Synthesis of Ranking Functions and Synthesis of Inductive Invariants and Synthesis of Recurrence Sets via Constraint Solving

Andreas Podelski

January 17, 2012

Program Verification and Constraints

- Reasoning about program computations
- Computation is a sequence of program states
- Sequences generated by transition relation
- Transition relation defined by assume & update statements
- Assume & update statements = transition constraints

Program Properties

- Non-reachability: given state is not reachable
- Termination: no infinite computation exists
- Linear-time properties (LTL):
  reduced to reachability and termination
  (in automata-theoretic approach)

Verification = finding auxiliary assertions

- Proving reachability = finding inductive invariant
- Proving termination = finding ranking relation

  (ranking relation defined by ranking function, i.e., an expression over program variables which bounds number of steps)

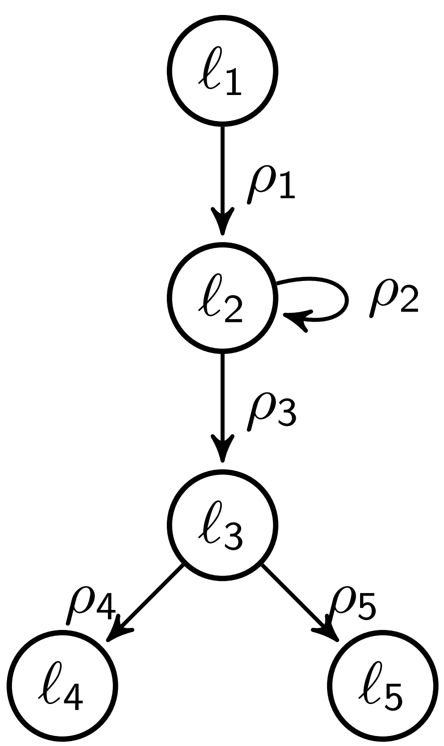
Running Example

```
main(int x, int y, int z) {
  assume(y >= z);
  while (x < y) {
    x++;
  }
  assert(x >= z);
}
```

- for constraint solving, treat x, y, and z as rationals

CFG and Transition Relations

```
main(int x, int y, int z) {
  assume(y >= z);
  while (x < y) {
    x++;
  }
  assert(x >= z);
}
```



$$\rho_1 = (y \geq z \wedge x' = x \wedge y' = y \wedge z' = z)$$

$$\rho_2 = (x + 1 \leq y \wedge x' = x + 1 \wedge y' = y \wedge z' = z)$$

$$\rho_3 = (x \geq y \wedge x' = x \wedge y' = y \wedge z' = z)$$

$$\rho_4 = (x \geq z \wedge x' = x \wedge y' = y \wedge z' = z)$$

$$\rho_5 = (x + 1 \leq z \wedge x' = x \wedge y' = y \wedge z' = z)$$

Transition Constraint => Matrix

$$\rho_2 = (x + 1 \leq y \wedge x' = x + 1 \wedge y' = y)$$

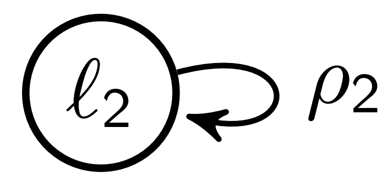
$$= (x - y \leq -1 \wedge -x + x' \leq 1 \wedge x - x' \leq -1 \wedge -y + y' \leq 0 \wedge y - y' \leq 0)$$

$$= \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \\ x' \\ y' \end{pmatrix} \leq \begin{pmatrix} -1 \\ 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}$$

Ranking Functions

- Ranking function, say *f*, maps states to distance until terminating state

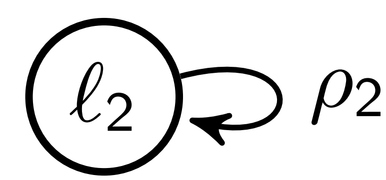
```
while (x < y) {
  x++;
}
```



- $f(x, y) = (y - x)$
- decrease at each step
- bounded from below

Ranking Function Constraint ∃∀

- ranking function $f(x, y) = f_x\, x + f_y\, y$
- lower bound δ_0
- decrease amount δ



$$\delta \geq 1 \wedge$$

$$\forall x\ \forall y\ \forall x'\ \forall y' :$$

$$\rho_2 \rightarrow (f_x x + f_y y \geq \delta_0 \wedge$$

$$f_x x' + f_y y' \leq f_x x + f_y y - \delta)$$

Quantifier Alternation ∃∀

$$\exists f_x \; \exists f_y \; \exists \delta_0 \; \exists \delta$$

$$\forall x \; \forall y \; \forall x' \; \forall y' :$$

$$\delta \geq 1 \wedge$$

$$\rho_2 \rightarrow (f_x x + f_y y \geq \delta_0 \wedge$$

$$f_x x' + f_y y' \leq f_x x + f_y y - \delta)$$

Farkas' Lemma

- implied inequalities are derivable as weighted$_{\geq 0}$ sums

$$(\exists x : Ax \leq b) \wedge (\forall x : Ax \leq b \rightarrow cx \leq \delta)$$

iff

$$\exists \lambda : \lambda \geq 0 \wedge \lambda A = c \wedge \lambda b \leq \delta$$

Transition Constraint => Matrix

$$\rho_2 = (x + 1 \leq y \wedge x' = x + 1 \wedge y' = y)$$

$$= (x - y \leq -1 \wedge -x + x' \leq 1 \wedge x - x' \leq -1 \wedge -y + y' \leq 0 \wedge y - y' \leq 0)$$

$$= \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \\ x' \\ y' \end{pmatrix} \leq \begin{pmatrix} -1 \\ 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}$$

Eliminating ∀-Quantifier (I)

$$\rho_2 = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \\ x' \\ y' \end{pmatrix} \leq \begin{pmatrix} -1 \\ 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}$$

implies

$$f_x x + f_y y \geq \delta_0 = \begin{pmatrix} -f_x & -f_y & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ x' \\ y' \end{pmatrix} \leq -\delta_0$$

Eliminating ∀-Quantifier (2)

$$\forall x\ \forall y\ \forall x'\ \forall y' : \rho_2 \rightarrow f_x x + f_y y \geq \delta_0$$

iff (by Farkas' lemma)

$$\exists \lambda : \lambda \geq 0 \wedge \lambda \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{pmatrix} = \begin{pmatrix} -f_x & -f_y & 0 & 0 \end{pmatrix} \wedge \lambda \begin{pmatrix} -1 \\ 1 \\ -1 \\ 0 \\ 0 \end{pmatrix} \leq -\delta_0$$

Ranking Function Constraint ∃

- Find solution for f_x , f_y , δ_0 , and δ

$$\delta \geq 1 \wedge$$

$$\exists \lambda \; \exists \mu :$$

$$\lambda \geq 0 \wedge \lambda \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{pmatrix} = \begin{pmatrix} -f_x & -f_y & 0 & 0 \end{pmatrix} \wedge \lambda \begin{pmatrix} -1 \\ 1 \\ -1 \\ 0 \\ 0 \end{pmatrix} \leq -\delta_0 \wedge$$

$$\mu \geq 0 \wedge \mu \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{pmatrix} = \begin{pmatrix} -f_x & -f_y & f_x & f_y \end{pmatrix} \wedge \mu \begin{pmatrix} -1 \\ 1 \\ -1 \\ 0 \\ 0 \end{pmatrix} \leq -\delta$$

ranking function $f(x, y) = f_x\, x + f_y\, y$ with bound δ_0, and gap δ

Ranking Function Constraint Solved

- solution for f_x , f_y , δ_0 , and δ

$$\lambda = (1\ 0\ 0\ 0\ 0)$$

$$\mu = (0\ 0\ 1\ 1\ 0)$$

$$f_x = -1$$

$$f_y = 1$$

$$\delta_0 = 1$$

$$\delta = 1$$

```
while (x < y) {
  x++;
}
```

- Ranking function $f(x, y) = (-1\ x + 1\ y) = y - x$

Ranking Function Algorithm

- Input $\rho(v, v') = R \begin{pmatrix} v \\ v' \end{pmatrix} \leq r$
- Defining constraint

$$\exists f \; \exists \delta_0 \; \exists \delta \; \forall v \; \forall v' : \delta \geq 1 \wedge \rho(v, v') \rightarrow (fv \geq \delta_0 \wedge fv' \leq fv - \delta)$$

- Linear constraint to solve

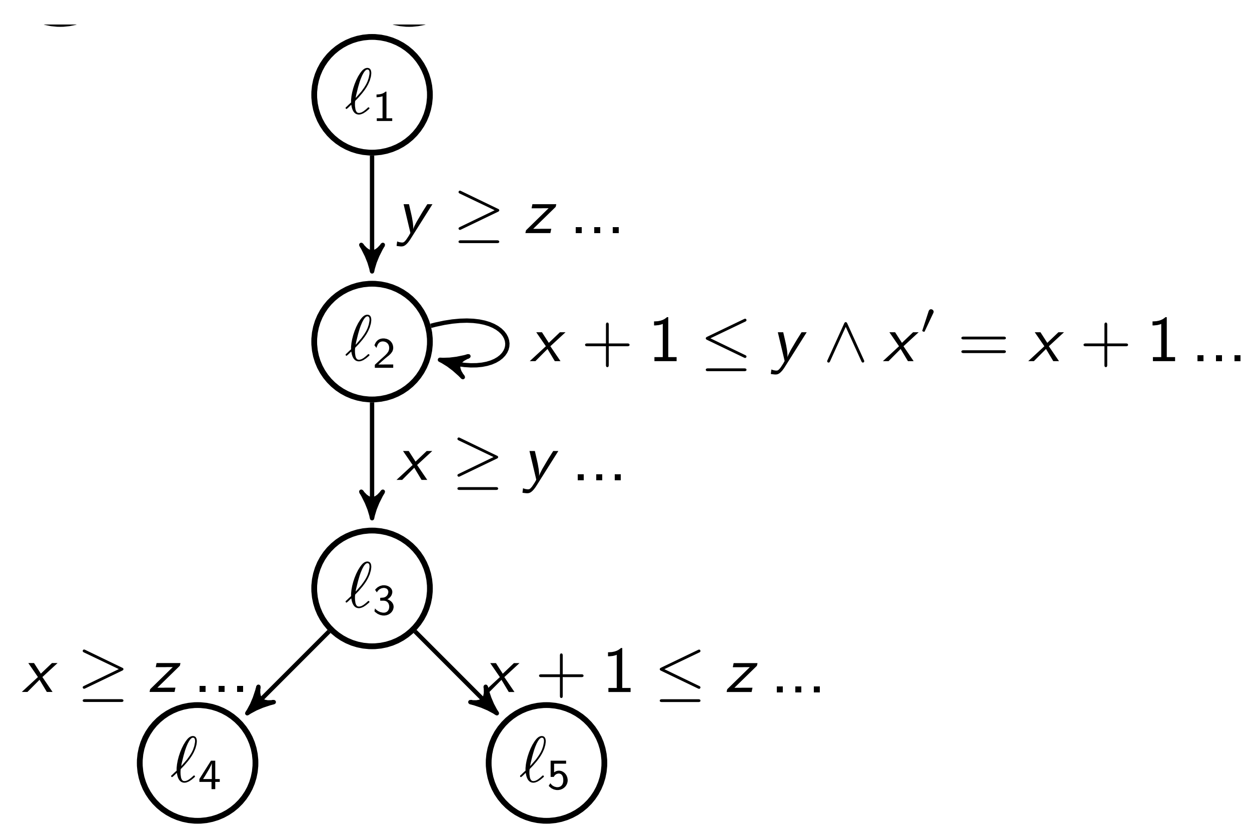
$$\exists f \; \exists \delta_0 \; \exists \delta \; \exists \lambda \; \exists \mu : \delta \geq 1 \wedge$$

$$\lambda \geq 0 \wedge \lambda R = (-f \;\; 0) \wedge \lambda r \leq -\delta_0 \wedge$$

$$\mu \geq 0 \wedge \mu R = (-f \;\; f) \wedge \mu r \leq -\delta$$

Invariants

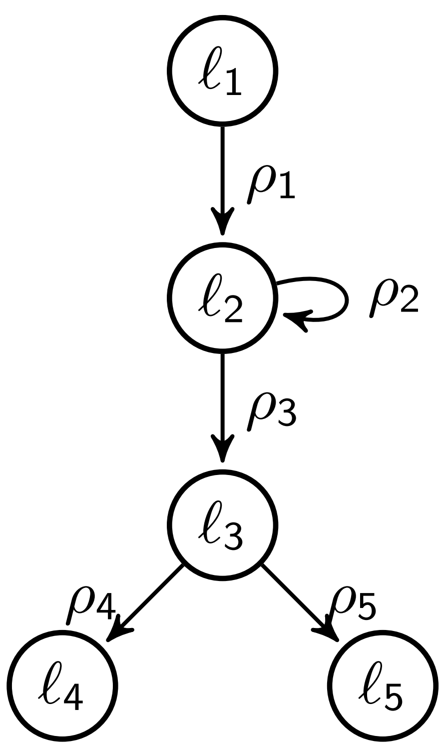
- Invariant for each control location:
  $l_1 : (0 \leq 0)$
  $l_2 : (z \leq y)$
  $l_3 : (z \leq x)$
  $l_4 : (0 \leq 0)$
  $l_5 : (1 \leq 0)$



- Inductiveness
  $l_2 : (z \leq y) \wedge (x{+}1 \leq y \wedge x'{=}x{+}1 \wedge y'{=}y) \Rightarrow (z' \leq y')$

Example Program

```
main(int x, int y, int z) {
  assume(y >= z);
  while (x < y) {
    x++;
  }
  assert(x >= z);
}
```



$$\rho_1 = (y \geq z \wedge x' = x \wedge y' = y \wedge z' = z)$$

$$\rho_2 = (x + 1 \leq y \wedge x' = x + 1 \wedge y' = y \wedge z' = z)$$

$$\rho_3 = (x \geq y \wedge x' = x \wedge y' = y \wedge z' = z)$$

$$\rho_4 = (x \geq z \wedge x' = x \wedge y' = y \wedge z' = z)$$

$$\rho_5 = (x + 1 \leq z \wedge x' = x \wedge y' = y \wedge z' = z)$$

Invariant Constraint ∃∀

- Find invariant at l_2 of the form $p_x\, x + p_y\, y + p_z\, z \leq p_0$ and invariant at l_3 of the form $q_x\, x + q_y\, y + q_z\, z \leq q_0$
- inductiveness of invariant at l_3 entails non-reachability of l_5

$$\forall x\ \forall y\ \forall z\ \forall x'\ \forall y'\ \forall z' :$$

$$(\rho_1 \rightarrow p_x x' + p_y y' + p_z z' \leq p_0) \wedge$$

$$((p_x x + p_y y + p_z z \leq p_0 \wedge \rho_2) \rightarrow p_x x' + p_y y' + p_z z' \leq p_0) \wedge$$

$$((p_x x + p_y y + p_z z \leq p_0 \wedge \rho_3) \rightarrow q_x x' + q_y y' + q_z z' \leq q_0) \wedge$$

$$((q_x x + q_y y + q_z z \leq p_0 \wedge \rho_4) \rightarrow 0 \leq 0) \wedge$$

$$((q_x x + q_y y + q_z z \leq p_0 \wedge \rho_5) \rightarrow 0 \leq -1)$$

Quantifier Alternation $\exists\forall$

- use matrix form

$$v = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$\rho_1 = R_1 \begin{pmatrix} v \\ v' \end{pmatrix} \leq r_1$$

$$\dots$$

$$\rho_5 = R_5 \begin{pmatrix} v \\ v' \end{pmatrix} \leq r_5$$

- eliminate $\forall$ by applying Farkas' lemma

Invariant Constraint ∃

- Find invariant at l_2 of the form $p_x\, x + p_y\, y + p_z\, z \leq p_0$ and invariant at l_3 of the form $q_x\, x + q_y\, y + q_z\, z \leq q_0$

$$\exists\lambda_1\ \exists\lambda_2\ \exists\lambda_3\ \exists\lambda_4\ \exists\lambda_5 :$$

$$\lambda_1 \geq 0 \wedge \lambda_1 R_1 = (0\ p_x\ p_y\ p_z) \wedge \lambda_1 r_1 \leq p_0 \wedge$$

$$\lambda_2 \geq 0 \wedge \lambda_2 \begin{pmatrix} p_x\ p_y\ p_z\ 0 \\ R_2 \end{pmatrix} = (0\ p_x\ p_y\ p_z) \wedge \lambda_2 \begin{pmatrix} p_0 \\ r_2 \end{pmatrix} \leq p_0 \wedge$$

$$\lambda_3 \geq 0 \wedge \lambda_3 \begin{pmatrix} p_x\ p_y\ p_z\ 0 \\ R_3 \end{pmatrix} = (0\ q_x\ q_y\ q_z) \wedge \lambda_3 \begin{pmatrix} p_0 \\ r_3 \end{pmatrix} \leq q_0 \wedge$$

$$\lambda_4 \geq 0 \wedge \lambda_4 \begin{pmatrix} q_x\ q_y\ q_z\ 0 \\ R_4 \end{pmatrix} = 0 \wedge \lambda_4 \begin{pmatrix} q_0 \\ r_4 \end{pmatrix} \leq 0 \wedge$$

$$\lambda_5 \geq 0 \wedge \lambda_5 \begin{pmatrix} q_x\ q_y\ q_z\ 0 \\ R_5 \end{pmatrix} = 0 \wedge \lambda_5 \begin{pmatrix} q_0 \\ r_5 \end{pmatrix} \leq -1$$

Invariant Constraint Solved

- Find $l_2 : p_x\, x + p_y\, y + p_z\, z \leq p_0$ and $l_3 : q_x\, x + q_y\, y + q_z\, z \leq q_0$

$$\lambda_1 = (1\ 1\ 1\ 1)$$
$$\lambda_2 = (1\ 0\ 1\ 1\ 1)$$
$$\lambda_3 = (1\ 1\ 1\ 1\ 1)$$
$$\lambda_4 = (0\ 0\ 0\ 0\ 0)$$
$$\lambda_5 = (1\ 1\ 0\ 0\ 0)$$

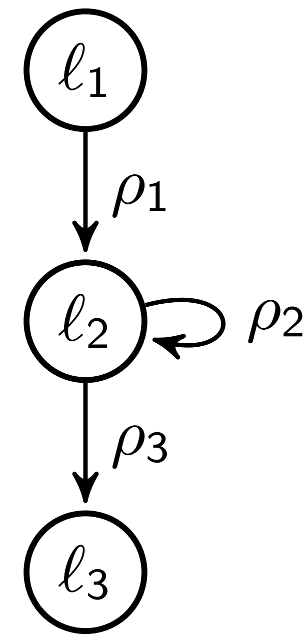
$$p_x = 0 \quad p_y = -1 \quad p_z = 1 \quad p_0 = 0$$
$$q_x = -1 \quad q_y = 0 \quad q_z = 1 \quad q_0 = 0$$

- Invariant at $l_2 : 0x + (-1)y + 1z \leq 0$ and $l_3 : (-1)x + 0y + 1z \leq 0$

  $l_2 : z \leq y$ and $l_3 : z \leq x$

Proving Non-Termination

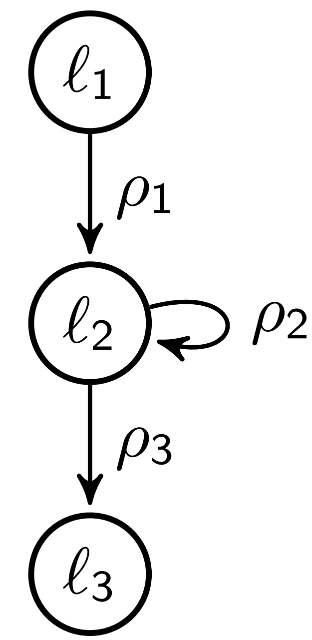
```
main(int x, int y, int z) {
  assume(y >= z);
  while (x < y) {
    x=x+1+z;
  }
}
```



- Non-terminating execution
  $(-1, 0, -1), (-1, 0, -1), \ldots$
- Recurrence set S is reachable and can always reach itself
- Example recurrence set S = (x+1 $\leq$ y /\ z $\leq$ -1)

Recurrence Set Constraint ∃∀∃

- Recurrence set $Sv \leq s$ is reachable and can always reach itself
- Let $v = (x\ y\ z)$
- Find $(Sv \leq s) = (p_x\, x + p_y\, y + p_z\, z \leq p_0\ \wedge$
  $q_x\, x + q_y\, y + q_z\, z \leq q_0)$



$\exists S\ \exists s :$

$$(\exists v\ \exists v' : \rho_1(v, v') \wedge Sv' \leq s) \wedge$$

$$(\forall v\ \exists v' : Sv \leq s \rightarrow (\rho_2(v, v') \wedge Sv' \leq s))$$

Quantifier Alternation ∃∀∃

- $\rho_1(v, v')$ and $\rho_2(v, v')$ define functional dependency between v' and v
  ... /\ $x' = x$ /\ $y' = y$ /\ $z' = z$
  ... /\ $x' = x+1+z$ /\ $y' = y$ /\ $z' = z$
- Useful for elimination of $\exists v'$

$$\exists S\ \exists s :$$

$$(\exists x\ \exists y\ \exists z : y \geq z \wedge S \begin{pmatrix} x \\ y \\ z \end{pmatrix} \leq s) \wedge$$

$$(\forall x\ \forall y\ \forall z : S \begin{pmatrix} x \\ y \\ z \end{pmatrix} \leq s \rightarrow (x + 1 \leq y \wedge S \begin{pmatrix} x+1+z \\ y \\ z \end{pmatrix} \leq s))$$

Quantifier Alternation $\exists\forall$

- Elimination of $\forall v$ produces:

$$\exists S \; \exists s :$$

$$(\exists x \; \exists y \; \exists z : y \geq z \wedge S \begin{pmatrix} x \\ y \\ z \end{pmatrix} \leq s) \wedge$$

$$(\exists \lambda : \lambda \geq 0 \wedge \lambda S = \begin{pmatrix} 1 & -1 & 0 \end{pmatrix} \wedge \lambda s \leq -1) \wedge$$

$$(\exists \Lambda : \Lambda \geq 0 \wedge \Lambda S = \begin{pmatrix} S_x & S_y & S_z + S_x \end{pmatrix} \wedge \Lambda s \leq (s - S_x))$$

Constraint on Recurrence Set, Solved

- Find ($p_x\, x + p_y\, y + p_z\, z \leq p_0$ /\ $q_x\, x + q_y\, y + q_z\, z \leq q_0$)

$$p = (1 \ -1\ 0) \qquad x = -2 \qquad \lambda = (1\ 0)$$

$$p_0 = -1 \qquad y = -1 \qquad \Lambda = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

$$q = (0\ 0\ 1) \qquad z = -1$$

$$q_0 = -1$$

- Non-terminating computation from (−2, −1, −1) not leaving (x+1 ≤ y) /\ (z ≤ −1)