The diamond lemma and its applications

Darij Grinberg

3 May 2018 // 20 May 2018

slides:
http://www.cip.ifi.lmu.de/~grinberg/algebra/diamond-talk.pdf

references:

- Eriksson, *Strong convergence and the polygon property of 1-player games*.
- MathOverflow answer #289320 (with list of references).
- Bremner/Dotsenko, *Algebraic Operads: An Algorithmic Companion*.

1.

Bubblesort

References:

- Galashin/Grinberg/Liu, arXiv:1509.03803v2 ancillary file, Section 4.2.

Bubblesort, the game

- Consider the following 1-player game:
  - **Start** with a list $(a_1, a_2, \ldots, a_n)$ of n numbers.
  - **Allowed move:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.

Bubblesort, the game

- Consider the following 1-player game:
  - **Start** with a list $(a_1, a_2, \dots, a_n)$ of n numbers.
  - **Allowed move:** Pick any $i \in \{1, 2, \dots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.

  Questions: Will the game always terminate?

  And will the final result depend on the choice of moves?

Bubblesort, the game

- Consider the following 1-player game:
  - **Start** with a list $(a_1, a_2, \ldots, a_n)$ of n numbers.
  - **Allowed move:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.

  Questions: Will the game always terminate?
  And will the final result depend on the choice of moves?
- Example: (we underline the two entries we are about to swap)

$$(\underline{6, 2}, 1) \to$$

Bubblesort, the game

- Consider the following 1-player game:
  - **Start** with a list $(a_1, a_2, \ldots, a_n)$ of n numbers.
  - **Allowed move:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.

  Questions: Will the game always terminate?

  And will the final result depend on the choice of moves?
- Example: (we underline the two entries we are about to swap)

$$(\underline{6, 2}, 1) \to (2, \underline{6, 1}) \to$$

- Consider the following 1-player game:
  - **Start** with a list $(a_1, a_2, \ldots, a_n)$ of n numbers.
  - **Allowed move:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.

  Questions: Will the game always terminate?
  And will the final result depend on the choice of moves?
- Example: (we underline the two entries we are about to swap)

$$(\underline{6, 2}, 1) \to (2, \underline{6, 1}) \to (\underline{2, 1}, 6) \to$$

Bubblesort, the game

- Consider the following 1-player game:
  - **Start** with a list $(a_1, a_2, \ldots, a_n)$ of n numbers.
  - **Allowed move:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.

  Questions: Will the game always terminate?
  And will the final result depend on the choice of moves?
- Example: (we underline the two entries we are about to swap)

$$(\underline{6, 2}, 1) \to (2, \underline{6, 1}) \to (\underline{2, 1}, 6) \to (1, 2, 6) .$$

- Consider the following 1-player game:
  - **Start** with a list $(a_1, a_2, \ldots, a_n)$ of n numbers.
  - **Allowed move:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.

  Questions: Will the game always terminate?
  And will the final result depend on the choice of moves?
- Example: (we underline the two entries we are about to swap)

$$(\underline{6, 2}, 1) \to (2, \underline{6, 1}) \to (\underline{2, 1}, 6) \to (1, 2, 6)\,.$$

Alternatively, from the same starting position:

$$(6, \underline{2, 1}) \to (\underline{6, 1}, 2) \to (1, \underline{6, 2}) \to (1, 2, 6)\,.$$

Bubblesort, the game

- Consider the following 1-player game:
  - **Start** with a list $(a_1, a_2, \ldots, a_n)$ of n numbers.
  - **Allowed move:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.

  Questions: Will the game always terminate?
  And will the final result depend on the choice of moves?
- Example: (we underline the two entries we are about to swap)

$$(\underline{6, 2}, 1) \to (2, \underline{6, 1}) \to (\underline{2, 1}, 6) \to (1, 2, 6)\,.$$

  Alternatively, from the same starting position:

$$(6, \underline{2, 1}) \to (\underline{6, 1}, 2) \to (1, \underline{6, 2}) \to (1, 2, 6)\,.$$

  Looks good so far; what about other tuples?

Bubblesort, the game

- Consider the following 1-player game:
  - **Start** with a list $(a_1, a_2, \ldots, a_n)$ of n numbers.
  - **Allowed move:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.

  Questions: Will the game always terminate?
  And will the final result depend on the choice of moves?
- **Yes**, it will terminate
  since the number of inversions decreases at each move.

Bubblesort, the game

- Consider the following 1-player game:
  - **Start** with a list $(a_1, a_2, \ldots, a_n)$ of n numbers.
  - **Allowed move:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.

  Questions: Will the game always terminate?

  And will the final result depend on the choice of moves?
- **Yes**, it will terminate

  since the number of inversions decreases at each move.

  No, the final result doesn't depend on the choice of moves,

  since an n-tuple has only one weakly increasing permutation.

Bubblesort, the game

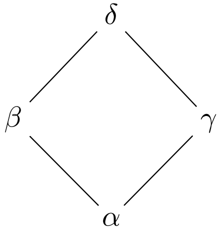
- Consider the following 1-player game:
  - **Start** with a list $(a_1, a_2, \ldots, a_n)$ of n numbers.
  - **Allowed move:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.

  Questions: Will the game always terminate?
  And will the final result depend on the choice of moves?
- **Yes**, it will terminate
  since the number of inversions decreases at each move.
  No, the final result doesn't depend on the choice of moves,
  since an n-tuple has only one weakly increasing permutation.
- This is a non-deterministic version of bubblesort (the simplest sorting algorithm ever).

Bubblesort, the game: partial order

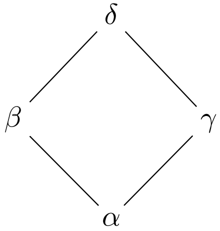
- Now what if $a_1, a_2, \ldots, a_n$ aren't numbers, but are elements of a poset instead?

- Now what if $a_1, a_2, \ldots, a_n$ aren't numbers, but are elements of a poset instead?
- Example:



Start with the list $(\delta, \gamma, \beta, \alpha)$.

- Now what if $a_1, a_2, \ldots, a_n$ aren't numbers, but are elements of a poset instead?
- Example:



Start with the list $(\delta, \gamma, \beta, \alpha)$.

$$(\underline{\delta, \gamma}, \beta, \alpha) \to (\gamma, \delta, \underline{\beta, \alpha}) \to (\gamma, \underline{\delta, \alpha}, \beta) \to (\underline{\gamma, \alpha}, \delta, \beta) \\ \to (\alpha, \gamma, \underline{\delta, \beta}) \to (\alpha, \gamma, \beta, \delta)\,.$$

We've got a linear extension of our poset, but is it still independent of the moves?

2.

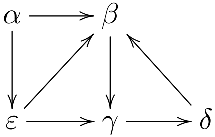
Chip-firing with a sink

References:

- Holroyd/Levine/Mészáros/Peres/Propp/Wilson, *Chip-Firing and Rotor-Routing on Directed Graphs*.
- Corry/Perkinson, *Divisors and Sandpiles*.
- Björner/Lovász, *Chip-firing games on directed graphs*.
- Spring 2017 Math 5707 homework set 5.
- more links.

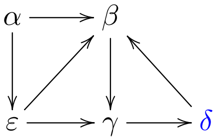
Chip-firing with a sink

- Start with a digraph (= directed graph).



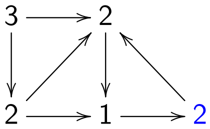
Chip-firing with a sink

- Start with a digraph (= directed graph).



Choose a vertex s that is *globally reachable* (i.e., for each vertex v, there is a path from v to s). Call it the *sink*. (Marked in blue above.)

- Start with a digraph (= directed graph).



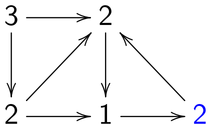
Choose a vertex s that is *globally reachable* (i.e., for each vertex v, there is a path from v to s). Call it the *sink*. (Marked in blue above.)

A *chip configuration* is a choice of nonnegative integer for each vertex. We consider a number i at vertex v to mean "i chips lying at v".

Chip-firing with a sink

- Start with a digraph (= directed graph).
  Choose a vertex s that is *globally reachable* (i.e., for each vertex v, there is a path from v to s). Call it the *sink*.
  (Marked in blue above.)
  A *chip configuration* is a choice of nonnegative integer for each vertex. We consider a number i at vertex v to mean "i chips lying at v".
- The *chip-firing game* is played as follows:
  - **Start** with a chip configuration.
  - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs,
    and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

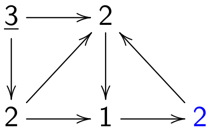
- Start with a digraph (= directed graph).



- The *chip-firing game* is played as follows:
  - **Start** with a chip configuration.
  - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs,
    and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

Chip-firing with a sink

- Start with a digraph (= directed graph).

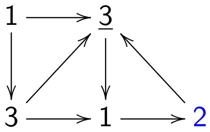


- The *chip-firing game* is played as follows:
  - **Start** with a chip configuration.
  - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs,
    and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

  Example: See image above (we underline the vertex about to be fired).

Chip-firing with a sink

- Start with a digraph (= directed graph).

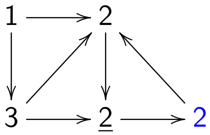


- The *chip-firing game* is played as follows:
  - **Start** with a chip configuration.
  - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs,
    and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

  Example: See image above (we underline the vertex about to be fired).

Chip-firing with a sink

- Start with a digraph (= directed graph).

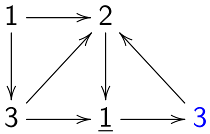


- The *chip-firing game* is played as follows:
  - **Start** with a chip configuration.
  - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs,
    and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

  Example: See image above (we underline the vertex about to be fired).

Chip-firing with a sink

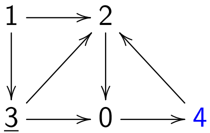
- Start with a digraph (= directed graph).



- The *chip-firing game* is played as follows:
  - **Start** with a chip configuration.
  - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs,
    and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

  Example: See image above (we underline the vertex about to be fired).

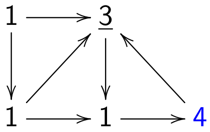
- Start with a digraph (= directed graph).



- The *chip-firing game* is played as follows:
  - **Start** with a chip configuration.
  - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs,
    and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

Example: See image above (we underline the vertex about to be fired).

- Start with a digraph (= directed graph).

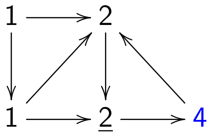


- The *chip-firing game* is played as follows:
  - **Start** with a chip configuration.
  - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs,
    and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

  Example: See image above (we underline the vertex about to be fired).

Chip-firing with a sink

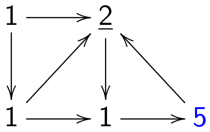
- Start with a digraph (= directed graph).



- The *chip-firing game* is played as follows:
  - **Start** with a chip configuration.
  - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs,
    and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

  Example: See image above (we underline the vertex about to be fired).

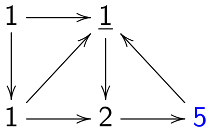
- Start with a digraph (= directed graph).



- The *chip-firing game* is played as follows:
  - **Start** with a chip configuration.
  - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs,
    and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

  Example: See image above (we underline the vertex about to be fired).

- Start with a digraph (= directed graph).

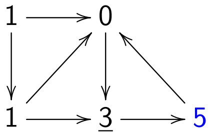


- The *chip-firing game* is played as follows:
  - **Start** with a chip configuration.
  - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs,
    and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

  Example: See image above (we underline the vertex about to be fired).

Chip-firing with a sink

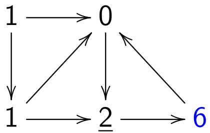
- Start with a digraph (= directed graph).



- The *chip-firing game* is played as follows:
  - **Start** with a chip configuration.
  - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs,
    and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

Example: See image above (we underline the vertex about to be fired).

- Start with a digraph (= directed graph).

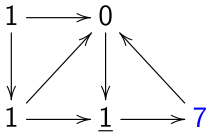


- The *chip-firing game* is played as follows:
  - **Start** with a chip configuration.
  - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs,
    and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

  Example: See image above (we underline the vertex about to be fired).

Chip-firing with a sink

- Start with a digraph (= directed graph).

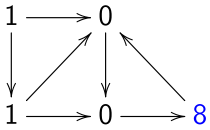


- The *chip-firing game* is played as follows:
  - **Start** with a chip configuration.
  - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs,
    and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

  Example: See image above (we underline the vertex about to be fired).

Chip-firing with a sink

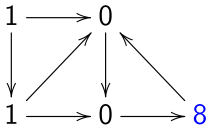
- Start with a digraph (= directed graph).



- The *chip-firing game* is played as follows:
    - **Start** with a chip configuration.
    - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs, and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

  Example: See image above (we underline the vertex about to be fired). Note: The sink cannot be fired!

- Start with a digraph (= directed graph).



- The *chip-firing game* is played as follows:
  - **Start** with a chip configuration.
  - **Allowed move:** Choose a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs,
    and "fire" it (i.e., for each arc $v \xrightarrow{a} w$, send a chip from v to w).

  Example: See image above (we underline the vertex about to be fired).

  Questions: Will the game always terminate?

  And will the final result depend on the choice of moves?

3.

The finite diamond lemma

References:

- Eriksson, *Strong convergence and the polygon property of 1-player games*.
- MathOverflow answer #289320 (with list of references).

Defining 1-player games

- Let us generalize these examples.

Defining 1-player games

- A *1-player game* consists of:
  - a set of *positions*;
  - a set of *moves*, each of which goes from one position to another.

Defining 1-player games

- A *1-player game* consists of:
  - a set of *positions*;
  - a set of *moves*, each of which goes from one position to another.

  In more familiar terms:
  - game = digraph (possibly infinite);
  - positions = vertices;
  - moves = arcs.

Defining 1-player games

- A *1-player game* consists of:
  - a set of *positions*;
  - a set of *moves*, each of which goes from one position to another.

  In more familiar terms:
  - game = digraph (possibly infinite);
  - positions = vertices;
  - moves = arcs.
- A position is said to be *terminal* if no moves are possible from this position (i.e., it is a vertex with outdegree 0).

Defining 1-player games

- A *1-player game* consists of:
  - a set of *positions*;
  - a set of *moves*, each of which goes from one position to another.

  In more familiar terms:
  - game = digraph (possibly infinite);
  - positions = vertices;
  - moves = arcs.
- A position is said to be *terminal* if no moves are possible from this position (i.e., it is a vertex with outdegree 0).
- If u and v are two positions (= vertices), then we write:
  - $u \longrightarrow v$ if we can get to v from u in one move (i.e., there is an arc $u \to v$);

Defining 1-player games

- A *1-player game* consists of:
  - a set of *positions*;
  - a set of *moves*, each of which goes from one position to another.

  In more familiar terms:
  - game = digraph (possibly infinite);
  - positions = vertices;
  - moves = arcs.
- A position is said to be *terminal* if no moves are possible from this position (i.e., it is a vertex with outdegree 0).
- If u and v are two positions (= vertices), then we write:
  - $u \longrightarrow v$ if we can get to v from u in one move (i.e., there is an arc $u \to v$);
  - $u \overset{*}{\longrightarrow} v$ if we can get to v from u in several moves (i.e., there is a walk $u \to v$).

Defining 1-player games

- A *1-player game* consists of:
  - a set of *positions*;
  - a set of *moves*, each of which goes from one position to another.

  In more familiar terms:
  - game = digraph (possibly infinite);
  - positions = vertices;
  - moves = arcs.
- A position is said to be *terminal* if no moves are possible from this position (i.e., it is a vertex with outdegree 0).
- If u and v are two positions (= vertices), then we write:
  - $u \longrightarrow v$ if we can get to v from u in one move (i.e., there is an arc $u \to v$);
  - $u \overset{*}{\longrightarrow} v$ if we can get to v from u in several moves (i.e., there is a walk $u \to v$).

  Note: $u \overset{*}{\longrightarrow} u$, since "several moves" includes "0 moves".

Monovariance, confluence

- A 1-player game is said to be:
  - *monovariant* if there is a map h from the set of all positions to $\mathbb{N}$ such that

$$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$

- A 1-player game is said to be:
  - *monovariant* if there is a map h from the set of all positions to $\mathbb{N}$ such that

    $$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$

    This means that there is a nonnegative-integer monovariant that decreases with each move. (Thus, the game always terminates.)

- A 1-player game is said to be:
  - *monovariant* if there is a map h from the set of all positions to $\mathbb{N}$ such that

    $$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$

    This means that there is a nonnegative-integer monovariant that decreases with each move. (Thus, the game always terminates.)
  - *confluent* if for each position u, there is a **unique** terminal position v such that $u \overset{*}{\longrightarrow} v$.

Monovariance, confluence

- A 1-player game is said to be:
  - *monovariant* if there is a map h from the set of all positions to $\mathbb{N}$ such that

    $$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$

    This means that there is a nonnegative-integer monovariant that decreases with each move. (Thus, the game always terminates.)
  - *confluent* if for each position u, there is a **unique** terminal position v such that $u \overset{*}{\longrightarrow} v$.
    This means that the terminal position does not depend on the choice of moves.

Monovariance, confluence

- A 1-player game is said to be:
  - *monovariant* if there is a map h from the set of all positions to $\mathbb{N}$ such that

    $$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$

    This means that there is a nonnegative-integer monovariant that decreases with each move. (Thus, the game always terminates.)
  - *confluent* if for each position u, there is a **unique** terminal position v such that $u \overset{*}{\longrightarrow} v$.
    This means that the terminal position does not depend on the choice of moves.
- So, we can answer our questions from the previous sections if we can show that our games are monovariant and confluent.

- A 1-player game is said to be:
  - *monovariant* if there is a map h from the set of all positions to $\mathbb{N}$ such that

    $$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$

    This means that there is a nonnegative-integer monovariant that decreases with each move. (Thus, the game always terminates.)
  - *confluent* if for each position u, there is a **unique** terminal position v such that $u \xrightarrow{*} v$.
    This means that the terminal position does not depend on the choice of moves.
- So, we can answer our questions from the previous sections if we can show that our games are monovariant and confluent.
- Note: Monovariance is not a necessary condition; we will loosen it later.

Local confluence

- A 1-player game is said to be:
  - *locally confluent* if for any positions u, v and w with

$$u \longrightarrow v \text{ and } u \longrightarrow w,$$

there exists a position t such that

$$v \xrightarrow{*} t \text{ and } w \xrightarrow{*} t.$$

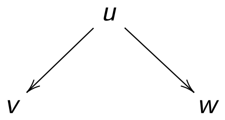
- A 1-player game is said to be:
  - *locally confluent* if for any positions u, v and w with

$$u \longrightarrow v \text{ and } u \longrightarrow w,$$

there exists a position t such that

$$v \xrightarrow{*} t \text{ and } w \xrightarrow{*} t.$$

Visually:



Local confluence

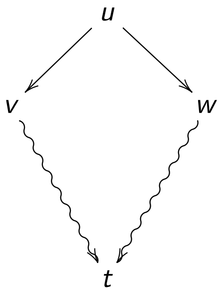
- A 1-player game is said to be:
  - *locally confluent* if for any positions u, v and w with

    $$u \longrightarrow v \text{ and } u \longrightarrow w,$$

    there exists a position t such that

    $$v \xrightarrow{*} t \text{ and } w \xrightarrow{*} t.$$

    Visually:

    .

    (Hence, local confluence is also called the "diamond condition".)

- A 1-player game is said to be:
  - *locally confluent* if for any positions u, v and w with

    $$u \longrightarrow v \text{ and } u \longrightarrow w,$$

    there exists a position t such that

    $$v \stackrel{*}{\longrightarrow} t \text{ and } w \stackrel{*}{\longrightarrow} t.$$

    This means that if a position u allows two possible moves $u \longrightarrow v$ and $u \longrightarrow w$, then there are a sequence of moves from v and a sequence of moves from w that lead to the same outcome.

Local confluence

- A 1-player game is said to be:
  - *locally confluent* if for any positions u, v and w with

$$u \longrightarrow v \text{ and } u \longrightarrow w,$$

there exists a position t such that

$$v \overset{*}{\longrightarrow} t \text{ and } w \overset{*}{\longrightarrow} t.$$

This means that if a position u allows two possible moves $u \longrightarrow v$ and $u \longrightarrow w$, then there are a sequence of moves from v and a sequence of moves from w that lead to the same outcome.
I.e., roughly speaking: There are no "watershed decisions" that lead into irreconcileable branches.

Newman's diamond lemma, finite case

- **Theorem (Newman's lemma, aka diamond lemma, in the finite case).**
  If a 1-player game is
  - **monovariant** and
  - **locally confluent**,

  then it is **confluent**.

Newman's diamond lemma, finite case

- **Theorem (Newman's lemma, aka diamond lemma, in the finite case).**
  If a 1-player game is
  - **monovariant** and
  - **locally confluent**,

  then it is **confluent**.
- In other words, in a monovariant 1-player game, confluence can be checked locally:
  If the result depends on the choice of moves, then we can pinpoint **one specific choice** that acts as a watershed.

- The diamond lemma is used in many places, and different cultures use different languages.
  Attempt at a dictionary:

our terminology	digraphs	computer science
1-player game	digraph	abstract rewriting system (ARS)
position	vertex	object
move	arc	reduction step
play sequence	walk	reduction sequence
terminal position	sink	normal form

  Here, "sink" means "vertex with outdegree 0"; this has nothing to do with the "sink" in chip-firing.

A Rosetta stone

- The diamond lemma is used in many places, and different cultures use different languages.
  Attempt at a dictionary:

our terminology	**digraphs**	**computer science**
1-player game	digraph	abstract rewriting system (ARS)
position	vertex	object
move	arc	reduction step
play sequence	walk	reduction sequence
terminal position	sink	normal form

  Here, "sink" means "vertex with outdegree 0"; this has nothing to do with the "sink" in chip-firing.
- This is related to finite-state machines, but our moves aren't determined by input.

Application: bubblesort, 1

- Recall the bubblesort game on a poset:
  - **Positions:** lists $(a_1, a_2, \ldots, a_n)$ of n elements of a poset P.
  - **Moves:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.

Application: bubblesort, 1

- Recall the bubblesort game on a poset:
  - **Positions:** lists $(a_1, a_2, \ldots, a_n)$ of n elements of a poset P.
  - **Moves:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.
- **Proposition.** The game always terminates, and the outcome does not depend on the choice of moves.

Application: bubblesort, 1

- Recall the bubblesort game on a poset:
  - **Positions:** lists $(a_1, a_2, \ldots, a_n)$ of n elements of a poset P.
  - **Moves:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.
- **Proposition.** The game always terminates, and the outcome does not depend on the choice of moves.
- *Proof:* By the diamond lemma, it suffices to prove monovariance and local confluence.

- Recall the bubblesort game on a poset:
  - **Positions:** lists $(a_1, a_2, \ldots, a_n)$ of n elements of a poset P.
  - **Moves:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.
- **Proposition.** The game always terminates, and the outcome does not depend on the choice of moves.
- *Proof:* By the diamond lemma, it suffices to prove monovariance and local confluence.

  Monovariance: Let

$$h(a_1, a_2, \ldots, a_n) = (\text{number of inversions of } (a_1, a_2, \ldots, a_n)) \\ = (\text{number of pairs } (i, j) \text{ with } i < j \text{ and } a_i > a_j).$$

  Easy to see:

$$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$

Application: bubblesort, 1

- Recall the bubblesort game on a poset:
  - **Positions:** lists $(a_1, a_2, \ldots, a_n)$ of n elements of a poset P.
  - **Moves:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.
- **Proposition.** The game always terminates, and the outcome does not depend on the choice of moves.
- *Proof:* By the diamond lemma, it suffices to prove monovariance and local confluence.

  Local confluence: If $u \longrightarrow v$ and $u \longrightarrow w$, then
  - v is obtained from u by swapping a_i with a_{i+1};
  - w is obtained from u by swapping a_j with a_{j+1}.

  Want to find t such that $v \overset{*}{\longrightarrow} t$ and $w \overset{*}{\longrightarrow} t$.
  WLOG assume $j \geq i$ (else swap v and w).

Application: bubblesort, 1

- Recall the bubblesort game on a poset:
  - **Positions:** lists $(a_1, a_2, \ldots, a_n)$ of n elements of a poset P.
  - **Moves:** Pick any $i \in \{1, 2, \ldots, n-1\}$ such that $a_i > a_{i+1}$, and swap a_i with a_{i+1}.
- **Proposition.** The game always terminates, and the outcome does not depend on the choice of moves.
- *Proof:* By the diamond lemma, it suffices to prove monovariance and local confluence.

  Local confluence: If $u \longrightarrow v$ and $u \longrightarrow w$, then
  - v is obtained from u by swapping a_i with a_{i+1};
  - w is obtained from u by swapping a_j with a_{j+1}.

  Want to find t such that $v \overset{*}{\longrightarrow} t$ and $w \overset{*}{\longrightarrow} t$.
  WLOG assume $j \geq i$ (else swap v and w).
  - **Case 1:** $j = i$. Then, $v \overset{*}{\longrightarrow} t$ and $w \overset{*}{\longrightarrow} t$ for $t = v = w$.

Application: bubblesort, 2

- *Proof (continued).* **Local confluence:**
  - **Case 2:** $j = i + 1$. Then, $a_i > a_{i+1} > a_{i+2}$ and

$$u = \left(\ldots, \underline{a_i, a_{i+1}}, a_{i+2}, \ldots\right) \longrightarrow (\ldots, a_{i+1}, a_i, a_{i+2}, \ldots) = v$$

$$u = \left(\ldots, a_i, \underline{a_{i+1}, a_{i+2}}, \ldots\right) \longrightarrow (\ldots, a_i, a_{i+2}, a_{i+1}, \ldots) = w.$$

- *Proof (continued).* **Local confluence:**
  - **Case 2:** $j = i + 1$. Then, $x > y > z$ and

$$u = (\dots, \underline{x, y}, z, \dots) \longrightarrow (\dots, y, x, z, \dots) = v \qquad \text{and}$$
$$u = (\dots, x, \underline{y, z}, \dots) \longrightarrow (\dots, x, z, y, \dots) = w.$$

(We have renamed a_i, a_{i+1}, a_{i+2} as x, y, z.)

Application: bubblesort, 2

- *Proof (continued).* **Local confluence:**
  - **Case 2:** $j = i + 1$. Then, $x > y > z$ and

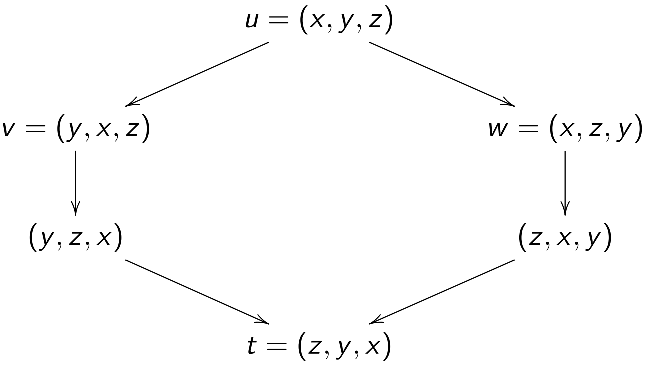
$$u = (\underline{x, y}, z) \longrightarrow (y, x, z) = v \qquad \text{and}$$
$$u = (x, \underline{y, z}) \longrightarrow (x, z, y) = w.$$

(We have omitted all other entries.)

- *Proof (continued).* **Local confluence:**
  - **Case 2:** $j = i + 1$. Then, $x > y > z$ and

$$u = (\underline{x, y}, z) \longrightarrow (y, x, z) = v \qquad \text{and}$$
$$u = (x, \underline{y, z}) \longrightarrow (x, z, y) = w.$$

"Reconcile" v and w as follows:

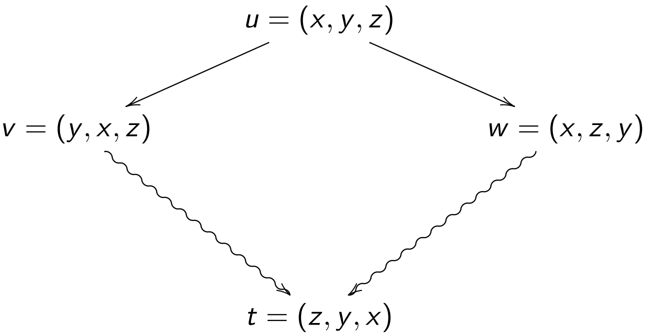


- *Proof (continued).* **Local confluence:**
  - **Case 2:** $j = i + 1$. Then, $x > y > z$ and

$$u = (\underline{x, y}, z) \longrightarrow (y, x, z) = v \qquad \text{and}$$

$$u = (x, \underline{y, z}) \longrightarrow (x, z, y) = w.$$

"Reconcile" v and w as follows:



- *Proof (continued).* **Local confluence:**
  - **Case 3:** $j > i + 1$. Then, $a_i > a_{i+1}$ and $a_j > a_{j+1}$ and

$$u = \left(\dots, \underline{a_i, a_{i+1}}, \dots, a_j, a_{j+1}, \dots\right) \longrightarrow (\dots, a_{i+1}, a_i, \dots, a_j, a_{j+1}, \dots) = v \qquad \text{and}$$

$$u = \left(\dots, a_i, a_{i+1}, \dots, \underline{a_j, a_{j+1}}, \dots\right) \longrightarrow (\dots, a_i, a_{i+1}, \dots, a_{j+1}, a_j, \dots) = w.$$

- *Proof (continued).* **Local confluence:**
  - **Case 3:** $j > i + 1$. Then, $p > q$ and $x > y$ and

$$\begin{aligned} u &= (\ldots, \underline{p, q}, \ldots, x, y, \ldots) \\ &\qquad \longrightarrow (\ldots, q, p, \ldots, x, y, \ldots) = v \qquad \text{and} \\ u &= (\ldots, p, q, \ldots, \underline{x, y}, \ldots) \\ &\qquad \longrightarrow (\ldots, p, q, \ldots, y, x, \ldots) = w. \end{aligned}$$

(We have renamed $a_i, a_{i+1}, a_j, a_{j+1}$ as p, q, x, y.)

Application: bubblesort, 3

- *Proof (continued).* **Local confluence:**
  - **Case 3:** $j > i + 1$. Then, $p > q$ and $x > y$ and

$$u = (\underline{p, q}, x, y) \longrightarrow (q, p, x, y) = v \qquad \text{and}$$

$$u = (p, q, \underline{x, y}) \longrightarrow (p, q, y, x) = w.$$

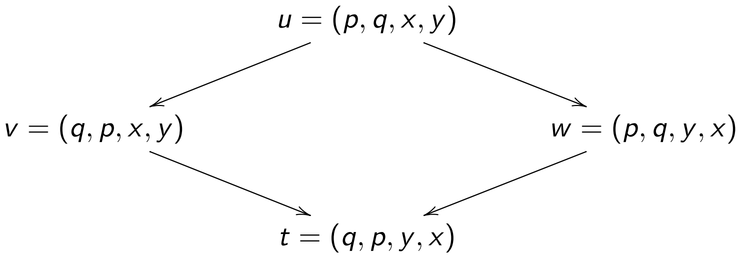
(We have omitted all other entries.)

- *Proof (continued).* **Local confluence:**
  - **Case 3:** $j > i + 1$. Then, $p > q$ and $x > y$ and

$$u = (\underline{p, q}, x, y) \longrightarrow (q, p, x, y) = v \qquad \text{and}$$

$$u = (p, q, \underline{x, y}) \longrightarrow (p, q, y, x) = w.$$

(We have omitted all other entries.)
"Reconcile" v and w as follows:



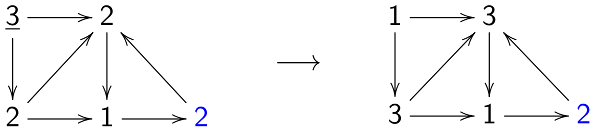
- *Proof (continued).* We have now checked both monovariance and local confluence.
  Hence, by the diamond lemma, the game is confluent, qed.

Application: bubblesort, 4

- *Proof (continued).* We have now checked both monovariance and local confluence.
  Hence, by the diamond lemma, the game is confluent, qed.
- This is a folklore fact; for a writeup, see Section 4.2 of Galashin/Grinberg/Liu, arXiv:1509.03803v2 ancillary file.

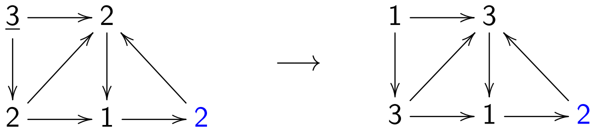
Application: chip-firing

- Recall the chip-firing game on a digraph D with vertex set V:



  - **Positions:** chip configurations, i.e., maps $f : V \to \mathbb{N}$.
  - **Moves:** “Firing” a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs.

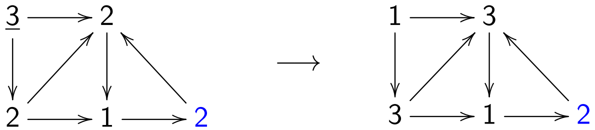
- Recall the chip-firing game on a digraph D with vertex set V:



  - **Positions:** chip configurations, i.e., maps $f : V \to \mathbb{N}$.
  - **Moves:** "Firing" a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs.
- **Proposition.** The game always terminates, and the outcome does not depend on the choice of moves.

Application: chip-firing

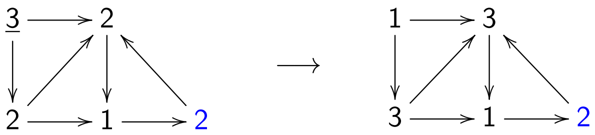
- Recall the chip-firing game on a digraph D with vertex set V:



  - **Positions:** chip configurations, i.e., maps $f : V \to \mathbb{N}$.
  - **Moves:** "Firing" a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs.
- **Proposition.** The game always terminates, and the outcome does not depend on the choice of moves.
- *Proof:* By the diamond lemma, it suffices to prove monovariance and local confluence.

Application: chip-firing

- Recall the chip-firing game on a digraph D with vertex set V:



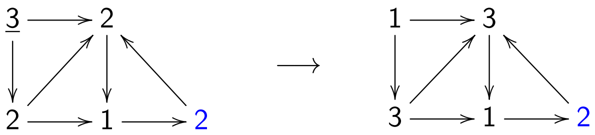
  - **Positions:** chip configurations, i.e., maps $f : V \to \mathbb{N}$.
  - **Moves:** "Firing" a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs.
- **Proposition.** The game always terminates, and the outcome does not depend on the choice of moves.
- *Proof:* By the diamond lemma, it suffices to prove monovariance and local confluence.

  Monovariance: Let $t = \sum\limits_{v \in V} f(v)$ and

$$h(f) = \sum_{v \in V} f(v) \cdot \left((t+1)^{|V|} - (t+1)^{|V| - d(v,s)} \right),$$

  where $d(v, s)$ is the minimum length of a path from v to s.

- Recall the chip-firing game on a digraph D with vertex set V:



  - **Positions:** chip configurations, i.e., maps $f : V \to \mathbb{N}$.
  - **Moves:** "Firing" a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs.
- **Proposition.** The game always terminates, and the outcome does not depend on the choice of moves.
- *Proof:* By the diamond lemma, it suffices to prove monovariance and local confluence.

  Local confluence: Easy: If two vertices can both be fired at the same time, then they can be fired in either order, and the outcome is the same.

- **Theorem (diamond lemma, in the finite case).**
  If a 1-player game is **monovariant** and **locally confluent**, then it is **confluent**.

- **Theorem (diamond lemma, in the finite case).**
  If a 1-player game is **monovariant** and **locally confluent**, then it is **confluent**.
- *Proof.* Let our game be monovariant (with function h) and locally confluent.
  We need to show that for each position u, there is a *unique* terminal position reachable from u.
  (I say that v is *reachable from* u if $u \overset{*}{\longrightarrow} v$.)
  We call this statement $\mathcal{S}(u)$.

- **Theorem (diamond lemma, in the finite case).** If a 1-player game is **monovariant** and **locally confluent**, then it is **confluent**.
- *Proof.* Let our game be monovariant (with function h) and locally confluent.
  We need to show that for each position u, there is a *unique* terminal position reachable from u.
  (I say that v is *reachable from* u if $u \overset{*}{\longrightarrow} v$.)
  We call this statement $\mathcal{S}(u)$.
  We prove $\mathcal{S}(u)$ by strong induction on $h(u)$.

- **Theorem (diamond lemma, in the finite case).**
  If a 1-player game is **monovariant** and **locally confluent**, then it is **confluent**.
- *Proof.* Let our game be monovariant (with function h) and locally confluent.
  We need to show that for each position u, there is a *unique* terminal position reachable from u.
  (I say that v is *reachable from* u if $u \xrightarrow{*} v$.)
  We call this statement $\mathcal{S}(u)$.
  We prove $\mathcal{S}(u)$ by strong induction on $h(u)$.
  Induction step: Let $h(u) = n$. Assume that $\mathcal{S}(x)$ holds for all positions x with $h(x) < n$.

- **Theorem (diamond lemma, in the finite case).**
  If a 1-player game is **monovariant** and **locally confluent**, then it is **confluent**.
- *Proof.* Let our game be monovariant (with function h) and locally confluent.
  We need to show that for each position u, there is a *unique* terminal position reachable from u.
  (I say that v is *reachable from* u if $u \overset{*}{\longrightarrow} v$.)
  We call this statement $\mathcal{S}(u)$.
  We prove $\mathcal{S}(u)$ by strong induction on $h(u)$.
  Induction step: Let $h(u) = n$. Assume that $\mathcal{S}(x)$ holds for all positions x with $h(x) < n$.
  Thus, for each position x with $h(x) < n$, there is a unique terminal position reachable from x. Call it x°; thus, $x \overset{*}{\longrightarrow} x^\circ$.

Newman's diamond lemma, finite case: proof, 1

- **Theorem (diamond lemma, in the finite case).**
  If a 1-player game is **monovariant** and **locally confluent**, then it is **confluent**.
- *Proof.* Let our game be monovariant (with function h) and locally confluent.
  We need to show that for each position u, there is a *unique* terminal position reachable from u.
  (I say that v is *reachable from* u if $u \overset{*}{\longrightarrow} v$.)
  We call this statement $\mathcal{S}(u)$.
  We prove $\mathcal{S}(u)$ by strong induction on $h(u)$.
  Induction step: Let $h(u) = n$. Assume that $\mathcal{S}(x)$ holds for all positions x with $h(x) < n$.
  Thus, for each position x with $h(x) < n$, there is a unique terminal position reachable from x. Call it x°; thus, $x \overset{*}{\longrightarrow} x^\circ$.
  WLOG u is not terminal (otherwise, $\mathcal{S}(u)$ is obvious).

- **Theorem (diamond lemma, in the finite case).**
  If a 1-player game is **monovariant** and **locally confluent**, then it is **confluent**.
- *Proof.* Let our game be monovariant (with function h) and locally confluent.
  We need to show that for each position u, there is a *unique* terminal position reachable from u.
  (I say that v is *reachable from u* if $u \xrightarrow{*} v$.)
  We call this statement $\mathcal{S}(u)$.
  We prove $\mathcal{S}(u)$ by strong induction on $h(u)$.
  Induction step: Let $h(u) = n$. Assume that $\mathcal{S}(x)$ holds for all positions x with $h(x) < n$.
  Thus, for each position x with $h(x) < n$, there is a unique terminal position reachable from x. Call it x°; thus, $x \xrightarrow{*} x^\circ$.
  WLOG u is not terminal (otherwise, $\mathcal{S}(u)$ is obvious).
  Thus there is a v such that $u \longrightarrow v$.

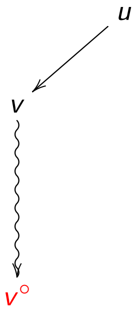
- **Theorem (diamond lemma, in the finite case).** If a 1-player game is **monovariant** and **locally confluent**, then it is **confluent**.
- *Proof.* Let our game be monovariant (with function h) and locally confluent.
  We need to show that for each position u, there is a *unique* terminal position reachable from u.
  (I say that v is *reachable from* u if $u \xrightarrow{*} v$.)
  We call this statement $\mathcal{S}(u)$.
  We prove $\mathcal{S}(u)$ by strong induction on $h(u)$.
  Induction step: Let $h(u) = n$. Assume that $\mathcal{S}(x)$ holds for all positions x with $h(x) < n$.
  Thus, for each position x with $h(x) < n$, there is a unique terminal position reachable from x. Call it x°; thus, $x \xrightarrow{*} x^\circ$.
  WLOG u is not terminal (otherwise, $\mathcal{S}(u)$ is obvious).
  Thus there is a v such that $u \longrightarrow v$.
  Hence, $h(v) < h(u) = n$, so that v° exists.

Newman's diamond lemma, finite case: proof, 2

- *Proof (continued).* We must prove $\mathcal{S}(u)$. So far we know:
  - u is a position with $h(u) = n$.
  - v is a position with $u \longrightarrow v$.
  - v° is a terminal position with $u \longrightarrow v \xrightarrow{*} v^\circ$.

- *Proof (continued).* We must prove $\mathcal{S}(u)$. So far we know:
  - u is a position with $h(u) = n$.
  - v is a position with $u \longrightarrow v$.
  - v° is a terminal position with $u \longrightarrow v \overset{*}{\longrightarrow} v^\circ$.

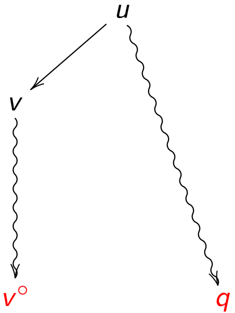
  Thus, there exists **some** terminal position reachable from u (namely, v°). Remains to prove its uniqueness.



- *Proof (continued).* We must prove $\mathcal{S}(u)$. So far we know:
  - u is a position with $h(u) = n$.
  - v is a position with $u \longrightarrow v$.
  - v° is a terminal position with $u \longrightarrow v \overset{*}{\longrightarrow} v^\circ$.

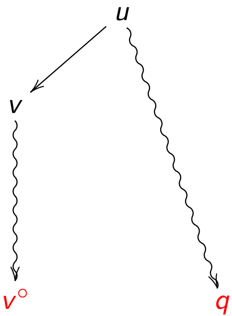
  Thus, there exists **some** terminal position reachable from u (namely, v°). Remains to prove its uniqueness.

  Let q be any other terminal position reachable from u. We want to prove $q = v^\circ$.



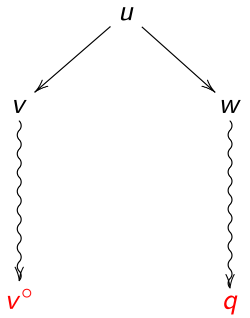
Newman's diamond lemma, finite case: proof, 2

- *Proof (continued).* We must prove $\mathcal{S}(u)$. So far we know:
  - u is a position with $h(u) = n$.
  - v is a position with $u \longrightarrow v$.
  - v° is a terminal position with $u \longrightarrow v \xrightarrow{*} v^\circ$.



Since q is terminal but u is not, we have $u \longrightarrow w \xrightarrow{*} q$ for some position w.

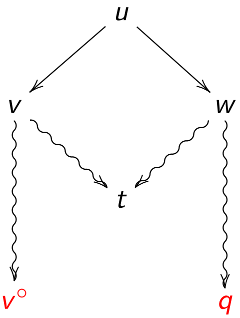
- *Proof (continued).* We must prove $\mathcal{S}(u)$. So far we know:
  - u is a position with $h(u) = n$.
  - v is a position with $u \longrightarrow v$.
  - v° is a terminal position with $u \longrightarrow v \xrightarrow{*} v^\circ$.



Local confluence shows that there is a t satisfying $v \xrightarrow{*} t$ and $w \xrightarrow{*} t$.

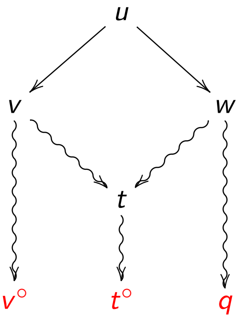
Newman's diamond lemma, finite case: proof, 2

- *Proof (continued).* We must prove $\mathcal{S}(u)$. So far we know:
  - u is a position with $h(u) = n$.
  - v is a position with $u \longrightarrow v$.
  - v° is a terminal position with $u \longrightarrow v \overset{*}{\longrightarrow} v^{\circ}$.



$h(t) \leq h(v) < h(u) = n$; thus, t° is well-defined.

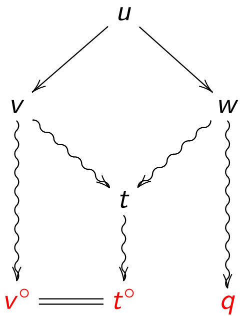
- *Proof (continued).* We must prove $\mathcal{S}(u)$. So far we know:
  - u is a position with $h(u) = n$.
  - v is a position with $u \longrightarrow v$.
  - v° is a terminal position with $u \longrightarrow v \overset{*}{\longrightarrow} v^\circ$.



$h(v) < h(u) = n$, so $\mathcal{S}(v)$ holds.

Thus, there is a unique terminal position reachable from v.
Since both v° and t° fit the bill, we thus obtain $v^\circ = t^\circ$.

- *Proof (continued).* We must prove $\mathcal{S}(u)$. So far we know:
  - u is a position with $h(u) = n$.
  - v is a position with $u \longrightarrow v$.
  - v° is a terminal position with $u \longrightarrow v \overset{*}{\longrightarrow} v^\circ$.

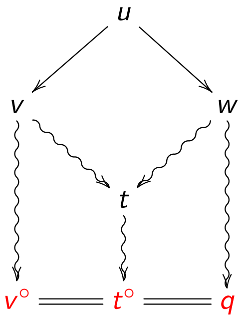


$h(w) < h(u) = n$, so $\mathcal{S}(w)$ holds.

Thus, there is a unique terminal position reachable from w.

Since both q and t° fit the bill, we thus obtain $q = t^\circ$.

- *Proof (continued).* We must prove $\mathcal{S}(u)$. So far we know:
  - u is a position with $h(u) = n$.
  - v is a position with $u \longrightarrow v$.
  - v° is a terminal position with $u \longrightarrow v \overset{*}{\longrightarrow} v^\circ$.



Thus, $q = t^\circ = v^\circ$, qed.

- **Theorem (diamond lemma, in the finite case).**
  If a 1-player game is monovariant and locally confluent, then it is confluent.

- **Theorem (diamond lemma, in the finite case).**
  If a 1-player game is monovariant and locally confluent, then it is confluent.
- **Theorem (Eriksson's polygon property theorem, in the finite case).**
  If a 1-player game is monovariant and locally confluent, with the additional property that the walks $v \xrightarrow{*} t$ and $w \xrightarrow{*} t$ in the local confluence condition have **equal lengths**,

- **Theorem (diamond lemma, in the finite case).**
  If a 1-player game is monovariant and locally confluent, then it is confluent.
- **Theorem (Eriksson's polygon property theorem, in the finite case).**
  If a 1-player game is monovariant and locally confluent, with the additional property that the walks $v \xrightarrow{*} t$ and $w \xrightarrow{*} t$ in the local confluence condition have **equal lengths**,
  then it is confluent, with the additional property that for each position v, all walks from v to the final position have **equal lengths**.

- **Theorem (diamond lemma, in the finite case).**
  If a 1-player game is monovariant and locally confluent, then it is confluent.
- **Theorem (Eriksson's polygon property theorem, in the finite case).**
  If a 1-player game is monovariant and locally confluent, with the additional property that the walks $v \xrightarrow{*} t$ and $w \xrightarrow{*} t$ in the local confluence condition have **equal lengths**,
  then it is confluent, with the additional property that for each position v, all walks from v to the final position have **equal lengths**.
  Proof idea. Let P be the set of positions.
  Define a new game, with
  - set of positions $P \times \mathbb{N}$;
  - moves $(u, k) \longrightarrow (v, k+1)$ whenever $u \longrightarrow v$ is a move of the original game and $k \in \mathbb{N}$.

  Apply the diamond lemma to this new game.

- Here is a similar, but simpler fact (exercise) also known as diamond lemma sometimes:

- **Theorem ("baby diamond lemma").** Assume that a 1-player game has the following property:
  - For any positions u, v and w with $u \longrightarrow v$ and $u \longrightarrow w$, there exists a position t such that $v \longrightarrow t$ and $w \longrightarrow t$.

- **Theorem ("baby diamond lemma").** Assume that a 1-player game has the following property:
  - For any positions u, v and w with $u \longrightarrow v$ and $u \longrightarrow w$, there exists a position t such that $v \longrightarrow t$ and $w \longrightarrow t$.

  Then:
  - For any positions u, v and w with

    $$u \overset{*}{\longrightarrow} v \text{ by a sequence of } n \text{ moves; and}$$

    $$u \overset{*}{\longrightarrow} w \text{ by a sequence of } m \text{ moves,}$$

    there exists a position t such that

    $$v \overset{*}{\longrightarrow} t \text{ by a sequence of } m \text{ moves; and}$$

    $$w \overset{*}{\longrightarrow} t \text{ by a sequence of } n \text{ moves.}$$

- **Theorem (“baby diamond lemma”).** Assume that a 1-player game has the following property:
  - For any positions u, v and w with $u \longrightarrow v$ and $u \longrightarrow w$, there exists a position t such that $v \longrightarrow t$ and $w \longrightarrow t$.

  Then:
  - For any positions u, v and w with

    $$u \overset{*}{\longrightarrow} v \text{ by a sequence of } n \text{ moves; and}$$

    $$u \overset{*}{\longrightarrow} w \text{ by a sequence of } m \text{ moves,}$$

    there exists a position t such that

    $$v \overset{*}{\longrightarrow} t \text{ by a sequence of } m \text{ moves; and}$$

    $$w \overset{*}{\longrightarrow} t \text{ by a sequence of } n \text{ moves.}$$

- Note that monovariance is not required.

- **Theorem ("baby diamond lemma").** Assume that a 1-player game has the following property:
  - For any positions u, v and w with $u \longrightarrow v$ and $u \longrightarrow w$, there exists a position t such that $v \longrightarrow t$ and $w \longrightarrow t$.

  Then:
  - For any positions u, v and w with

$$u \overset{*}{\longrightarrow} v \text{ by a sequence of } n \text{ moves; and}$$

$$u \overset{*}{\longrightarrow} w \text{ by a sequence of } m \text{ moves,}$$

    there exists a position t such that

$$v \overset{*}{\longrightarrow} t \text{ by a sequence of } m \text{ moves; and}$$

$$w \overset{*}{\longrightarrow} t \text{ by a sequence of } n \text{ moves.}$$

- Note that monovariance is not required.
- Chip-firing satisfies the above property. Bubblesort does not.

- **Theorem ("baby diamond lemma").** Assume that a 1-player game has the following property:
  - For any positions u, v and w with $u \longrightarrow v$ and $u \longrightarrow w$, there exists a position t such that $v \longrightarrow t$ and $w \longrightarrow t$.

  Then:
  - For any positions u, v and w with

    $$u \overset{*}{\longrightarrow} v \text{ by a sequence of } n \text{ moves; and}$$

    $$u \overset{*}{\longrightarrow} w \text{ by a sequence of } m \text{ moves,}$$

    there exists a position t such that

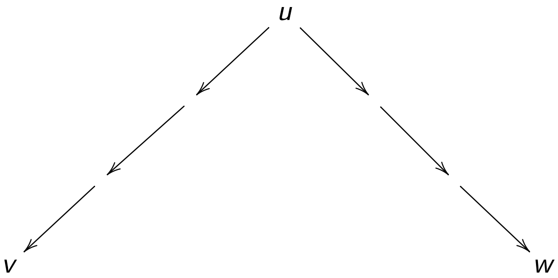
    $$v \overset{*}{\longrightarrow} t \text{ by a sequence of } m \text{ moves; and}$$

    $$w \overset{*}{\longrightarrow} t \text{ by a sequence of } n \text{ moves.}$$

- Note that monovariance is not required.
- Chip-firing satisfies the above property. Bubblesort does not.
- Some call only this theorem the "diamond lemma".

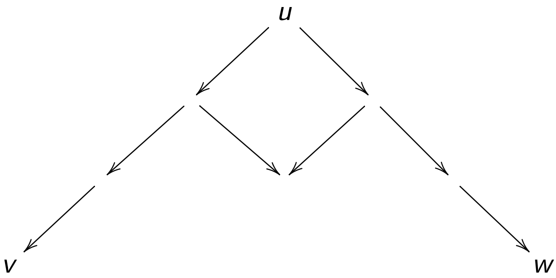
"Baby diamond lemma": proof idea

- *Proof idea for "baby diamond lemma":*



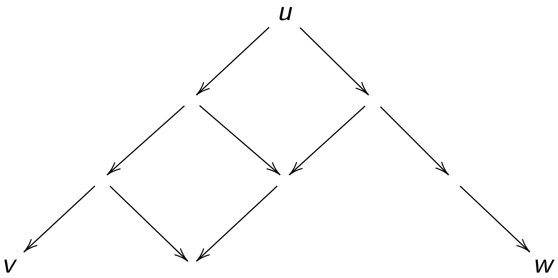
“Baby diamond lemma”: proof idea

- *Proof idea for “baby diamond lemma”:*



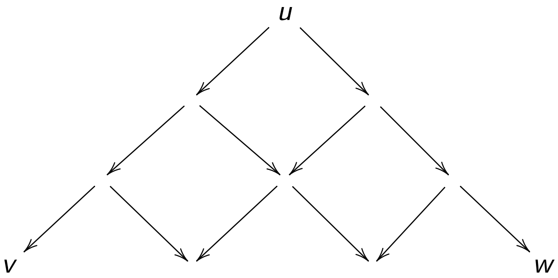
"Baby diamond lemma": proof idea

- *Proof idea for "baby diamond lemma":*



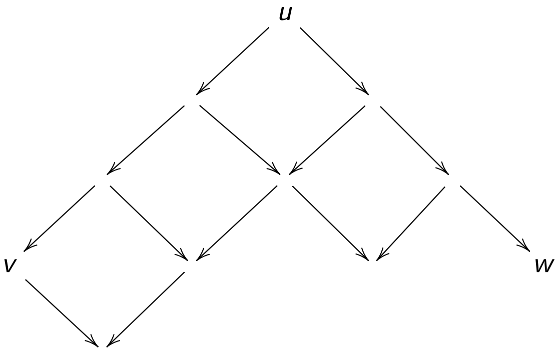
“Baby diamond lemma”: proof idea

- *Proof idea for “baby diamond lemma”:*



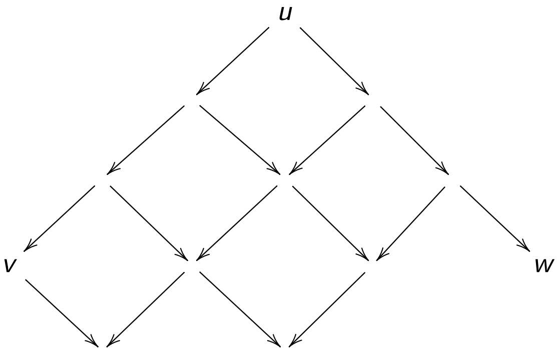
“Baby diamond lemma”: proof idea

- *Proof idea for “baby diamond lemma”:*



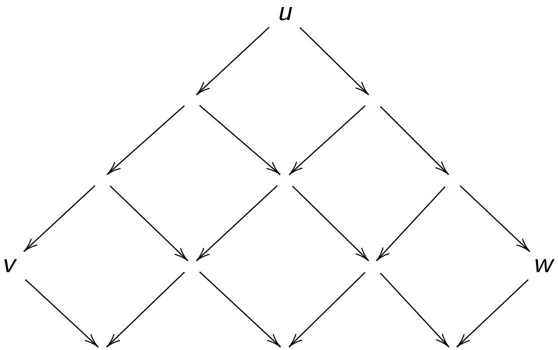
"Baby diamond lemma": proof idea

- *Proof idea for "baby diamond lemma":*



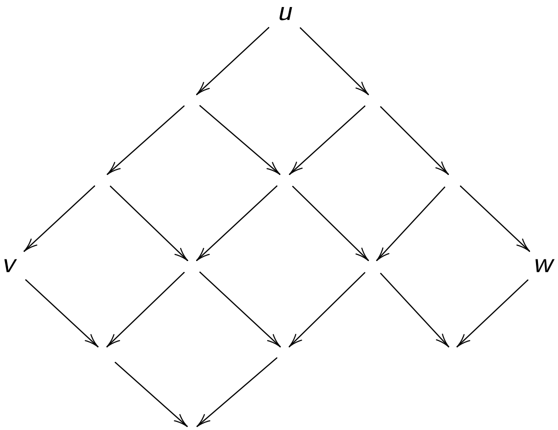
"Baby diamond lemma": proof idea

- *Proof idea for "baby diamond lemma":*



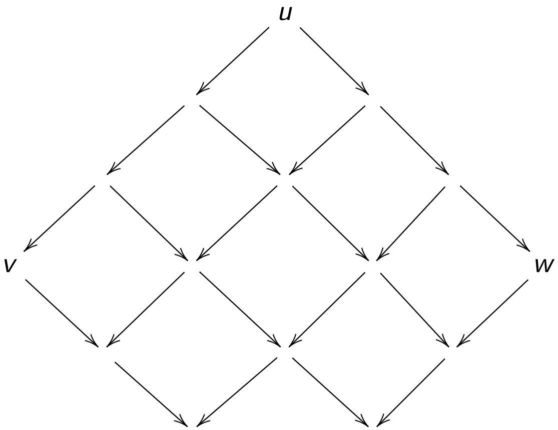
"Baby diamond lemma": proof idea

- *Proof idea for "baby diamond lemma":*



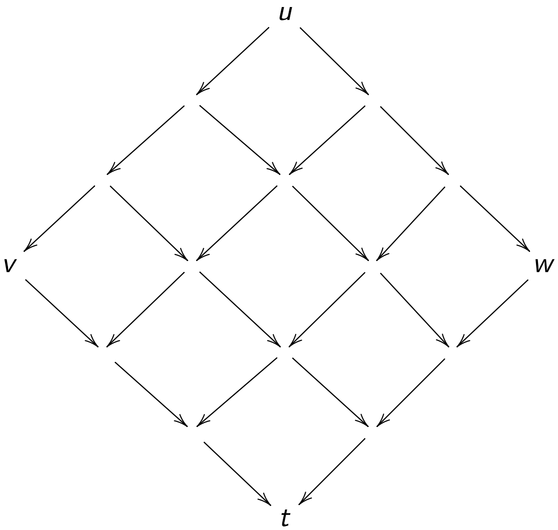
"Baby diamond lemma": proof idea

- *Proof idea for "baby diamond lemma":*



"Baby diamond lemma": proof idea

- *Proof idea for "baby diamond lemma":*



4.

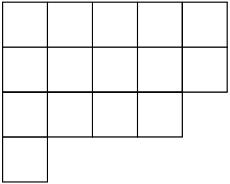
Application: the domino game

References:

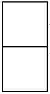
- Eriksson, *Strong convergence and the polygon property of 1-player games*.
- Olsson, *Combinatorics and Representations of Finite Groups*, sections 1–3.

Application: The domino game, 1

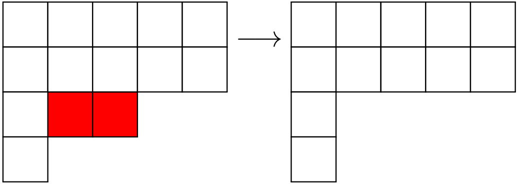
- Consider the following game:
  - **Positions:** integer partitions (drawn as Young diagrams).

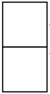


Application: The domino game, 1

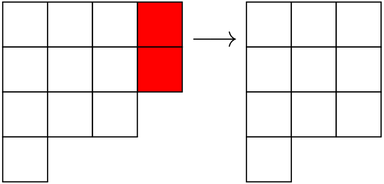
- Consider the following game:
  - **Positions:** integer partitions (drawn as Young diagrams).
  - **Moves:** Remove a domino (i.e., either  or )

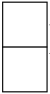
    from the outer rim (i.e., the diagram must have no cells to the right or below the domino).



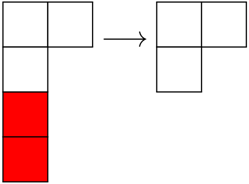
- Consider the following game:
  - **Positions:** integer partitions (drawn as Young diagrams).
  - **Moves:** Remove a domino (i.e., either  or )

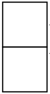
    from the outer rim (i.e., the diagram must have no cells to the right or below the domino).

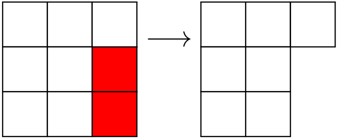


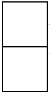
- Consider the following game:
  - **Positions:** integer partitions (drawn as Young diagrams).
  - **Moves:** Remove a domino (i.e., either  or )

    from the outer rim (i.e., the diagram must have no cells to the right or below the domino).

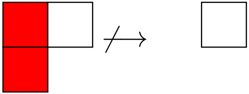


- Consider the following game:
  - **Positions:** integer partitions (drawn as Young diagrams).
  - **Moves:** Remove a domino (i.e., either  or ) from the outer rim (i.e., the diagram must have no cells to the right or below the domino).



- Consider the following game:
  - **Positions:** integer partitions (drawn as Young diagrams).
  - **Moves:** Remove a domino (i.e., either  or )

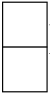
    from the outer rim (i.e., the diagram must have no cells to the right or below the domino).



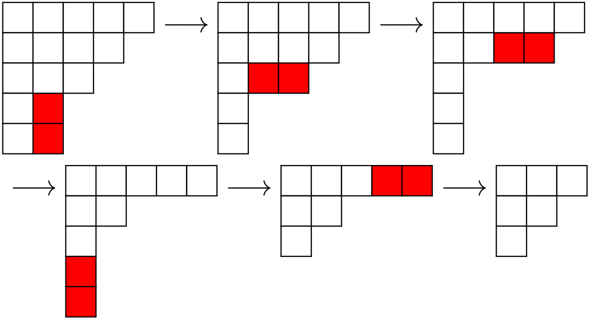
not a valid move!

[**Note:** The "outer rim" condition ensures that the result of removing the domino is still a Young diagram, **without shifting**.]

Application: The domino game, 1

- Consider the following game:
  - **Positions:** integer partitions (drawn as Young diagrams).
  - **Moves:** Remove a domino (i.e., either  or ) from the outer rim (i.e., the diagram must have no cells to the right or below the domino).

Example of the game:



- Consider the following game:
  - **Positions:** integer partitions (drawn as Young diagrams).
  - **Moves:** Remove a domino (i.e., either  or ) from the outer rim (i.e., the diagram must have no cells to the right or below the domino).

- Consider the following game:
  - **Positions:** integer partitions (drawn as Young diagrams).
  - **Moves:** Remove a domino (i.e., either  or ) from the outer rim (i.e., the diagram must have no cells to the right or below the domino).
- **Proposition.** The game terminates and is confluent (i.e., the result does not depend on the choice of moves).

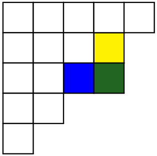
Application: The domino game, 2

- Consider the following game:
  - **Positions:** integer partitions (drawn as Young diagrams).
  - **Moves:** Remove a domino (i.e., either  or ) from the outer rim (i.e., the diagram must have no cells to the right or below the domino).
- **Proposition.** The game terminates and is confluent (i.e., the result does not depend on the choice of moves).
- *Proof.* Apply the diamond lemma.

- Consider the following game:
  - **Positions:** integer partitions (drawn as Young diagrams).
  - **Moves:** Remove a domino (i.e., either  or ) from the outer rim (i.e., the diagram must have no cells to the right or below the domino).
- **Proposition.** The game terminates and is confluent (i.e., the result does not depend on the choice of moves).
- *Proof.* Apply the diamond lemma.
  Monovariance: $h(\lambda) = |\lambda|$ decreases by 2 with each move.

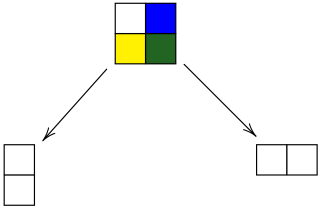
Application: The domino game, 2

- Consider the following game:
  - **Positions:** integer partitions (drawn as Young diagrams).
  - **Moves:** Remove a domino (i.e., either  or ) from the outer rim (i.e., the diagram must have no cells to the right or below the domino).
- **Proposition.** The game terminates and is confluent (i.e., the result does not depend on the choice of moves).
- *Proof.* Apply the diamond lemma.
  Local confluence: Easy check. The only nontrivial case: two overlapping dominos that can be removed simultaneously:



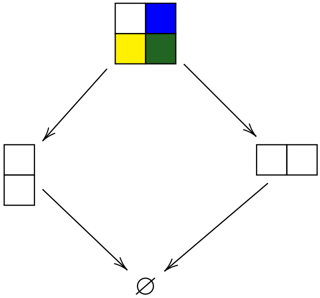
Application: The domino game, 2

- Consider the following game:
  - **Positions:** integer partitions (drawn as Young diagrams).
  - **Moves:** Remove a domino (i.e., either  or ) from the outer rim (i.e., the diagram must have no cells to the right or below the domino).
- **Proposition.** The game terminates and is confluent (i.e., the result does not depend on the choice of moves).
- *Proof.* Apply the diamond lemma.
  Local confluence: Easy check. The only nontrivial case:
  (local view:)



Application: The domino game, 2

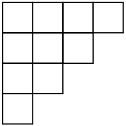
- Consider the following game:
  - **Positions:** integer partitions (drawn as Young diagrams).
  - **Moves:** Remove a domino (i.e., either  or ) from the outer rim (i.e., the diagram must have no cells to the right or below the domino).
- **Proposition.** The game terminates and is confluent (i.e., the result does not depend on the choice of moves).
- *Proof.* Apply the diamond lemma.
  Local confluence: Easy check. The only nontrivial case:
  (local view:)



Application: The domino game: terminal positions

- The terminal positions are called the 2-*cores*, aka *staircases*. They are the partitions of the form

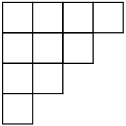
$$(m, m-1, m-2, \ldots, 1) \qquad \text{for } m \in \mathbb{N}.$$



Application: The domino game: terminal positions

- The terminal positions are called the 2-*cores*, aka *staircases*. They are the partitions of the form

$$(m, m-1, m-2, \ldots, 1) \qquad \text{for } m \in \mathbb{N}.$$



Proof idea. If a Young diagram has no dominos to remove, then it can have neither two equal-length rows, nor two equal-length columns. Thus, each row is by 1 shorter than the previous row.

Application: The domino game, generalized

- More generally, instead of removing dominos, one can remove "p-rim hooks" for any given positive integer p.
  (Eriksson calls this the "p-snake game".)
  This gives rise to "p-cores" (useful in characteristic-p representation theory of symmetric groups).

5.

The general diamond lemma

References:

- Bezem, Coquand, *Newman's Lemma – a Case Study in Proof Automation and Geometric Logic*.

- **Recall:** A 1-player game is said to be:
  - *monovariant* if there is a map h from the set of all positions to $\mathbb{N}$ such that

$$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$

- **Recall:** A 1-player game is said to be:
  - *monovariant* if there is a map h from the set of all positions to $\mathbb{N}$ such that

$$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$

- This is often too restrictive in practice.

- **Recall:** A 1-player game is said to be:
  - *monovariant* if there is a map h from the set of all positions to $\mathbb{N}$ such that

$$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$

- This is often too restrictive in practice.
- Standard answer: replace monovariance by "termination":
- A 1-player game is said to be:
  - *terminating* if there is no infinite chain $u_0 \longrightarrow u_1 \longrightarrow u_2 \longrightarrow \cdots$.

- **Recall:** A 1-player game is said to be:
  - *monovariant* if there is a map h from the set of all positions to $\mathbb{N}$ such that

$$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$

- This is often too restrictive in practice.
- Standard answer: replace monovariance by "termination":
- A 1-player game is said to be:
  - *terminating* if there is no infinite chain $u_0 \longrightarrow u_1 \longrightarrow u_2 \longrightarrow \cdots$.
- **Theorem (Newman's lemma, classical version).**
  If a 1-player game is
  - **terminating** and
  - **locally confluent**,

  then it is **confluent**.

Monovariance revisited, 1

- **Recall:** A 1-player game is said to be:
  - *monovariant* if there is a map h from the set of all positions to $\mathbb{N}$ such that

$$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$

- This is often too restrictive in practice.
- Standard answer: replace monovariance by "termination":
- A 1-player game is said to be:
  - *terminating* if there is no infinite chain $u_0 \longrightarrow u_1 \longrightarrow u_2 \longrightarrow \cdots$.
- **Theorem (Newman's lemma, classical version).**

  If a 1-player game is
  - **terminating** and
  - **locally confluent**,

  then it is **confluent**.
- This is actually an "if and only if".

- **Recall:** A 1-player game is said to be:
  - *monovariant* if there is a map h from the set of all positions to $\mathbb{N}$ such that
  $$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$
- This is often too restrictive in practice.
- Standard answer: replace monovariance by "termination":
- A 1-player game is said to be:
  - *terminating* if there is no infinite chain
  $u_0 \longrightarrow u_1 \longrightarrow u_2 \longrightarrow \cdots$.
- **Theorem (Newman's lemma, classical version).**
  If a 1-player game is
  - **terminating** and
  - **locally confluent**,

  then it is **confluent**.
- This is actually an "if and only if".
- **Bad news:** This theorem is no longer constructive, and the proof uses tricky logic.

Monovariance revisited, 2

- Fortunately, we can turn the theorem constructive **and** make the proof simple again.

Monovariance revisited, 2

- Fortunately, we can turn the theorem constructive **and** make the proof simple again.
  Trick (Bezem/Coquand): Replace “terminating” by “Noetherian”, and define the latter constructively by requiring an induction rule to work.

- Fortunately, we can turn the theorem constructive **and** make the proof simple again.
  Trick (Bezem/Coquand): Replace "terminating" by "Noetherian", and define the latter constructively by requiring an induction rule to work.
  (You might have seen Noetherian induction. Imagine defining a Noetherian space as a space on which Noetherian induction works, rather than using chains of subspaces!)

Monovariance revisited, 2

- Fortunately, we can turn the theorem constructive **and** make the proof simple again.
  Trick (Bezem/Coquand): Replace "terminating" by "Noetherian", and define the latter constructively by requiring an induction rule to work.
  (You might have seen Noetherian induction. Imagine defining a Noetherian space as a space on which Noetherian induction works, rather than using chains of subspaces!)
- I will use my own notations, but the idea is from Bezem/Coquand.

- Fortunately, we can turn the theorem constructive **and** make the proof simple again.
  Trick (Bezem/Coquand): Replace "terminating" by "Noetherian", and define the latter constructively by requiring an induction rule to work.
  (You might have seen Noetherian induction. Imagine defining a Noetherian space as a space on which Noetherian induction works, rather than using chains of subspaces!)
- I will use my own notations, but the idea is from Bezem/Coquand.
- We will use posets (= partially ordered sets); but totally ordered sets are enough for what we want to do.
  You may read "totally ordered set" for "poset" in the following.

Monovariance revisited, 3

- A poset S is said to be *Noetherian* if and only if it allows (strong) induction over $s \in S$, i.e., if the following rule holds:
  - If $\mathcal{A}(s)$ is a statement for each $s \in S$, and if each $s \in S$ satisfies

    $$(\mathcal{A}(t) \text{ for all } t < s) \Longrightarrow \mathcal{A}(s),$$

    then each $s \in S$ satisfies $\mathcal{A}(s)$.

Monovariance revisited, 3

- A poset S is said to be *Noetherian* if and only if it allows (strong) induction over $s \in S$, i.e., if the following rule holds:
  - If $\mathcal{A}(s)$ is a statement for each $s \in S$, and if each $s \in S$ satisfies

$$(\mathcal{A}(t) \text{ for all } t < s) \Longrightarrow \mathcal{A}(s),$$

    then each $s \in S$ satisfies $\mathcal{A}(s)$.
- A 1-player game is said to be:
  - *Noetherian* if there is a map h from the set of all positions to a Noetherian poset S such that

$$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$

- A poset S is said to be *Noetherian* if and only if it allows (strong) induction over $s \in S$, i.e., if the following rule holds:
  - If $\mathcal{A}(s)$ is a statement for each $s \in S$, and if each $s \in S$ satisfies

$$(\mathcal{A}(t) \text{ for all } t < s) \Longrightarrow \mathcal{A}(s),$$

    then each $s \in S$ satisfies $\mathcal{A}(s)$.
- A 1-player game is said to be:
  - *Noetherian* if there is a map h from the set of all positions to a Noetherian poset S such that

$$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$

- **Theorem (Newman's lemma, constructive version).**
  If a 1-player game is
  - **Noetherian** and
  - **locally confluent**,

  then it is **confluent**.

- A poset S is said to be *Noetherian* if and only if it allows (strong) induction over $s \in S$, i.e., if the following rule holds:
  - If $\mathcal{A}(s)$ is a statement for each $s \in S$, and if each $s \in S$ satisfies
    $$(\mathcal{A}(t) \text{ for all } t < s) \Longrightarrow \mathcal{A}(s),$$
    then each $s \in S$ satisfies $\mathcal{A}(s)$.
- A 1-player game is said to be:
  - *Noetherian* if there is a map h from the set of all positions to a Noetherian poset S such that
    $$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$
- **Theorem (Newman's lemma, constructive version).**
  If a 1-player game is
  - **Noetherian** and
  - **locally confluent**,

  then it is **confluent**.
- Note how lazy we are: All but the blue parts are copied from the finite case! The proof, too, can be directly copied over.

Examples of Noetherian posets

- This is only useful if we can find Noetherian posets S.

Examples of Noetherian posets

- This is only useful if we can find Noetherian posets S.
- In classical logic, a poset S is Noetherian if it has no infinite chains $s_0 > s_1 > s_2 > \cdots$.
  So this is just the obvious way to force the game to be terminating.

Examples of Noetherian posets

- This is only useful if we can find Noetherian posets S.
- In classical logic, a poset S is Noetherian if it has no infinite chains $s_0 > s_1 > s_2 > \cdots$.
  So this is just the obvious way to force the game to be terminating.
- In constructive logic:
  First of all, $\mathbb{N}$ is Noetherian.

Examples of Noetherian posets

- This is only useful if we can find Noetherian posets S.
- In classical logic, a poset S is Noetherian if it has no infinite chains $s_0 > s_1 > s_2 > \cdots$.
  So this is just the obvious way to force the game to be terminating.
- In constructive logic:
  First of all, $\mathbb{N}$ is Noetherian.
  So is each finite poset.

Examples of Noetherian posets

- This is only useful if we can find Noetherian posets S.
- In classical logic, a poset S is Noetherian if it has no infinite chains $s_0 > s_1 > s_2 > \cdots$.
  So this is just the obvious way to force the game to be terminating.
- In constructive logic:
  First of all, $\mathbb{N}$ is Noetherian.
  So is each finite poset.
  What else?

Lexicographic product: definition

- Let P and Q be two posets.
  The *lexicographic product* of P and Q is the poset $P \times Q$ with ordering given by

$$\big((p,q) < (p',q')\big) \iff \big((p < p') \text{ or } (p = p' \text{ and } q < q')\big) .$$

Lexicographic product: definition

- Let P and Q be two posets.
  The *lexicographic product* of P and Q is the poset $P \times Q$ with ordering given by

$$\big((p,q) < (p',q')\big) \iff \big((p < p') \text{ or } (p = p' \text{ and } q < q')\big) .$$

- If P and Q are totally ordered, then so is $P \times Q$.

Lexicographic product: definition

- Let P and Q be two posets.
  The *lexicographic product* of P and Q is the poset $P \times Q$ with ordering given by

$$\left((p,q) < \left(p', q'\right)\right) \iff \left(\left(p < p'\right) \text{ or } \left(p = p' \text{ and } q < q'\right)\right).$$

- If P and Q are totally ordered, then so is $P \times Q$.
- The lexicographic product is associative, and thus extends to several posets, yielding $P_1 \times P_2 \times \cdots \times P_k$ with lexicographic order:

Lexicographic product: definition

- Let P and Q be two posets.
  The *lexicographic product* of P and Q is the poset $P \times Q$ with ordering given by

  $$\left((p,q) < \left(p',q'\right)\right) \iff \left(\left(p < p'\right) \text{ or } \left(p = p' \text{ and } q < q'\right)\right).$$

- If P and Q are totally ordered, then so is $P \times Q$.
- The lexicographic product is associative, and thus extends to several posets, yielding $P_1 \times P_2 \times \cdots \times P_k$ with lexicographic order:

$$\begin{aligned} &((a_1, a_2, \ldots, a_k) > (b_1, b_2, \ldots, b_k)) \\ \iff &(\text{there is some } i \text{ such that } a_i > b_i, \text{ and} \\ &\text{each } j < i \text{ satisfies } a_j = b_j). \end{aligned}$$

Lexicographic product: definition

- Let P and Q be two posets.
  The *lexicographic product* of P and Q is the poset $P \times Q$ with ordering given by

$$((p,q) < (p',q')) \iff ((p < p') \text{ or } (p = p' \text{ and } q < q'))\,.$$

- If P and Q are totally ordered, then so is $P \times Q$.
- The lexicographic product is associative, and thus extends to several posets, yielding $P_1 \times P_2 \times \cdots \times P_k$ with lexicographic order:

$$\begin{aligned} &((a_1, a_2, \ldots, a_k) > (b_1, b_2, \ldots, b_k)) \\ \iff &(\text{there is some } i \text{ such that } a_i > b_i, \text{ and} \\ &\text{each } j < i \text{ satisfies } a_j = b_j)\,. \end{aligned}$$

- **Theorem.** If P and Q are Noetherian posets, then so is their lexicographic product $P \times Q$.

Lexicographic product: proof of Noetherianness

- **Theorem.** If P and Q are Noetherian posets, then so is their lexicographic product $P \times Q$.

Lexicographic product: proof of Noetherianness

- **Theorem.** If P and Q are Noetherian posets, then so is their lexicographic product $P \times Q$.
- *Proof idea.* Assume P and Q are Noetherian.

- **Theorem.** If P and Q are Noetherian posets, then so is their lexicographic product $P \times Q$.
- *Proof idea.* Assume P and Q are Noetherian.
  Let $\mathcal{A}(p, q)$ be a statement for each $(p, q) \in P \times Q$.
  Assume that each $(p, q) \in P \times Q$ satisfies

$$\left(\mathcal{A}\left(p', q'\right) \text{ for all } \left(p', q'\right) < (p, q)\right) \Longrightarrow (\mathcal{A}(p, q)).$$

  Goal: Show that each $(p, q) \in P \times Q$ satisfies $\mathcal{A}(p, q)$.

Lexicographic product: proof of Noetherianness

- **Theorem.** If P and Q are Noetherian posets, then so is their lexicographic product $P \times Q$.
- *Proof idea.* Assume P and Q are Noetherian.
  Let $\mathcal{A}(p,q)$ be a statement for each $(p,q) \in P \times Q$.
  Assume that each $(p,q) \in P \times Q$ satisfies
  $$\left(\mathcal{A}(p',q') \text{ for all } (p',q') < (p,q)\right) \Longrightarrow \left(\mathcal{A}(p,q)\right).$$
  Goal: Show that each $(p,q) \in P \times Q$ satisfies $\mathcal{A}(p,q)$.
- Prove $\mathcal{A}(p,q)$ by induction on p (thanks to Noetherianness of P, using
  $$\mathcal{A}'(p) := \left(\mathcal{A}(p,q) \text{ holds for all } q \in Q\right)$$
  as the statement) and, inside it, an induction on q (thanks to Noetherianness of Q).

Lexicographic product: proof of Noetherianness

- **Theorem.** If P and Q are Noetherian posets, then so is their lexicographic product $P \times Q$.
- *Proof idea.* Assume P and Q are Noetherian.
  Let $\mathcal{A}(p,q)$ be a statement for each $(p,q) \in P \times Q$.
  Assume that each $(p,q) \in P \times Q$ satisfies
  $$\left(\mathcal{A}(p',q') \text{ for all } (p',q') < (p,q)\right) \Longrightarrow (\mathcal{A}(p,q)).$$
  Goal: Show that each $(p,q) \in P \times Q$ satisfies $\mathcal{A}(p,q)$.
- Prove $\mathcal{A}(p,q)$ by induction on p (thanks to Noetherianness of P, using
  $$\mathcal{A}'(p) := (\mathcal{A}(p,q) \text{ holds for all } q \in Q)$$
  as the statement) and, inside it, an induction on q (thanks to Noetherianness of Q).
- This proves the Theorem.

Lexicographic product: proof of Noetherianness

- **Theorem.** If P and Q are Noetherian posets, then so is their lexicographic product $P \times Q$.
- *Proof idea.* Assume P and Q are Noetherian.
  Let $\mathcal{A}(p,q)$ be a statement for each $(p,q) \in P \times Q$.
  Assume that each $(p,q) \in P \times Q$ satisfies

$$\left(\mathcal{A}(p',q') \text{ for all } (p',q') < (p,q)\right) \Longrightarrow \left(\mathcal{A}(p,q)\right).$$

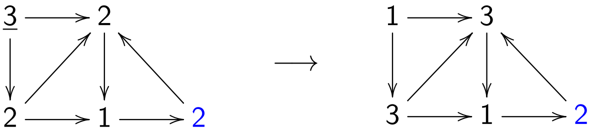
  Goal: Show that each $(p,q) \in P \times Q$ satisfies $\mathcal{A}(p,q)$.
- Prove $\mathcal{A}(p,q)$ by induction on p (thanks to Noetherianness of P, using

$$\mathcal{A}'(p) := \left(\mathcal{A}(p,q) \text{ holds for all } q \in Q\right)$$

  as the statement) and, inside it, an induction on q (thanks to Noetherianness of Q).
- This proves the Theorem.
- **Corollary.** If $P_1, P_2, \ldots, P_k$ are finitely many Noetherian posets, then their lexicographic product $P_1 \times P_2 \times \cdots \times P_k$ is Noetherian as well.

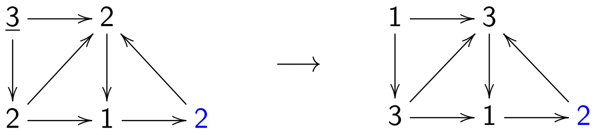
Chip-firing revisited

- **Example** for use of a lexicographic product:
- Recall the chip-firing game on a digraph D with vertex set V:



  - **Positions:** chip configurations, i.e., maps $f : V \to \mathbb{N}$.
  - **Moves:** "Firing" a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs.

- **Example** for use of a lexicographic product:
- Recall the chip-firing game on a digraph D with vertex set V:



  - **Positions:** chip configurations, i.e., maps $f : V \to \mathbb{N}$.
  - **Moves:** "Firing" a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs.
- We proved monovariance using

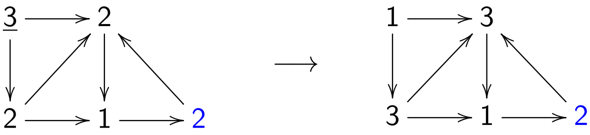
$$h : \{\text{positions}\} \to \mathbb{N},$$

$$f \mapsto \sum_{v \in V} f(v) \cdot \left((t+1)^{|V|} - (t+1)^{|V| - d(v,s)} \right),$$

where $t = \sum_{v \in V} f(v)$ and where $d(v, s)$ is the minimum length of a path from v to s.

Chip-firing revisited

- **Example** for use of a lexicographic product:
- Recall the chip-firing game on a digraph D with vertex set V:



  - **Positions:** chip configurations, i.e., maps $f : V \to \mathbb{N}$.
  - **Moves:** "Firing" a vertex $v \neq s$ that has at least as many chips as it has outgoing arcs.
- We can more easily prove Noetherianness using

h : {positions} $\to$ (lexicographic product of $m+1$ copies of $\mathbb{N}$),

$$f \mapsto \left(\sum_{v \in V;\ d(v,s) > k} f(v) \right)_{0 \leq k \leq m},$$

where $m = \max_{v \in V} d(v, s)$.
(The monovariance was an afterthought of this.)

Finite subsets form a Noetherian order: statement

- Here comes another way of constructing Noetherian totally ordered sets.

- Here comes another way of constructing Noetherian totally ordered sets.
- Let S be a totally ordered set.
  Let $\mathcal{P}_{\text{fin}}(S)$ be the set of all finite subsets of S.
  Equip $\mathcal{P}_{\text{fin}}(S)$ with a total order as follows:

  $$(A \leq B) \iff (A \subseteq B \text{ or } \max(A \setminus B) < \max(B \setminus A)).$$

  (We understand $\max(A \setminus B) < \max(B \setminus A)$ to be false if $B \subseteq A$.)

- Here comes another way of constructing Noetherian totally ordered sets.
- Let S be a totally ordered set.
  Let $\mathcal{P}_{\text{fin}}(S)$ be the set of all finite subsets of S.
  Equip $\mathcal{P}_{\text{fin}}(S)$ with a total order as follows:

$$(A \leq B) \iff (A \subseteq B \text{ or } \max(A \setminus B) < \max(B \setminus A)).$$

  (We understand $\max(A \setminus B) < \max(B \setminus A)$ to be false if $B \subseteq A$.)
  In other words, $A \leq B$ if and only if A can be obtained from B by repeatedly
  - removing an element;
  - replacing an element by (possibly several) smaller elements.

Finite subsets form a Noetherian order: statement

- Here comes another way of constructing Noetherian totally ordered sets.
- Let S be a totally ordered set.
  Let $\mathcal{P}_{\text{fin}}(S)$ be the set of all finite subsets of S.
  Equip $\mathcal{P}_{\text{fin}}(S)$ with a total order as follows:

  $$(A \leq B) \iff (A \subseteq B \text{ or } \max(A \setminus B) < \max(B \setminus A)).$$

  (We understand $\max(A \setminus B) < \max(B \setminus A)$ to be false if $B \subseteq A$.)
  In other words, $A \leq B$ if and only if A can be obtained from B by repeatedly
  - removing an element;
  - replacing an element by (possibly several) smaller elements.
- It is easy to see that $\mathcal{P}_{\text{fin}}(S)$ is totally ordered.
- **Theorem.** If S is Noetherian, then so is $\mathcal{P}_{\text{fin}}(S)$.

Finite subsets form a Noetherian order: proof, 1

- **Theorem.** If S is Noetherian, then so is $\mathcal{P}_{\text{fin}}(S)$.
- *Proof idea.* Assume S is Noetherian.

Finite subsets form a Noetherian order: proof, 1

- **Theorem.** If S is Noetherian, then so is $\mathcal{P}_{\text{fin}}(S)$.
- *Proof idea.* Assume S is Noetherian.
  For each $a \in S$, we let $S_{\leq a}$ be the subset $\{s \in S \mid s \leq a\}$ of S (a totally ordered set, with order inherited from S).
  Thus, $\mathcal{P}_{\text{fin}}(S_{\leq a})$ is a sub-totally ordered set of $\mathcal{P}_{\text{fin}}(S)$.

- **Theorem.** If S is Noetherian, then so is $\mathcal{P}_{\text{fin}}(S)$.
- *Proof idea.* Assume S is Noetherian.
  For each $a \in S$, we let $S_{\leq a}$ be the subset $\{s \in S \mid s \leq a\}$ of S (a totally ordered set, with order inherited from S).
  Thus, $\mathcal{P}_{\text{fin}}(S_{\leq a})$ is a sub-totally ordered set of $\mathcal{P}_{\text{fin}}(S)$.
- For each $a \in S$, let $\mathcal{G}(a)$ be the statement
  $(\mathcal{P}_{\text{fin}}(S_{\leq a})$ is Noetherian$)$.
  We shall prove that $\mathcal{G}(a)$ holds for all $a \in S$.
  This will easily yield the claim (since the $S_{\leq a}$ for all a cover S).

- **Theorem.** If S is Noetherian, then so is $\mathcal{P}_{\text{fin}}(S)$.
- *Proof idea.* Assume S is Noetherian.
  For each $a \in S$, we let $S_{\leq a}$ be the subset $\{s \in S \mid s \leq a\}$ of S (a totally ordered set, with order inherited from S).
  Thus, $\mathcal{P}_{\text{fin}}(S_{\leq a})$ is a sub-totally ordered set of $\mathcal{P}_{\text{fin}}(S)$.
- For each $a \in S$, let $\mathcal{G}(a)$ be the statement
  ($\mathcal{P}_{\text{fin}}(S_{\leq a})$ is Noetherian).
  We shall prove that $\mathcal{G}(a)$ holds for all $a \in S$.
  This will easily yield the claim (since the $S_{\leq a}$ for all a cover S).
- We shall prove $\mathcal{G}(a)$ by induction on a (since S is Noetherian). So we assume that $\mathcal{G}(b)$ holds for all $b < a$.

Finite subsets form a Noetherian order: proof, 1

- **Theorem.** If S is Noetherian, then so is $\mathcal{P}_{\text{fin}}(S)$.
- *Proof idea.* Assume S is Noetherian.
  For each $a \in S$, we let $S_{\leq a}$ be the subset $\{s \in S \mid s \leq a\}$ of S (a totally ordered set, with order inherited from S).
  Thus, $\mathcal{P}_{\text{fin}}(S_{\leq a})$ is a sub-totally ordered set of $\mathcal{P}_{\text{fin}}(S)$.
- For each $a \in S$, let $\mathcal{G}(a)$ be the statement
  $(\mathcal{P}_{\text{fin}}(S_{\leq a})$ is Noetherian$)$.
  We shall prove that $\mathcal{G}(a)$ holds for all $a \in S$.
  This will easily yield the claim (since the $S_{\leq a}$ for all a cover S).
- We shall prove $\mathcal{G}(a)$ by induction on a (since S is Noetherian). So we assume that $\mathcal{G}(b)$ holds for all $b < a$.
- We must prove $\mathcal{G}(a)$.
  In other words, we must prove that $\mathcal{P}_{\text{fin}}(S_{\leq a})$ is Noetherian.

Finite subsets form a Noetherian order: proof, 1

- **Theorem.** If S is Noetherian, then so is $\mathcal{P}_{\text{fin}}(S)$.
- *Proof idea.* Assume S is Noetherian.
  For each $a \in S$, we let $S_{\leq a}$ be the subset $\{s \in S \mid s \leq a\}$ of S (a totally ordered set, with order inherited from S).
  Thus, $\mathcal{P}_{\text{fin}}(S_{\leq a})$ is a sub-totally ordered set of $\mathcal{P}_{\text{fin}}(S)$.
- For each $a \in S$, let $\mathcal{G}(a)$ be the statement
  ($\mathcal{P}_{\text{fin}}(S_{\leq a})$ is Noetherian).
  We shall prove that $\mathcal{G}(a)$ holds for all $a \in S$.
  This will easily yield the claim (since the $S_{\leq a}$ for all a cover S).
- We shall prove $\mathcal{G}(a)$ by induction on a (since S is Noetherian). So we assume that $\mathcal{G}(b)$ holds for all $b < a$.
- We must prove $\mathcal{G}(a)$.
  In other words, we must prove that $\mathcal{P}_{\text{fin}}(S_{\leq a})$ is Noetherian.
- Let $\mathcal{A}(M)$ be a statement for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
  Assume that $\mathcal{A}(M)$ holds whenever all $N < M$ satisfy $\mathcal{A}(N)$.
  We must show that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.

Finite subsets form a Noetherian order: proof, 2

- What we know so far:
  (1) $\mathcal{G}(b)$ holds (that is, $\mathcal{P}_{\text{fin}}(S_{\leq b})$ is Noetherian) for all $b < a$.
  (2) $\mathcal{A}(M)$ is a statement for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
  (3) $\mathcal{A}(M)$ holds whenever all $N < M$ satisfy $\mathcal{A}(N)$.
  We must show that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.

- What we know so far:
  (1) $\mathcal{G}(b)$ holds (that is, $\mathcal{P}_{\text{fin}}(S_{\leq b})$ is Noetherian) for all $b < a$.
  (2) $\mathcal{A}(M)$ is a statement for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
  (3) $\mathcal{A}(M)$ holds whenever all $N < M$ satisfy $\mathcal{A}(N)$.
  We must show that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
- First, we claim that
  (4) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq b})$ for each $b < a$.
  Indeed, this is proven by induction on M, which is allowed by (1), and which uses (3) for the induction step.

Finite subsets form a Noetherian order: proof, 2

- What we know so far:
  (1) $\mathcal{G}(b)$ holds (that is, $\mathcal{P}_{\text{fin}}(S_{\leq b})$ is Noetherian) for all $b < a$.
  (2) $\mathcal{A}(M)$ is a statement for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
  (3) $\mathcal{A}(M)$ holds whenever all $N < M$ satisfy $\mathcal{A}(N)$.
  We must show that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
- First, we claim that
  (4) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq b})$ for each $b < a$.
  Indeed, this is proven by induction on M, which is allowed by (1), and which uses (3) for the induction step.
- Rewrite (4) (and the obvious fact that $\mathcal{A}(\varnothing)$ holds, which again follows from (3)) as
  (5) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \notin M$.

- What we know so far:
  (1) $\mathcal{G}(b)$ holds (that is, $\mathcal{P}_{\text{fin}}(S_{\leq b})$ is Noetherian) for all $b < a$.
  (2) $\mathcal{A}(M)$ is a statement for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
  (3) $\mathcal{A}(M)$ holds whenever all $N < M$ satisfy $\mathcal{A}(N)$.
  We must show that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
- First, we claim that
  (4) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq b})$ for each $b < a$.
  Indeed, this is proven by induction on M, which is allowed by (1), and which uses (3) for the induction step.
- Rewrite (4) (and the obvious fact that $\mathcal{A}(\varnothing)$ holds, which again follows from (3)) as
  (5) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \notin M$.
- Thus, (3) yields that $\mathcal{A}(\{a\})$ holds.

- What we know so far:
  (1) $\mathcal{G}(b)$ holds (that is, $\mathcal{P}_{\text{fin}}(S_{\leq b})$ is Noetherian) for all $b < a$.
  (2) $\mathcal{A}(M)$ is a statement for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
  (3) $\mathcal{A}(M)$ holds whenever all $N < M$ satisfy $\mathcal{A}(N)$.
  (5) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \notin M$.

  We must show that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.

- What we know so far:
  (1) $\mathcal{G}(b)$ holds (that is, $\mathcal{P}_{\text{fin}}(S_{\leq b})$ is Noetherian) for all $b < a$.
  (2) $\mathcal{A}(M)$ is a statement for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
  (3) $\mathcal{A}(M)$ holds whenever all $N < M$ satisfy $\mathcal{A}(N)$.
  (5) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \notin M$.
  We must show that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
- Next, we claim that
  (6) $\mathcal{A}(M \cup \{a\})$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq b})$ for each $b < a$.
  Indeed, this is proven by induction on M, which is allowed by (1), and which uses (3) and (5) for the induction step (since each set in $\mathcal{P}_{\text{fin}}(S_{\leq a})$ that is $< M \cup \{a\}$ is either of the form $N \cup \{a\}$ with $N < M$, or does not contain a).

Finite subsets form a Noetherian order: proof, 3

- What we know so far:

  (1) $\mathcal{G}(b)$ holds (that is, $\mathcal{P}_{\text{fin}}(S_{\leq b})$ is Noetherian) for all $b < a$.

  (2) $\mathcal{A}(M)$ is a statement for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.

  (3) $\mathcal{A}(M)$ holds whenever all $N < M$ satisfy $\mathcal{A}(N)$.

  (5) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \notin M$.

  We must show that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.

- Next, we claim that

  (6) $\mathcal{A}(M \cup \{a\})$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq b})$ for each $b < a$.

  Indeed, this is proven by induction on M, which is allowed by (1), and which uses (3) and (5) for the induction step (since each set in $\mathcal{P}_{\text{fin}}(S_{\leq a})$ that is $< M \cup \{a\}$ is either of the form $N \cup \{a\}$ with $N < M$, or does not contain a).

- Rewrite (6) (and the fact that $\mathcal{A}(\{a\})$ holds) as

  (7) $\mathcal{A}(M \cup \{a\})$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{<a})$ satisfying $a \notin M$.

Finite subsets form a Noetherian order: proof, 4

- What we know so far:
  (2) $\mathcal{A}(M)$ is a statement for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
  (5) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \notin M$.
  (7) $\mathcal{A}(M \cup \{a\})$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \notin M$.

  We must show that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.

Finite subsets form a Noetherian order: proof, 4

- What we know so far:
  (2) $\mathcal{A}(M)$ is a statement for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
  (5) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \notin M$.
  (7) $\mathcal{A}(M \cup \{a\})$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \notin M$.

  We must show that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
- Rewrite (7) as
  (8) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \in M$.

- What we know so far:
  (2) $\mathcal{A}(M)$ is a statement for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
  (5) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \notin M$.
  (7) $\mathcal{A}(M \cup \{a\})$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \notin M$.

  We must show that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
- Rewrite (7) as
  (8) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \in M$.
- Combine (5) with (8) to conclude that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.

  Thus, we proved that $\mathcal{P}_{\text{fin}}(S_{\leq a})$ is Noetherian.

Finite subsets form a Noetherian order: proof, 4

- What we know so far:
  (2) $\mathcal{A}(M)$ is a statement for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
  (5) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \notin M$.
  (7) $\mathcal{A}(M \cup \{a\})$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \notin M$.

  We must show that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
- Rewrite (7) as
  (8) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \in M$.
- Combine (5) with (8) to conclude that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.

  Thus, we proved that $\mathcal{P}_{\text{fin}}(S_{\leq a})$ is Noetherian.
- To prove that $\mathcal{P}_{\text{fin}}(S)$ is Noetherian, it suffices to notice that each $M \in \mathcal{P}_{\text{fin}}(S)$ belongs to $\mathcal{P}_{\text{fin}}(S_{\leq a})$ for some $a \in S$ (or is empty).

Finite subsets form a Noetherian order: proof, 4

- What we know so far:

  (2) $\mathcal{A}(M)$ is a statement for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.

  (5) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \notin M$.

  (7) $\mathcal{A}(M \cup \{a\})$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \notin M$.

  We must show that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.
- Rewrite (7) as

  (8) $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$ satisfying $a \in M$.
- Combine (5) with (8) to conclude that $\mathcal{A}(M)$ holds for each $M \in \mathcal{P}_{\text{fin}}(S_{\leq a})$.

  Thus, we proved that $\mathcal{P}_{\text{fin}}(S_{\leq a})$ is Noetherian.
- To prove that $\mathcal{P}_{\text{fin}}(S)$ is Noetherian, it suffices to notice that each $M \in \mathcal{P}_{\text{fin}}(S)$ belongs to $\mathcal{P}_{\text{fin}}(S_{\leq a})$ for some $a \in S$ (or is empty).
- (I've got the idea from Mines/Richman/Ruitenburg, *A Course in Constructive Algebra*, proof of Theorem 6.4. They work with different notations and prove a more general result.)

More posets

- In classical logic, there are several nontrivial Noetherian posets:
  - weakly decreasing tuples of arbitrary size with lexicographic order;
  - infinite sequences with lexicographic order;
  - trees (infamous hydra theorem);
  - graphs w.r.t. minor relation,
  - etc.

  (Correct me if/where I'm wrong.)
  I don't know which of these are still Noetherian in constructive logic.

6.

Application: Gröbner bases

References:

- Bremner/Dotsenko, *Algebraic Operads: An Algorithmic Companion*.
- Becker/Weispfennig, *Gröbner Bases: A computational approach to commutative algebra*.
- Cox/Little/O'Shea, *Ideals, Varieties, and Algorithms*.

- Fix a commutative ring $\mathbb{K}$, and a monic polynomial

$$d = x^m - d_1 x^{m-1} - d_2 x^{m-2} - \cdots - d_m x^0 \in \mathbb{K}[x].$$

- The polynomial division game:
  - **Positions:** polynomials $f \in \mathbb{K}[x]$.
  - **Moves:** Pick any $n \geq m$ such that x^n appears in f; call c its coefficient; subtract $cx^{n-m}d$ from f (that is, subtract the multiple of d that kills the x^n-term in f and leaves the higher terms unchanged).

("Appears in f" means "appears with nonzero coefficient in f".)

Introduction: polynomial division, the game, 1

- Fix a commutative ring $\mathbb{K}$, and a monic polynomial

$$d = x^m - d_1 x^{m-1} - d_2 x^{m-2} - \cdots - d_m x^0 \in \mathbb{K}[x].$$

- The polynomial division game:
  - **Positions:** polynomials $f \in \mathbb{K}[x]$.
  - **Moves:** Pick any $n \geq m$ such that x^n appears in f; call c its coefficient; subtract $cx^{n-m}d$ from f (that is, subtract the multiple of d that kills the x^n-term in f and leaves the higher terms unchanged).

  ("Appears in f" means "appears with nonzero coefficient in f".)
- **Proposition.** The game always terminates, and the outcome does not depend on the choice of moves.

- The polynomial division game:
  - **Positions:** polynomials $f \in \mathbb{K}[x]$.
  - **Moves:** Pick any $n \geq m$ such that x^n appears in f; call c its coefficient in f; subtract $cx^{n-m}d$ from f.
- **Proposition.** The game always terminates, and the outcome does not depend on the choice of moves.

- The polynomial division game:
  - **Positions:** polynomials $f \in \mathbb{K}[x]$.
  - **Moves:** Pick any $n \geq m$ such that x^n appears in f; call c its coefficient in f; subtract $cx^{n-m}d$ from f.
- **Proposition.** The game always terminates, and the outcome does not depend on the choice of moves.
- *Proof:* By the (general) diamond lemma, it suffices to prove Noetherianness and local confluence.

- The polynomial division game:
  - **Positions:** polynomials $f \in \mathbb{K}[x]$.
  - **Moves:** Pick any $n \geq m$ such that x^n appears in f; call c its coefficient in f; subtract $cx^{n-m}d$ from f.
- **Proposition.** The game always terminates, and the outcome does not depend on the choice of moves.
- *Proof:* By the (general) diamond lemma, it suffices to prove Noetherianness and local confluence.

  Noetherianness: Let

$$h : \mathbb{K}[x] \to \mathcal{P}_{\text{fin}}(\mathbb{N}),$$
$$f \mapsto \{n \in \mathbb{N} \mid x^n \text{ appears in } f\}.$$

  Easy to see:

$$h(u) > h(v) \text{ whenever } u \longrightarrow v.$$

- The polynomial division game:
  - **Positions:** polynomials $f \in \mathbb{K}[x]$.
  - **Moves:** Pick any $n \geq m$ such that x^n appears in f; call c its coefficient in f; subtract $cx^{n-m}d$ from f.
- **Proposition.** The game always terminates, and the outcome does not depend on the choice of moves.
- *Proof:* By the (general) diamond lemma, it suffices to prove Noetherianness and local confluence.

  Local confluence: Exercise.

- The polynomial division game:
  - **Positions:** polynomials $f \in \mathbb{K}[x]$.
  - **Moves:** Pick any $n \geq m$ such that x^n appears in f; call c its coefficient in f; subtract $cx^{n-m}d$ from f.
- **Proposition.** The game always terminates, and the outcome does not depend on the choice of moves.
- This shouldn't come as a surprise: The "game" is just polynomial division by d, but done in an unsystematic (and slow) fashion.

Introduction: polynomial division, modified

- Let us modify the game somewhat to make it more predictable.

- Let us modify the game somewhat to make it more predictable.
- The polynomial division game:
  - **Positions:** polynomials $f \in \mathbb{K}[x]$.
  - **Moves:** Pick any $n \geq m$ such that x^n appears in f; call c its coefficient in f; subtract $cx^{n-m}d$ from f.

- Let us modify the game somewhat to make it more predictable.
- The polynomial division game:
  - **Positions:** polynomials $f \in \mathbb{K}[x]$.
  - **Moves:** Pick any $n \geq m$ such that x^n appears in f; call c its coefficient in f; subtract $cx^{n-m}d$ from f.
- The move requires the coefficient of x^n to be nonzero. This is fickle and not very constructive. Better: Keep track of powers of x that have already been killed in previous moves (but not by random cancellation), and only require x^n to be not one of them.

- Let us modify the game somewhat to make it more predictable.
- The modified polynomial division game:
  - **Positions:** pairs (M, f) consisting of an $M \in \mathcal{P}_{\text{fin}}(\mathbb{N})$ and a polynomial $f \in \mathbb{K}[x]$.
  - **Moves:** Pick any $n \geq m$ such that $n \in M$;
    call c its coefficient in f;
    replace n by $n-1, n-2, \ldots, n-m$ in M;
    subtract $cx^{n-m}d$ from f.

- Let us modify the game somewhat to make it more predictable.
- The modified polynomial division game:
  - **Positions:** pairs (M, f) consisting of an $M \in \mathcal{P}_{\text{fin}}(\mathbb{N})$ and a polynomial $f \in \mathbb{K}[x]$.
  - **Moves:** Pick any $n \geq m$ such that $n \in M$;
    call c its coefficient in f;
    replace n by $n-1, n-2, \ldots, n-m$ in M;
    subtract $cx^{n-m}d$ from f.
- All changes are in blue.
  The set M keeps track of all powers of x that can possibly still appear in f, but random cancellations do not get removed from M.

- Let us modify the game somewhat to make it more predictable.
- The modified polynomial division game:
  - **Positions:** pairs (M, f) consisting of an $M \in \mathcal{P}_{\text{fin}}(\mathbb{N})$ and a polynomial $f \in \mathbb{K}[x]$.
  - **Moves:** Pick any $n \geq m$ such that $n \in M$;
    call c its coefficient in f;
    replace n by $n-1, n-2, \ldots, n-m$ in M;
    subtract $cx^{n-m}d$ from f.
- If we forget about M, then each move of the modified game either corresponds to a move or the original game, or leaves f unchanged.

- Let us modify the game somewhat to make it more predictable.
- The modified polynomial division game:
  - **Positions:** pairs (M, f) consisting of an $M \in \mathcal{P}_{\text{fin}}(\mathbb{N})$ and a polynomial $f \in \mathbb{K}[x]$.
  - **Moves:** Pick any $n \geq m$ such that $n \in M$;
    call c its coefficient in f;
    replace n by $n-1, n-2, \ldots, n-m$ in M;
    subtract $cx^{n-m}d$ from f.
- If we forget about M, then each move of the modified game either corresponds to a move or the original game, or leaves f unchanged.
- **Proposition.** The modified game always terminates, and the outcome does not depend on the choice of moves.

- The modified polynomial division game:
  - **Positions:** pairs (M, f) consisting of an $M \in \mathcal{P}_{\text{fin}}(\mathbb{N})$ and a polynomial $f \in \mathbb{K}[x]$.
  - **Moves:** Pick any $n \geq m$ such that $n \in M$;
    call c its coefficient in f;
    replace n by $n-1, n-2, \ldots, n-m$ in M;
    subtract $cx^{n-m}d$ from f.
- **Proposition.** The modified game always terminates, and the outcome does not depend on the choice of moves.
- *Proof.* As for the previous game, but easier.
  Noetherianness: Let

$$h : \{\text{positions}\} \to \mathcal{P}_{\text{fin}}(\mathbb{N}),$$
$$(M, f) \mapsto M.$$

  Local confluence: Even easier than before.

Multiple variables

- Let us generalize:

Multiple variables

- Consider a polynomial ring $\mathbb{K}[x_1, x_2, \ldots, x_n]$ in n variables.
- *Monomials* are formal expressions $x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$ with $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$.

- Consider a polynomial ring $\mathbb{K}[x_1, x_2, \ldots, x_n]$ in n variables.
- *Monomials* are formal expressions $x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$ with $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$.
- Monomials can be multiplied in the obvious way.
- We say that a monomial $\mathfrak{m} = x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$ *divides* a monomial $\mathfrak{n} = x_1^{b_1} x_2^{b_2} \cdots x_n^{b_n}$ if $a_i \leq b_i$ for all i.
  In this case, $\mathfrak{n}/\mathfrak{m} := x_1^{b_1 - a_1} x_2^{b_2 - a_2} \cdots x_n^{b_n - a_n}$.

- Consider a polynomial ring $\mathbb{K}[x_1, x_2, \ldots, x_n]$ in n variables.
- *Monomials* are formal expressions $x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$ with $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$.
- Monomials can be multiplied in the obvious way.
- We say that a monomial $\mathfrak{m} = x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$ *divides* a monomial $\mathfrak{n} = x_1^{b_1} x_2^{b_2} \cdots x_n^{b_n}$ if $a_i \leq b_i$ for all i.
  In this case, $\mathfrak{n}/\mathfrak{m} := x_1^{b_1 - a_1} x_2^{b_2 - a_2} \cdots x_n^{b_n - a_n}$.
- The *lcm* (lowest common multiple) of two monomials $\mathfrak{m} = x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$ and $\mathfrak{n} = x_1^{b_1} x_2^{b_2} \cdots x_n^{b_n}$ is defined to be the monomial $\operatorname{lcm}(\mathfrak{m}, \mathfrak{n}) := x_1^{\max\{a_1, b_1\}} x_2^{\max\{a_2, b_2\}} \cdots x_n^{\max\{a_n, b_n\}}$.

Multiple variables

- Consider a polynomial ring $\mathbb{K}[x_1, x_2, \ldots, x_n]$ in n variables.
- *Monomials* are formal expressions $x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$ with $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$.
- We equip $\mathbb{N}^n$ with the lexicographic order (i.e., the total order obtained as the lexicographic product $\mathbb{N} \times \mathbb{N} \times \cdots \times \mathbb{N}$).
- We transfer this order to monomials. Thus,

$$\begin{aligned} & \left(x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n} > x_1^{b_1} x_2^{b_2} \cdots x_n^{b_n} \right) \\ \iff & \left((a_1, a_2, \ldots, a_n) > (b_1, b_2, \ldots, b_n) \right) \\ \iff & (\text{there is some } i \text{ such that } a_i > b_i, \text{ and} \\ & \qquad \text{each } j < i \text{ satisfies } a_j = b_j) . \end{aligned}$$

- Consider a polynomial ring $\mathbb{K}[x_1, x_2, \ldots, x_n]$ in n variables.
- *Monomials* are formal expressions $x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$ with $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$.
- We equip $\mathbb{N}^n$ with the lexicographic order (i.e., the total order obtained as the lexicographic product $\mathbb{N} \times \mathbb{N} \times \cdots \times \mathbb{N}$).
- We transfer this order to monomials. Thus,

$$\left(x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n} > x_1^{b_1} x_2^{b_2} \cdots x_n^{b_n}\right)$$
$$\Longleftrightarrow ((a_1, a_2, \ldots, a_n) > (b_1, b_2, \ldots, b_n))$$
$$\Longleftrightarrow (\text{there is some } i \text{ such that } a_i > b_i, \text{ and each } j < i \text{ satisfies } a_j = b_j).$$

- This is a total order.
  Thus, every nonzero polynomial p has a unique *leading monomial* (i.e., maximum monomial appearing with nonzero coefficient).
  We say that p is *monic* if the coefficient of its leading monomial is 1.

Multivariate polynomial division, the game

- Fix a commutative ring $\mathbb{K}$, and finitely many monic polynomials $g_1, g_2, \ldots, g_k$ in $\mathbb{K}[x_1, x_2, \ldots, x_n]$.

Multivariate polynomial division, the game

- Fix a commutative ring $\mathbb{K}$, and finitely many monic polynomials $g_1, g_2, \ldots, g_k$ in $\mathbb{K}[x_1, x_2, \ldots, x_n]$. For each i, let $\mathfrak{h}_i$ be the leading monomial of g_i.

Multivariate polynomial division, the game

- Fix a commutative ring $\mathbb{K}$, and finitely many monic polynomials $g_1, g_2, \ldots, g_k$ in $\mathbb{K}[x_1, x_2, \ldots, x_n]$.
  For each i, let $\mathfrak{h}_i$ be the leading monomial of g_i.
- The multivariate polynomial division game:
  - **Positions:** polynomials $f \in \mathbb{K}[x_1, x_2, \ldots, x_n]$.
  - **Moves:** Pick any monomial $\mathfrak{m}$ that appears in f and any i such that $\mathfrak{h}_i \mid \mathfrak{m}$. Call c the coefficient of $\mathfrak{m}$ in f.
    Subtract $c\,(\mathfrak{m}/\mathfrak{h}_i)\,g_i$ from f (that is, subtract the multiple of g_i that kills the $\mathfrak{m}$-term in f and leaves higher terms unchanged).

Multivariate polynomial division, the game

- Fix a commutative ring $\mathbb{K}$, and finitely many monic polynomials $g_1, g_2, \ldots, g_k$ in $\mathbb{K}[x_1, x_2, \ldots, x_n]$.
  For each i, let $\mathfrak{h}_i$ be the leading monomial of g_i.
- The multivariate polynomial division game:
  - **Positions:** polynomials $f \in \mathbb{K}[x_1, x_2, \ldots, x_n]$.
  - **Moves:** Pick any monomial $\mathfrak{m}$ that appears in f and any i such that $\mathfrak{h}_i \mid \mathfrak{m}$. Call c the coefficient of $\mathfrak{m}$ in f.
    Subtract $c\,(\mathfrak{m}/\mathfrak{h}_i)\,g_i$ from f (that is, subtract the multiple of g_i that kills the $\mathfrak{m}$-term in f and leaves higher terms unchanged).
- The game always terminates.

Multivariate polynomial division, the game

- Fix a commutative ring $\mathbb{K}$, and finitely many monic polynomials $g_1, g_2, \ldots, g_k$ in $\mathbb{K}[x_1, x_2, \ldots, x_n]$.
  For each i, let $\mathfrak{h}_i$ be the leading monomial of g_i.
- The multivariate polynomial division game:
  - **Positions:** polynomials $f \in \mathbb{K}[x_1, x_2, \ldots, x_n]$.
  - **Moves:** Pick any monomial $\mathfrak{m}$ that appears in f and any i such that $\mathfrak{h}_i \mid \mathfrak{m}$. Call c the coefficient of $\mathfrak{m}$ in f.
    Subtract $c(\mathfrak{m}/\mathfrak{h}_i) g_i$ from f (that is, subtract the multiple of g_i that kills the $\mathfrak{m}$-term in f and leaves higher terms unchanged).
- The game always terminates.
- When is it confluent?

Multivariate polynomial division, the game: Example 1

- **Example 1:**
  - $n = 2$. Write x and y for x_1 and x_2.
  - $k = 2$. Let $g_1 = x^2y - x$ and $g_2 = y^2x - y$.

- **Example 1:**
  - $n = 2$. Write x and y for x_1 and x_2.
  - $k = 2$. Let $g_1 = x^2y - x$ and $g_2 = y^2x - y$.
- Start the game with $f = x^2y^2$:

$$\underline{x^2y^2} \xrightarrow{g_1} xy \text{ (terminal)}$$

(where $\xrightarrow{g_i}$ means that the move uses g_i) versus

$$\underline{x^2y^2} \xrightarrow{g_2} xy \text{ (terminal)}.$$

Looks good so far.

- **Example 1:**
  - $n = 2$. Write x and y for x_1 and x_2.
  - $k = 2$. Let $g_1 = x^2y - x$ and $g_2 = y^2x - y$.
- Start the game with $f = x^3y^3$:

$$\underline{x^3y^3} \xrightarrow{g_1} \underline{x^2y^2} \xrightarrow{g_1} xy \text{ (terminal)}$$

versus

$$\underline{x^3y^3} \xrightarrow{g_2} \underline{x^2y^2} \xrightarrow{g_1} xy \text{ (terminal).}$$

Looks good so far.

Multivariate polynomial division, the game: Example 1

- **Example 1:**
  - $n = 2$. Write x and y for x_1 and x_2.
  - $k = 2$. Let $g_1 = x^2y - x$ and $g_2 = y^2x - y$.
- Not hard to see: This one is confluent.

- **Example 2:**
  - $n = 2$. Write x and y for x_1 and x_2.
  - $k = 2$. Let $g_1 = x^2y - y$ and $g_2 = y^2x - x$.

- **Example 2:**
  - $n = 2$. Write x and y for x_1 and x_2.
  - $k = 2$. Let $g_1 = x^2y - y$ and $g_2 = y^2x - x$.
- Start the game with $f = x^2y^2$:

$$\underline{x^2y^2} \xrightarrow{g_1} y^2 \text{ (terminal)}$$

versus

$$\underline{x^2y^2} \xrightarrow{g_2} x^2 \text{ (terminal)}.$$

- **Example 2:**
  - $n = 2$. Write x and y for x_1 and x_2.
  - $k = 2$. Let $g_1 = x^2y - y$ and $g_2 = y^2x - x$.
- Start the game with $f = x^2y^2$:

$$\underline{x^2y^2} \xrightarrow{g_1} y^2 \text{ (terminal)}$$

versus

$$\underline{x^2y^2} \xrightarrow{g_2} x^2 \text{ (terminal)}.$$

Not confluent!

Multivariate polynomial division, the game: Example 3

- **Example 3:**
  - $n = 2$. Write x and y for x_1 and x_2.
  - $k = 2$. Let $g_1 = x^2 - x - y$ and $g_2 = y^2 - x - y$.

- **Example 3:**
  - $n = 2$. Write x and y for x_1 and x_2.
  - $k = 2$. Let $g_1 = x^2 - x - y$ and $g_2 = y^2 - x - y$.
- Start the game with $f = x^2y^2$:

$$\underline{x^2y^2} \xrightarrow{g_1} \underline{xy^2} + y^3 \xrightarrow{g_2} x^2 + xy + \underline{y^3} \xrightarrow{g_2} \underline{x^2} + 2xy + y^2$$
$$\xrightarrow{g_1} x + y + 2xy + \underline{y^2} \xrightarrow{g_2} 2x + 2y + 2xy \text{ (terminal)}$$

versus

$$\underline{x^2y^2} \xrightarrow{g_2} \underline{x^3} + x^2y \xrightarrow{g_1} x^2 + xy + \underline{x^2y} \xrightarrow{g_1} x^2 + 2xy + \underline{y^2}$$
$$\xrightarrow{g_2} x + y + \underline{x^2} + 2xy \xrightarrow{g_1} 2x + 2y + 2xy. \text{ (terminal)}$$

- **Example 3:**
  - $n = 2$. Write x and y for x_1 and x_2.
  - $k = 2$. Let $g_1 = x^2 - x - y$ and $g_2 = y^2 - x - y$.
- Start the game with $f = x^2y^2$:

$$\underline{x^2y^2} \xrightarrow{g_1} \underline{xy^2} + y^3 \xrightarrow{g_2} x^2 + xy + \underline{y^3} \xrightarrow{g_2} \underline{x^2} + 2xy + y^2$$
$$\xrightarrow{g_1} x + y + 2xy + \underline{y^2} \xrightarrow{g_2} 2x + 2y + 2xy \text{ (terminal)}$$

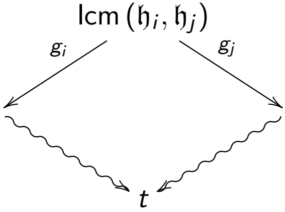
versus

$$\underline{x^2y^2} \xrightarrow{g_2} \underline{x^3} + x^2y \xrightarrow{g_1} x^2 + xy + \underline{x^2y} \xrightarrow{g_1} x^2 + 2xy + \underline{y^2}$$
$$\xrightarrow{g_2} x + y + \underline{x^2} + 2xy \xrightarrow{g_1} 2x + 2y + 2xy. \text{ (terminal)}$$

Looks confluent so far. But how to prove it in general?

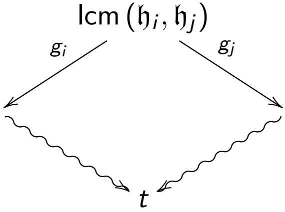
- Fix a commutative ring $\mathbb{K}$, and finitely many monic polynomials $g_1, g_2, \ldots, g_k$ in $\mathbb{K}[x_1, x_2, \ldots, x_n]$.
  For each i, let $\mathfrak{h}_i$ be the leading monomial of g_i.
- The multivariate polynomial division game:
  - **Positions:** polynomials $f \in \mathbb{K}[x_1, x_2, \ldots, x_n]$.
  - **Moves:** Pick any monomial $\mathfrak{m}$ that appears in f and any i such that $\mathfrak{h}_i \mid \mathfrak{m}$. Call c the coefficient of $\mathfrak{m}$ in f.
    Subtract $c(\mathfrak{m}/\mathfrak{h}_i) g_i$ from f (that is, subtract the multiple of g_i that kills the $\mathfrak{m}$-term in f and leaves higher terms unchanged).

- Fix a commutative ring $\mathbb{K}$, and finitely many monic polynomials $g_1, g_2, \ldots, g_k$ in $\mathbb{K}[x_1, x_2, \ldots, x_n]$.
  For each i, let $\mathfrak{h}_i$ be the leading monomial of g_i.
- **Theorem (Buchberger).** The game is confluent if and only if for each i and j, there is a position t such that



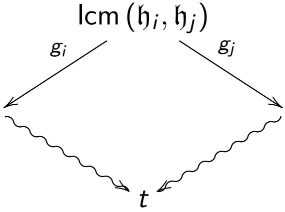
(where $\xrightarrow{g_i}$ means "move using g_i", while $\xrightarrow{g_j}$ means "move using g_j").

- Fix a commutative ring $\mathbb{K}$, and finitely many monic polynomials $g_1, g_2, \ldots, g_k$ in $\mathbb{K}[x_1, x_2, \ldots, x_n]$.
  For each i, let $\mathfrak{h}_i$ be the leading monomial of g_i.
- **Theorem (Buchberger).** The game is confluent if and only if for each i and j, there is a position t such that



  (where $\xrightarrow{g_i}$ means "move using g_i", while $\xrightarrow{g_j}$ means "move using g_j").
- It suffices to consider the case $i < j$.

- Fix a commutative ring $\mathbb{K}$, and finitely many monic polynomials $g_1, g_2, \ldots, g_k$ in $\mathbb{K}[x_1, x_2, \ldots, x_n]$.
  For each i, let $\mathfrak{h}_i$ be the leading monomial of g_i.
- **Theorem (Buchberger).** The game is confluent if and only if for each i and j, there is a position t such that

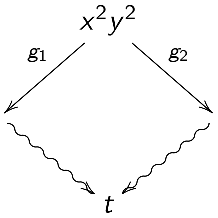


  (where $\xrightarrow{g_i}$ means "move using g_i", while $\xrightarrow{g_j}$ means "move using g_j").
- Again, the game can be modified so it no longer depends on the nonvanishing of coefficients.
  The modified game has the same properties.

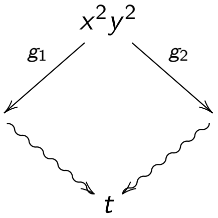
- **Example 3:**
  - $n = 2$. Write x and y for x_1 and x_2.
  - $k = 2$. Let $g_1 = x^2 - x - y$ and $g_2 = y^2 - x - y$.
- Let us prove that this game is confluent.

- **Example 3:**
  - $n = 2$. Write x and y for x_1 and x_2.
  - $k = 2$. Let $g_1 = x^2 - x - y$ and $g_2 = y^2 - x - y$.
- Let us prove that this game is confluent.
  $\operatorname{lcm}(\mathfrak{h}_1, \mathfrak{h}_2) = x^2 y^2$.

- **Example 3:**
  - $n = 2$. Write x and y for x_1 and x_2.
  - $k = 2$. Let $g_1 = x^2 - x - y$ and $g_2 = y^2 - x - y$.
- Let us prove that this game is confluent.
  $\operatorname{lcm}(\mathfrak{h}_1, \mathfrak{h}_2) = x^2 y^2$.
  We only need to consider the case $i < j$; thus $i = 1$ and $j = 2$.
  Thus we need to find t such that

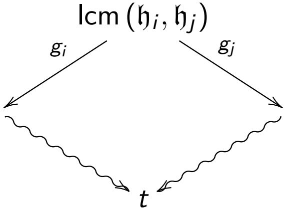


- **Example 3:**
  - $n = 2$. Write x and y for x_1 and x_2.
  - $k = 2$. Let $g_1 = x^2 - x - y$ and $g_2 = y^2 - x - y$.
- Let us prove that this game is confluent.
  $\operatorname{lcm}(\mathfrak{h}_1, \mathfrak{h}_2) = x^2y^2$.
  We only need to consider the case $i < j$; thus $i = 1$ and $j = 2$.
  Thus we need to find t such that



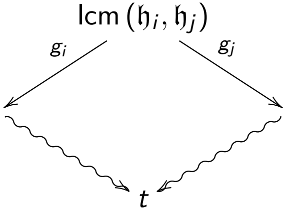
But we did that a few slides ago! ($t = 2x + 2y + 2xy$.)
So the game is confluent.

- Actually, this holds more generally:
  Theorem (Buchberger's 1st criterion). If the monomials $\mathfrak{h}_i$ and $\mathfrak{h}_j$ have no indeterminates in common (i.e., no variable appears in both; equivalently, $\operatorname{lcm}(\mathfrak{h}_i, \mathfrak{h}_j) = \mathfrak{h}_i \mathfrak{h}_j$), then there is a position t such that



Multivariate polynomial division, the game: Buchberger 1

- Actually, this holds more generally:
  Theorem (Buchberger's 1st criterion). If the monomials $\mathfrak{h}_i$ and $\mathfrak{h}_j$ have no indeterminates in common (i.e., no variable appears in both; equivalently, $\operatorname{lcm}(\mathfrak{h}_i, \mathfrak{h}_j) = \mathfrak{h}_i \mathfrak{h}_j$), then there is a position t such that



- Thus, for example, the game is always confluent if $\mathfrak{h}_i = x_i^{\text{something}}$ for each i.

Multivariate polynomial division, the game: Gröbner bases

- Fix a commutative ring $\mathbb{K}$, and finitely many monic polynomials $g_1, g_2, \ldots, g_k$ in $\mathbb{K}[x_1, x_2, \ldots, x_n]$.
  For each i, let $\mathfrak{h}_i$ be the leading monomial of g_i.

Multivariate polynomial division, the game: Gröbner bases

- Fix a commutative ring $\mathbb{K}$, and finitely many monic polynomials $g_1, g_2, \ldots, g_k$ in $\mathbb{K}[x_1, x_2, \ldots, x_n]$.
  For each i, let $\mathfrak{h}_i$ be the leading monomial of g_i.
- When the game is confluent, the polynomials $g_1, g_2, \ldots, g_k$ are said to form a *Gröbner basis*.
  The terminal position obtained in the game is then called the *remainder of f upon division by $g_1, g_2, \ldots, g_k$*.

- Fix a commutative ring $\mathbb{K}$, and finitely many monic polynomials $g_1, g_2, \ldots, g_k$ in $\mathbb{K}[x_1, x_2, \ldots, x_n]$.
  For each i, let $\mathfrak{h}_i$ be the leading monomial of g_i.
- When the game is confluent, the polynomials $g_1, g_2, \ldots, g_k$ are said to form a *Gröbner basis*.
  The terminal position obtained in the game is then called the *remainder of f upon division by $g_1, g_2, \ldots, g_k$*.
- Gröbner bases are often defined with respect to other orders (not just the lexicographic one).
  The theory is then similar.

Multivariate polynomial division, the game: Gröbner bases

- Fix a commutative ring $\mathbb{K}$, and finitely many monic polynomials $g_1, g_2, \ldots, g_k$ in $\mathbb{K}[x_1, x_2, \ldots, x_n]$.
  For each i, let $\mathfrak{h}_i$ be the leading monomial of g_i.
- When the game is confluent, the polynomials $g_1, g_2, \ldots, g_k$ are said to form a *Gröbner basis*.
  The terminal position obtained in the game is then called the *remainder of f upon division by $g_1, g_2, \ldots, g_k$*.
- Gröbner bases are often defined with respect to other orders (not just the lexicographic one).
  The theory is then similar.
- Used throughout computational algebraic geometry and beyond.

- Fix a commutative ring $\mathbb{K}$, and finitely many monic polynomials $g_1, g_2, \ldots, g_k$ in $\mathbb{K}[x_1, x_2, \ldots, x_n]$.
  For each i, let $\mathfrak{h}_i$ be the leading monomial of g_i.
- When the game is confluent, the polynomials $g_1, g_2, \ldots, g_k$ are said to form a *Gröbner basis*.
  The terminal position obtained in the game is then called the *remainder of f upon division by $g_1, g_2, \ldots, g_k$*.
- Gröbner bases are often defined with respect to other orders (not just the lexicographic one).
  The theory is then similar.
- Used throughout computational algebraic geometry and beyond.
- There is a noncommutative version, where monomials are replaced by *words* (and the indeterminates don't commute).