of its concentration to be assigned the new interval of *S2E4*. This does not occur in at least one potential state, resulting in "sat" for all four steps. For this query, there also exists at least one state such that the property is not satisfied.