Lecturecise 8
Recursion and Fixpoints
Algebraic Data Types Introduction

2013

Recall: Least fixpoint of a recursive function

Assumptions:

- C - a collection (set) of sets (e.g. sets of pairs, i.e. relations)
- $E : C \to C$ that is ω-continuous: for $r_0 \subseteq r_1 \subseteq r_2 \ldots$,

$$E(\bigcup_i r_i) = \bigcup_i E(r_i)$$

THEOREM: $s = \bigcup_i E^i(\emptyset)$ is such that

1. $E(s) = s$ — s is a **fixpoint** of E
2. if r is such that $E(r) \subseteq r$, then $s \subseteq r$
   — s is the smallest

We call s the **least fixpoint** of E and write $s = lfp(E)$
The least fixpoint is always unique: if s_1 and s_2 are least fixpoints, then $s_1 \subseteq s_2$ and $s_2 \subseteq s_1$, so $s_1 = s_2$

Example 1: Prove that recursive function meets spec

Prove that if s is the relation denoting the recursive function below, then

$$((x, y), (x', y')) \in s \rightarrow y' \geq y$$

```
def f =
  if (x > 0) {
    x = x − 1
    f
    y = y + 2
  }
```

$$
\begin{aligned}
E(r_f) = \; & (\Delta_{S(x>0)} \circ (\\
& \quad \rho(x = x - 1) \circ \\
& \quad r_f \circ \\
& \quad \rho(y = y + 2)) \\
&) \cup \Delta_{S(x \leq 0)}
\end{aligned}
$$

Example 1: Prove that recursive function meets spec

$$E(r_f) = (\Delta_{S(x>0)} \circ \big(\rho(x = x - 1) \circ r_f \circ \rho(y = y + 2)\big)) \cup \Delta_{S(x \leq 0)}$$

$$((x, y), (x', y')) \in s \rightarrow y' \geq y$$

Example 1: Prove that recursive function meets spec

$$E(r_f) = (\Delta_{S(x>0)} \circ \big(\rho(x = x - 1) \circ r_f \circ \rho(y = y + 2)\big)) \cup \Delta_{S(x \leq 0)}$$

$$((x, y), (x', y')) \in s \rightarrow y' \geq y$$

Since it holds

$$E(r) \subseteq r \rightarrow lfp(E) \subseteq r$$

we only need to show that for $r = \{((x, y), (x', y'))|y' \geq y\}$ it holds that $E(r) \subseteq r$.

Example 1: Prove that recursive function meets spec

$$E(r_f) = (\Delta_{S(x>0)} \circ (\rho(x = x - 1) \circ r_f \circ \rho(y = y + 2))) \cup \Delta_{S(x \leq 0)}$$

$$((x, y), (x', y')) \in s \rightarrow y' \geq y$$

Since it holds

$$E(r) \subseteq r \rightarrow lfp(E) \subseteq r$$

we only need to show that for $r = \{((x, y), (x', y'))|y' \geq y\}$ it holds that $E(r) \subseteq r$.

$$\begin{aligned} E(r) = \quad & \{((x, y), (x', y'))|\exists x_1, x_2, y_1, y_2 . x > 0 \wedge x_1 = x - 1 \wedge y_1 = y \wedge \\ & x_1 = x_2 \wedge y_2 > y_1 \wedge x' = x_2 \wedge y' = y_2 + 2\} \cup \Delta_{S(x>0)} \end{aligned}$$

Example 1: Prove that recursive function meets spec

$$E(r_f) = (\Delta_{S(x>0)} \circ (\rho(x = x - 1) \circ r_f \circ \rho(y = y + 2))) \cup \Delta_{S(x \leq 0)}$$

$$((x, y), (x', y')) \in s \rightarrow y' \geq y$$

Since it holds

$$E(r) \subseteq r \rightarrow lfp(E) \subseteq r$$

we only need to show that for $r = \{((x, y), (x', y'))|y' \geq y\}$ it holds that $E(r) \subseteq r$.

$$\begin{aligned} E(r) = \quad & \{((x, y), (x', y'))|\exists x_1, x_2, y_1, y_2 . x > 0 \wedge x_1 = x - 1 \wedge y_1 = y \wedge \\ & x_1 = x_2 \wedge y_2 > y_1 \wedge x' = x_2 \wedge y' = y_2 + 2\} \cup \Delta_{S(x>0)} \\ \subseteq \quad & r \end{aligned}$$

Example 2: Computing the least fixpoint is harder

Compute the least fixpoint of the recursive function:

```
def f =
  if (x > 0) {
    x = x − 1
    f
    y = y + 2
  }
```

$$E(r_f) = (\Delta_{S(x>0)} \circ (\rho(x = x - 1) \circ r_f \circ \rho(y = y + 2)) \cup \Delta_{S(x \leq 0)}$$

Computing the elements the sequence

$$E(r_f) = (\Delta_{S(x>0)} \circ (\rho(x = x - 1) \circ r_f \circ \rho(y = y + 2))) \cup \Delta_{S(x \leq 0)}$$

$$r_k = E^k(\emptyset)$$

Computing the elements the sequence

$$E(r_f) = (\Delta_{S(x>0)} \circ (\rho(x = x - 1) \circ r_f \circ \rho(y = y + 2))) \cup \Delta_{S(x \leq 0)}$$

$r_k = E^k(\emptyset)$
Plan:

1. Find a mathematical formula describing the relations r_k, containing x, x', y, y', k.
2. Find a mathematical formula for $\bigcup_{k \geq 0} r_k$

- $r_1 = E(\emptyset) =$

Computing the elements the sequence

$$E(r_f) = (\Delta_{S(x>0)} \circ (\rho(x = x - 1) \circ r_f \circ \rho(y = y + 2))) \cup \Delta_{S(x\leq 0)}$$

$r_k = E^k(\emptyset)$
Plan:

1. Find a mathematical formula describing the relations r_k, containing x, x', y, y', k.
2. Find a mathematical formula for $\bigcup_{k\geq 0} r_k$

- $r_1 = E(\emptyset) =$
  $\Delta_{S(x\leq 0)} = \{((x, y), (x', y')) \mid x \leq 0 \wedge x' = x \wedge y' = y\}$

Computing the elements the sequence

$$E(r_f) = (\Delta_{S(x>0)} \circ (\rho(x = x - 1) \circ r_f \circ \rho(y = y + 2))) \cup \Delta_{S(x\leq 0)}$$

$r_k = E^k(\emptyset)$
Plan:

1. Find a mathematical formula describing the relations r_k, containing x, x', y, y', k.
2. Find a mathematical formula for $\bigcup_{k\geq 0} r_k$

- $r_1 = E(\emptyset) =$
  $\Delta_{S(x\leq 0)} = \{((x, y), (x', y')) \mid x \leq 0 \wedge x' = x \wedge y' = y\}$
- $r_2 = E(\Delta_{S(x\leq 0)}) =$
  $(\Delta_{S(x>0)} \circ (\rho(x = x-1) \circ \Delta_{S(x\leq 0)} \circ \rho(y = y+2))) \cup \Delta_{S(x\leq 0)} =$

Computing the elements the sequence

$$E(r_f) = (\Delta_{S(x>0)} \circ (\rho(x = x - 1) \circ r_f \circ \rho(y = y + 2))) \cup \Delta_{S(x\leq 0)}$$

$r_k = E^k(\emptyset)$
Plan:

1. Find a mathematical formula describing the relations r_k, containing x, x', y, y', k.
2. Find a mathematical formula for $\bigcup_{k\geq 0} r_k$

- $r_1 = E(\emptyset) =$
  $\Delta_{S(x\leq 0)} = \{((x, y), (x', y')) \mid x \leq 0 \wedge x' = x \wedge y' = y\}$
- $r_2 = E(\Delta_{S(x\leq 0)}) =$
  $(\Delta_{S(x>0)} \circ (\rho(x = x-1) \circ \Delta_{S(x\leq 0)} \circ \rho(y = y+2))) \cup \Delta_{S(x\leq 0)} =$

$$\{((x, y), (x', y')) | \quad (x > 0 \wedge x' = x - 1 \wedge x - 1 \leq 0 \wedge y' = y + 2) \\ \vee (x \leq 0 \wedge x' = x \wedge y' = y)\}$$

Computing the elements the sequence

$$E(r_f) = (\Delta_{S(x>0)} \circ (\rho(x = x - 1) \circ r_f \circ \rho(y = y + 2))) \cup \Delta_{S(x\leq 0)}$$

$$r_2 = \{((x, y), (x', y'))|(x > 0 \wedge x' = x - 1 \wedge x - 1 \leq 0 \wedge y' = y + 2) \\ \vee (x \leq 0 \wedge x' = x \wedge y' = y)\}$$

$$r_3 = (\Delta_{S(x>0)} \circ (\rho(x = x - 1) \circ r_2 \circ \rho(y = y + 2))) \cup \Delta_{S(x\leq 0)}$$

$$= \{((x, y), (x', y'))|(x = 2 \wedge x' = x - 2 \wedge y' = y + 4) \\ \vee (x = 1 \wedge x' = x - 1 \wedge y' = y + 2) \\ \vee (x \leq 0 \wedge x' = x \wedge y' = y)\}$$

Proof by induction

$$r_k = \left\{ ((x,y),(x',y')) \middle| \bigvee_{i=0}^{k-1} F_i \right\}$$

F_0 : $x \leq 0 \wedge x' = x \wedge y' = y$, for $i > 0$:

$$F_i \equiv x = i \wedge x' = 0 \wedge y' = y + 2i$$

$$E(r_f) = (\Delta_{S(x>0)} \circ (\rho(x = x-1) \circ r_f \circ \rho(y = y+2))) \cup \Delta_{S(x\leq 0)}$$

$k = 1$: We computed
$r_1 = E(\emptyset) = \Delta_{S(x\leq 0)} = \{((x,y),(x',y')) \mid F_0\}$
Induction step:

$$E(r_k) = E\left(\left\{ ((x,y),(x',y')) \middle| \bigvee_{i=0}^{k-1} F_i \right\} \right)$$

Proof by induction

$$r_k = \left\{ ((x,y),(x',y')) \middle| \bigvee_{i=0}^{k-1} F_i \right\}$$

$F_0 : \quad x \leq 0 \wedge x' = x \wedge y' = y$, for $i > 0$:

$$F_i \equiv x = i \wedge x' = 0 \wedge y' = y + 2i$$

$$E(r_f) = (\Delta_{S(x>0)} \circ (\rho(x = x-1) \circ r_f \circ \rho(y = y+2))) \cup \Delta_{S(x\leq 0)}$$

$k = 1$: We computed
$r_1 = E(\emptyset) = \Delta_{S(x\leq 0)} = \{((x,y),(x',y')) \mid F_0\}$
Induction step:

$$\begin{aligned} E(r_k) &= E\left(\left\{ ((x,y),(x',y')) \middle| \bigvee_{i=0}^{k-1} F_i \right\} \right) \\ &= \bigcup_{i=1}^{k-1} E(\{((x,y),(x',y')) \mid F_i\} \end{aligned}$$

Proof by induction

$$r_k = \left\{ ((x,y),(x',y')) \middle| \bigvee_{i=0}^{k-1} F_i \right\}$$

F_0 : $x \leq 0 \wedge x' = x \wedge y' = y$, for $i > 0$:

$$F_i \equiv x = i \wedge x' = 0 \wedge y' = y + 2i$$

$$E(r_f) = (\Delta_{S(x>0)} \circ (\rho(x = x-1) \circ r_f \circ \rho(y = y+2))) \cup \Delta_{S(x\leq 0)}$$

$k = 1$: We computed
$r_1 = E(\emptyset) = \Delta_{S(x\leq 0)} = \{((x,y),(x',y')) \mid F_0\}$
Induction step:

$$\begin{aligned} E(r_k) &= E\left(\left\{ ((x,y),(x',y')) \middle| \textstyle\bigvee_{i=0}^{k-1} F_i \right\} \right) \\ &= \textstyle\bigcup_{i=1}^{k-1} E(\{((x,y),(x',y')) \mid F_i\} \\ &= \{((x,y),(x',y')) \mid \textstyle\bigvee_{i=1}^{k-1}(F_0 \vee F_{i+1})\} \end{aligned}$$

Proof by induction

$$r_k = \left\{ ((x,y),(x',y')) \middle| \bigvee_{i=0}^{k-1} F_i \right\}$$

F_0 : $x \leq 0 \wedge x' = x \wedge y' = y$, for $i > 0$:

$$F_i \equiv x = i \wedge x' = 0 \wedge y' = y + 2i$$

$$E(r_f) = (\Delta_{S(x>0)} \circ (\rho(x = x-1) \circ r_f \circ \rho(y = y+2))) \cup \Delta_{S(x\leq 0)}$$

$k = 1$: We computed
$r_1 = E(\emptyset) = \Delta_{S(x\leq 0)} = \{((x,y),(x',y')) \mid F_0\}$
Induction step:

$$\begin{aligned} E(r_k) &= E\left(\left\{ ((x,y),(x',y')) \middle| \textstyle\bigvee_{i=0}^{k-1} F_i \right\} \right) \\ &= \textstyle\bigcup_{i=1}^{k-1} E(\{((x,y),(x',y')) \mid F_i\} \\ &= \{((x,y),(x',y')) \mid \textstyle\bigvee_{i=1}^{k-1}(F_0 \vee F_{i+1})\} \\ &= \{((x,y),(x',y')) \mid \textstyle\bigvee_{i=0}^{k} F_i\} = r_{k+1} \end{aligned}$$

Constructing least fixpoint as union of sequence

$F_0 : \quad x \leq 0 \wedge x' = x \wedge y' = y$, for $i > 0$:

$$F_i \equiv x = i \wedge x' = 0 \wedge y' = y + 2i$$

By fixpoint theorem

$$s = \bigcup_{k=1}^{\infty} r_k = \bigcup_{i=1}^{\infty} \left\{ ((x, y), (x', y')) \middle| \bigvee_{i=0}^{k-1} F_i \right\}$$

$$=$$

Constructing least fixpoint as union of sequence

$F_0 : \quad x \leq 0 \wedge x' = x \wedge y' = y$, for $i > 0$:

$$F_i \equiv x = i \wedge x' = 0 \wedge y' = y + 2i$$

By fixpoint theorem

$$\begin{aligned} s &= \bigcup_{k=1}^{\infty} r_k = \bigcup_{i=1}^{\infty} \left\{ ((x,y),(x',y')) \middle| \bigvee_{i=0}^{k-1} F_i \right\} \\ &= \bigcup_{k=1}^{\infty} \left\{ ((x,y),(x',y')) \middle| F_i \right\} \\ &= \end{aligned}$$

Constructing least fixpoint as union of sequence

$F_0 : \quad x \leq 0 \wedge x' = x \wedge y' = y$, for $i > 0$:

$$F_i \equiv x = i \wedge x' = 0 \wedge y' = y + 2i$$

By fixpoint theorem

$$\begin{aligned} s &= \bigcup_{k=1}^{\infty} r_k = \bigcup_{i=1}^{\infty} \left\{ ((x,y),(x',y')) \middle| \bigvee_{i=0}^{k-1} F_i \right\} \\ &= \bigcup_{k=1}^{\infty} \left\{ ((x,y),(x',y')) \middle| F_i \right\} \\ &= \left\{ ((x,y),(x',y')) \middle| F_0 \vee \exists k.\ k > 0 \wedge F_k \right\} \end{aligned}$$

Constructing least fixpoint as union of sequence

$F_0 : \quad x \leq 0 \wedge x' = x \wedge y' = y$, for $i > 0$:

$$F_i \equiv x = i \wedge x' = 0 \wedge y' = y + 2i$$

By fixpoint theorem

$$\begin{aligned} s &= \bigcup_{k=1}^{\infty} r_k = \bigcup_{i=1}^{\infty} \left\{ ((x,y),(x',y')) \middle| \bigvee_{i=0}^{k-1} F_i \right\} \\ &= \bigcup_{k=1}^{\infty} \left\{ ((x,y),(x',y')) \middle| F_i \right\} \\ &= \left\{ ((x,y),(x',y')) \middle| F_0 \vee \exists k.\ k > 0 \wedge F_k \right\} \end{aligned}$$

$$\exists k.\ 0 < k \wedge F_k \equiv \exists k.\ 0 < k \wedge x = k \wedge x' = 0 \wedge y' = y + 2k$$

Constructing least fixpoint as union of sequence

$F_0 : \quad x \leq 0 \wedge x' = x \wedge y' = y$, for $i > 0$:

$$F_i \equiv x = i \wedge x' = 0 \wedge y' = y + 2i$$

By fixpoint theorem

$$\begin{aligned} s &= \bigcup_{k=1}^{\infty} r_k = \bigcup_{i=1}^{\infty} \left\{ ((x,y),(x',y')) \middle| \bigvee_{i=0}^{k-1} F_i \right\} \\ &= \bigcup_{k=1}^{\infty} \left\{ ((x,y),(x',y')) \middle| F_i \right\} \\ &= \left\{ ((x,y),(x',y')) \middle| F_0 \vee \exists k.\ k > 0 \wedge F_k \right\} \end{aligned}$$

$$\begin{aligned} \exists k.\ 0 < k \wedge F_k \equiv \exists k.\ 0 < k \wedge x = k \wedge x' = 0 \wedge y' = y + 2k \\ 0 < x \wedge x' = 0 \wedge y' = y + 2x \end{aligned}$$

Constructing least fixpoint as union of sequence

$F_0 : \quad x \leq 0 \wedge x' = x \wedge y' = y$, for $i > 0$:

$$F_i \equiv x = i \wedge x' = 0 \wedge y' = y + 2i$$

By fixpoint theorem

$$\begin{aligned} s &= \bigcup_{k=1}^{\infty} r_k = \bigcup_{i=1}^{\infty} \left\{ ((x,y),(x',y')) \middle| \bigvee_{i=0}^{k-1} F_i \right\} \\ &= \bigcup_{k=1}^{\infty} \left\{ ((x,y),(x',y')) \middle| F_i \right\} \\ &= \left\{ ((x,y),(x',y')) \middle| F_0 \vee \exists k.\ k > 0 \wedge F_k \right\} \end{aligned}$$

$$\begin{aligned} \exists k.\ 0 < k \wedge F_k \equiv\ & \exists k.\ 0 < k \wedge x = k \wedge x' = 0 \wedge y' = y + 2k \\ & 0 < x \wedge x' = 0 \wedge y' = y + 2x \end{aligned}$$

$$\begin{aligned} s = \{ ((x,y),(x',y')) \mid\ & (x \leq 0 \wedge x' = x \wedge y' = 0) \\ & \vee (0 < x \wedge x' = 0 \wedge y' = y + 2x) \} \end{aligned}$$

Alternative Acceptable Method

$$E(r_f) = \Delta_{S(x>0)} \circ \rho(x = x - 1) \circ r_f \circ \rho(y = y + 2) \ \cup \ \Delta_{S(x \leq 0)}$$

1. Guess a fixpoint, in this case:

$$s' = \{((x,y),(x',y')) \mid (x \leq 0 \wedge x' = x \wedge y' = 0) \vee (0 < x \wedge x' = 0 \wedge y' = y + 2x)\}$$

2. Verify $E(s') \subseteq s'$
3. Show that, for every initial state (x, y), if (x', y') is such that $((x,y),(x',y')) \in s$, then there is an n (e.g. number of execution steps) such that $((x,y),(x',y')) \in E^n(\emptyset)$.

Note that, if $s = lfp(E)$ then 2. implies $s \subseteq s'$ and 3. implies

$$s' \subseteq \bigcup_{n \geq 0} E^n(\emptyset) = s$$

Together, they imply $s = s'$, so the guessed s' is the least fixpoint.

Observation

It is much simpler to check that a procedure satisfies a specification

$$F(r) \subseteq r$$

Than to find the least s such that

$$F(s) = s$$

Replacing Calls with Specs

```
def f = if (x > 0) {
    x = x − 1
    f
    y = y + 2
}
ensuring (!(0 < old(x)) || (x == 0 && y == old(y) + 2∗old(x)))

def fNoRec = if (x > 0) {
    x = x − 1
    { var x0 = x, y0 = y
      havoc(x,y)
      assume(!(0 < x0) || (x == 0 && y == y0 + 2∗x0))
    }
    y = y + 2
  }
ensuring (!(0 < old(x)) || (x == 0 && y == old(y) + 2∗old(x)))
```

If fNoRec satisfies postcondition r, so does f

Replacing Calls with Specs

```
def f = if (x > 0) {
    x = x − 1
    f
    y = y + 2
}
ensuring (!(0 < old(x)) || (x == 0 && y == old(y) + 2∗old(x)))

def fNoRec = if (x > 0) {
    x = x − 1
    { var x0 = x, y0 = y
      havoc(x,y)
      assume(!(0 < x0) || (x == 0 && y == y0 + 2∗x0))
    }
    y = y + 2
  }
ensuring (!(0 < old(x)) || (x == 0 && y == old(y) + 2∗old(x)))
```

If fNoRec satisfies postcondition r, so does f

Reason: if fNoRec verifies, then $E(r) \subseteq r$, so $lfp(E) \subseteq r$

Least Fixpoint Reasoning Rules

1. $$E(lfp(E)) = E$$

2. $$\frac{E(r) \subseteq r}{lfp(E) \subseteq r}$$

Example 3: Multiple Fixpoints

Previous example tested if $x > 0$. Now we test $x! = 0$.

```
def g =
  if (x != 0) {
    x = x − 1
    g
    y = y + 2
  }
```

$$
\begin{aligned}
E'(r_g) = \; & (\Delta_{S(x \neq 0)} \circ (\\
& \quad \rho(x = x - 1) \circ \\
& \quad r_g \circ \\
& \quad \rho(y = y + 2)) \\
&) \cup \Delta_{S(x=0)}
\end{aligned}
$$

Example 3: Multiple Fixpoints

Previous example tested if $x > 0$. Now we test $x! = 0$.

```
def g =
  if (x != 0) {
    x = x − 1
    g
    y = y + 2
  }
```

$$E'(r_g) = (\Delta_{S(x\neq 0)} \circ (\rho(x = x - 1) \circ r_g \circ \rho(y = y + 2))) \cup \Delta_{S(x=0)}$$

What does g do when called in state where $x < 0$?

Example 3: Multiple Fixpoints

Previous example tested if $x > 0$. Now we test $x! = 0$.

```
def g =
  if (x != 0) {
    x = x − 1
    g
    y = y + 2
  }
```

$$E'(r_g) = (\Delta_{S(x\neq 0)} \circ (\rho(x = x - 1) \circ r_g \circ \rho(y = y + 2))) \cup \Delta_{S(x=0)}$$

What does g do when called in state where $x < 0$?

TASK: Find two different fixpoint relations: s_1 and s_2 where $s_1 \neq s_2$, $E'(s_1) = s_1$, and $E'(s_2) = s_2$.

Finding two Fixpoints

```
def g =
  if (x != 0) {
    x = x − 1
    g
    y = y + 2
  }
```

$$
\begin{aligned}
E'(r_g) = \quad & (\Delta_{S(x \neq 0)} \circ (\\
& \quad \rho(x = x - 1) \circ \\
& \quad r_g \circ \\
& \quad \rho(y = y + 2)) \\
&) \cup \Delta_{S(x=0)}
\end{aligned}
$$

Finding two Fixpoints

```
def g =
  if (x != 0) {
    x = x − 1
    g
    y = y + 2
  }
```

$$E'(r_g) = (\Delta_{S(x\neq 0)} \circ (\rho(x = x - 1) \circ r_g \circ \rho(y = y + 2))) \cup \Delta_{S(x=0)}$$

$$s_1 = \{((x,y),(x',y')) \mid (x = 0 \wedge x' = 0 \wedge y' = y) \vee (0 < x \wedge x' = 0 \wedge y' = y + 2x)\}$$

$$s_2 = s_1 \cup \{((x,y),(x',y')) \mid x < 0\}$$

Finding two Fixpoints

```
def g =
  if (x != 0) {
    x = x − 1
    g
    y = y + 2
  }
```

$$E'(r_g) = (\Delta_{S(x\neq 0)} \circ (\rho(x = x - 1) \circ r_g \circ \rho(y = y + 2))) \cup \Delta_{S(x=0)}$$

$$s_1 = \{((x,y),(x',y')) \mid (x = 0 \wedge x' = 0 \wedge y' = y) \vee (0 < x \wedge x' = 0 \wedge y' = y + 2x)\}$$

$$s_2 = s_1 \cup \{((x,y),(x',y')) \mid x < 0\}$$

Multiple fixpoints differ in pairs of states $((x,y),(x',y'))$ for which execution from (x,y) does not terminate.

Finding two Fixpoints

```
def g =
  if (x != 0) {
    x = x − 1
    g
    y = y + 2
  }
```

$$E'(r_g) = (\Delta_{S(x\neq 0)} \circ (\rho(x = x-1) \circ r_g \circ \rho(y = y+2)) \cup \Delta_{S(x=0)}$$

$$s_1 = \{((x,y),(x',y')) \mid (x = 0 \wedge x' = 0 \wedge y' = y) \vee (0 < x \wedge x' = 0 \wedge y' = y + 2x)\}$$

$$s_2 = s_1 \cup \{((x,y),(x',y')) \mid x < 0\}$$

Multiple fixpoints differ in pairs of states $((x,y),(x',y'))$ for which execution from (x,y) does not terminate.

Least fixpoint (s_1) contains no states for which execution from (x,y) does not terminate.

Finding two Fixpoints

```
def g =
  if (x != 0) {
    x = x − 1
    g
    y = y + 2
  }
```

$$E'(r_g) = (\Delta_{S(x\neq 0)} \circ (\rho(x = x-1) \circ r_g \circ \rho(y = y+2)) \cup \Delta_{S(x=0)}$$

$$s_1 = \{((x,y),(x',y')) \mid (x = 0 \wedge x' = 0 \wedge y' = y) \vee (0 < x \wedge x' = 0 \wedge y' = y + 2x)\}$$

$$s_2 = s_1 \cup \{((x,y),(x',y')) \mid x < 0\}$$

Multiple fixpoints differ in pairs of states $((x,y),(x',y'))$ for which execution from (x,y) does not terminate.

Least fixpoint (s_1) contains no states for which execution from (x,y) does not terminate.

Can we put any junk for non-terminating states and it will be fixpoint?

Fixpoints

Suppose we assign y to 2.

$$
\begin{aligned}
E''(r_g) = \quad & (\Delta_{S(x \neq 0)} \circ (\\
& \quad \rho(x = x - 1) \circ \\
& \quad r_g \circ \\
& \quad \rho(y = 2)) \\
&) \cup \Delta_{S(x=0)}
\end{aligned}
$$

Is s_3 a fixpoint of E'':

$$
\begin{aligned}
s_3 \quad = \{((x, y), (x', y')) \mid \; & (x < 0) \\
& \vee (x = 0 \wedge x' = 0 \wedge y' = y) \\
& \vee (0 < x \wedge x' = 0 \wedge y' = 2)\}
\end{aligned}
$$

Fixpoints

Suppose we assign y to 2.

$$\begin{aligned} E''(r_g) = \; & (\Delta_{S(x\neq 0)} \circ (\\ & \rho(x = x - 1) \circ \\ & r_g \circ \\ & \rho(y = 2)) \\ &) \cup \Delta_{S(x=0)} \end{aligned}$$

Is s_3 a fixpoint of E'':

$$\begin{aligned} s_3 \quad = \{((x,y),(x',y'))\big| \; & (x < 0) \\ & \vee (x = 0 \wedge x' = 0 \wedge y' = y) \\ & \vee (0 < x \wedge x' = 0 \wedge y' = 2)\} \end{aligned}$$

Does $((-1,0),(-1,0)) \in s_3$ hold?

Fixpoints

Suppose we assign y to 2.

$$\begin{aligned} E''(r_g) = \; & (\Delta_{S(x\neq 0)} \circ (\\ & \rho(x = x - 1) \circ \\ & r_g \circ \\ & \rho(y = 2)) \\ &) \cup \Delta_{S(x=0)} \end{aligned}$$

Is s_3 a fixpoint of E'':

$$\begin{aligned} s_3 \quad = \{((x,y),(x',y'))\big| \; & (x < 0) \\ & \vee (x = 0 \wedge x' = 0 \wedge y' = y) \\ & \vee (0 < x \wedge x' = 0 \wedge y' = 2)\} \end{aligned}$$

Does $((-1,0),(-1,0)) \in s_3$ hold?
Does $((-1,0),(-1,0)) \in E''(s_3)$ hold?

Fixpoints

Suppose we assign y to 2.

$$\begin{aligned} E''(r_g) = \; & (\Delta_{S(x\neq 0)} \circ (\\ & \rho(x = x - 1) \circ \\ & r_g \circ \\ & \rho(y = 2)) \\ &) \cup \Delta_{S(x=0)} \end{aligned}$$

Is s_3 a fixpoint of E'':

$$\begin{aligned} s_3 \quad = \{((x,y),(x',y'))\big| \; & (x < 0) \\ & \vee (x = 0 \wedge x' = 0 \wedge y' = y) \\ & \vee (0 < x \wedge x' = 0 \wedge y' = 2)\} \end{aligned}$$

Does $((-1,0),(-1,0)) \in s_3$ hold?
Does $((-1,0),(-1,0)) \in E''(s_3)$ hold?
Find a non-least fixpoint of E''

More on Inductively Defined Sets

Induction Principle = Set is the Least Fixpoint

Set of natural numbers N is defined like this:

- $0 \in N$
- if $x \in N$ then $x + 1 \in N$

Following this definition, define function F from sets to sets:

$$F(S) = \{0\} \cup \{x + 1 \mid x \in S\}$$

Then the definition above says $F(N) \subseteq N$.

Induction Principle = Set is the Least Fixpoint

Set of natural numbers N is defined like this:

- $0 \in N$
- if $x \in N$ then $x + 1 \in N$

Following this definition, define function F from sets to sets:

$$F(S) = \{0\} \cup \{x + 1 \mid x \in S\}$$

Then the definition above says $F(N) \subseteq N$.
What definition **really means** is

1. $F(N) \subseteq N$
2. for any set P such that $F(P) \subseteq P$, we have $N \subseteq P$

This of set P as a property that we wish to show holds for all natural numbers. The fact that N is the least set means it suffices to show

$$\{0\} \cup \{x + 1 \mid x \in P\} \subseteq P$$

that is, $0 \in P$ and $x + 1 \in P$ for every $x \in P$. From there $N \subseteq P$

Least Fixpoints as Definition Mechanism

$$F(S) = \{0\} \cup \{x + 1 \mid x \in S\}$$

Then $F(\emptyset) = \{0\}$. Generally, $F^k(\emptyset) = \{0, 1, \ldots, k\}$
Least fixpoint is union over all $F^k(\emptyset)$, set of natural numbers.

$$G(S) = \{1\} \cup \{x + 2 \mid x \in S\}$$

What is $G^k(S) =$

Least Fixpoints as Definition Mechanism

$$F(S) = \{0\} \cup \{x + 1 \mid x \in S\}$$

Then $F(\emptyset) = \{0\}$. Generally, $F^k(\emptyset) = \{0, 1, \ldots, k\}$
Least fixpoint is union over all $F^k(\emptyset)$, set of natural numbers.

$$G(S) = \{1\} \cup \{x + 2 \mid x \in S\}$$

What is $G^k(S) = \{1, 3, \ldots, 2k + 1\}$
What is least fixpoint of G?

Least Fixpoints as Definition Mechanism

$$F(S) = \{0\} \cup \{x + 1 \mid x \in S\}$$

Then $F(\emptyset) = \{0\}$. Generally, $F^k(\emptyset) = \{0, 1, \ldots, k\}$
Least fixpoint is union over all $F^k(\emptyset)$, set of natural numbers.

$$G(S) = \{1\} \cup \{x + 2 \mid x \in S\}$$

What is $G^k(S) = \{1, 3, \ldots, 2k + 1\}$
What is least fixpoint of G? Set of positive odd numbers
$A = \{1, 3, 5, 7, \ldots\}$

Least Fixpoints as Definition Mechanism

$$F(S) = \{0\} \cup \{x + 1 \mid x \in S\}$$

Then $F(\emptyset) = \{0\}$. Generally, $F^k(\emptyset) = \{0, 1, \ldots, k\}$
Least fixpoint is union over all $F^k(\emptyset)$, set of natural numbers.

$$G(S) = \{1\} \cup \{x + 2 \mid x \in S\}$$

What is $G^k(S) = \{1, 3, \ldots, 2k + 1\}$
What is least fixpoint of G? Set of positive odd numbers
$A = \{1, 3, 5, 7, \ldots\}$

Every well-behaved function F gives a recursive definition and the corresponding recursion principle.

- a theorem guarantees that the object being defined exists (it is the least fixpoint s of F)
- being the least implies we can establish approximation r of s by showing approximation satisfies $F(r) \subseteq r$

Proofs by induction = age-old approximation effort.

Non-standard Models

$$F(S) = \{0\} \cup \{e + 1 \mid e \in S\}$$

If our domain allows real numbers then also $F(\mathbb{R}) = \mathbb{R}$.
So, the set of real numbers is also a fixpoint of S, but not the least one.

Another example of non-least model: take the set of all polynomials $ax + b$ where a, b are integers and where x is a formal variable. Constants are just $0 \cdot x + c$

- $(a_1x + b_1) + (a_2x + b_2) = ((a_1 + a_2)x + (b_1 + b_2))$
- $(a_1x + b_1) < (a_2x + b_2)$ iff $a_1 < a_2 \vee (a_1 = a_2 \wedge b_1 < b_2)$

Multiplication by constant is repeated addition, divisibility similar
Does this set and operations satisfy properties of Presburger arithmetic? If yes, we call the result a non-standard model of Presburger arithmetic. Satisfies same formulas.

Algebraic Data Types

Example: Shallow Tree Flip

```
case object Leaf extends Tree
case class Node(t1:Tree, x:BigInt, t2:Tree) extends Tree
def flip(t:Tree):Tree = t match {
  case Leaf => Leaf
  case Node(st1,x1,st2) => Node(st2,x1,st1)
}
def test(t:Tree):Boolean = { flip(flip(t)) == t }
```

Negated verification condition

$$t1 = \mathit{flipBody}(t) \wedge t2 = \mathit{flipBody}[t := t1] \wedge t2 \neq t$$

We would like to prove it is not satisfiable.
Is there a decision procedure to do this?

Inductive Definition of Binary Trees of Integers

case object Leaf **extends** Tree
case class Node(t1:Tree, x:BigInt, t2:Tree) **extends** Tree

$\mathbb{Z}$ - integers
$Trees = lfp(F)$ where

$$F(S) = \{Leaf\} \cup \{Node(t_1, x, t_2) \mid t_1 \in S, x \in \mathbb{Z}, t_2 \in S\}$$

If we know neither fixpoints nor Scala, we may try to say stuff like:
Trees are constructed using the following rules:

1. Leaf is a tree.
2. if t_1 is a tree, x is an integer, and t_2 is a tree, then $Node(t_1, x, t_2)$ is a tree.

"Nothing else is a tree."
"Tree is generated using only the rules above."

Algebraic Data Types, also known as Term Algebras

case object Leaf **extends** Tree
case object Flower **extends** Tree
case object Spike **extends** Tree
case class Node(t1:Tree,t2:Tree) **extends** Tree
case class Succ(t3:Tree) **extends** Tree
case class Oak(t4:Tree,t5:Tree,t6:Tree) **extends** Tree
case class Pine(t7:Tree,t8:Tree) **extends** Tree
Oak(Oak(Leaf,Leaf,Flower),Node(Succ(Leaf),Leaf),Pine(Spike,Spike)) : Tree

$$\{Leaf\} \subseteq Tree$$
$$\ldots$$
$$\{Node(t1, t2) \mid t1 \in Tree, t2 \in Tree\} \subseteq Tree$$
$$\{Succ(t3) \mid t3 \in Tree\} \subseteq Tree$$
$$\ldots$$

Collect LHSs, we obtain F s.t. above is same as $F(Tree) \subseteq Tree$
Constructors: Leaf, Flower, Spike, Node, Succ, Oak, Pine
Selectors: t1,t2,...,t8

Term Algebras

Σ - (finite) set of constructors, $f \in \Sigma$ has arity $ar(f) \geq 0$
If $ar(f) = 0$ then f is constant, $ar(f) = 1$: unary function,
$ar(f) = 2$: binary
Set of (ground) terms (trees) $Terms_\Sigma$ is least set S such that

$$\{f(t_1, \ldots, t_n) \mid n = ar(f), t_1, \ldots, t_n \in S\} \subseteq S$$

Example: Let $\Sigma = \{f, c\}$ with $ar(f) = 1$, $ar(c) = 0$.

$$Terms_\Sigma = \{c, f(c), f(f(c)), \ldots\}$$

Comparison to integers

	integers	*terms*
domain	$\mathbb{Z}$	*Terms*
constants	$0, 1, \ldots$	$\{f \mid ar(f) = 0\}$
operations	$+, -$	$\{f \mid ar(f) > 0\}$
relations	$=, <, \mid$	$=$

Example: if we apply f to term $f(c)$ we obtain bigger term $f(f(c))$

Properties of Term Algebras

$$f(t_1, \ldots, t_n) \neq g(s_1, \ldots, s_m), \quad \text{if } f \neq g$$

$$f(t_1, \ldots, t_n) = f(s_1, \ldots, s_n), \quad \text{iff } \bigwedge_{i=1}^{n} t_i = s_i$$

Clearly if t_1 is contained as a term inside t_2, then they are distinct. Therefore, it cannot be the case that e.g. $f(f(f(x))) = x$

Term Algebra Constraints

Equations in Presburger arithmetic are equalities that contain constants, operations, and uknowns like

$$3x + 2y = 7$$

Here we also have equations that contain constants and operations, like

$$Node(x, y) = Node(y, x)$$

Observe that the above constraint is equivalent to $x = y$

We can solve constraints in term algebra using unification

Unification Algorithm

A set of equations is in *solved form* if it is of the form $\{x_1 \doteq t_1, \ldots, x_n \doteq t_n\}$ where variables x_i do not appear in terms t_j, that is $\{x_1, \ldots, x_n\} \cap (FV(t_1) \cup \ldots FV(t_n)) = \emptyset$
We obtain a solved form in finite time using the algorithm that applies the following rules in any order as long as no clash is reported and as long as the equations are not in solved form.

- **Orient**: Select $t \doteq x$ where t is not x, and replace it with $x \doteq t$.
- **Delete**: Select $x \doteq x$, remove it.
- **Eliminate**: Given $x \doteq t$ where x does not occur in t, substitute x with t in all remaining equations.
- **Occurs Check**: Given $x \doteq t$ where x occurs in t, report clash.
- **Decomposition**: Given $f(t_1, \ldots, t_n) \doteq f(s_1, \ldots, s_n)$, replace it with $t_1 \doteq s_1, \ldots, t_n \doteq s_n$.
- **Clash**: Given $f(t_1, \ldots, t_n) \doteq g(s_1, \ldots, s_m)$ for f not g, report clash

Run Unification Algorithm

$\Sigma = \{h, f, a, b\}$ with arities $2, 2, 0, 0$

$$h(x, f(x, y)) = h(f(a, v), f(f(u, b), f(u, u)))$$

$$h(x, f(x, x)) = h(f(a, v), f(f(u, b), f(u, u)))$$

$$h(x, f(x, y)) = h(f(u, v), v)$$

Example from Verification

$\Sigma = \{Leaf, Node\}$, $ar(Leaf) = 0, ar(Node) = 2$
Consider 'flip' of a tree invoked twice $z_1 \rightsquigarrow z_2 \rightsquigarrow z_3$
Show that the following implication holds for all variables
$z_1, z_2, z_3, x_1, y_1, x_2, y_2$ whose values range over $Terms_\Sigma$

$$(((z_1 = Leaf \wedge z_2 = Leaf) \vee (z_1 = Node(x_1, y_1) \wedge z_2 = Node(y_1, x_1)))$$
$$\wedge((z_3 = Leaf \wedge z_3 = Leaf) \vee (z_2 = Node(x_2, y_2) \wedge z_3 = Node(y_2, x_3))))$$
$$\rightarrow z_3 = z_1$$

Unification Algorithm: Consequences

Solved form describes all solutions

How to handle disequalities?

How to handle disjunctions?

Can we also support quantifiers?