6.2.2 Simplex

The Simplex algorithm is the prime algorithm for solving optimization problems of systems of linear inequations [59] over the rationals. For automated reasoning optimization at the level of conjunctions of inequations is not in focus. Rather, solvability of a set of linear inequations as a subproblem of some theory combination is the typical application. In this context the simplex algorithm is useful as well, due to its incremental nature. If an inequation $t \circ c$, $\circ \in \{\leq, \geq, <, >\}$, $t = \sum a_i x_i$, $a_i, c \in \mathbb{Q}$, is added to a set N of inequations where the simplex algorithm has already found a solution for N, the algorithm needs not to start from scratch. Instead it continues with the solution found for N. In practice, it turns out that then typically only few steps are needed to derive a solution for $N \cup \{t \circ d\}$ if it exists.

The simplex algorithm introduced in this section is a simplified version of the classical dual simplex used for solving optimization problems.

First, I show the case for non-strict inequations. Starting point is a set N (conjunction) of (non-strict) inequations of the form $(\sum_{x_j \in X} a_{i,j} x_j) \circ_i c_i$ where $\circ_i \in \{\geq, \leq\}$ for all i. Note that an equation $\sum a_i x_i = c$ can be encoded by two inequations $\{\sum a_i x_i \leq c, \sum a_i x_i \geq c\}$.

The variables occurring in N are assumed to be totally ordered by some ordering $\prec$. The ordering $\prec$ will eventually guarantee termination of the simplex algorithm, see Definition 6.2.10 and Theorem 6.2.11 below. I assume the x_j to be all different, without loss of generality $x_j \prec x_{j+1}$, and I assume that all coefficients are normalized by the gcd of the $a_{i,j}$ for all j: if the gcd is different from 1 for one inequation, it is used for division of all coefficients of the inequation.

The goal is to decide whether there exists an assignment β from the x_j into $\mathbb{Q}$ such that $\mathrm{LRA}(\beta) \models \bigwedge_i [(\sum_{x_j \in X} a_{i,j} x_j) \circ_i c_i]$, or equivalently, $\mathrm{LRA}(\beta) \models N$. So the x_j are free variables, i.e., placeholders for concrete values, i.e., existentially quantified.

The first step is to transform the set N of inequations into two disjoint sets E, B of equations and simple bounds, respectively. The set E contains equations of the form $y_i \approx \sum_{x_j \in X} a_{i,j} x_j$, where the y_i are fresh and the set B contains the respective simple bounds $y_i \circ_i c_i$. In case the original inequation from N was already a simple bound, i.e., of the form $x_j \circ_j c_j$ it is simply moved to B. If in N left hand sides of inequations $(\sum_{x_j \in X} a_{i,j} x_j) \circ_i c_i$ are shared, it is sufficient to introduce one equation for the respective left hand side. The y_i are also part of the total ordering $\prec$ on all variables. Clearly, for any assignment β and its respective extension on the y_i, the two representations are equivalent:

$$\mathrm{LRA}(\beta) \models N$$

iff

$$\mathrm{LRA}(\beta[y_i \mapsto \beta(\textstyle\sum_{x_j \in X} a_{i,j} x_j)]) \models E$$

and

$$\mathrm{LRA}(\beta[y_i \mapsto \beta(\textstyle\sum_{x_j \in X} a_{i,j} x_j)]) \models B.$$

Given E and B a variable z is called *dependent* if it occurs on the left hand side of an equation in E, i.e., there is an equation $(z \approx \sum_{x_j \in X} a_{i,j} x_j) \in E$, and in case such a defining equation for z does not exist in E the variable z is called *independent*. Note that by construction the initial y_i are all dependent and do not occur on the right hand side of an equation.

Given a dependant variable x, an independent variable y, and a set of equations E, the *pivot* operation exchanges the roles of x, y in E where y occurs with non-zero coefficient in the defining equation of x. Let $(x \approx ay + t) \in E$ be the defining equation of x in E. When writing $(x \approx ay + t)$ for some equation, I always assume that $y \notin \mathrm{vars}(t)$. Let E' be E without the defining equation of x. Then

$$\mathrm{piv}(E, x, y) := \{y \approx \frac{1}{a}x + \frac{1}{-a}t\} \cup E'\{y \mapsto (\frac{1}{a}x + \frac{1}{-a}t)\}.$$

Given an assignment β, an independent variable y, a rational value c, and a set of equations E then the *update* of β with respect to y, c, and E is

$$\mathrm{upd}(\beta, y, c, E) := \beta[y \mapsto c, \{x \mapsto \beta[y \mapsto c](t) \mid x \approx t \in E\}].$$

A Simplex problem state is a quintuple $(E; B; \beta; S; s)$ where E is a set of equations; B a set of simple bounds; β an assignment to all variables in E, B; S a set of derived bounds, and s the status of the problem with $s \in \{\top, \mathrm{IV}, \mathrm{DV}, \bot\}$. The state $s = \top$ indicates that $\mathrm{LRA}(\beta) \models S$; the state $s = \mathrm{IV}$ that potentially

$\mathrm{LRA}(\beta) \not\models x \circ c$ for some independent variable x, $x \circ c \in S$; the state $s = \mathrm{DV}$ that $\mathrm{LRA}(\beta) \models x \circ c$ for all independent variables x, $x \circ c \in S$, but potentially $\mathrm{LRA}(\beta) \not\models x' \circ c'$ for some dependent variable x', $x' \circ c' \in S$; and the state $s = \bot$ that the problem is unsatisfiable. In particular, the following states can be distinguished:

$(E; B; \beta_0; \emptyset; \top)$	is the start state for N and its transformation into E, B, and assignment $\beta_0(x) := 0$ for all $x \in \mathrm{vars}(E \cup B)$
$(E; \emptyset; \beta; S; \top)$	is a final state, where $\mathrm{LRA}(\beta) \models E \cup S$ and hence the problem is solvable
$(E; B; \beta; S; \bot)$	is a final state, where $E \cup B \cup S$ has no model

Important invariants of the simplex rules are: (i) for every dependent variable there is exactly one equation in E defining the variable and (ii) dependent variables do not occur on the right hand side of an equation, (iii) $\mathrm{LRA}(\beta) \models E$. These invariants are maintained by a pivot (piv) or an update (upd) operation. Here are the rules:

EstablishBound $(E; B \uplus \{x \circ c\}; \beta; S; \top) \Rightarrow_{\mathrm{SIMP}} (E; B; \beta; S \cup \{x \circ c\}; \mathrm{IV})$

AckBounds $(E; B; \beta; S; s) \Rightarrow_{\mathrm{SIMP}} (E; B; \beta; S; \top)$
if $\mathrm{LRA}(\beta) \models S$, $s \in \{\mathrm{IV}, \mathrm{DV}\}$

FixIndepVar $(E; B; \beta; S; \mathrm{IV}) \Rightarrow_{\mathrm{SIMP}} (E; B; \mathrm{upd}(\beta, x, c, E); S; \mathrm{IV})$
if $(x \circ c) \in S$, $\mathrm{LRA}(\beta) \not\models x \circ c$, x independent

AckIndepBound $(E; B; \beta; S; \mathrm{IV}) \Rightarrow_{\mathrm{SIMP}} (E; B; \beta; S; \mathrm{DV})$
if $\mathrm{LRA}(\beta) \models x \circ c$, for all independent variables x with bounds $x \circ c$ in S

FixDepVar$\leq$ $(E; B; \beta; S; \mathrm{DV}) \Rightarrow_{\mathrm{SIMP}} (E'; B; \mathrm{upd}(\beta, x, c, E'); S; \mathrm{DV})$
if $(x \leq c) \in S$, x dependent, $\mathrm{LRA}(\beta) \not\models x \leq c$, there is an independent variable y and equation $(x \approx ay + t) \in E$ where ($a < 0$ and $\beta(y) < c'$ for all $(y \leq c') \in S$) or ($a > 0$ and $\beta(y) > c'$ for all $(y \geq c') \in S$) and $E' := \mathrm{piv}(E, x, y)$

FixDepVar$\geq$ $(E; B; \beta; S; \mathrm{DV}) \Rightarrow_{\mathrm{SIMP}} (E'; B; \mathrm{upd}(\beta, x, c, E'); S; \mathrm{DV})$
if $(x \geq c) \in S$, x dependent, $\mathrm{LRA}(\beta) \not\models x \geq c$, there is an independent variable y and equation $(x \approx ay + t) \in E$ where ($a > 0$ and $\beta(y) < c'$ for all $(y \leq c') \in S$) or ($a < 0$ and $\beta(y) > c'$ for all $(y \geq c') \in S$) and $E' := \mathrm{piv}(E, x, y)$

FailBounds $(E; B; \beta; S; \top) \Rightarrow_{\mathrm{SIMP}} (E; B; \beta; S; \bot)$
if there are two contradicting bounds $x \leq c_1$ and $x \geq c_2$ in $B \cup S$ for some variable x

FailDepVar$\leq$ $(E; B; \beta; S; \mathrm{DV}) \Rightarrow_{\mathrm{SIMP}} (E; B; \beta; S; \bot)$

if $(x \leq c) \in S$, x dependent, LRA(β) $\not\models x \leq c$ and there is no independent variable y and equation $(x \approx ay + t) \in E$ where ($a < 0$ and $\beta(y) < c'$ for all $(y \leq c') \in S$) or ($a > 0$ and $\beta(y) > c'$ for all $(y \geq c') \in S$)

FailDepVar≥ $\quad (E; B; \beta; S; \mathrm{DV}) \Rightarrow_{\mathrm{SIMP}} (E; B; \beta; S; \bot)$

if $(x \geq c) \in S$, x dependent, $\beta \not\models_{\mathrm{LA}} x \geq c$ and there is no independent variable y and equation $(x \approx ay+t) \in E$ where (if $a > 0$ and $\beta(y) < c'$ for all $(y \leq c') \in S$) or (if $a < 0$ and $\beta(y) > c'$ for all $(y \geq c') \in S$)

The simplex rules satisfy a number of invariants that eventually lead to proofs for soundness, completeness and termination. A state $(E; B; \beta; \emptyset; \top)$ is called an *start state* if E is a finite set of equations $x_i \approx \sum a_{i,j} y_j$ such that the x_i occur only on left hand sides and only once in E, and B is a finite set of simple bounds $z_i \circ c$ where z_i occurs in E and $\circ \in \{leq, \geq\}$, and β maps all variables to 0.

Example 6.2.5 (Simplex Detecting Satisfiability)**.** Consider the equational system $E = \{2y + x \geq 1, y - x \leq -2, x \geq 0\}$ which results after preprocessing in the sets $E_0 = \{z_1 \approx 2y + x, z_2 \approx y - x\}$ and $B_0 = \{z_1 \geq 1, z_2 \leq -2, x \geq 0\}$. Starting with an initial assignment β_0 that maps all variables to 0 and hence satisfies E_0, a Simplex run is as follows. Each line gets a number and I make references to the components of the simplex state of previous lines with respect to the line number.

$(E_0, B_0, \beta_0, \emptyset, \top)$
(1) $\Rightarrow_{\mathrm{SIMP}}^{\mathrm{EstablishBound}}$ $(E_0, B_0 \setminus \{x \geq 0\}, \beta_0, \{x \geq 0\}, \mathrm{IV})$
(2) $\Rightarrow_{\mathrm{SIMP}}^{\mathrm{AckBounds}}$ $(E_0, B_1, \beta_0, \{x \geq 0\}, \top)$
(3) $\Rightarrow_{\mathrm{SIMP}}^{\mathrm{EstablishBound}}$ $(E_0, \{z_2 \leq -2\}, \beta_0, \{x \geq 0, z_1 \geq 1\}, \mathrm{IV})$
(4) $\Rightarrow_{\mathrm{SIMP}}^{\mathrm{AckIndepBound}}$ $(E_0, \{z_2 \leq -2\}, \beta_0, \{x \geq 0, z_1 \geq 1\}, \mathrm{DV})$

Now the bound $z_1 \geq 1$ is clearly not satisfied by β_0, so in order to fix it rule FixDepVar≥ is applied. In order to increase z_1 with respect to $z_1 \approx 2y+x$ either y or x need to be increased. Variable y, is not contained in S_4 and x is only bound from below, so both variables can be selected for pivoting. Here I select x, resulting in the new equational system $E_5 = \{x \approx -2y + z_1, z_2 \approx 3y - z_1\}$ and assignment $\beta_5 = \{z_1 \mapsto 1, y \mapsto 0, x \mapsto 1, z_2 \mapsto -1\}$.

(5) $\Rightarrow_{\mathrm{SIMP}}^{\mathrm{FixDepVar}\geq}$ $(E_5, \{z_2 \leq -2\}, \beta_5, \{x \geq 0, z_1 \geq 1\}, \mathrm{DV})$
(6) $\Rightarrow_{\mathrm{SIMP}}^{\mathrm{AckBounds}}$ $(E_5, \{z_2 \leq -2\}, \beta_5, S_5, \top)$
(7) $\Rightarrow_{\mathrm{SIMP}}^{\mathrm{EstablishBound}}$ $(E_5, \emptyset, \beta_5, S_5 \cup \{z_2 \leq -2\}, \mathrm{IV})$
(8) $\Rightarrow_{\mathrm{SIMP}}^{\mathrm{AckIndepBound}}$ $(E_5, \emptyset, \beta_5, S_7, \mathrm{DV})$

Now the bound $z_2 \leq -2$ is not satisfied by β_5, because $\beta_5(z_2) = -1$. Pivoting on $z_2 \approx 3y - z_1$ on y yields $E_9 = \{x \approx -\frac{2}{3}z_2 + \frac{1}{3}z_1, y \approx \frac{1}{3}(z_2 + z_1)\}$ and assignment $\beta_9 = \{z_2 \mapsto -2, z_1 \mapsto 1, x \mapsto \frac{5}{3}, y \mapsto -\frac{1}{3}\}$.

(9) $\Rightarrow_{\mathrm{SIMP}}^{\mathrm{FixDepVar}\leq}$ $(E_9, \emptyset, \beta_9, \{z_1 \geq 1, z_2 \leq -2, x \geq 0\}, \mathrm{DV})$
(10) $\Rightarrow_{\mathrm{SIMP}}^{\mathrm{AckBounds}}$ $(E_9, \emptyset, \beta_9, S_9, \top)$

Now B_{10} is empty and β_{10} satisfies all bounds and hence constitutes a solution to the initial problem.

The equational system and the respective bounds of Example 6.2.5 can be interpreted geometrically. Then a FixDepVar rule application corresponds to testing the intersection points between two of the three initial straights for a solution.

Example 6.2.6 (Simplex Detecting Unsatisfiability)**.** Consider the equational system $E = \{x + 2y \geq 1, x - y \leq 3, x \geq 0, y \leq -1\}$ which results after preprocessing in the sets $E_0 = \{z_1 \approx x + 2y, z_2 \approx x - y\}$ and $B_0 = \{z_1 \geq 1, z_2 \leq 3, x \geq 0, y \leq -1\}$. Starting with an initial assignment β_0 that maps all variables to 0 and hence satisfies E_0, a Simplex run is as follows. Again, each line gets a number and I make references to the components of the simplex state of previous lines with respect to the line number.

$(E_0, B_0, \beta_0, \emptyset, \top)$
(1) $\Rightarrow_{\text{SIMP}}^{\text{EstablishBound}}$ $(E_0, B_0 \setminus \{x \geq 0\}, \beta_0, \{x \geq 0\}, \text{IV})$
(2) $\Rightarrow_{\text{SIMP}}^{\text{AckBounds}}$ $(E_0, B_1, \beta_0, \{x \geq 0\}, \top)$
(3) $\Rightarrow_{\text{SIMP}}^{\text{EstablishBound}}$ $(E_0, B_1 \setminus \{y \leq -1\}, \beta_0, \{x \geq 0, y \leq -1\}, \text{IV})$
(4) $\Rightarrow_{\text{SIMP}}^{\text{FixIndepVar}}$ $(E_0, B_3, \{x \mapsto 0, y \mapsto -1, z_1 \mapsto -2, z_2 \mapsto 1\}, S_3, \text{IV})$
(5) $\Rightarrow_{\text{SIMP}}^{\text{AckBounds}}$ $(E_0, B_3, \beta_4, S_3, \top)$
(6) $\Rightarrow_{\text{SIMP}}^{\text{EstablishBound}}$ $(E_0, B_3 \setminus \{z_1 \geq 1\}, \beta_4, S_3 \cup \{z_1 \geq 1\}, \text{IV})$
(7) $\Rightarrow_{\text{SIMP}}^{\text{AckIndepBound}}$ $(E_0, B_6, \beta_4, S_6, \text{DV})$

The bound $z_1 \geq 1$ is not satisfied by β_7 because $\beta_7(z_1) = -2$. Pivoting on x in $z_1 \approx x + 2y$ yields $E_8 = \{x \approx z_1 - 2y, z_2 \approx z_1 - 3y\}$ and $\beta_8 = \{z_1 \mapsto 1, y \mapsto -1, x \mapsto 3, z_2 \mapsto 4\}$.

(8) $\Rightarrow_{\text{SIMP}}^{\text{FixDepVar}\geq}$ $(E_8, B_6, \beta_8, \{x \geq 0, y \leq -1, z_1 \geq 1\}, \text{DV})$
(9) $\Rightarrow_{\text{SIMP}}^{\text{AckBounds}}$ $(E_8, B_6, \beta_8, S_8, \top)$
(10) $\Rightarrow_{\text{SIMP}}^{\text{EstablishBound}}$ $(E_8, \emptyset, \beta_8, S_8 \cup \{z_2 \leq 3\}, \text{IV})$
(11) $\Rightarrow_{\text{SIMP}}^{\text{AckIndepBound}}$ $(E_8, \emptyset, \beta_8, S_{10}, \text{DV})$
(12) $\Rightarrow_{\text{SIMP}}^{\text{FailDepVar}\leq}$ $(E_8, \emptyset, \beta_8, S_{10}, \bot)$

The bound $z_2 \leq 3$ is not satisfied by β_8 because $\beta_8(z_2) = 4$. In order to meet the bound the value of z_2 needs to be decreased using the equation $z_2 \approx z_1 - 3y$. So either z_1 needs to be decreased, but $\beta_8(z_1) = 1$ and z_1 is bounded below by $z_1 \geq 1$, or y needs to be increased, but $\beta_8(y) = -1$ and y is bounded above by $y \leq -1$. Therefore, rule FailDepVar≤ is applicable, the initial system is unsatisfiable.

Lemma 6.2.7 (Simplex State Invariants)**.** The following invariants hold for any state $(E_i; B_i; \beta_i; S_i; s_i)$ derived by $\Rightarrow_{\text{SIMP}}$ on a start state $(E_0; B_0; \beta_0; \emptyset; \top)$:

1. for every dependent variable there is exactly one equation in E defining the variable

2. dependent variables do not occur on the right hand side of an equation

3. $\text{LRA}(\beta) \models E_i$

4. for all in dependant variables x either $\beta_i(x) = 0$ or $\beta_i(x) = c$ for some bound $x \circ c \in S_i$

5. for all assignments α it holds $\text{LRA}(\alpha) \models E_0$ iff $\text{LRA}(\alpha) \models E_i$

Proof. 1, 2. By induction on the length of a $\Rightarrow_{\text{SIMP}}$ derivation. A consequence of the definition of piv.

3. By induction on the length of a $\Rightarrow_{\text{SIMP}}$ derivation. A consequence of the definition of upd.

4. By induction on the length of a $\Rightarrow_{\text{SIMP}}$ derivation and a case analysis for all rules changing β_i. Recall that initially β_0 maps all variables to 0.

5. The piv operation is equivalence preserving, i.e., an assignment α satisfies E iff it satisfies $\text{piv}(E, x, y)$ for a dependent variable x and an independent variable y. □

Lemma 6.2.8 (Simplex Run Invariants)**.** For any run of $\Rightarrow_{\text{SIMP}}$ from start state $(E_0; B_0; \beta_0; \emptyset; \top) \Rightarrow_{\text{SIMP}} (E_1; B_1; \beta_1; S_1; s_1) \Rightarrow_{\text{SIMP}} \ldots$:

1. the set $\{\beta_o, \beta_1, \ldots\}$ is finite

2. if the sets of dependent and independent variables for two equational systems E_i, E_j coincide, then $E_i = E_j$

3. the set $\{E_o, E_1, \ldots\}$ is finite

4. let S_i not contain contradictory bounds, then $(E_i; B_i; \beta_i; S_i; s_i) \Rightarrow_{\text{SIMP}}^{\text{FixIndepVar},*}$ is finite

Proof. 1. By induction on the length of a $\Rightarrow_{\text{SIMP}}$ derivation. Variables are bound by the β_i to constants occurring B_0. This set is finite. Furthermore, the domain of each β_i is constant. Hence the set $\{\beta_o, \beta_1, \ldots\}$ is finite.

2. By Lemma 6.2.7.1 and 2, for any dependent variable z there is exactly one equation $z \approx a_1x_1 + \ldots + a_nx_n$ in every E. Now assume that dependent and independent variables for two equational systems E_i, E_j coincide but actually E_i and E_j differ in one equation $(z \approx a_1x_1 + \ldots + a_nx_n) \in E_i$ and $(z \approx b_1y_1 + \ldots + b_my_m) \in E_j$. By Lemma 6.2.7.5 it must hold $x_i = y_i$ and $n = m$. It remains to show that the coefficients are identical. For $n = 1$ this is obvious. For $n \geq 2$ this follows again from Lemma 6.2.7.5 by the following two assignments γ, γ', assuming $a_1 \neq b_1$. The first assignment is defined by $\gamma(z) = n$, and $\gamma(x_k) = \frac{1}{a_k}$ for $1 \leq k \leq n$ and the second by $\gamma'(z) = n - 2$, $\gamma'(x_1) = -\frac{1}{a_1}$ and $\gamma'(x_k) = \frac{1}{a_k}$ for $2 \leq k \leq n$. Both assignments satisfy the defining equations for z and can be extended to satisfy E_i and E_j. Then from γ we can conclude

$$a_1\frac{1}{a_1} > b_1\frac{1}{a_1} \qquad \text{iff} \qquad a_2\frac{1}{a_2} + \ldots + a_n\frac{1}{a_n} < b_2\frac{1}{a_2} + \ldots + b_n\frac{1}{a_n}$$

and from γ' accordingly

$$a_1\frac{1}{a_1} > b_1\frac{1}{a_1} \qquad \text{iff} \qquad a_2\frac{1}{a_2} + \ldots + a_n\frac{1}{a_n} > b_2\frac{1}{a_2} + \ldots + b_n\frac{1}{a_n}$$

a contradiction.

3. A consequence of 2.

4. The independent variables are in fact independent from each other. Thus any bound on an independent can be eventually satisfied by rule FixIndepVar. □

Corollary 6.2.9 (Infinite Runs Contain a Cycle)**.** Let $(E_0; B_0; \beta_0; \emptyset; \top) \Rightarrow_{\text{SIMP}} (E_1; B_1; \beta_1; S_1; s_1) \Rightarrow_{\text{SIMP}} \ldots$ be an infinite run. Then there are two states $(E_i; B_i; \beta_i; S_i; s_i)$, $(E_k; B_k; \beta_k; S_k; s_k)$ such that $i \neq k$ and $(E_i; B_i; \beta_i; S_i; s_i) = (E_k; B_k; \beta_k; S_k; s_k)$.

Proof. The initial sets are all finite. No rule adds a simple bound to any B_i , they can only be moved to some S_i and stay there. So there are only finitely many such configurations B_i , S_i during a run. By Lemma 6.2.8.1 there are only finitely many different β_i. By Lemma 6.2.8.3 there are only finitely many different E_i. In sum, any infinite run must contain two identical states, a cycle. □

Definition 6.2.10 (Reasonable Strategy)**.** A *reasonable* strategy prefers FailBounds over EstablishBounds and the FixDepVar rules select minimal variables x, y in the ordering $\prec$.

Theorem 6.2.11 (Simplex Soundness, Completeness & Termination)**.** Given a reasonable strategy and initial set N of inequations and its separation into E and B :

1. $\Rightarrow_{\text{SIMP}}$ terminates on $(E_0; B_0; \beta_0; \emptyset; \top)$
2. if $(E; B; \beta_0; \emptyset; \top) \Rightarrow^*_{\text{SIMP}} (E'; B'; \beta; S; \bot)$ then N has no solution
3. if $(E; B; \beta_0; \emptyset; \top) \Rightarrow^*_{\text{SIMP}} (E'; \emptyset; \beta; B; \top)$ and $(E; \emptyset; \beta; B; \top)$ is a normal form, then $\text{LRA}(\beta) \models N$
4. all final states $(E; B; \beta; S; s)$ match either 2. or 3.

Proof. 1. (Idea) An infinite run must contain a cycle due to Corollary 6.2.9. Runs always selecting minimal variables for the FixDepVar rules cannot contain cycles.

2. (Scetch) The fail rules are correct, given Lemma 6.2.7.5.

3. By Lemma 6.2.7.5 and all initial bounds are satisfied by β, because AckBounds is the only rule generating $\top$.

4. A state $(E; B; \beta; S; \text{IV})$ can always be rewritten to a state $(E; B; \beta'; S; \top)$ or $(E; B; \beta'; S; \text{DV})$. Any state $(E; B; \beta; S; \text{DV})$ is either rewritten to a final state $(E; B; \beta; S; \bot)$ or again a state $(E'; B; \beta'; S; \text{DV})$. The rest follows from termination. □

In case of strict bounds the idea is to introduce an infinitesimal small constant $\delta > 0$ and to replace the strict bound by a non-strict one. So, for example, a bound $x < 5$ is replaced by $x \leq 5 - \delta$. Now δ is treated symbolically through the overall computation, i.e., we extend $\mathbb{Q}$ to $\mathbb{Q}_\delta$ with new pairs (q, k) with $q, k \in \mathbb{Q}$ where (q, k) represents $q + k\delta$ and the operations, relations on $\mathbb{Q}$ are lifted to $\mathbb{Q}_\delta$:

$$\begin{aligned}
(q_1, k_1) + (q_2, k_2) &:= (q_1 + q_2, k_1 + k_2)\\
p(q, k) &:= (pq, pk)\\
(q_1, k_1) \leq (q_2, k_2) &:= (q_1 < q_2) \vee (q_1 = q_2 \wedge k_1 \leq k_2)
\end{aligned}$$