User's Guide for the `sturm` Method

Manuel Eberl <eberlm@in.tum.de>
Institut für Informatik, Technische Universität München

June 26, 2014

Contents

1 Introduction

The `sturm` method uses Sturm's theorem to determine the number of distinct real roots of a polynomial (with rational coefficients) within a certain interval. It also provides some preprocessing to decide a number of statements that can be reduced to real roots of polynomials, such as simple polynomial inequalities and logical combinations of polynomial equations.

2 Usage

2.1 Examples

The following examples should give a good overview of what the `sturm` method can do:

lemma "card $\{x :: \text{real}.\ (x-1)^2 * (x+1) = 0\}\ =\ 2$" **by** sturm

lemma "card $\{x :: \text{real}.\ -0.010831 < x\ \wedge\ x < 0.010831\ \wedge$
$\text{poly}\ [:\ 0, -17/2097152, -49/16777216, 1/6, 1/24, 1/120\ :]\ x\ =\ 0\}\ =\ 3$" **by** sturm

lemma "card $\{x :: \text{real}.\ x^3 + x = 2 * x^2\ \wedge\ x^3 - 6 * x^2 + 11 * x = 6\}\ =\ 1$" **by** sturm

lemma "card $\{x :: \text{real}.\ x^3 + x = 2 * x^2\ \vee\ x^3 - 6 * x^2 + 11 * x = 6\}\ =\ 4$" **by** sturm

lemma "$(x :: \text{real})^2 + 1 > 0$" **by** sturm

lemma "$(x :: \text{real}) > 0\ \Longrightarrow\ x^2 + 1 > 0$" **by** sturm

lemma "$[\![(x :: \text{real}) > 0; x \leq 2/3]\!]\ \Longrightarrow\ x * x \neq\ x$" **by** sturm

lemma "$(x :: \text{real}) > 1\ \Longrightarrow\ x * x > x$" **by** sturm

lemma "$x \geq -0.5\ \Longrightarrow\ x \leq 1\ \Longrightarrow\ x^3 - 5 * x^2 + x + 3 \geq (0 :: \text{real})$" **by** sturm

lemma "mono $(\lambda x :: \text{real}.\ x^3)$" **by** sturm

lemma "strict_mono $(\lambda x :: real.$
$x^7/7 - 14/5 * x^5 + 3 * x^4 + 49/3 * x^3 - 42 * x^2 + 36 * x - 13)$" **by** sturm

2.2 Determining the number of real roots

The "classical" application of Sturm's theorem is to count the number of real roots of a polynomial in a certain interval. The `sturm` method supports this for any polynomial with rational coefficients and any real interval, i. e.$[a; b]$, $(a; b]$, $[a; b)$, and $(a; b)$ where $a \in \mathbb{Q} \cup \{-\infty\}$ and $b \in \mathbb{Q} \cup \{\infty\}$.[1] The general form of the theorems the method expects is:

$$\text{card}\ \{x :: \text{real}.\ a < x \wedge x < b \wedge p\ x = 0\}\ = ?n$$

$?n$ should be replaced by the actual number of such roots and p may be any polynomial real function in x with rational coefficients. The bounds $a < x$ and $x < b$ can be omitted for the "∞" case.

Furthermore, the `sturm` method can instantiate the number $?n$ on the right-hand side automatically if it is left unspecified (as a schematic variable in a schematic lemma). However, due to technical restrictions this also takes twice as long as simply proving that the specified number is correct.

[1]The restriction to rational numbers for the coefficients and interval bounds is to the fact that the code generator is used internally, which, of course, does not support computations on irrational real numbers.

2.3 Inequalities

A simple special case of root counting is the statement that a polynomial $p \in \mathbb{R}[X]$ has no roots in a certain interval, which can be written as:

$$\forall x :: \text{real}.\ x > a \wedge x < b \longrightarrow p\,x \neq 0$$

The `sturm` method can be directly applied to statements such as this and prove them.

2.4 More complex expressions

By using some simple preprocessing, the `sturm` method can also decide more complex statements:

$$\text{card}\ \{x :: \text{real}.\ x > a\ \wedge\ x < b\ \wedge\ P\,x\}\ =\ n$$

where $P\,x$ is a "polynomial expression", which is defined as:

1. $p\,x = q\,x$, where p and q are polynomial functions, such as $\lambda x.\ a$, $\lambda x.\ x$, $\lambda x.\ x^2$, poly p, and so on

2. $P\ x\ \wedge\ Q\ x$ or $P\ x\ \vee\ Q\ x$, where $P\ x$ and $Q\ x$ are polynomial expressions

Of course, by reduction to the case of zero roots, the following kind of statement is also provable by `sturm` :

$$\forall x :: \text{real}.\ x > a\ \wedge\ x < b\ \longrightarrow\ P\ x$$

where $P\ x$ is a "negated polynomial expression", which is defined as:

1. $p\,x \neq q\,x$, where p and q are polynomial functions

2. $P\ x\ \wedge\ Q\ x$ or $P\ x\ \vee\ Q\ x$, where $P\ x$ and $Q\ x$ are negated polynomial expressions

2.5 Simple strict linear inequalities

For any polynomial $p \in \mathbb{R}[X]$, the question whether $p(x) > 0$ for all $x \in I$ for a non-empty real interval I can obviously be reduced to the question of whether $p(x) \neq 0$ for all $x \in I$, i. e.p has no roots in I, and $p(x) > 0$ for some arbitrary fixed $x \in I$, the first of which can be decided using Sturm's theorem and the second by choosing an arbitrary $x \in I$ and evaluating $p(x)$.

Using this reduction, the `sturm` method can also decide single "less than"/"greater than" inequalities of the form

$$\forall x :: \text{real}.\ x > a\ \wedge\ x < b\ \longrightarrow\ p\,x < q\,x$$

2.6 Simple nonstrict linear inequalities and monotonicity

The `sturm` method can also decide nonstrict inequalities, e.g. $p(x) \geq 0$ for all $x \in I$ where I is a non-empty real interval. In contrast to the previous cases, proving nonstrict inequalities is done with help from an ML function that computes a witness, which is then used in the proof by the actual proof method. The witness is a decomposition of the interval in question into disjoint intervals such that every interval contains exactly one root, which makes it possible to check nonnegativity by verifying it on each interval.

Note that while this does not compromise soundness in any way (as witnesses still have to go through checking in Isabelle), it may compromise completeness if the ML code, due to a bug, fails to provide an adequate witness.

Additionally, the `sturm` method can also prove monotonicity and strict monotonicity of polynomials (over the entire real line) by reducing the problem to nonnegativity of the derivative.

2.7 A note on meta logic versus object logic

While statements like $\forall x :: \text{real}.\ x^2 + 1 > 0$ were expressed in their HOL notation in this guide, the `sturm` method can also prove the meta logic equivalents $\bigwedge x :: \text{real}.\ x^2 + 1 > 0$ and $(x :: \text{real})^2 + 1 > 0$ directly.

3 Troubleshooting

Should you find that the `sturm` method fails to prove a statement that it should, according to the above text, be able to prove, please go through the following steps:

1. ensure that your function is indeed a *real* polynomial. Add an appropriate type annotation if necessary.
2. use a computer algebra system to ensure that the property is indeed correct
3. if this did not help, send the statement in question to `eberlm@in.tum.de`; it may be a bug in the preprocessing of the proof method or

Note: in case of monotonicity and nonnegativity, the proof method will output a counterexample if the statement is false. Should proving nonnegativity or monotonicity fail without a counterexample being given, please also report this as a bug.