A DECIDABLE THEORY: \((\mathbb{Q}, <)\)
Consider the following language.
Consider the following language.

1. The logical symbols $\land$, $\neg$, $(\exists)$.
Consider the following language.

1. The logical symbols $\land, \neg, (\exists)$.
2. Variables $x, y, z, \ldots$ that range over $\mathbb{Q}$.
Consider the following language.

1. The logical symbols $\land$, $\neg$, $(\exists)$.
2. Variables $x, y, z, \ldots$ that range over $\mathbb{Q}$.
3. Constants: all elements of $\mathbb{Q}$. 

Variables and Symbols for $(\mathbb{Q}, <)$
Consider the following language.

1. The logical symbols $\land$, $\neg$, $(\exists)$.
2. Variables $x, y, z, \ldots$ that range over $\mathbb{Q}$.
3. Constants: all elements of $\mathbb{Q}$.
4. The symbols $<$ and $=$. **Note** We do not have $+$ or $\times$. 
Atomic Formulas

An Atomic Formula is:
Atomic Formulas

An **Atomic Formula** is:

1. For any variables $x, y$, 
Atomic Formulas

An **Atomic Formula** is:

1. For any variables $x, y$,

   $x < y$
Atomic Formulas

An **Atomic Formula** is:

1. For any variables $x, y$,

   $$x < y$$

   and
Atomic Formulas

An **Atomic Formula** is:

1. For any variables $x, y$,

   $x < y$

   and

   $x = y$
Atomic Formulas

An **Atomic Formula** is:

1. For any variables $x, y$,

   $x < y$

   and

   $x = y$

are Atomic Formulas.
QL Formulas

A \((\mathbb{Q}, <)\) Formula is:

1. Any Atomic Formula is a \((\mathbb{Q}, <)\) Formula.
2. If \(\phi_1, \phi_2\) are \((\mathbb{Q}, <)\) Formulas then so are
   2.1 \(\phi_1 \land \phi_2\),
   2.2 \(\phi_1 \lor \phi_2\),
   2.3 \(\neg \phi_1\).
3. If \(\phi(x_1, \ldots, x_n)\) is a QL Formula then so is \((\exists x_i)[\phi(x_1, \ldots, x_n)]\).
QL Formulas

A $(\mathbb{Q}, <)$ Formula is:

1. Any Atomic Formula is a $(\mathbb{Q}, <)$ Formula.
A \((\mathbb{Q}, <)\) Formula is:

1. Any Atomic Formula is a \((\mathbb{Q}, <)\) Formula.
2. If \(\phi_1, \phi_2\) are \((\mathbb{Q}, <)\) Formulas then so are

\[\phi_1 \land \phi_2, \quad \phi_1 \lor \phi_2, \quad \neg \phi_1\]
QL Formulas

A \((\mathbb{Q}, <)\) Formula is:

1. Any Atomic Formula is a \((\mathbb{Q}, <)\) Formula.
2. If \(\phi_1, \phi_2\) are \((\mathbb{Q}, <)\) Formulas then so are
   2.1 \(\phi_1 \land \phi_2\),
QL Formulas

A \((\mathbb{Q}, <)\) Formula is:

1. Any Atomic Formula is a \((\mathbb{Q}, <)\) Formula.
2. If \(\phi_1, \phi_2\) are \((\mathbb{Q}, <)\) Formulas then so are
   
   2.1 \(\phi_1 \land \phi_2\),
   
   2.2 \(\phi_1 \lor \phi_2\)
QL Formulas

A \((\mathbb{Q}, <)\) Formula is:

1. Any Atomic Formula is a \((\mathbb{Q}, <)\) Formula.
2. If \(\phi_1, \phi_2\) are \((\mathbb{Q}, <)\) Formulas then so are
   
   2.1 \(\phi_1 \land \phi_2\),
   2.2 \(\phi_1 \lor \phi_2\)
   2.3 \(\neg \phi_1\)
A \((\mathbb{Q}, <)\) Formula is:

1. Any Atomic Formula is a \((\mathbb{Q}, <)\) Formula.
2. If \(\phi_1, \phi_2\) are \((\mathbb{Q}, <)\) Formulas then so are
   2.1 \(\phi_1 \land \phi_2\),
   2.2 \(\phi_1 \lor \phi_2\)
   2.3 \(\neg \phi_1\)
3. If \(\phi(x_1, \ldots, x_n)\) is a QL Formula then so is \((\exists x_i)[\phi(x_1, \ldots, x_n)]\)
The Theory of \((\mathbb{Q}, <)\)

The following problem is decidable.
The Theory of \((\mathbb{Q}, <)\)

The following problem is decidable.

- Input \(\phi\), a sentence in \((\mathbb{Q}, <)\).
The following problem is decidable.

- **Input** $\phi$, a sentence in $(\mathbb{Q}, <)$.
- **Determine if** $\phi$ is TRUE.
An Example of Quantifier Elimination

Example of Procedure

(∃w)(∀x)(∃y)[(w < x) ∧ (w < y)]

What orderings on x, y, z are consistent with w < x ∧ w < y? Note that = is allowed.

w < y < x
w < x < y
w < x = y

Hence (∃w)(∀x)(∃y)[(w < x) ∧ (w < y)] is equivalent to

(∃w)(∀x)(∃y)[(w < x < y) ∨ (w < y < x) ∨ (w < y = x)]

which is equivalent to

(∃w)(∀x)[(∃y)[w < x < y] ∨ (∃y)[w < y < x] ∨ (∃y)[w < y = x]]
An Example of Quantifier Elimination

Example of Procedure

\((\exists w)(\forall x)(\exists y)[(w < x) \land (w < y)]\)
An Example of Quantifier Elimination

Example of Procedure

\((\exists w)(\forall x)(\exists y)[(w < x) \land (w < y)]\)

Question  What orderings on \(x, y, z\) are consistent with \(w < x \land w < y\)? Note that \(=\) is allowed.
An Example of Quantifier Elimination

Example of Procedure

$(\exists w)(\forall x)(\exists y)[(w < x) \land (w < y)]$

Question  What orderings on $x, y, z$ are consistent with $w < x \land w < y$? Note that $=$ is allowed.

$w < y < x$
An Example of Quantifier Elimination

Example of Procedure
\((\exists w)(\forall x)(\exists y)[(w < x) \land (w < y)]\)

Question  What orderings on \(x, y, z\) are consistent with \(w < x \land w < y\)? Note that \(=\) is allowed.

\(w < y < x\)

\(w < x < y\)
An Example of Quantifier Elimination

Example of Procedure

$(\exists w)(\forall x)(\exists y) [(w < x) \land (w < y)]$

Question  What orderings on $x, y, z$ are consistent with $w < x \land w < y$? Note that $=$ is allowed.

- $w < y < x$
- $w < x < y$
- $w < x = y$
An Example of Quantifier Elimination

Example of Procedure

\((\exists w)(\forall x)(\exists y)[(w < x) \land (w < y)]\)

Question  What orderings on \(x, y, z\) are consistent with \(w < x \land w < y\)? Note that \(=\) is allowed.

\(w < y < x\)
\(w < x < y\)
\(w < x = y\)

Hence \((\exists w)(\forall x)(\exists y)[(w < x) \land (w < y)]\) is equiv to
An Example of Quantifier Elimination

Example of Procedure

$(\exists w)(\forall x)(\exists y)[(w < x) \land (w < y)]$

**Question** What orderings on $x$, $y$, $z$ are consistent with $w < x \land w < y$? Note that $=$ is allowed.

- $w < y < x$
- $w < x < y$
- $w < x = y$

Hence $(\exists w)(\forall x)(\exists y)[(w < x) \land (w < y)]$ is equiv to

$$(\exists w)(\forall x)(\exists y)[(w < x < y) \lor (w < y < x) \lor (w < y = x)]$$
An Example of Quantifier Elimination

Example of Procedure

$$\exists w (\forall x) (\exists y) [(w < x) \land (w < y)]$$

**Question** What orderings on $x$, $y$, $z$ are consistent with $w < x \land w < y$? Note that $=$ is allowed.

- $w < y < x$
- $w < x < y$
- $w < x = y$

Hence $(\exists w)(\forall x)(\exists y)[(w < x) \land (w < y)]$ is equiv to

$$(\exists w)(\forall x)(\exists y)[(w < x < y) \lor (w < y < x) \lor (w < y = x)]$$

which is equiv to
An Example of Quantifier Elimination

Example of Procedure

$$(\exists w)(\forall x)(\exists y)[(w < x) \land (w < y)]$$

Question  What orderings on $x, y, z$ are consistent with $w < x \land w < y$? Note that $=$ is allowed.

$w < y < x$

$w < x < y$

$w < x = y$

Hence $(\exists w)(\forall x)(\exists y)[(w < x) \land (w < y)]$ is equiv to

$$(\exists w)(\forall x)(\exists y)[(w < x < y) \lor (w < y < x) \lor (w < y = x)]$$

which is equiv to

$$(\exists w)(\forall x)(\exists y)[w < x < y] \lor (\exists y)[w < y < x] \lor (\exists y)[w < y = x]$$
An Example of Quantifier Elimination

Example of Procedure

$$(\exists w)(\forall x)(\exists y)[(w < x) \land (w < y)]$$

**Question**  What orderings on $x, y, z$ are consistent with $w < x \land w < y$? Note that $=$ is allowed.

- $w < y < x$
- $w < x < y$
- $w < x = y$

Hence $(\exists w)(\forall x)(\exists y)[(w < x) \land (w < y)]$ is equiv to

$$(\exists w)(\forall x)(\exists y)[(w < x < y) \lor (w < y < x) \lor (w < y = x)]$$

which is equiv to

$$(\exists w)(\forall x)[(\exists y)[w < x < y] \lor (\exists y)[w < y < x] \lor (\exists y)[w < y = x]]$$

Can then look at each piece separately.
An Example of Quantifier Elimination (cont)

\[(\exists y) \[ w < x < y \] \] is TRUE iff \[ w < x \] is TRUE. So can ELIM \[ y \].

\[(\exists y) \[ w < y < x \] \] is TRUE iff \[ w < x \] is TRUE. So can ELIM \[ y \].

\[(\exists w, x, y) \[ w < y = x \] \] is TRUE iff \[ w < x \] is TRUE. So ELIM \[ y \].

So \[(\exists w) (\forall x) [(\exists y) \[ w < x < y \] \lor (\exists y) \[ w < y < x \] \lor (\exists y) \[ w < y = x \]] \equiv (\exists w) (\forall x) \[ w < x \] \lor (\exists w) \[ w < x \] \lor (\exists w) \[ w < x \] \equiv (\exists w) (\forall x) \[ w < x \]

Key
We elim a \[ \exists y \]! That elim clauses is incidental.
An Example of Quantifier Elimination (cont)

\((\exists y)[w < x < y]\) is TRUE iff \(w < x\) is TRUE. So can ELIM \(y\).
An Example of Quantifier Elimination (cont)

$(\exists y)[w < x < y]$ is TRUE iff $w < x$ is TRUE. So can ELIM $y$.

$(\exists y)[w < y < x]$ is TRUE iff $w < x$ is TRUE. So can ELIM $y$. 

Key: We elim a $\exists y$! That elim clauses is incidental.
An Example of Quantifier Elimination (cont)

$(\exists y) [w < x < y]$ is TRUE iff $w < x$ is TRUE. So can ELIM $y$.

$(\exists y) [w < y < x]$ is TRUE iff $w < x$ is TRUE. So can ELIM $y$.

$(\exists w, x, y) [w < y = x]$ is TRUE iff $w < x$ is TRUE. So ELIM $y$. 

Key
We elim a $\exists y$! That elim clauses is incidental.
An Example of Quantifier Elimination (cont)

(∃y)[w < x < y] is TRUE iff w < x is TRUE. So can ELIM y.
(∃y)[w < y < x] is TRUE iff w < x is TRUE. So can ELIM y.
(∃w, x, y)[w < y = x] is TRUE iff w < x is TRUE. So ELIM y.
So

(∃w)(∀x)[(∃y)[w < x < y] ∨ (∃y)[w < y < x] ∨ (∃y)[w < y = x]] \equiv
(∃y)[w < x < y] is TRUE iff w < x is TRUE. So can ELIM y.
(∃y)[w < y < x] is TRUE iff w < x is TRUE. So can ELIM y.
(∃w, x, y)[w < y = x] is TRUE iff w < x is TRUE. So ELIM y.

So

(∃w)(∀x)[(∃y)[w < x < y] ∨ (∃y)[w < y < x] ∨ (∃y)[w < y = x]] ≡

(∃w)(∀x)[(∃y)[w < x] ∨ (∃y)[w < x] ∨ (∃y)[w < x]] ≡
An Example of Quantifier Elimination (cont)

$(\exists y)[w < x < y]$ is TRUE iff $w < x$ is TRUE. So can ELIM $y$.

$(\exists y)[w < y < x]$ is TRUE iff $w < x$ is TRUE. So can ELIM $y$.

$(\exists w, x, y)[w < y = x]$ is TRUE iff $w < x$ is TRUE. So ELIM $y$.

So

$(\exists w)(\forall x)[(\exists y)[w < x < y] \lor (\exists y)[w < y < x] \lor (\exists y)[w < y = x]] \equiv$

$(\exists w)(\forall x)[(\exists y)[w < x] \lor (\exists y)[w < x] \lor (\exists y)[w < x]] \equiv$

$(\exists w)(\forall x)[(w < x) \lor (w < x) \lor (w < x))] \equiv (\exists w)(\forall x)[w < x]$

**Key** We elim a $\exists y$! That elim clauses is incidental.
An Example of Quantifier Elimination (cont)

\[(\exists w)(\forall x)[w < x]\]
An Example of Quantifier Elimination (cont)

$(\exists w)(\forall x)[w < x]$

We can ELIM a $\exists$ quantifier. Yeah
An Example of Quantifier Elimination (cont)

\[(\exists w)(\forall x)[w < x]\]

We can ELIM a \(\exists\) quantifier. Yeah
But we have a \(\forall\) quantifier. Boo
An Example of Quantifier Elimination (cont)

\[(\exists w)(\forall x)[w < x]\]

We can ELIM a \(\exists\) quantifier. **Yeah**
But we have a \(\forall\) quantifier. **Boo**
But recall that \(\forall \equiv \neg \exists \neg\). **Yeah**
An Example of Quantifier Elimination (cont)

$$(\exists w)(\forall x)[w < x]$$

We can ELIM a $\exists$ quantifier. **Yeah**

But we have a $\forall$ quantifier. **Boo**

But recall that $\forall \equiv \neg \exists \neg$. **Yeah**

$$(\exists w)\neg(\exists x)\neg[w < x] \equiv$$
An Example of Quantifier Elimination (cont)

$$(\exists w)(\forall x)[w < x]$$

We can ELIM a $\exists$ quantifier. **Yeah**
But we have a $\forall$ quantifier. **Boo**
But recall that $\forall \equiv \neg \exists \neg$. **Yeah**

$$(\exists w)\neg (\exists x)\neg[w < x] \equiv$$

$$(\exists w)\neg (\exists x)[x \leq w]$$

Look at the inner part:
An Example of Quantifier Elimination (cont)

\[(\exists w)(\forall x)[w < x]\]

We can ELIM a \(\exists\) quantifier.  **Yeah**
But we have a \(\forall\) quantifier.  **Boo**
But recall that \(\forall \equiv \neg \exists \neg\).  **Yeah**

\[(\exists w)\neg(\exists x)\neg[w < x] \equiv\]

\[(\exists w)\neg(\exists x)[x \leq w]\]

Look at the inner part:

\[(\exists x)[x \leq w] \equiv \text{TRUE}\]
An Example of Quantifier Elimination (cont)

\((\exists w) \neg (\exists x)[x \leq w]\)

Look at the inner part:

\((\exists w) \neg (\exists x)[x \leq w]\) \equiv (\exists w)[\neg TRUE] \equiv (\exists w)[FALSE] \equiv FALSE

So the original statement is FALSE.
(\exists w \neg (\exists x)[x \leq w]

Look at the inner part:
An Example of Quantifier Elimination (cont)

\[(\exists w)\neg(\exists x)[x \leq w]\]

Look at the inner part:

\[(\exists w)\neg(\exists x)[x \leq w] \equiv (\exists w)[\neg{\text{TRUE}}] \equiv \]

\[(\exists w)[{\text{FALSE}}] \equiv \text{FALSE}\]
An Example of Quantifier Elimination (cont)

$$\exists w \neg (\exists x) [x \leq w]$$

Look at the inner part:

$$\exists w \neg (\exists x) [x \leq w] \equiv (\exists w)[\neg \text{TRUE}] \equiv$$

$$\exists w)[\text{FALSE}] \equiv \text{FALSE}$$

So the original statement is FALSE.
**Lemma on Quantifier Elimination**

**Lemma** ∃ an algorithm that will, given a sentence of the form

$$(Q_1x_1)\cdots(Q_{n-1}x_{n-1})(\exists x_n)[\phi(x_1, \ldots, x_n)]$$

(where the $Q_i$ are quantifiers) return a sentence of the form

$$(Q_1x_1)\cdots(Q_{n-1}x_{n-1})[\phi'(x_1, \ldots, x_{n-1})]$$
Lemma on Quantifier Elimination

**Lemma**  \( \exists \) an algorithm that will, given a sentence of the form

\[
(Q_1 x_1) \cdots (Q_{n-1} x_{n-1}) (\exists x_n)[\phi(x_1, \ldots, x_n)]
\]

(where the \( Q_i \) are quantifiers) return a sentence of the form

\[
(Q_1 x_1) \cdots (Q_{n-1} x_{n-1})[\phi'(x_1, \ldots, x_{n-1})]
\]

Replace \( \phi(x_1, \ldots, x_n) \) with an OR of all poss. orderings of \( x_1, \ldots, x_n \).
Lemma on Quantifier Elimination

**Lemma** ∃ an algorithm that will, given a sentence of the form

\[(Q_1x_1) \cdots (Q_{n-1}x_{n-1})(∃x_n)[ϕ(x_1, \ldots , x_n)]\]

(where the \(Q_i\) are quantifiers) return a sentence of the form

\[(Q_1x_1) \cdots (Q_{n-1}x_{n-1})[ϕ'(x_1, \ldots , x_{n-1})]\]

Replace \(ϕ(x_1, \ldots , x_n)\) with an OR of all poss. orderings of \(x_1, \ldots , x_n\).

Then replace

\[(∃x_n)[L_1(x_1, \ldots , x_n) \lor \cdots \lor L_m(x_1, \ldots , x_n)]\]

with

\[(∃x_n)[L_1(x_1, \ldots , x_n)] \lor \cdots \lor (∃x_n)[L_m(x_1, \ldots , x_n)].\]
Lemma on Quantifier Elimination

Lemma \( \exists \) an algorithm that will, given a sentence of the form

\[
(Q_1x_1) \cdots (Q_{n-1}x_{n-1})(\exists x_n)[\phi(x_1, \ldots, x_n)]
\]

(where the \( Q_i \) are quantifiers) return a sentence of the form

\[
(Q_1x_1) \cdots (Q_{n-1}x_{n-1})[\phi'(x_1, \ldots, x_{n-1})]
\]

Replace \( \phi(x_1, \ldots, x_n) \) with an OR of all poss. orderings of \( x_1, \ldots, x_n \).

Then replace

\[
(\exists x_n)[L_1(x_1, \ldots, x_n) \lor \cdots \lor L_m(x_1, \ldots, x_n)]
\]

with

\[
(\exists x_n)[L_1(x_1, \ldots, x_n)] \lor \cdots \lor (\exists x_n)[L_m(x_1, \ldots, x_n)].
\]

Each part is either \( \equiv \) to the part with \( x_n \) removed OR T or F.
(\mathbb{Q}, <) is Decidable: The Algorithm

1. \( (Q_1 \times 1) \cdots (Q_n \times n) \)\[ \phi(x_1, \ldots, x_n) \]. Replace \( \forall \) with \( \neg \exists \neg \).

2. Apply the Quant Elim Lemma over and over again until either you end up with a TRUE or a FALSE or a sentence with one variable whose truth will be easily discerned (see next slide for more on that).
(\mathbb{Q}, <) is Decidable: The Algorithm

Algorithm

1. (Q_1 \times_1) \cdots (Q_n \times_n) [\varphi(x_1, \ldots, x_n)]. Replace \forall with \neg \exists \neg.

2. Apply the Quant Elim Lemma over and over again until either you end up with a TRUE or a FALSE or a sentence with one variable whose truth will be easily discerned (see next slide for more on that).
(\mathbb{Q}, <) \text{ is Decidable: The Algorithm}

\textbf{Algorithm}

1. \((Q_1x_1) \cdots (Q_nx_n)[\phi(x_1, \ldots, x_n)]\). Replace \(\forall\) with \(\neg\exists\neg\).
(\mathbb{Q}, <) is Decidable: The Algorithm

**Algorithm**

1. \((Q_1 x_1) \cdots (Q_n x_n)[\phi(x_1, \ldots, x_n)]\). Replace \(\forall\) with \(\neg \exists \neg\).
2. Apply the Quant Elim Lemma over and over again until either you end up with a TRUE or a FALSE or a sentence with one variable whose truth will be easily discerned (see next slide for more on that).
One Variable Sentences

We allow constants in the language, which are rationals.
One Variable Sentences

We allow constants in the language, which are rationals.

We list all possible sentences with one variable. Let $q \in \mathbb{Q}$. 
One Variable Sentences

We allow constants in the language, which are rationals.

We list all possible sentences with one variable. Let $q \in \mathbb{Q}$.

1. $(\exists x)[x = q]$, $(\exists x)[x < q]$, $(\exists x)[x > q]$. These are all TRUE.
We allow constants in the language, which are rationals.

We list all possible sentences with one variable. Let $q \in \mathbb{Q}$.

1. $(\exists x)[x = q]$, $(\exists x)[x < q]$, $(\exists x)[x > q]$. These are all TRUE.
2. $(\forall x)[x = q]$, $(\exists x)[x < q]$, $(\exists x)[x > q]$. These are all FALSE.
Is the Decidability Result Interesting?

\((\mathbb{Q}, <)\) is decidable! **Great!** We can take all of the open questions about \((\mathbb{Q}, <)\) and use the decision procedure to solve them!
Is the Decidability Result Interesting?

$(\mathbb{Q}, <)$ is decidable! **Great!** We can take all of the open questions about $(\mathbb{Q}, <)$ and use the decision procedure to solve them!

Two problems with this

A contrast to H10:

1. H10 is undec. / since interesting math can be stated.
2. $(\mathbb{Q}, <)$ is dec., but no math of interest can be stated /.

Are there any dec theories where you can state interesting math? Can such theories be used to solve interesting open problems? No.
Is the Decidability Result Interesting?

$(\mathbb{Q}, <)$ is decidable! **Great!** We can take all of the open questions about $(\mathbb{Q}, <)$ and use the decision procedure to solve them!

Two problems with this

1. The procedure to decide $(\mathbb{Q}, <)$ is slow. This might not be so bad- there are better algorithms, and we have fast machines.

2. There are no interesting open questions about $(\mathbb{Q}, <)$. That's a bigger problem.

A contrast to H10:

1. H10 is undec. Since interesting math can be stated.
2. $(\mathbb{Q}, <)$ is dec., but no math of interest can be stated. 

Are there any dec theories where you can state interesting math? Can such theories be used to solve interesting open problems? No.
Is the Decidability Result Interesting?

$(\mathbb{Q}, <)$ is decidable! **Great!** We can take all of the open questions about $(\mathbb{Q}, <)$ and use the decision procedure to solve them!

Two problems with this

1. The procedure to decide $(\mathbb{Q}, <)$ is slow. This might not be so bad- there are better algorithms, and we have fast machines.
2. There are no interesting open questions about $(\mathbb{Q}, <)$. That's a bigger problem.
Is the Decidability Result Interesting?

$(\mathbb{Q}, <)$ is decidable! **Great!** We can take all of the open questions about $(\mathbb{Q}, <)$ and use the decision procedure to solve them!

Two problems with this

1. The procedure to decide $(\mathbb{Q}, <)$ is slow. This might not be so bad- there are better algorithms, and we have fast machines.
2. There are no interesting open questions about $(\mathbb{Q}, <)$. That's a bigger problem.

A contrast to H10:

1. H10 is undecidable since interesting math can be stated.
2. $(\mathbb{Q}, <)$ is decidable, but no math of interest can be stated.
Is the Decidability Result Interesting?

(\(\mathbb{Q}, <\)) is decidable! **Great!** We can take all of the open questions about \((\mathbb{Q}, <)\) and use the decision procedure to solve them!

Two problems with this

1. The procedure to decide \((\mathbb{Q}, <)\) is slow. This might not be so bad- there are better algorithms, and we have fast machines.
2. There are no interesting open questions about \((\mathbb{Q}, <)\). Thats a bigger problem.

A contrast to H10:

1. H10 is undec. 😞 since interesting math can be stated.
Is the Decidability Result Interesting?

$(\mathbb{Q}, <)$ is decidable! **Great!** We can take all of the open questions about $(\mathbb{Q}, <)$ and use the decision procedure to solve them!

Two problems with this

1. The procedure to decide $(\mathbb{Q}, <)$ is slow. This might not be so bad- there are better algorithms, and we have fast machines.
2. There are no interesting open questions about $(\mathbb{Q}, <)$. That's a bigger problem.

A contrast to H10:

1. H10 is undec. 😞 since interesting math can be stated.
2. $(\mathbb{Q}, <)$ is dec. 😊 but no math of interest can be stated 😞.
Is the Decidability Result Interesting?

\((\mathbb{Q}, <)\) is decidable! **Great!** We can take all of the open questions about \((\mathbb{Q}, <)\) and use the decision procedure to solve them!

Two problems with this

1. The procedure to decide \((\mathbb{Q}, <)\) is slow. This might not be so bad- there are better algorithms, and we have fast machines.
2. There are no interesting open questions about \((\mathbb{Q}, <)\). That's a bigger problem.

A contrast to H10:

1. H10 is undec. 😞 since interesting math can be stated.
2. \((\mathbb{Q}, <)\) is dec. 😊 but no math of interest can be stated 😞.

Are there any dec theories where you can state interesting math?
Is the Decidability Result Interesting?

$(\mathbb{Q}, <)$ is decidable! **Great!** We can take all of the open questions about $(\mathbb{Q}, <)$ and use the decision procedure to solve them!

Two problems with this

1. The procedure to decide $(\mathbb{Q}, <)$ is slow. This might not be so bad- there are better algorithms, and we have fast machines.
2. There are no interesting open questions about $(\mathbb{Q}, <)$. That's a bigger problem.

A contrast to H10:

1. H10 is undec. 😞 since interesting math can be stated.
2. $(\mathbb{Q}, <)$ is dec. 😊 but no math of interest can be stated 😞.

Are there any dec theories where you can state interesting math? Can such theories be used to solve interesting open problems?
Is the Decidability Result Interesting?

$(\mathbb{Q}, <)$ is decidable! **Great!** We can take all of the open questions about $(\mathbb{Q}, <)$ and use the decision procedure to solve them!

Two problems with this

1. The procedure to decide $(\mathbb{Q}, <)$ is slow. This might not be so bad- there are better algorithms, and we have fast machines.
2. There are no interesting open questions about $(\mathbb{Q}, <)$. That's a bigger problem.

A contrast to H10:

1. H10 is undec. 😞 since interesting math can be stated.
2. $(\mathbb{Q}, <)$ is dec. 😊 but no math of interest can be stated 😞.

Are there any dec theories where you can state interesting math? Can such theories be used to solve interesting open problems? **No.**
Interesting Combinatorics

Some interesting combinatorics arises from the dec procedure for \((\mathbb{Q}, <)\).
Interesting Combinatorics

Some interesting combinatorics arises from the dec procedure for $(\mathbb{Q}, <)$.

1. How many ways you order $x_1, \ldots, x_n$. 
Interesting Combinatorics

Some interesting combinatorics arises from the dec procedure for $(\mathbb{Q}, <)$.

1. How many ways you order $x_1, \ldots, x_n$. We all know this is $n!$. 

Next slide for examples and the first few numbers.
Some interesting combinatorics arises from the dec procedure for $(\mathbb{Q}, <)$.

1. How many ways you order $x_1, \ldots, x_n$. We all know this is $n!$.
2. How many ways you order $x_1, \ldots, x_n$ if you allow $=$? Next slide for examples and the first few numbers.
The Horse Numbers and $H(3)$

$H(n)$ is the number of ways that $n$ horses can finish a race. Note that some could be tied.
The Horse Numbers and $H(3)$

$H(n)$ is the number of ways that $n$ horses can finish a race. Note that some could be tied.

$H(2) = 3$: $x_1 < x_2$, $x_2 < x_1$, $x_1 = x_2$.

$H(3)$ we will derive. If $x_1$ is unique least:

- $x_1 < x_2 < x_3$
- $x_1 < x_2 = x_3$
- $x_1 < x_3 < x_2$

There are 3 where $x_1$ is unique least.

There are 3 where $x_2$ is unique least.

There are 3 where $x_3$ is unique least.

If $x_1 = x_2$ is least:

- $x_1 = x_2 < x_3$

There is 1.

If $x_1 = x_3$ is least:

- $x_1 = x_3 < x_2$

There is 1.

If $x_2 = x_3$ is least:

- $x_2 = x_3 < x_1$

There is 1.

If $x_1 = x_2 = x_3$ there is 1.

Total $H(3) = 3 + 3 + 3 + 1 + 1 + 1 + 1 = 13$. 

The Horse Numbers and $H(3)$

$H(n)$ is the number of ways that $n$ horses can finish a race. Note that some could be tied.

$H(2) = 3$: $x_1 < x_2$, $x_2 < x_1$, $x_1 = x_2$.

$H(3)$ we will derive. If $x_1$ is unique least:

If $x_1 = x_2$ is least: $x_1 = x_2 < x_3$. There is 1.

If $x_1 = x_3$ is least: $x_1 = x_3 < x_2$. There is 1.

If $x_2 = x_3$ is least: $x_2 = x_3 < x_1$. There is 1.

If $x_1 = x_2 = x_3$ there is 1.

Total $H(3) = 3 + 3 + 3 + 1 + 1 + 1 + 1 = 13$. 
The Horse Numbers and $H(3)$

$H(n)$ is the number of ways that $n$ horses can finish a race. Note that some could be tied.

$H(2) = 3$: $x_1 < x_2$, $x_2 < x_1$, $x_1 = x_2$.

$H(3)$ we will derive. If $x_1$ is unique least:

1. $x_1 < x_2 < x_3$
2. $x_1 < x_2 = x_3$
3. $x_1 < x_3 < x_2$

Total $H(3) = 3 + 3 + 3 + 1 + 1 + 1 + 1 = 13$. 
The Horse Numbers and $H(3)$

$H(n)$ is the number of ways that $n$ horses can finish a race. Note that some could be tied.

$H(2) = 3$: $x_1 < x_2$, $x_2 < x_1$, $x_1 = x_2$.

$H(3)$ we will derive. If $x_1$ is unique least:

- $x_1 < x_2 < x_3$
- $x_1 < x_2 = x_3$
- $x_1 < x_3 < x_2$

There are 3 where $x_1$ is unique least.

Total $H(3) = 3 + 3 + 3 + 1 + 1 + 1 + 1 = 13$. 
The Horse Numbers and \( H(3) \)

\( H(n) \) is the number of ways that \( n \) horses can finish a race. Note that some could be tied.

\( H(2) = 3: x_1 < x_2, x_2 < x_1, x_1 = x_2. \)

\( H(3) \) we will derive. If \( x_1 \) is unique least:

\( x_1 < x_2 < x_3 \)
\( x_1 < x_2 = x_3 \)
\( x_1 < x_3 < x_2 \)

There are 3 where \( x_1 \) is unique least.

There are 3 where \( x_2 \) is unique least.
The Horse Numbers and $H(3)$

$H(n)$ is the number of ways that $n$ horses can finish a race. Note that some could be tied.

$H(2) = 3$: $x_1 < x_2$, $x_2 < x_1$, $x_1 = x_2$.

$H(3)$ we will derive. If $x_1$ is unique least:

- $x_1 < x_2 < x_3$
- $x_1 < x_2 = x_3$
- $x_1 < x_3 < x_2$

There are 3 where $x_1$ is unique least.

There are 3 where $x_2$ is unique least.

There are 3 where $x_3$ is unique least.
The Horse Numbers and $H(3)$

$H(n)$ is the number of ways that $n$ horses can finish a race. Note that some could be tied.

$H(2) = 3$: $x_1 < x_2$, $x_2 < x_1$, $x_1 = x_2$.

$H(3)$ we will derive. If $x_1$ is unique least:

- $x_1 < x_2 < x_3$
- $x_1 < x_2 = x_3$
- $x_1 < x_3 < x_2$

There are 3 where $x_1$ is unique least.
There are 3 where $x_2$ is unique least.
There are 3 where $x_3$ is unique least.

If $x_1 = x_2$ is least: $x_1 = x_2 < x_3$. There is 1.
The Horse Numbers and $H(3)$

$H(n)$ is the number of ways that $n$ horses can finish a race. Note that some could be tied.

$H(2) = 3$: $x_1 < x_2$, $x_2 < x_1$, $x_1 = x_2$.

$H(3)$ we will derive. If $x_1$ is unique least:

$x_1 < x_2 < x_3$

$x_1 < x_2 = x_3$

$x_1 < x_3 < x_2$

There are 3 where $x_1$ is unique least.

There are 3 where $x_2$ is unique least.

There are 3 where $x_3$ is unique least.

If $x_1 = x_2$ is least: $x_1 = x_2 < x_3$. There is 1.

If $x_1 = x_3$ is least: $x_1 = x_3 < x_2$. There is 1.
The Horse Numbers and $H(3)$

$H(n)$ is the number of ways that $n$ horses can finish a race. Note that some could be tied.

$H(2) = 3$: $x_1 < x_2$, $x_2 < x_1$, $x_1 = x_2$.

$H(3)$ we will derive. If $x_1$ is unique least:

- $x_1 < x_2 < x_3$
- $x_1 < x_2 = x_3$
- $x_1 < x_3 < x_2$

There are 3 where $x_1$ is unique least.

There are 3 where $x_2$ is unique least.

There are 3 where $x_3$ is unique least.

If $x_1 = x_2$ is least: $x_1 = x_2 < x_3$. There is 1.

If $x_1 = x_3$ is least: $x_1 = x_3 < x_2$. There is 1.

If $x_2 = x_3$ is least: $x_2 = x_3 < x_1$. There is 1.

Total $H(3) = 3 + 3 + 3 + 1 + 1 + 1 + 1 = 13$. 
The Horse Numbers and $H(3)$

$H(n)$ is the number of ways that $n$ horses can finish a race. Note that some could be tied.

$H(2) = 3$: $x_1 < x_2$, $x_2 < x_1$, $x_1 = x_2$.

$H(3)$ we will derive. If $x_1$ is unique least:

$x_1 < x_2 < x_3$
$x_1 < x_2 = x_3$
$x_1 < x_3 < x_2$

There are 3 where $x_1$ is unique least.

There are 3 where $x_2$ is unique least.

There are 3 where $x_3$ is unique least.

If $x_1 = x_2$ is least: $x_1 = x_2 < x_3$. There is 1.

If $x_1 = x_3$ is least: $x_1 = x_3 < x_2$. There is 1.

If $x_2 = x_3$ is least: $x_2 = x_3 < x_1$. There is 1.

If $x_1 = x_2 = x_3$ there is 1.
The Horse Numbers and $H(3)$

$H(n)$ is the number of ways that $n$ horses can finish a race. Note that some could be tied.

$H(2) = 3$: $x_1 < x_2$, $x_2 < x_1$, $x_1 = x_2$.

$H(3)$ we will derive. If $x_1$ is unique least:

$x_1 < x_2 < x_3$
$x_1 < x_2 = x_3$
$x_1 < x_3 < x_2$

There are 3 where $x_1$ is unique least.
There are 3 where $x_2$ is unique least.
There are 3 where $x_3$ is unique least.

If $x_1 = x_2$ is least: $x_1 = x_2 < x_3$. There is 1.
If $x_1 = x_3$ is least: $x_1 = x_3 < x_2$. There is 1.
If $x_2 = x_3$ is least: $x_2 = x_3 < x_1$. There is 1.
If $x_1 = x_2 = x_3$ there is 1.

Total $H(3) = 3 + 3 + 3 + 1 + 1 + 1 + 1 = 13$. 
The Horse Numbers: $H(4)$

$H(1) = 1 \quad H(2) = 3 \quad H(3) = 13.$
The Horse Numbers: $H(4)$

$H(1) = 1$  $H(2) = 3$  $H(3) = 13$.

Work with your neighbor to try to derive $H(4)$.
Hint: You use $H(2)$ and $H(3)$. 

The Horse Numbers: $H(4)$

\[ H(0) = 1 \quad H(1) = 1 \quad H(2) = 3 \quad H(3) = 13. \]
The Horse Numbers: $H(4)$

$$H(0) = 1 \quad H(1) = 1 \quad H(2) = 3 \quad H(3) = 13.$$  

1. There is ONE min. $\binom{4}{1} \times H(3)$. 

Total $H(4) = \binom{4}{1} \times H(3) + \binom{4}{2} \times H(2) + \binom{4}{3} \times H(1) + \binom{4}{4} \times H(0) = 75$. 

The Horse Numbers: $H(4)$

$$H(0) = 1 \quad H(1) = 1 \quad H(2) = 3 \quad H(3) = 13.$$ 

1. There is ONE min. \( \binom{4}{1} \times H(3) \).

2. There are TWO mins. \( \binom{4}{2} \times H(2) \).
The Horse Numbers: $H(4)$

$H(0) = 1 \quad H(1) = 1 \quad H(2) = 3 \quad H(3) = 13.$

1. There is ONE min. $\binom{4}{1} \times H(3)$.
2. There are TWO mins. $\binom{4}{2} \times H(2)$.
3. There are THREE mins. $\binom{4}{3} \times H(1)$. 

Total $H(4) = \binom{4}{1} \times H(3) + \binom{4}{2} \times H(2) + \binom{4}{3} \times H(1) = 75$. 


The Horse Numbers: $H(4)$

$H(0) = 1 \quad H(1) = 1 \quad H(2) = 3 \quad H(3) = 13.$

1. There is ONE min. $\binom{4}{1} \times H(3)$.
2. There are TWO mins. $\binom{4}{2} \times H(2)$.
3. There are THREE mins. $\binom{4}{3} \times H(1)$.
4. There are FOUR mins. $\binom{4}{4} \times H(0)$. 
The Horse Numbers: \( H(4) \)

\[
H(0) = 1 \quad H(1) = 1 \quad H(2) = 3 \quad H(3) = 13.
\]

1. There is ONE min. \( \binom{4}{1} \times H(3) \).
2. There are TWO mins. \( \binom{4}{2} \times H(2) \).
3. There are THREE mins. \( \binom{4}{3} \times H(1) \).
4. There are FOUR mins. \( \binom{4}{4} \times H(0) \).

Total

\[
H(4) = \binom{4}{1} \times H(3) + \binom{4}{2} \times H(2) + \binom{4}{3} \times H(1) + \binom{4}{0} \times H(0) = 75.
\]
The Horse Numbers: Recurrence

\[ H(n): \]

1) There is ONE min. \[ \times H(n-1). \]
2) There are TWO mins. \[ \times H(n-2). \]
(...)
\[ \times H(1). \]
\[ \times H(0). \]
\[ H(n) = \sum \]
The Horse Numbers: Recurrence

$H(n)$:

1) There is ONE min. $\binom{n}{1} \times H(n - 1)$. 

$H(n) = \binom{n}{1} H(n - 1) + \cdots + \binom{n}{n} H(0)$. 
The Horse Numbers: Recurrence

\(H(n):\)

1) There is ONE min. \(\binom{n}{1} \times H(n - 1).\)

2) There are TWO mins. \(\binom{n}{2} \times H(n - 2).\)
The Horse Numbers: Recurrence

\[ H(n): \]

1) There is ONE min. \( \binom{n}{1} \times H(n - 1) \).

2) There are TWO mins. \( \binom{n}{2} \times H(n - 2) \).
The Horse Numbers: Recurrence

\[ H(n): \]

1) There is ONE min. \( \binom{n}{1} \times H(n - 1). \)

2) There are TWO mins. \( \binom{n}{2} \times H(n - 2). \)

\[ \vdots \]

n - 1) There are \( n - 1 \) mins. \( \binom{n}{n - 1} \times H(1). \)
The Horse Numbers: Recurrence

\[ H(n): \]

1) There is ONE min. \( \binom{n}{1} \times H(n - 1) \).

2) There are TWO mins. \( \binom{n}{2} \times H(n - 2) \).

\ldots

n - 1) There are \( n - 1 \) mins. \( \binom{n}{n-1} \times H(1) \).

n) There are \( n \) mins. \( \binom{n}{n} \times H(0) \).

\[ H(n) = \binom{n}{1}H(n - 1) + \cdots + \binom{n}{n}H(0). \]
The Bill Numbers

\[ B(n) \] is the number of ways \( n \) horses can finish GIVEN that \( x_1 < x_2 \).
The Bill Numbers

\[ B(n) \] is the number of ways \( n \) horses can finish \textit{GIVEN} that \( x_1 < x_2 \).

\[ B(2) = 1 \]
The Bill Numbers

\(B(n)\) is the number of ways \(n\) horses can finish \textit{GIVEN} that \(x_1 < x_2\).

\[B(2) = 1\]
\[B(3) = 5.\]

\(x_1 < x_2 < x_3\)
\(x_1 < x_2 = x_3\)
\(x_1 < x_3 < x_2\)
\(x_1 = x_3 < x_2\)
\(x_3 < x_1 < x_2\)
The Bill Numbers

$B(n)$ is the number of ways $n$ horses can finish GIVEN that $x_1 < x_2$.

$B(2) = 1$

$B(3) = 5$.

$x_1 < x_2 < x_3$

$x_1 < x_2 = x_3$

$x_1 < x_3 < x_2$

$x_1 = x_3 < x_2$

$x_3 < x_1 < x_2$

There may be a HW where you find $B(4)$ and get a recurrence for $B(n)$. (The recurrence will also use the $H$ numbers.)