The program termination problem a.k.a. (uniform) halting problem:

*Given any computer program, *decide* whether the program will finish running or could run forever*

*Decide* is used in the technical sense

- Use only a finite amount of time
- Return either “yes” or “no”
Introduction

The program termination problem a.k.a. (uniform) halting problem:

Given any computer program, decide whether its transition relation is well founded

Decide is used in the technical sense

- Use only a finite amount of time
- Return either “yes” or “no”
Myth: It is *always impossible to* prove terminating programs terminating

Truth: It is *impossible to always* prove terminating programs terminating
Introduction

Myth: It is always impossible to prove terminating programs terminating.

Truth: It is impossible to always prove terminating programs terminating.

while(n>1) {
    if (n % 2 == 0) {
        n = n/2;
    } else {
        n = 3*n +1;
    }
}
Introduction

- The program termination problem *a.k.a.* (uniform) halting problem:

  Given any computer program, *decide* whether its transition relation is well founded

- *Decide* is used in the technical sense
  - Use only a finite amount of time
  - Return either “yes” or “no”
Introduction

→ The program termination problem a.k.a. (uniform) halting problem:


Given any computer program, decide whether its transition relation is well founded

Try to

→ Decide is used in the technical sense

▪ Use only a finite amount of time
▪ Return either “yes” or “no” or “don’t know”
Automatically discovering abstraction is key to a solution.

Try to decide with abstraction
- Use only a finite amount of time
- Return either “yes”, “no”, or “don’t know”
Automatically discovering abstraction is key to a solution.

Try to decide with abstraction
- Use only a finite amount of time
- Return either “yes”, “no”, or “don’t know”
Automatically discovering abstraction is key to a solution.

Try to decide with abstraction
- Use only a finite amount of time
- Return either "yes", "no", or "don't know"

Underapproximation
Termination is a matter of practical importance:

- Liveness: *Is every call to AcquireLock() followed by a call to ReleaseLock()*?

- Pure termination: *Does the mouse driver’s dispatch routine always return control back to the OS*?

Recent advances allow us to prove termination in many practical cases.
A Year Ticks Over, and Zunes Get Hiccups

BY JENA WORTHAM
Published: December 31, 2008

It has been nine years since the Y2K computer glitch inspired apocalyptic fears. On Wednesday, another time-related bug created its own small-scale panic. For some owners of the Zune, the portable media player that is Microsoft’s answer to the Apple iPod, it was a day without Kanye West and Girl Talk.

Owners of 30-gigabyte Zunes began flooding Zune-related Web sites with complaints early Wednesday morning. They said their players had suddenly stopped working, displaying only a frozen start-up screen.

After spending much of the day digging into the problem, Microsoft said that it had traced it to a software bug “related to the way the device handles a leap year.” Apparently the Zune was expecting 2008 to have 365 days, not 366.

The fix for the glitch? Patience. The company said the internal clock on the players should reset itself at 7 a.m. Eastern time on Thursday. Microsoft advised Zune owners to drain the battery and then turn the players back on after that time. Those who were hoping to provide the soundtrack to New Year’s Eve parties had no choice but to find a friend with an iPod.

Some Zune owners, like Geoffrey House, a 55-year-old entrepreneur in Las Vegas, were frustrated by the breakdown. “It’s inexcusable,” Mr. House said. “It’s surprising that..."
A Year Ticks Over, and Zunes Get Hiccups

By JENNA WORTHAM
Published: December 31, 2008

It has been nine years since the Y2K computer glitch inspired apocalyptic fears. On Wednesday, another time-related bug created its own small-scale panic. For some owners of the Zune, the portable media player that is Microsoft’s answer to the Apple iPod, it was a day without Kanye West and Girl Talk.

Owners of 30-gigabyte Zunes began flooding Zune-related Web sites with complaints early Wednesday morning. They said their players had suddenly stopped working, displaying only a frozen start-up screen.

After spending much of the day digging into the problem, Microsoft said that it had traced it to a software bug “related to the way the device handles a leap year.” Apparently the
A Year Ticks Over, and Zunes Get Hiccups

Glitch freezes Microsoft Zune music devices

By Jenna Wortham

It has been a year since Microsoft Corp.'s Zune digital music player became an option for music lovers. The device, aimed at the same niche as Apple Inc.'s iPod, has struggled to find a place with consumers, and its sales have barely moved the needle.

After spending months and thousands of dollars, Apple has a near monopoly on digital music. Microsoft, which entered the market a year ago, has so far failed to gain much traction. The company has yet to release a version of its Zune software for desktop computers, but it has announced plans to do so before the end of the year.

Despite the challenges, some analysts believe that Microsoft's Zune is a good product, but that it has failed to take off because of marketing and distribution issues. Others say that the device is too expensive, and that it lacks the features that consumers expect.

Zune's sales have been disappointing, but they have not been entirely unexpected. Microsoft has been more successful with its Xbox gaming systems, and Zune is seen as a complementary product.

The company has also been working to improve the software for its Zune, and has recently released a new version that includes better support for streaming music. But many analysts believe that Microsoft needs to do more to make its Zune a viable competitor in the digital music market.
BOOL ConvertDays(UINT32 days, SYSTEMTIME *lpTime)
{
    int dayOfWeek, month, year;
    UINT8 *month_tab;

    // Calculate current day of the week
    dayOfWeek = GetDayOfWeek(days);
    year = ORIGIN_YEAR;

    while (days > 365)
    {
        if (IsLeapYear(year))
        {
            if (days > 366)
            {
                days -= 366;
                year += 1;
            }
        }
        else
        {
            days -= 365;
            year += 1;
        }
    }

    // Determine whether it is a leap year
    month_tab = (UINT8 *)((IsLeapYear(year)) ? monthtable_leap : monthtable);

    // Convert days to months and days
    month = days / 30;
    days %= 30;

    // Convert months to days
    month = month * 30;
    month += days;

    // Convert months to weeks
    month += daysOfWeek;

    // Convert weeks to months
    month = month / 4;

    // Convert months to years
    year = month / 12;
}

Related Bits: The Hard Truth on Microsoft's Zune Music Devices

As of last week, Microsoft has acknowledged there was a problem with some Zunes and said customers it was working to address it. Later, the company said engineers had identified that the cause of the problem was an issue with the device’s internal clock driver and how it handles a leap year, such as 2008.

The issue relates to a part of the Zune hardware, and only affects the 30-gigabyte Zunes, a Microsoft spokeswoman said. The issue, Microsoft said, would resolve itself as the device ages.

Microsoft initially acknowledged there was a problem with some Zunes and said customers were working to address it. Later, the company said engineers had identified that the cause of the problem was an issue with the device’s internal clock driver and how it handles a leap year, such as 2008.

The issue relates to a part of the Zune hardware, and only affects the 30-gigabyte Zunes, a Microsoft spokeswoman said. The issue, Microsoft said, would resolve itself as the device ages.

Zune music devices account for a very small proportion of Microsoft’s roughly $60 billion in annual sales and its market share is dwarfed by that of Apple, whose iPods have more than 70% of the market, according to NPD figures.

Matt Rosoff, an analyst with Directions on Microsoft, a Seattle-based research firm that tracks the company, said the Zune machine underscored the device’s weakness in the market. Microsoft has “missed the boat” in the market for digital music players, Mr. Rosoff said. “That game is over.”

Rosoff expects the company to shift its strategy early in 2009 to focus on improving the Zune software and service offerings, and joining with mobile-phone handset makers.

The Zune problem closed out a year in which Microsoft’s problems had been more prominent than its successes. Microsoft shares lost around 45% in 2008. They closed Wednesday up 0.5% at $19.44.

The company is still struggling to excite customers about its latest operating system, Windows Vista. That may force the company to release Vista’s successor, Windows 7, earlier than initially planned.

Microsoft has also failed to gain traction in building market share for its Internet-search advertising operations, an area Chief Executive Steve Ballmer has indicated is a priority for the company.
Do you see the bug?
for (entry = DeviceExtension->ReadQueue.Flink; 
    entry != &DeviceExtension->ReadQueue; 
    entry = entry->Flink) {

    irp = CONTAINING_RECORD (entry, IRP, Tail.Overlay.ListEntry);
    stack = IoGetCurrentIrpStackLocation (irp);
    if (stack->FileObject == fileObject) {
        RemoveEntryList (entry);

        oldCancelRoutine = IoSetCancelRoutine (irp, NULL);

        //
        // IoCancelIrp() could have just been called on this IRP.
        // What we're interested in is not whether IoCancelIrp() was called
        // (i.e., nextIrp->Cancel is set), but whether IoCancelIrp() calls
        // is about to call) our cancel routine. To check that, check to
        // of the test-and-set macro IoSetCancelRoutine.
        //
        if (oldCancelRoutine) {
            //
            // Cancel routine not called for this IRP. Return this IRP.
            //
            return irp;
        }
        else {
            //
            // This IRP was just cancelled and the cancel routine was (or will
            // be) called. The cancel routine will complete this IRP as soon as
            // we drop the spinlock. So don't do anything with the IRP.
            //
            // Also, the cancel routine will try to dequeue the IRP, so make the
            // IRP's listEntry point to itself.
            //
            ASSERT (irp->Cancel);
            InitializeListHead (&irp->Tail.Overlay.ListEntry);
        }
    }
}
for (entry = DeviceExtension->ReadQueue.Flink;
    entry != &DeviceExtension->ReadQueue;
    entry = entry->Flink) {
    irp = CONTAINING_RECORD(entry, IRP, Tail.Overlay.ListEntry);
    stack = IoGetCurrentIrpStackLocation (irp);
    if (stack->FileObject == fileObject) {
        RemoveEntryList (entry);
        oldCancelRoutine = IoSetCancelRoutine (irp, NULL);
        //
        // IoCancelIrp() could have just been called on this IRP.
        // What we're interested in is not whether IoCancelIrp() was called
        // (ie, nextIrp->Cancel is set), but whether IoCancelIrp() called
        // is about to call) our cancel routine. To check that, check to
        // of the test-and-set macro IoSetCancelRoutine.
        //
        //
        if (oldCancelRoutine) {
            //
            // Cancel routine not called for this IRP. Return this IRP.
            //
            return irp;
        }
        else {
            //
            // This IRP was just cancelled and the cancel routine was (or will
            // be) called. The cancel routine will complete this IRP as soon as
            // we drop the spinlock. So don't do anything with the IRP.
            //
            // Also, the cancel routine will try to dequeue the IRP, so make the
            // IRP's listEntry point to itself.
            //
            ASSERT (irp->Cancel);
            InitializeListHead (&irp->Tail.Overlay.ListEntry);
            }
        }
    
    }
Perhaps recent advances will help unlock solutions to other problems

- Search for thread-scheduling that guarantees termination (operating systems)
- Wang’s tiling problem (graphics)
- Synthesis of compounds that kill targeted cells (medicine)
- ........
Outline of lectures

- Lecture 1:
  - Principles
  - Rank function synthesis

- Lecture 2:
  - Checking & refinement
  - Termination analysis

- Lecture 3:
  - Recursion
  - WP synthesis
  - Non-termination

- Lecture 4:
  - Fair termination
  - Data structures
  - Concurrency
Outline

- Introduction
- Well-founded relations and ranking functions
- Disjunctive well-foundedness
- Decomposition
- Notes on rank function synthesis
Outline

→ Introduction

→ Well-founded relations and ranking functions

→ Disjunctive well foundedness

→ Decomposition

→ Notes on rank function synthesis

References:

- Cantor
- Turing
- Jones et al.
- Podelski & Rybalchenko
- Colon & Sipma
- Floyd
Introduction

Well-founded relations and ranking functions

Disjunctive well foundedness

Decomposition

Notes on rank function synthesis
Well-founded relations
Well-founded relations
Well-founded relations
Well-founded relations do not permit infinite sequences
Well-founded relations

$R \subseteq S \times S$ is well-founded if for any infinite $S$-sequence $s_0, s_1, \ldots$ there exists a $j$ such that $(s_j, s_{j+1}) \not\in R$.

Well-founded relations do not permit infinite sequences
Well-founded relations

$$R \triangleq x' \geq x + 1 \land x < 100$$
Well-founded relations

\[ R \triangleq \{(s, t) \mid t(x) \geq s(x) + 1 \land s(x) < 100\} \]

\[ R \triangleq x' \geq x + 1 \land x < 100 \]
Well-founded relations

\[ R \triangleq \left[ x' \geq x + 1 \land x < 100 \right] \]

\[ R \triangleq \{(s, t) \mid t(x) \geq s(x) + 1 \land s(x) < 100\} \]

\[ R \triangleq x' \geq x + 1 \land x < 100 \]
Questions

- If $R$ is WF, is $R; R$ WF?

- If $R; R$ is WF, is $R$ WF?

- If $R$ is WF and $R \subseteq R'$, is $R$ WF?

- If $R$ is WF and $R' \subseteq R$, is $R$ WF?
Questions

\[ R^n \triangleq \begin{cases} R; R^{n-1} & \text{if } n \geq 1 \\ \text{ID} & \text{otherwise} \end{cases} \]

\[ R^+ \triangleq \{ (s, t) \mid \exists n > 0. (s, t) \in R^n \} \]

\[ R^* \triangleq R^+ \cup \text{ID} \]
If $R$ is WF, is $R^*$ WF? What about $R^+$?

\[
R^n \triangleq \begin{cases} 
R; R^{n-1} & \text{if } n \geq 1 \\
\text{ID} & \text{otherwise}
\end{cases}
\]

\[
R^+ \triangleq \{(s, t) \mid \exists n > 0. (s, t) \in R^n\}
\]

\[
R^* \triangleq R^+ \cup \text{ID}
\]
If $R^+$ is WF, is $R$ WF? What about $R^*$?

\[
R^n \triangleq \begin{cases} 
R; R^{n-1} & \text{if } n \geq 1 \\
\text{ID} & \text{otherwise}
\end{cases}
\]

\[
R^+ \triangleq \{(s, t) \mid \exists n > 0. (s, t) \in R^n \}
\]

\[
R^* \triangleq R^+ \cup \text{ID}
\]
Outline

- Introduction

- Well-founded relations and ranking functions

- Disjunctive well foundedness

- Decomposition

- Notes on rank function synthesis
Well-ordered sets

$(S, \leq)$ is a well-ordered set iff it is

- reflexive $(a \leq a)$
- antisymmetric $(a \leq b \land b \leq a \Rightarrow a = b)$
- transitive $(a \leq b \land b \leq c \Rightarrow a \leq c)$
- comparable $(a \leq b \lor b \leq a)$
- every nonempty subset of $S$ has a least element.
Well-ordered sets

- The natural numbers are a well-ordered set, as in the worst case 0 is the least element of the subset.

- The integers are not well-ordered because there is no least element.

- For any integer constant \( b \in \mathbb{N} \), the set \( \{x \mid x \in \mathbb{N} \land x \geq b\} \) is a well-ordered set.

- The non-negative real numbers are not a well-ordered set because there is no least element in the open interval (0,1).
\( f : S \rightarrow Y \) is a \textit{ranking function} if \( Y \) is a well-ordered set.

We define the \textit{ranking relation} of \( f \) to be:

\[
\succsim_f \triangleq \{ (s, t) \mid f(s) > f(t) \}
\]

Theorem. \( \text{WF}(R) \iff \exists f. \ R \subseteq \succsim_f \)
\[ f : S \rightarrow Y \] is a \textit{ranking function} if \( Y \) is a well-ordered set.

\[ \overset{\triangle}{\succeq_f} \triangleq \{(s, t) \mid f(s) > f(t)\} \]

\[ \text{Theorem. } \WF(R) \text{ iff } \exists f. \ R \subseteq \succeq_f \]
*Ranking functions and ranking relations*

$f : S \to Y$ is a *ranking function* if $Y$ is a well-ordered set.

We define the *ranking relation* of $f$ to be:

\[
\begin{align*}
\succcurlyeq_f & \triangleq \{(s,t) \mid f(s) > f(t)\} \\
\succcurlyeq_{f,b} & \triangleq \{(s,t) \mid f(s) > f(t) \land f(s) \geq b\} \\
\succcurlyeq_{f,b,d} & \triangleq \{(s,t) \mid f(s) \geq f(t) + d \land f(s) \geq b\}
\end{align*}
\]

**Theorem.** \(WF(R) \iff \exists f. \ R \subseteq \succcurlyeq_f\)
Example:

\[ R \triangleq x' = x - 1 \land x > 0 \]
Example:

\[ R \triangleq x' = x - 1 \land x > 0 \]

Is \( R \) well founded?
Example:

\[ R \triangleq x' = x - 1 \land x > 0 \]

\[ R \subseteq \geq_{x,-1} \]
Example:

\[ R \triangleq x' = x - 1 \land x > 0 \]

\[ R \subseteq \mathbb{N}_{x,-1} \]

Shorthand for \( f : S \to \mathbb{N} \) where \( f(s) \triangleq s(x) \)
Example:

\[ R \triangleq x' = x - 1 \land x > 0 \]

\[ R \subseteq \geq_{x,-1} \]
Example

\[ x' = x - 1 \land x > 0 \subseteq x' > x \land x \geq -1 \]

\[ R \subseteq \geq_{x,-1} \]
\[ \forall x, x'. \ x' = x - 1 \land x > 0 \Rightarrow x' > x \land x \geq -1 \]

\[ x' = x - 1 \land x > 0 \subseteq x' > x \land x \geq -1 \]

\[ R \subseteq \geq_{x,-1} \]
\[
\forall x, x'. \ x' = x - 1 \land x > 0 \Rightarrow x' > x \land x \geq -1
\]

\[
x' = x - 1 \land x > 0 \subseteq x' > x \land x \geq -1
\]

\[\Rightarrow R \subseteq \geq_{x,-1}\]

**Connection is made precise in the handout**
Questions

(a) $1 < 0$
(b) $0 < 1$
(c) $x' > x \land x' < 1000$
(d) $x' > x \land x' > 1000$
(e) $x' \geq x + 1 \land x' < 1000$
(f) $x' \geq x - 1 \land x' < 1000$
(g) $y' \geq y + 1 \land z' = z \land z < 1000$
(h) $y' + 1 \geq y \land z' = z \land z < 1000$
(i) $(x' = x - 1 \lor x' = x + 1) \land x < 1000$
(j) $x' = x - z \land x > 0$
Questions

(a) $1 < 0$
(b) $0 < 1$
(c) $x' > x \land x' < 1000$
(d) $x' > x \land x' > 1000$
(e) $x' \geq x + 1 \land x' < 1000$
(f) $x' \geq x - 1 \land x' < 1000$
(g) $y' \geq y + 1 \land z' = z \land z < 1000$
(h) $y' + 1 \geq y \land z' = z \land z < 1000$
(i) $(x' = x - 1 \lor x' = x + 1) \land x < 1000$
(j) $x' = x - z \land x > 0$

Let's try to prove these with a decision procedure
Transition systems

Notation:

\( \rightarrow \) Transition system: \( P = (I, R, S) \)

\( \rightarrow \) Update relation: \( R \)

\( \rightarrow \) Initial states: \( I \)

\( \rightarrow \) Transition relation: \( R^* \big|_I \triangleq R \cap (R^*(I) \times R^*(I)) \)
Transition systems

Notation:

- Transition system: \( P = (I, R, S) \)
- Update relation: A transition system is terminating if its \textit{transition relation} is WF
- Initial states: \( I \)
- Transition relation: \( R|_{I} \triangleq R \cap (R^*(I) \times R^*(I)) \)
Transition systems

Notation:

- Transition system:
- Update relation:
- Initial states:
- Transition relation:

A transition system is terminating if its transition relation is WF.

\[
\begin{align*}
R_{\downarrow Q} & \triangleq R \cap (Q \times Q) \\
R_{\ast I} & \triangleq R_{\downarrow R^*(I)}
\end{align*}
\]
Supporting invariants

- Update relations are typically not well founded, even when the transition relation is

- Computing a precise $R^*(l)$ is very hard (technically, undecidable)
Supporting invariants

In practice, we must find a supporting invariant that is "computable" yet still powerful enough for termination:

\[ Q \text{ is an invariant of } I \iff Q \supseteq R^*(I) \]

Much of the hard part when proving WF is finding the right invariant
Example

\[ R \triangleq x' = x + y \land y' = y \land x > 0 \]
\[ I \triangleq y' < -1 \]

Is the update relation well founded?
Example

\[ R \triangleq x' = x + y \land y' = y \land x > 0 \]
\[ I \triangleq y' < -1 \]

→ Is the update relation well founded?
→ Is the transition relation well founded?
Example

\[ R \triangleq x' = x + y \land y' = y \land x > 0 \]
\[ I \triangleq y' < -1 \]

→ Is the update relation well founded?
→ Is the transition relation well founded?
→ How would you prove this with a decision procedure?
Outline

→ Introduction

→ Well-founded relations and ranking functions

→ Disjunctive well foundedness

→ Decomposition techniques

→ Rank function synthesis (if time permits)
Example

Is the following relation well founded?

\[ \land \left\{ \begin{array}{l}
  x > 0, \\
  y > 0, \\
  (x' = x - 1 \land y' = y + 1) \lor (x' = x \land y' = y - 1)
\end{array} \right\} \]

**Yes?** What’s the ranking function?

**No?** Show me a counterexample
Example

Is the following relation well founded?

\[ \land \left\{ \begin{array}{l}
  x > 0, \\
  y > 0, \\
  (x' = x - 1 \land y' = y + 1) \lor (x' = x \land y' = y - 1)
\end{array} \right\} \]

Yes? What’s the ranking function?

No? Show me a counterexample

\[ f(x, y) \triangleq ?? \]
Example

Is the following relation well founded?

\[ x > 0, \]
\[ y > 0, \]
\[ (x' = x - 1 \land y' = y + 1) \lor (x' = x \land y' = y - 1) \]

Yes? What’s the ranking function?

No? Show me a counterexample

\[ f(x, y) \equiv 2x + y \]
Example

Is the following relation well founded?

\[
\land \begin{cases} 
  x > 0, \\
  y > 0, \\
  (x' = x - 1 \land \ \\
  \text{Yes? What's the range?} \\
  \text{No? Show me a counterexample.}
\end{cases}
\]

\[f(x, y) \triangleq 2x + y\]
Is the following relation well founded?

\[ \bigwedge \left\{ \begin{array}{l}
  x > 0, \\
  y > 0, \\
  (x' = x - 1) \lor (x' = x \land y' = y - 1)
\end{array} \right\} \]

Yes? What’s the ranking function?

No? Show me a counterexample
Is the following relation well founded?

\[ y := \neq \]

\[ \land \left\{ \begin{align*}
x & > 0, \\
y & > 0, \\
(x' = x - 1) \lor (x' = x \land y' = y - 1)
\end{align*} \right\} \]

**Yes?** What’s the ranking function?

**No?** Show me a counterexample
Is the following relation well founded?

\[ \land \left\{ \begin{aligned}
x &> 0, \\
y &> 0, \\
(x' = x - 1) \lor (x' = x \land y' = y - 1)
\end{aligned} \right\} \]

Yes? What’s the ranking function?

No? Show me a counterexample

\[ f(x, y) \triangleq \text{???} \]
Temptation!!!!!!

→ It's tempting to look for ranking functions by examining some cases

\[ \left\{ \begin{array}{l} x > 0, \\ y > 0, \\ (x' = x - 1) \lor (x' = x \land y' = y - 1) \end{array} \right\} \]
Its tempting to look for ranking functions by examining some cases

\[ \land \left\{ \begin{array}{l}
x > 0, \\
y > 0, \\
(x' = x - 1) \lor (y' = y - 1)
\end{array} \right\} \]
Its tempting to look for ranking functions by examining some cases

\[\land \left\{ \begin{align*}
x &> 0, \\
y &> 0, \\
(x' = x - 1) &\lor (y' = y - 1) \end{align*} \right\}\]

Is this relation well founded?
Its tempting to look for ranking functions by examining some cases

\[ \wedge \left\{ \begin{array}{l}
x > 0, \\
y > 0,
\end{array} \right. \bigvee (x' = x - 1) \lor (y' = y - 1) \]

Is this relation well founded?

\[ f(x, y) = x \]
Temptation!!!!

→ It's tempting to look for ranking functions by examining some cases

\[
\land \left\{ \begin{array}{l}
    x > 0, \\
    y > 0, \\
    (x' = x \land y' = y - 1)
\end{array} \right\}
\]

\[f(x, y) = x\]
Its tempting to look for ranking functions by examining some cases

\[
\land \left\{ \begin{array}{l}
x > 0, \\
y > 0, \\
(x' = y) \lor (x' = x \land y' = y - 1)
\end{array} \right\}
\]

\[ f(x, y) = y \]

\[ f(x, y) = x \]
Temptation!!!!!

\[ \land \left\{ \begin{array}{l}
x > 0, \\
y > 0, \\
(x') \lor (x' = x \land y' = y - 1)
\end{array} \right\} \]

\[ f(x, y) = y \]

\[ f(x, y) = x \]

How can we combine these?
Temptation!!!!!!

It's tempting to look for ranking functions by examining some cases:

\[ R \subseteq \geq_x \cup \geq_y \]

\[\wedge \left\{ \begin{array}{l}
  x > 0, \\
  y > 0, \\
  (x' = x \land y' = y - 1)
\end{array} \right\} \]

\[ f(x, y) = y \]

\[ f(x, y) = x \]

How can we combine these?
It's tempting to look for ranking functions by examining some cases.

\[ P \subseteq \bigcup_{y \in \mathbb{Y}} \{ x' \} \]

\[ \bigwedge \left\{ \begin{array}{l} x > 0, \\ y > 0, \\ (x', y') \lor (x' = x \land y' = y - 1) \end{array} \right\} \]

\[ f(x, y) = y \]

\[ f(x, y) = x \]

How can we combine these?
If $Q$ and $R$ are well founded, is $Q \cup R$ well founded?
Disjunctive well foundedness

If $Q$ and $R$ are well founded, is $Q \cup R$ well founded?

Counterexample:

$$\land \left\{ \begin{array}{l}
x > 0 \\
y > 0 \\
(x' = x + 1 \land y' = y - 1) \lor (x' = x - 1 \land y' = y + 1)
\end{array} \right\}$$
Counterexample:

\[
\land \left\{ \begin{array}{l}
x > 0 \\
y > 0 \\
(x' = x + 1 \land y' = y - 1) \lor (x' = x - 1 \land y' = y + 1)
\end{array} \right\}
\]
Counterexample:

\[ (x' = x + 1 \land y' = y - 1) \lor (x' = x - 1 \land y' = y + 1) \]
Disjunctive well foundedness

Theorem

• Assume that $Q_1, Q_2, \ldots Q_n$ are well-founded relations.

• $R$ is well-founded iff $R^+ \subseteq Q_1 \cup Q_2 \cup \ldots \cup Q_n$. 
Theorem

- Assume that $Q_1, Q_2, \ldots Q_n$ are well-founded relations
- $R$ is well-founded iff $R^+ \subseteq Q_1 \cup Q_2 \cup \ldots \cup Q_n$.

This is key!
Disjunctive well foundedness

Constructing the argument is much easier
- Simply union based on examples rather than a holistic synthesis
- Many (hopefully) easier problems, rather than one big one

Checking the argument is harder
- Even checking WF without invariants is now no longer decidable

Checking is something we know how to do
- The check is a very difficult (but solvable) invariance property.........more later.........
Disjunctive well foundedness

➔ **Constructing** the argument is much easier
  - Simply union based on examples rather than a holistic synthesis
  - Many (hopefully) easier problems, rather than one big one

➔ **Checking** the argument is harder
  - Even checking WF without invariants is now no longer decidable

➔ Checking is something we know how to do
  - The check is a very difficult (but solvable) invariance property...........more later.........
Lemma.

\[ R^+ \subseteq Q \text{ if } R \subseteq Q \text{ and } (Q; R) \subseteq Q \]
\[ R \triangleq x > 0 \land y > 0 \land x' = x - 1 \land y' = y \]
\[ \lor \quad x > 0 \land y > 0 \land y' = y - 1 \]

\[ Q \triangleq (\geq_{y,0} \cap \geq_{x,0}) \cup \geq_{(y,0)} \]

\[ R \subseteq Q, \quad (Q; R) \subseteq Q, \text{ thus } R^+ \subseteq Q \]
\[ R \triangleq x > 0 \land x' = x - y \land y' = y + 1 \]
\[ R \triangleq x > 0 \land x' = x - y \land y' = y + 1 \]

Can you prove termination with induction?
Outline

→ Introduction

→ Well-founded relations and ranking

→ Disjunctive well foundedness

→ Decomposition techniques

→ Rank function synthesis (if time permits)
Most systems we’re interested in proving terminating have at least some finite structure we can make use of
Most systems we’re interested in proving terminating have at least some finite structure we can make use of.
Most systems we’re interested in proving terminating have at least some finite structure we can make use of.
Most systems we’re interested in proving terminating have at least some finite structure we can make use of.

- All states at location 10
- All states at circuit reset
Most systems we’re interested in proving terminating have at least some finite structure we can make use of.

We can prove termination one slice at a time.
Decomposition
Decomposition

\[ R^+ \downarrow_{pc=10} \]
Decomposition

$R^+ \downarrow_{pc=10}$
Theorem

- Assume \( v \in \text{VAR} \)
- Assume \( L = R^*(I)(v) \) is finite
- \( R^I \) is well-founded if for all \( l \in L \), \( (R^I)^+\big|_v=l \) is well-founded.
Decomposition

- Assume $v \in \text{VAR}$

- Assume $L = R^*(I)(v)$ is finite.

- Let $k_1$ and $k_2$ be constants from $\text{VAL}$.

- Assume that, if $R(s, t)$ and $t(v) = k_2$ then $s(v) = k_1$.

- $\forall l \in L. (R^+_I)_{v=l}$ is well founded iff $\forall l \in L - \{k_2\}. (R^+_I)_{v=l}$ is well founded
Is the following relation well founded?

\[ R \triangleq (b' = 1 \land b = 0) \lor (b' = 0 \land b = 1) \land (b = 1 \land x' = x - 1 \land x > 0) \lor (b = 0 \land x' = x) \]

**Yes?** What’s the ranking function?

**No?** Show me a counterexample
Example

\[ R^+ \mid_{b=1} \subseteq Q_1 \]
\[ \land \]
\[ R^+ \mid_{b=0} \subseteq Q_2 \]

Is the following relation bounded?

\[
R \triangleq (b' = 1 \land b = 0) \lor (b' = 0 \land b = 1) \\
\land \quad (b = 1 \land x' = x - 1 \land x > 0) \lor (b = 0 \land x' = x)
\]

**Yes?** What’s the ranking function?

**No?** Show me a counterexample
Is the following relation bounded?

\[ R^+ \mid_{b=1} \subseteq Q_1 \]
\[ \land \]
\[ R^- \mid_{b=1} \subseteq Q_2 \]

Yes? What’s the ranking function?

No? Show me a counterexample
Example

\[ R^+_{b=1} \subseteq Q_1 \]
\[ \wedge \]
\[ R^+_{b=0} \subseteq Q_2 \]

\[ R^+_{\neq b=1} = b' = 1 \wedge b = 1 \wedge x' < x \wedge x > 0 \]
\[ \subseteq \succsim_{x,0} \]
Outline

➤ Introduction

➤ Well-founded relations and ranking

➤ Disjunctive well foundedness

➤ Decomposition techniques

➤ Notes on rank function synthesis
Goal: find an $f$ such that

$$\forall V, V'. \ R(V, V') \Rightarrow f(V) > f(V')$$
We consider a search for affine functions over generic parameters, e.g. \( f(g_1, g_2, g_3) = 1g_1 + 2g_2 + 0g_3 + 5 \)

**Goal:** Find an \( f \) such that

\[
\forall V, V'. \ R(V, V') \Rightarrow f(V) > f(V')
\]
Thus $f$ is the vector of coefficients 1, 2, 0, 5, and $f(V)$ is $f_1 V_1 + f_2 V_2 + f_3 V_3 + f_4$

$$\forall V, V'. \ R(V, V') \Rightarrow f(V) > f(V')$$
Rank function synthesis

The difficulty is that search for $f_1, f_2, f_3, \text{etc}$ is a non-linear problem.

Thus $f$ is the vector of coefficients $1, 2, 0, 5$, and $f(V)$ is $f_1V_1 + f_2V_2 + f_3V_3 + f_4$

$\forall V, V'. \ R(V, V') \Rightarrow f(V) > f(V')$
Theorem. Assume that

- $M$ is a matrix,
- $v$ is a column vector,
- $f$ is a row vector,
- and $Mv \leq 0$ is satisfiable.

Then:

$$Mv \leq 0 \Rightarrow fv \leq 0$$

iff

$$\exists \lambda. \lambda M = f \land \lambda \geq 0$$
Theorem. Assume that

- $M$ is a matrix,
- $v$ is a column vector,
- $f$ is a row vector,
- and $Mv \leq 0$ is satisfiable.

Let

$$Mv \leq 0 \Rightarrow fv \leq 0$$

iff

$$\exists \lambda. \lambda M = f \land \lambda \geq 0$$
Farkas’ lemma

**Theorem.** Assume that

- $M$ is a matrix,
- $v$ is a column vector,
- $f$ is a row vector,
- and $Mv \leq 0$ is satisfiable.

Then

$$Mv \leq 0 \Rightarrowfv \leq 0$$

iff

$$\exists \lambda. \lambda M = f \land \lambda \geq 0$$
A Complete Method for the Synthesis of Linear Ranking Functions

Andreas Podelski and Andrey Rybalchenko

Max-Planck-Institut für Informatik
Saarbrücken, Germany

Abstract. We present an automated method for proving the termination of an unnested program loop by synthesizing linear ranking functions. The method is complete. Namely, if a linear ranking function exists then it will be discovered by our method. The method relies on the fact that we can obtain the linear ranking functions of the program loop as the solutions of a system of linear inequalities that we derive from the program loop. The method is used as a subroutine in a method for proving termination and other liveness properties of more general programs via transition invariants; see [PR03].
Goal: find an \( f \) such that

\[
\forall V, V'. \ R(V, V') \Rightarrow f(V) > f(V')
\]
Goal: find an $f$ such that

$$\forall V, V'. \ R(V, V') \Rightarrow f(V) > f(V')$$

or $-f(V) \leq -f(V')$, or $-f(V) + f(V') \leq 0$
Goal: find an $f$ such that

$$\forall V, V'. \ R(V, V') \Rightarrow -f(V) + f(V') \leq 0$$

or $-f(V) \leq -f(V')$, or $-f(V) + f(V') \leq 0$
Goal: find an $f$ such that

$$\forall V, V'. \ R(V, V') \Rightarrow -f(V) + f(V') \leq 0$$
Theorem. Assume that

- $M$ is a matrix,
- $v$ is a column vector,
- $f$ is a row vector,
- and $Mv \leq 0$ is satisfiable.

Then:

$$Mv \leq 0 \Rightarrow fv \leq 0$$

iff

$$\exists \lambda. \lambda M = f \land \lambda \geq 0$$

Goal: find an $f$ such that

$$\forall V, V'. R(V, V') \Rightarrow -f(V) + f(V') \leq 0$$
**Theorem.** Assume that

- $M$ is a matrix,
- $v$ is a column vector,
- $f$ is a row vector,
- and $Mv \leq 0$ is satisfiable.

Then:

$$Mv \leq 0 \implies fv \leq 0$$

iff

$$\exists \lambda. \lambda M = f \land \lambda \geq 0$$

**Goal:** find an $f$ such that

$$\forall V, V'. R(V, V') \implies -f(V) + f(V') \leq 0$$
**Theorem.** Assume that
- $M$ is a matrix,
- $v$ is a column vector,
- $f$ is a row vector,
- and $Mv \leq 0$ is satisfiable.

Then:

\[ Mv \leq 0 \Rightarrow fv \leq 0 \]

iff

\[ \exists \lambda. \lambda M = f \land \lambda \geq 0 \]

---

**Goal:** find an $f$ such that

\[ \forall V, V'. R(V, V') \Rightarrow -f(V) + f(V') \leq 0 \]
Farkas' lemma

Theorem. Assume that

- $M$ is a matrix,
- $v$ is a column vector,
- $f$ is a row vector,
- and $Mv \leq 0$ is satisfiable.

Then:

\[ Mv \leq 0 \iff fv \leq 0 \]

iff

\[ \exists \lambda. \lambda M = f \wedge \lambda \geq 0 \]

Goal: find an $f$ such that

\[ \forall V, V'. R(V, V') \Rightarrow -f(V) + f(V') \leq 0 \]

Now we can use SMT solvers (e.g. Z3)
assume(x>0);
x := x - 1;
assume(y>0);
y := y + pos();
Rank function synthesis

\[
\begin{align*}
\text{assume}(x & > 0); & \quad x \ > \ 0 \\
x & := x - 1; & \quad x' \ = \ x - 1 \\
\text{assume}(y & > 0); & \quad y \ > \ 0 \\
y & := y + \text{pos}(); & \quad y' \ > \ y
\end{align*}
\]
Rank function synthesis

```
assume(x>0);
  x  >  0  \text{ \hspace{0.5cm}} x  >  0
x := x - 1;
  x' = x - 1  \text{ \hspace{0.5cm}} x' \geq x - 1
assume(y>0);
  y  >  0  \text{ \hspace{0.5cm}} x' \leq x - 1
y := y + \text{pos}();
  y' > y  \text{ \hspace{0.5cm}} y' > 0
```

\[ y' > y \]
assume(x>0);
\[ x > 0 \quad x > 0 \]
\[ x := x - 1; \quad x' = x - 1 \quad x' \geq x - 1 \]
assume(y>0);
\[ y > 0 \quad x' \leq x - 1 \]
y := y + pos();
\[ y' > y \quad y > 0 \]
\[ y' > y \]

\[
\begin{align*}
0x' & \quad + \quad 0y' & \quad + \quad -1x & \quad + \quad 0y & \quad + \quad 1 & \quad \leq & \quad 0 \\
1x' & \quad + \quad 0y' & \quad + \quad -1x & \quad + \quad 0y & \quad + \quad 1 & \quad \leq & \quad 0 \\
-1x' & \quad + \quad 0y' & \quad + \quad 1x & \quad + \quad 0y & \quad + \quad -1 & \quad \leq & \quad 0 \\
0x' & \quad + \quad 0y' & \quad + \quad 0x & \quad + \quad -1y & \quad + \quad 1 & \quad \leq & \quad 0 \\
0x' & \quad + \quad -1y' & \quad + \quad 0x & \quad + \quad 1y & \quad + \quad 1 & \quad \leq & \quad 0
\end{align*}
\]
Rank function synthesis

\[0x' + 0y' + -1x + 0y + 1 \leq 0\]

\[1x' + 0y' + -1x + 0y + 1 \leq 0\]

\[-1x' + 0y' + 1x + 0y + -1 \leq 0\]

\[0x' + 0y' + 0x + -1y + 1 \leq 0\]

\[0x' + -1y' + 0x + 1y + 1 \leq 0\]
Rank function synthesis

\[
\begin{align*}
0x' & \quad + \quad 0y' & \quad + \quad -1x & \quad + \quad 0y & \quad + \quad 1 & \quad \leq & \quad 0 \\
1x' & \quad + \quad 0y' & \quad + \quad -1x & \quad + \quad 0y & \quad + \quad 1 & \quad \leq & \quad 0 \\
-1x' & \quad + \quad 0y' & \quad + \quad 1x & \quad + \quad 0y & \quad + \quad -1 & \quad \leq & \quad 0 \\
0x' & \quad + \quad 0y' & \quad + \quad 0x & \quad + \quad -1y & \quad + \quad 1 & \quad \leq & \quad 0 \\
0x' & \quad + \quad -1y' & \quad + \quad 0x & \quad + \quad 1y & \quad + \quad 1 & \quad \leq & \quad 0 
\end{align*}
\]
Rank function synthesis

\[
\begin{align*}
0x' + 0y' + -1x + 0y + 1 & \leq 0 \\
1x' + 0y' + -1x + 0y + 1 & \leq 0 \\
-1x' + 0y' + 1x + 0y + -1 & \leq 0 \\
0x' + 0y' + 0x + -1y + 1 & \leq 0 \\
0x' + -1y' + 0x + 1y + 1 & \leq 0
\end{align*}
\]

\[
\subset \\
\supseteq f, b
\]
Rank function synthesis

\[ \begin{align*}
0x' & \quad + \quad 0y' & \quad + \quad -1x & \quad + \quad 0y & \quad + \quad 1 & \quad \leq & \quad 0 \\
1x' & \quad + \quad 0y' & \quad + \quad -1x & \quad + \quad 0y & \quad + \quad 1 & \quad \leq & \quad 0 \\
-1x' & \quad + \quad 0y' & \quad + \quad 1x & \quad + \quad 0y & \quad + \quad -1 & \quad \leq & \quad 0 \\
0x' & \quad + \quad 0y' & \quad + \quad 0x & \quad + \quad -1y & \quad + \quad 1 & \quad \leq & \quad 0 \\
0x' & \quad + \quad -1y' & \quad + \quad 0x & \quad + \quad 1y & \quad + \quad 1 & \quad \leq & \quad 0
\end{align*} \]

\[ \subseteq \]

\[ \geq f, b \]

Can we find such an \( f \) and \( b \)?
Rank function synthesis

\[
\begin{align*}
0x' + 0y' + -1x + 0y + 1 & \leq 0 \\
1x' + 0y' + -1x + 0y + 1 & \leq 0 \\
-1x' + 0y' + 1x + 0y + -1 & \leq 0 \\
0x' + 0y' + 0x + -1y + 1 & \leq 0 \\
0x' + -1y' + 0x + 1y + 1 & \leq 0
\end{align*}
\]

\[
\subseteq
\]

\[
\begin{align*}
f(x, y) & > f(x', y') \\
f(x', y') & \geq b
\end{align*}
\]

Can we find such an \( f \) and \( b \)?
Rank function synthesis

\[
\begin{align*}
0x' & + 0y' + -1x + 0y + 1 \leq 0 \\
1x' & + 0y' + -1x + 0y + 1 \leq 0 \\
-1x' & + 0y' + 1x + 0y + -1 \leq 0 \\
0x' & + 0y' + 0x + -1y + 1 \leq 0 \\
0x' & + -1y' + 0x + \\
\end{align*}
\]

\[f(x, y) \geq f(x', y') \geq b \]

\[f(x', y') + -f(x, y) + 1 \leq 0 \]

\[-f(x', y') + b \leq 0 \]

Can we find such an \(f\) and \(b\)?
Rank function synthesis

\[ f(a, b) \triangleq c_1 a + c_2 b \]
\[ -f(a, b) \triangleq c_3 b + c_4 b \]
\[ c_1 = -1c_3 \]
\[ c_2 = -1c_4 \]

\[ f(x', y') + f(x, y) + 1 \leq 0 \]
\[ -f(x', y') + b \leq 0 \]
Rank function synthesis

\[ f(a, b) \triangleq c_1 a + c_2 b \]

\[ -f(a, b) \triangleq c_3 a + c_4 b \]

\[ c_1 = -1c_3 \]

\[ c_2 = -1c_4 \]

\[
\begin{align*}
c_1 x' + c_2 y' + c_3 x + c_4 y + 1 & \leq 0 \\
c_3 x' + c_4 y' + b & \leq 0 \\
1c_1 + 1c_3 + 0 & \leq 0 \\
-1c_1 + -1c_3 + 0 & \leq 0 \\
1c_2 + 1c_4 + 0 & \leq 0 \\
-1c_2 + -1c_4 + 0 & \leq 0
\end{align*}
\]
Rank function synthesis

\[\begin{align*}
0x' & + 0y' + -1x + 0y + 1 \leq 0 \\
1x' & + 0y' + -1x + 0y + 1 \leq 0 \\
-1x' & + 0y' + 1x + 0y + -1 \leq 0 \\
0x' & + 0y' + 0x + -1y + 1 \leq 0 \\
0x' & + -1y' + 0x + 1y + 1 \leq 0
\end{align*}\]

\[\subseteq\]

\[\begin{align*}
c_1x' & + c_2y' + c_3x + c_4y + 1 \leq 0 \\
c_3x' & + c_4y' + b \leq 0 \\
1c_1 & + 1c_3 + 0 \leq 0 \\
-1c_1 & + -1c_3 + 0 \leq 0 \\
1c_2 & + 1c_4 + 0 \leq 0 \\
-1c_2 & + -1c_4 + 0 \leq 0
\end{align*}\]
Rank function synthesis

\[
\begin{align*}
0x' & + 0y' + -1x + 0y + 1 \leq 0 \\
1x' & + 0y' + -1x + 0y + 1 \leq 0 \\
-1x' & + 0y' + 1x + 0y + -1 \leq 0 \\
0x' & + 0y' + 0x + -1y + 1 \leq 0 \\
0x' & + -1y' + 0x + 1y + 1 \leq 0 \\
\end{align*}
\]

\[
\Rightarrow
\]

\[
\begin{align*}
c_1x' & + c_2y' + c_3x + c_4y + 1 \leq 0 \\
\end{align*}
\]

\[
\begin{align*}
c_3x' & + c_4y' + b \leq 0 \\
1c_1 & + 1c_3 + 0 \leq 0 \\
-1c_1 & + -1c_3 + 0 \leq 0 \\
1c_2 & + 1c_4 + 0 \leq 0 \\
-1c_2 & + -1c_4 + 0 \leq 0 \\
\end{align*}
\]
Rank function synthesis

\[ \begin{align*}
0x' & + 0y' + -1x + 0y + 1 \leq 0 \\
1x' & + 0y' + -1x + 0y + 1 \leq 0 \\
-1x' & + 0y' + 1x + 0y + -1 \leq 0 \\
0x' & + 0y' + 0x + -1y + 1 \leq 0 \\
0x' & + -1y' + 0x + 1y + 1 \leq 0 \\
\end{align*} \]

\[ \Rightarrow \]

\[ \begin{align*}
c_1x' & + c_2y' + c_3x' + c_4y + 1 \leq 0 \\
c_3x' & + c_4y' + b \leq 0 \\
1c_1 & + 1c_3 + 0 \leq 0 \\
-1c_1 & + -1c_3 + 0 \leq 0 \\
1c_2 & + 1c_4 + 0 \leq 0 \\
-1c_2 & + -1c_4 + 0 \leq 0 \\
\end{align*} \]

Nonlinear
Rank function synthesis

\[
\begin{align*}
0x' &+ 0y' + -1x + 0y + 1 \leq 0 \\
1x' &+ 0y' + -1x + 0y + 1 \leq 0 \\
-1x' &+ 0y' + 1x + 0y + -1 \leq 0 \\
0x' &+ 0y' + 0x + -1y + 1 \leq 0 \\
0x' &+ -1y' + 0x + 1y + 1 \leq 0 \\
\end{align*}
\]

\[
\begin{align*}
c_1x' &+ c_2y' + -c_3x + c_4y + 1 \leq 0 \\
c_3x' &+ c_4y' + b \leq 0 \\
1c_1 &+ 1c_3 + 0 \leq 0 \\
-1c_1 &+ -1c_3 + 0 \leq 0 \\
1c_2 &+ 1c_4 + 0 \leq 0 \\
-1c_2 &+ -1c_4 + 0 \leq 0 \\
\end{align*}
\]

Non convex
Rank function synthesis

\[
\begin{align*}
0x' + 0y' + -1x + 0y + 1 & \leq 0 \\
1x' + 0y' + -1x + 0y + 1 & \leq 0 \\
-1x' + 0y' + 1x + 0y + -1 & \leq 0 \\
0x' + 0y' + 0x + -1y + 1 & \leq 0 \\
0x' + -1y' + 0x + 1y + 1 & \leq 0
\end{align*}
\]

\[\Rightarrow\]

\[
\begin{align*}
c_1 x' + c_2 y' + c_3 x & + c_4 y + 1 & \leq 0 \\
c_3 x' + c_4 y' + b & \leq 0 \\
1c_1 + 1c_3 + 0 & \leq 0 \\
-1c_1 + -1c_3 + 0 & \leq 0 \\
1c_2 + 1c_4 + 0 & \leq 0 \\
-1c_2 + -1c_4 + 0 & \leq 0
\end{align*}
\]

mixture of \( \mathcal{E} \) \& \( \mathcal{A} \) variables
Rank function synthesis

\begin{align*}
0x' & + 0y' + -1x + 0y + 1 \leq 0 \\
1x' & + 0y' + -1x + 0y + 1 \leq 0 \\
-1x' & + 0y' + 1x + 0y + -1 \leq 0 \\
0x' & + 0y' + 0x + -1y + 1 \leq 0 \\
0x' & + -1y' + 0x + 1y + 1 \leq 0
\end{align*}

\[\Rightarrow\]

\begin{align*}
c_1x' & + c_2y' + c_3x + c_4y + 1 \leq 0 \\
c_3x' & + c_4y' + b \leq 0 \\
1c_1 & + 1c_3 + 0 \leq 0 \\
-1c_1 & + -1c_3 + 0 \leq 0 \\
1c_2 & + 1c_4 + 0 \leq 0 \\
-1c_2 & + -1c_4 + 0 \leq 0
\end{align*}

mixture of $E$ and $A$ variables
Rank function synthesis

\[
\begin{align*}
0x' &+ 0y' + -1x + 0y + 1 \leq 0 \\
1x' &+ 0y' + -1x + 0y + 1 \leq 0 \\
-1x' &+ 0y' + 1x + 0y + -1 \leq 0 \\
0x' &+ 0y' + 0x + -1y + 1 \leq 0 \\
0x' &+ -1y' + 0x + 1y + 1 \leq 0
\end{align*}
\]

\[
\Rightarrow
\]

\[
c_1x' + c_2y' + c_3x + c_4y + 1 \leq 0
\]

(for simplicity)
Rank function synthesis

\[
\begin{align*}
0x' & + 0y' + -1x + 0y + 1 \leq 0 \\
1x' & + 0y' + -1x + 0y + 1 \leq 0 \\
-1x' & + 0y' + 1x + 0y + -1 \leq 0 \\
0x' & + 0y' + 0x + -1y + 1 \leq 0 \\
0x' & + -1y' + 0x + 1y + 1 \leq 0
\end{align*}
\]

\[
\Rightarrow \\
\begin{align*}
c_1x' & + c_2y' + c_3x + c_4y + 1 \leq 0
\end{align*}
\]
Rank function synthesis

\[
R \triangleq 
\begin{align*}
0x' & + 0y' + -1x + 0y + 1 \leq 0 \\
1x' & + 0y' + -1x + 0y + 1 \leq 0 \\
-1x' & + 0y' + 1x + 0y + -1 \leq 0 \\
0x' & + 0y' + 0x + -1y + 1 \leq 0 \\
0x' & + -1y' + 0x + 1y + 1 \leq 0 
\end{align*}
\]

\[\Rightarrow\]

\[
\psi \triangleq c_1 x' + c_2 y' + c_3 x + c_4 y + 1 \leq 0
\]

Farkas’ lemma. \( R \Rightarrow \psi \) iff there exist real multipliers \( \lambda_1, \ldots, \lambda_5 \geq 0 \) such that

\[
c_1 = \sum_{i=1}^{5} \lambda_i a_{i,1} \quad \land \quad c_4 = \sum_{i=1}^{5} \lambda_i a_{i,4} \quad \land \quad 1 \leq (\sum_{i=0}^{5} \lambda_i b_i)
\]
Rank function synthesis

\[ R \triangleq \begin{array}{cccccc}
0x' & + & 0y' & + & -1x & + & 0y & + & 1 & \leq & 0 \\
1x' & + & 0y' & + & -1x & + & 0y & + & 1 & \leq & 0 \\
-1x' & + & 0y' & + & 1x & + & 0y & + & -1 & \leq & 0 \\
0x' & + & 0y' & + & 0x & + & -1y & + & 1 & \leq & 0 \\
0x' & + & -1y' & + & 0x & + & 1y & + & 1 & \leq & 0 
\end{array} \]

\[ \Rightarrow \]

\[ \psi \triangleq c_1 x' + c_2 y' + c_3 x + c_4 y + 1 \leq 0 \]

Farkas’ lemma. \( R \Rightarrow \psi \) iff there exist real multipliers \( \lambda_1, \ldots, \lambda_5 \geq 0 \) such that

\[ c_1 = \sum_{i=1}^{5} \lambda_i a_{i,1} \wedge \cdots \wedge c_4 = \sum_{i=1}^{5} \lambda_i a_{i,4} \wedge 1 \leq (\sum_{i=0}^{5} \lambda_i b_i) \]
Rank function synthesis

\[
\begin{align*}
R & \triangleq 0x' + 0y' + -1x + 0y + 1 \leq 0 \\
& \triangleq 1x' + 0y' + -1x + 0y + 1 \leq 0 \\
& \triangleq -1x' + 0y' + 1x + 0y + -1 \leq 0 \\
& \triangleq 0x' + 0y' + 0x + -1y + 1 \leq 0 \\
& \triangleq 0x' + -1y' + 0x + 1y + 1 \leq 0 \\
\Rightarrow \\
\psi & \triangleq c_1 x' + c_2 y' + c_3 x + c_4 y + 1 \leq 0 \\
\end{align*}
\]

Farkas' lemma. \( R \Rightarrow \psi \) iff there exist real multipliers \( \lambda_1, \ldots, \lambda_5 \geq 0 \) such that

\[
\begin{align*}
c_1 & = \sum_{i=1}^{5} \lambda_i a_{i,1} \quad \wedge \cdots \wedge \\
c_4 & = \sum_{i=1}^{5} \lambda_i a_{i,4} \quad \wedge \quad 1 \leq (\sum_{i=0}^{5} \lambda_i b_i)
\end{align*}
\]
Rank function synthesis

\[ R \triangleq \begin{align*}
0x' & + 0y' + -1x + 0y + 1 & \leq & 0 \\
1x' & + 0y' + -1x + 0y + 1 & \leq & 0 \\
-1x' & + 0y' + 1x + 0y + -1 & \leq & 0 \\
0x' & + 0y' + 0x + -1y + 1 & \leq & 0 \\
0x' & + -1y' + 0x + 1y + 1 & \leq & 0
\end{align*} \]

\[ \Rightarrow \]

\[ \psi \triangleq c_1 x' + c_2 y' + c_3 x + c_4 y + 1 \leq 0 \]

**Farkas’ lemma.** \( R \Rightarrow \psi \) iff there exist real multipliers \( \lambda_1, \ldots, \lambda_5 \geq 0 \) such that

\[ c_1 = \sum_{i=1}^{5} \lambda_i a_{i,1} \quad \wedge \cdots \wedge \quad c_4 = \sum_{i=1}^{5} \lambda_i a_{i,4} \quad \wedge \quad 1 \leq (\sum_{i=0}^{5} \lambda_i b_i) \]
Rank function synthesis

\[ R \triangleq \begin{align*}
0x' &+ 0y' + -1x' + 0y + 1 &\leq 0 \\
1x' &+ 0y' + -1x + 0y + 1 &\leq 0 \\
-1x' &+ 0y' + 1x + 0y + -1 &\leq 0 \\
0x' &+ 0y' + 0x + -1y + 1 &\leq 0 \\
0x' &+ -1y' + 0x + 1y + 1 &\leq 0
\end{align*} \]

\[ \Rightarrow \]

\[ \psi \triangleq c_1 x' + c_2 y' + c_3 x + c_4 y + 1 \leq 0 \]

**Farkas' lemma.** \( R \Rightarrow \psi \) iff there exist real multipliers \( \lambda_1, \ldots, \lambda_5 \geq 0 \) such that

\[ c_1 = \sum_{i=1}^{5} \lambda_i a_{i,1} \land \cdots \land c_4 = \sum_{i=1}^{5} \lambda_i a_{i,4} \land 1 \leq (\sum_{i=0}^{5} \lambda_i b_i) \]
Rank function synthesis

\[ R \triangleq \begin{align*}
0x' & + 0y' + -1x + 0y + 1 \leq 0 \\
1x' & + 0y' + -1x + 0y + 1 \leq 0 \\
0x' & + 0y' + 0x + -1y + 1 \leq 0 \\
0x' & + -1y' + 0x + 1y + 1 \leq 0
\end{align*} \]

\( \psi \triangleq \begin{align*}
C_1 &= 0\lambda_1 + 1\lambda_2 + -1\lambda_3 + 0\lambda_4 + 0\lambda_5 \\
C_2 &= 0\lambda_1 + 0\lambda_2 + 0\lambda_3 + 0\lambda_4 + -1\lambda_5 \\
C_3 &= -1\lambda_1 + -1\lambda_2 + 1\lambda_3 + 0\lambda_4 + 0\lambda_5 \\
C_4 &= 0\lambda_1 + 0\lambda_2 + 0\lambda_3 + -1\lambda_4 + 1\lambda_5 \\
1 &\leq 1\lambda_1 + 1\lambda_2 + -1\lambda_3 + 1\lambda_4 + 1\lambda_5 \\
C_1 &= -1c_3 \wedge \lambda_1 \geq 0 \wedge \lambda_2 \geq 0 \wedge \lambda_3 \geq 0 \\
C_2 &= -1c_4 \wedge \lambda_4 \geq 0 \wedge \lambda_5 \geq 0
\end{align*} \]

Farkas’ lemma:
\( \lambda_1, \ldots, \lambda_5 \geq 0 \)

\[ c_1 = \sum_{i=1}^{5} \lambda_i a_{i,1} \wedge \cdots \wedge c_4 = \sum_{i=1}^{5} \lambda_i a_{i,4} \wedge 1 \leq (\sum_{i=0}^{5} \lambda_i b_i) \]
Rank function synthesis

Linear, convex, with only 3 variables ........

\[ R \triangleq 0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \]

\[ 0x' + 0y' = 0 \]
\[ 0x + -1y + 1 = 0 \]
\[ 0x' + -1y + 1 = 0 \]

\[ \psi = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} \]
\[ \begin{align*}
  c_1 &= 0\lambda_1 + 1\lambda_2 + -1\lambda_3 + 0\lambda_4 + 0\lambda_5 \\
  c_2 &= 0\lambda_1 + 0\lambda_2 + 0\lambda_3 + 0\lambda_4 + -1\lambda_5 \\
  c_3 &= -1\lambda_1 + -1\lambda_2 + 1\lambda_3 + 0\lambda_4 + 0\lambda_5 \\
  c_4 &= 0\lambda_1 + 0\lambda_2 + 0\lambda_3 + -1\lambda_4 + 1\lambda_5 \\
  1 &\leq 1\lambda_1 + 1\lambda_2 + -1\lambda_3 + 1\lambda_4 + 1\lambda_5 \\
  c_1 &= -1c_3 \land \lambda_1 \geq 0 \land \lambda_2 \geq 0 \land \lambda_3 \geq 0 \\
  c_2 &= -1c_4 \land \lambda_4 \geq 0 \land \lambda_5 \geq 0 \\
\end{align*} \]

Farkas' lemma

\[ \lambda_1, \ldots, \lambda_5 \geq 0 \]

\[ c_1 = \sum_{i=1}^{5} \lambda_i a_{i,1} \land \cdots \land c_4 = \sum_{i=1}^{5} \lambda_i a_{i,4} \land 1 \leq (\sum_{i=0}^{5} \lambda_i b_i) \]
Rank function synthesis

\[ R \triangleq -1x' + 0y' + 1x \]

Satisfying assignment:

\[
\begin{align*}
c_1 &= 1 \\
c_2 &= 0 \\
c_3 &= -1 \\
c_4 &= 0 \\
\lambda_1 &= 0 \\
\lambda_2 &= 1 \\
\lambda_3 &= 0 \\
\lambda_4 &= 0 \\
\lambda_5 &= 0
\end{align*}
\]

Farkas’ lemma:

\[
\begin{align*}
\lambda_1, \ldots, \lambda_5 &> 0 \\
c_1 &= \sum_{i=1}^{5} \lambda_i a_{i,1} \land \cdots \land c_4 = \sum_{i=1}^{5} \lambda_i a_{i,4} \land 1 \leq (\sum_{i=0}^{5} \lambda_i b_i)
\end{align*}
\]
Rank function synthesis

\[ R \triangleq -1x' + 0y' + 1x, \quad 0x' + 0y' + 0x + \ldots + 0x + 1 \leq 0 \]

\[ \Rightarrow \]

\[ \psi \triangleq c_1 x' + c_2 y' + c_3 x + c_4 y + 1 \leq 0 \]

**Farkas’ lemma.**  \( R \Rightarrow \psi \) iff there exist real multipliers \( \lambda_1, \ldots, \lambda_5 \geq 0 \) such that

\[ c_1 = \sum_{i=1}^{5} \lambda_i a_{i,1} \land \cdots \land c_4 = \sum_{i=1}^{5} \lambda_i a_{i,4} \land 1 \leq (\sum_{i=0}^{5} \lambda_i b_i) \]
\[
\begin{align*}
    f(a, b) & \triangleq c_1 a + c_2 b \\
    -f(a, b) & \triangleq c_3 b + c_4 b \\
    c_1 &= -1c_3 \\
    c_2 &= -1c_4
\end{align*}
\]

Satisfying assignment:
\[
\begin{align*}
    c_1 &= 1 & c_2 &= 0 \\
    c_3 &= -1 & c_4 &= 0 \\
    \lambda_1 &= 0 & \lambda_2 &= 1 \\
    \lambda_3 &= 0 & \lambda_4 &= 0 \\
    \lambda_5 &= 0 & \quad & \leq 0
\end{align*}
\]

\[
0x + -1y \quad \leq 0
\]
\[
0x + 1y + 1 \quad \leq 0
\]

\[
\psi \triangleq c_1 x' + c_2 y' + c_3 x + c_4 y + 1 \quad \leq 0
\]

**Farkas’ lemma.** \( R \Rightarrow \psi \) iff there exist real multipliers \( \lambda_1, \ldots, \lambda_5 \geq 0 \) such that
\[
c_1 = \sum_{i=1}^{5} \lambda_i a_{i,1} \quad \land \cdots \land \quad c_4 = \sum_{i=1}^{5} \lambda_i a_{i,4} \quad \land \quad 1 \leq (\sum_{i=0}^{5} \lambda_i b_i)
\]
\[ f(a, b) \triangleq c_1 a + c_2 b \]
\[ -f(a, b) \triangleq c_3 b + c_4 b \]
\[ c_1 = -1c_3 \]
\[ c_2 = -1c_4 \]

Satisfying assignment:
\[
\begin{align*}
c_1 &= 1 \\
c_2 &= 0 \\
c_3 &= -1 \\
c_4 &= 0 \\
\lambda_1 &= 0 \\
\lambda_2 &= 1 \\
\lambda_3 &= 0 \\
\lambda_4 &= 0 \\
\lambda_5 &= 0
\end{align*}
\]

\[
0x + 0x + 0y + 1y + 1 \leq 0
\]

\[ f(x, y) \triangleq \frac{1}{2}x + 0y \]
\[ -f(x, y) \triangleq \frac{1}{2}x + 0y \]

\[ \psi \triangleq c_1 x' + c_2 y' + c_3 x + c_4 y + 1 \leq 0 \]

**Farkas’ lemma.** \( R \Rightarrow \psi \) iff there exist real multipliers \( \lambda_1, \ldots, \lambda_5 \geq 0 \) such that
\[
\begin{align*}
c_1 &= \sum_{i=1}^{5} \lambda_i a_{i,1} \land \cdots \land c_4 &= \sum_{i=1}^{5} \lambda_i a_{i,4} \land 1 \leq (\sum_{i=0}^{5} \lambda_i b_i)
\end{align*}
\]
Rank function synthesis

\[ R \triangleq \begin{align*}
0x' &\quad + \quad 0y' &\quad + \quad -1x &\quad + \quad 0y &\quad + \quad 1 &\quad \leq &\quad 0 \\
1x' &\quad + \quad 0y' &\quad + \quad -1x &\quad + \quad 0y &\quad + \quad 1 &\quad \leq &\quad 0 \\
-1x' &\quad + \quad 0y' &\quad + \quad 1x &\quad + \quad 0y &\quad + \quad -1 &\quad \leq &\quad 0 \\
0x' &\quad + \quad 0y' &\quad + \quad 0x &\quad + \quad -1y &\quad + \quad 1 &\quad \leq &\quad 0 \\
0x' &\quad + \quad -1y' &\quad + \quad 0x &\quad + \quad 1y &\quad + \quad 1 &\quad \leq &\quad 0
\end{align*} \]

\[ \Rightarrow \]

\[ \psi \triangleq c_1 x' + c_2 y' + c_3 x + c_4 y + 1 \leq 0 \]
Rank function synthesis

\[ R \triangleq \begin{align*}
0x' &+ 0y' &+ \; -1x &+ 0y &+ \; 1 &\leq 0 \\
1x' &+ 0y' &+ \; -1x &+ 0y &+ \; 1 &\leq 0 \\
-1x' &+ 0y' &+ \; 1x &+ 0y &+ \; -1 &\leq 0 \\
0x' &+ 0y' &+ \; 0x &+ \; -1y &+ \; 1 &\leq 0 \\
0x' &+ \; -1y' &+ \; 0x &+ \; 1y &+ \; 1 &\leq 0
\end{align*} \]

\[ \implies f(x', y') + f(x, y) + 1 \leq 0 \\
\quad -f(x', y') + b \leq 0 \]
Rank function synthesis

\[
R \triangleq \begin{align*}
0x' &+ 0y' &+ &-1x &+ &0y &+ &1 &
\leq &0 \\
1x' &+ 0y' &+ &-1x &+ &0y &+ &1 &
\leq &0 \\
-1x' &+ 0y' &+ &1x &+ &0y &+ &-1 &
\leq &0 \\
0x' &+ 0y' &+ &0x &+ &-1y &+ &1 &
\leq &0 \\
0x' &+ &-1y' &+ &0x &+ &1y &+ &1 &
\leq &0
\end{align*}
\]

\[
\Rightarrow \\
\begin{align*}
\rho(x, y) &\triangleq \frac{1}{2}x + 0y \\
-f(x, y) &\triangleq -\frac{1}{2}x + 0y \\
b &=-1
\end{align*}
\]

\[
f(x', y') &+ -f(x, y) &+ &1 &
\leq &0 \\
-f(x', y') &+ b &
\leq &0
\]
Rank function synthesis

\[ R \triangleq \]

\[
\begin{align*}
0x' &+ 0y' + -1x + 0y + 1 \leq 0 \\
1x' &+ 0y' + -1x + 0y + 1 \leq 0 \\
-1x' &+ 0y' + 1x + 0y + -1 \leq 0 \\
0x' &+ 0y' + 0x + -1y + 1 \leq 0 \\
0x' &+ -1y' + 0x + 1y + 1 \leq 0 \\
\end{align*}
\]

\[
\Rightarrow \quad \begin{align*}
\rho(x, y) &\triangleq \frac{1}{2}x + oy \\
-\rho(x, y) &\triangleq -\frac{1}{2}x + oy \\
\end{align*}
\]

\[
b = -1
\]

\[
x' + -x + 1 \leq 0 \\
-x' + -1 \leq 0
\]
Rank function synthesis

\[ R \triangleq \begin{aligned}
0x' &+ 0y' + -1x + 0y + 1 \leq 0 \\
1x' &+ 0y' + -1x + 0y + 1 \leq 0 \\
-1x' &+ 0y' + 1x + 0y + -1 \leq 0 \\
0x' &+ 0y' + 0x + -1y + 1 \leq 0 \\
0x' &+ -1y' + 0x + 1y + 1 \leq 0
\end{aligned} \]

\[ \subseteq \]

\[ \geq x,0 \]
\[ 0x' + 0y' + -1x + 0y + 1 \leq 0 \]

\[ R \triangleq \]

```
$ cat input.txt
relation< from<X,Y>
    , to<Xp,Yp>
    , constraint< [ X>0
                    , Xp=X-1
                    , Y>0
                    , Yp>Y
                    ]>
    , dump('output.txt')>

$ rankfinder -extrarank input.txt -primed
RankFinder: Synthesis of linear ranking functions.
Computing primed boundedness constraint \( rx' \geq d \_0 \)
Ranking \( r = [1,0] \)
Bounded by \( d_0 = 0 \)
Min decrease \( d = 1 \)
```

157
Rank function synthesis

\[ 0x' + 0y' + -1x + 0y + 1 \leq 0 \]

\[ R \triangleq \]

```
$ cat input.txt
relation( from< X, Y >
   , to< X_p, Y_p >
   , constraint< [ X>0
   , X_p=X-1
   , Y>0
   , Y_p>Y
   ]>
   , dump('output.txt'))

$ rankfinder -extrarank input.txt -primed
RankFinder: Synthesis of linear ranking functions.
Computing primed boundedness constraint \( rx' \) >= d 0
Ranking \( r = [1,0] \)
Bounded by \( d0 = 0 \)
Min decrease \( d = 1 \)
```

- \( X_p \) is \( x' \)
- \( R \)
- Ranking function
- Bound
- Goes down by 1
  (fixed in our formulation)
Question: can we automatically synthesize $f$s if we limit their form?

- Linear ranking functions from linear convex relations: Yes, *always*!
- Linear ranking functions from linear non-convex relations: *Yes, sometimes*......
- Linear ranking functions from non-linear convex relations: *Yes, sometimes*.....
- Linear ranking functions with invariants from convex relations: *Yes, always*.....
- Non-linear ranking functions from non-linear convex relations: *Yes, sometimes*.....
- ................
Linear ranking functions

Not all WF linear relations have linear ranking functions

Example 1: \( R \triangleq x \geq 0 \land x' = -2x + 10 \)
- No linear \( f \) exists s.t. \( R \subseteq \geq f \)
- \( R^+ \subseteq \geq_{x,0} \cup \geq_{(-x,-10)} \)

Example 2: \( R \triangleq x > 0 \land x' = x - y \land y' = y + 1 \)
- No linear \( f \) exists s.t. \( R \subseteq \geq f \)
- \( R^+ \subseteq \geq_{x,0} \cup \geq_{(-y,0)} \)

Other examples: Ackermann’s function and most programs with complex nested loops
Today

- Introduction
- Well-founded relations and ranking functions
- Disjunctive well foundedness
- Decomposition
- Notes on rank function synthesis