מתמטיקה בדידה
Discrete Math

Lecture 11

Last Week: Strict Partial Order

Definition: A binary relation, R, on a set A is said to be a strict partial order if it is

Asymmetric: $\forall a,b \in A$, $aRb \rightarrow \neg(bRa)$.

Transitive: $\forall a,b,c \in A$, $(aRb \wedge bRc) \rightarrow aRc$.

Examples:

- The < relation on numbers: aRb iff $a < b$.
- The $\subset$ relation on subsets: ARB iff $A \subset B$.

Strict partial order	יחס סדר חלקי ממש
Partially ordered set	קבוצה סדורה חלקית (קס"ח)

Weak Partial Order

Definition: A binary relation, R, on a set A is said to be a weak partial order if it is

Reflexive: $\forall a \in A$, aRa.

Anti-symmetric*: $\forall a,b \in A$, $(aRb \wedge a \neq b) \rightarrow \neg(bRa)$.

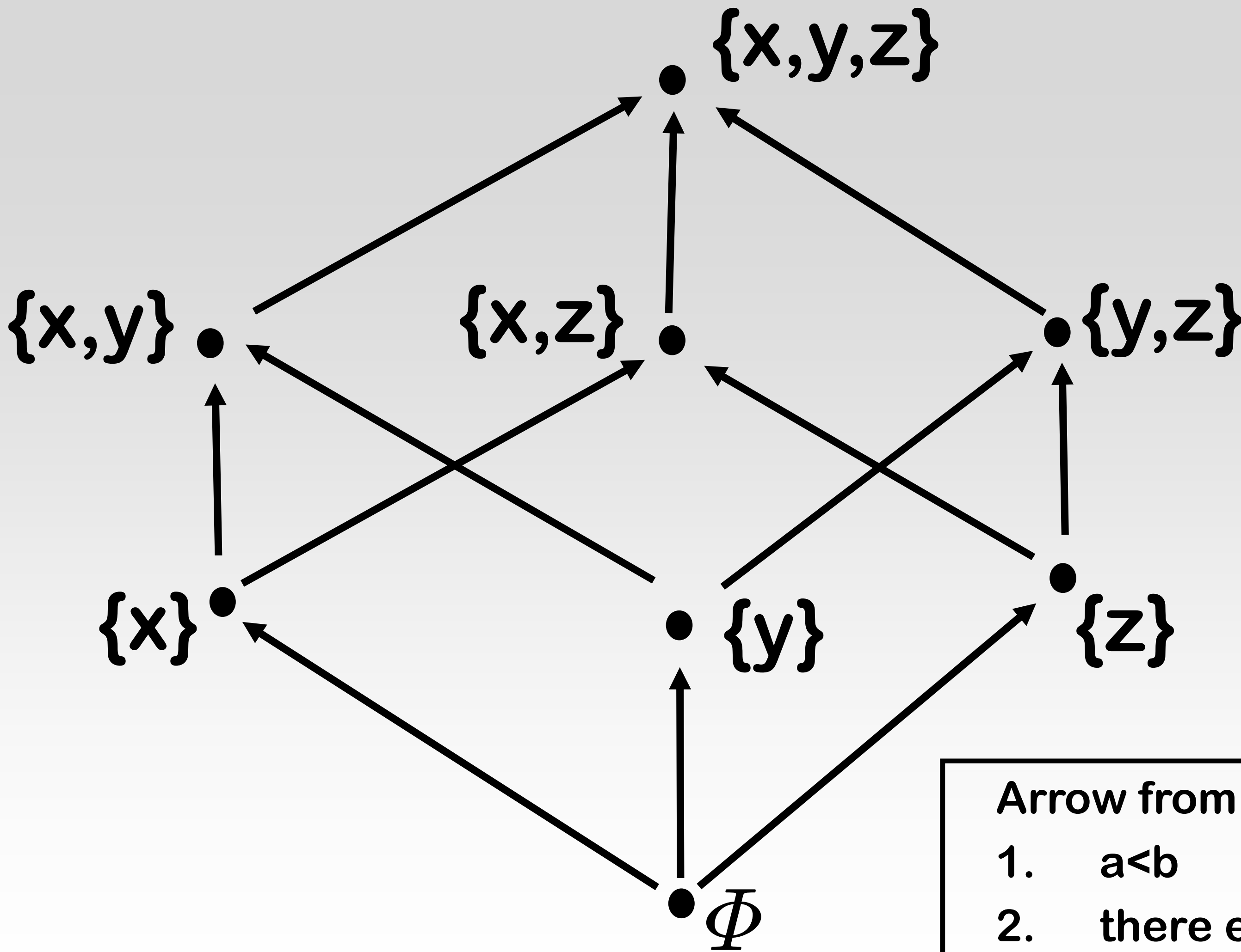
Transitive: $\forall a,b,c \in A$, $(aRb \wedge bRc) \rightarrow aRc$.

Examples:

- The $\leq$ relation on numbers: aRb iff $a \leq b$.
- The $\subseteq$ relation on subsets: ARB iff $A \subseteq B$.
- The "divides" relation. mRn iff $\exists k$ so that $n = km$.

Weak partial order יחס סדר חלקי

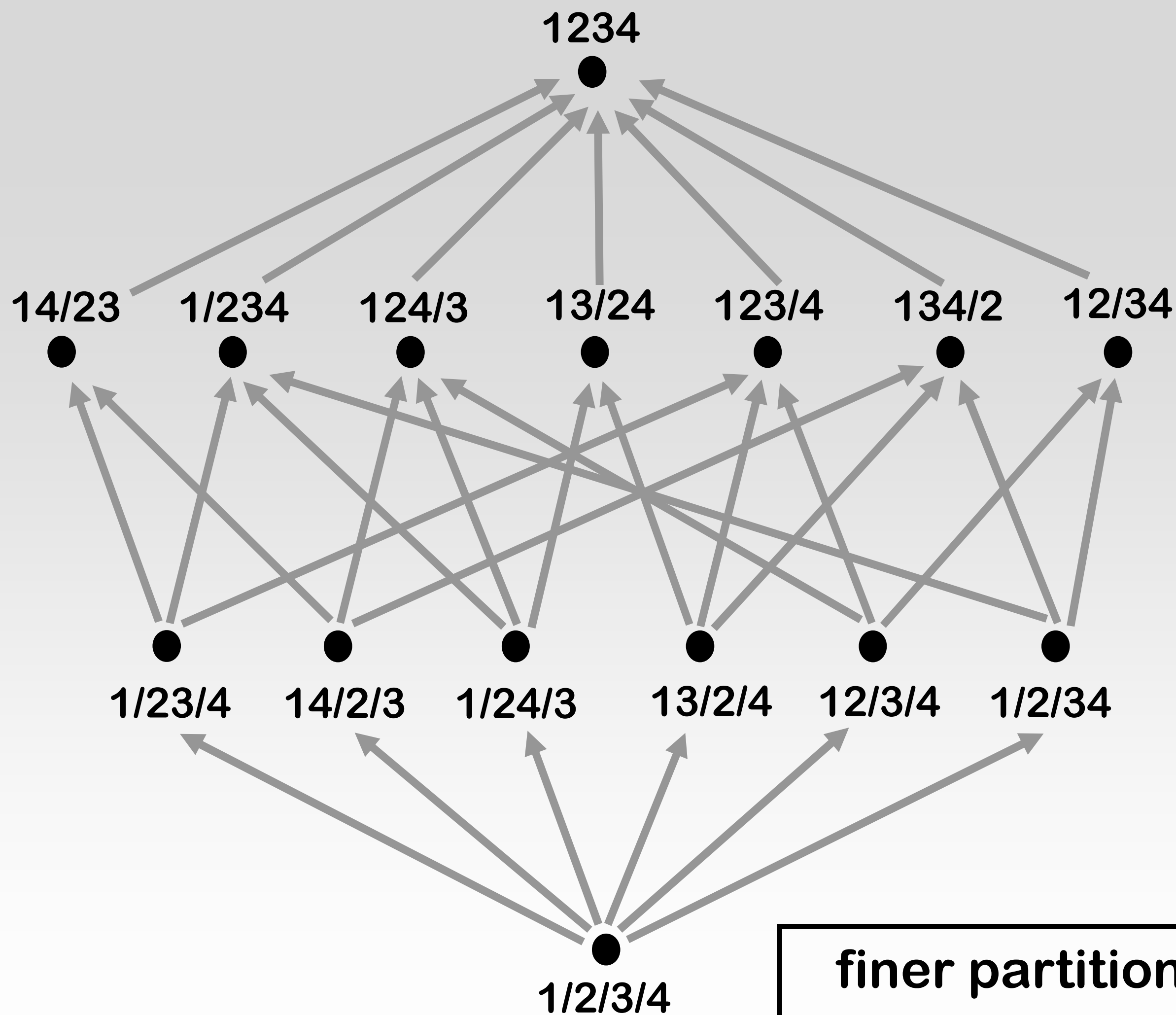
Hasse Diagram



Arrow from a to b if:

1. a<b
2. there exists no c such that a<c<b

Example: Partitions of {1,2,3,4}



finer partition is < than coarser partition

Total Order

Definition: A partial order R is said to be total if

$$\forall a,b \in A,\ a \neq b \rightarrow (aRb) \text{ or } (bRa)$$

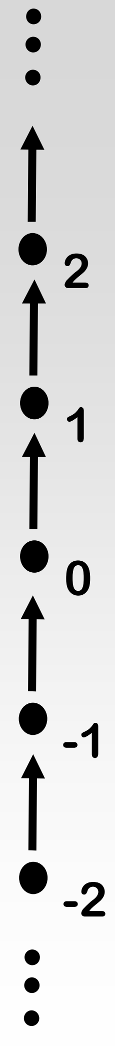
Every two different elements $a,b \in A$ are comparable.

Examples: The $\leq$, < relations on numbers.

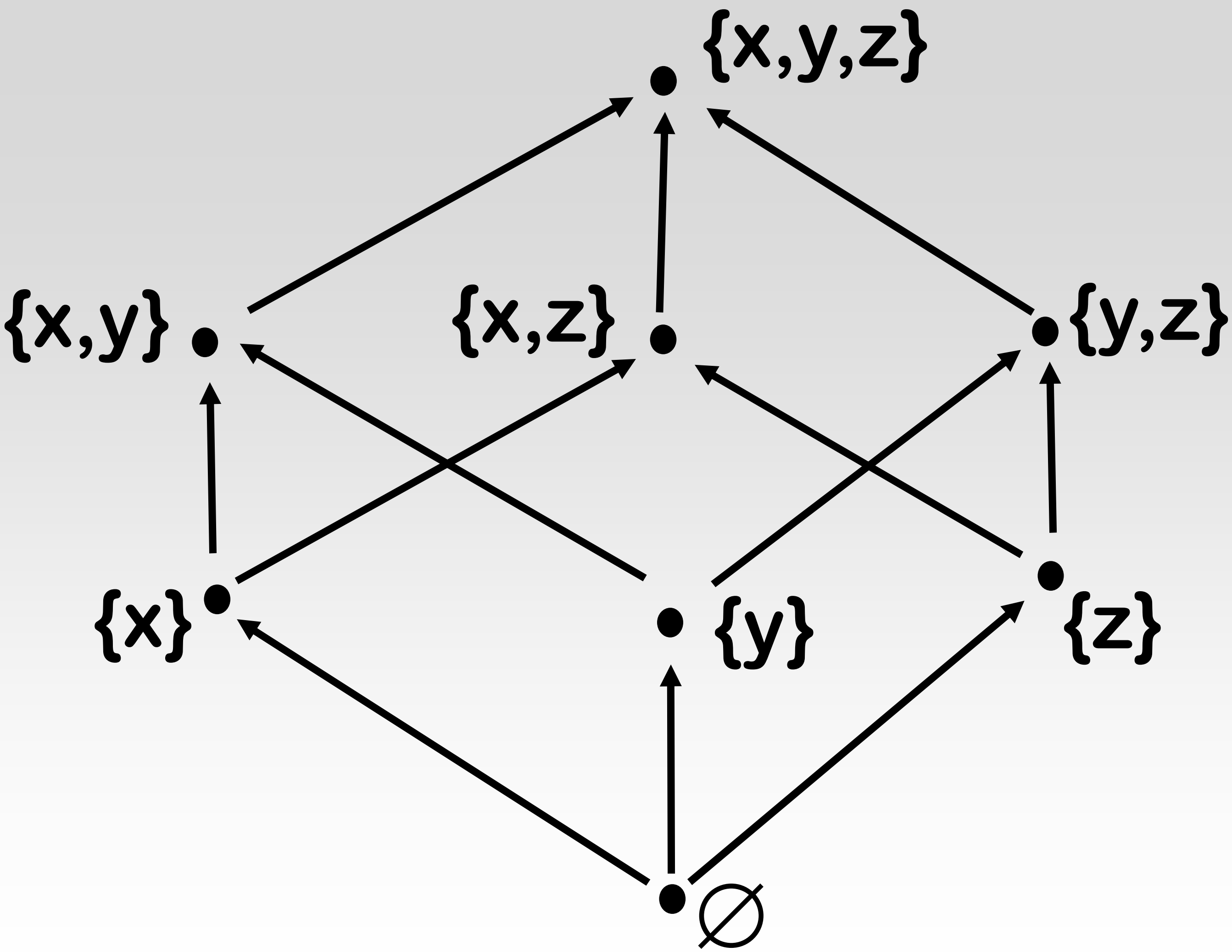
Non-examples: The $\subset$, $\subseteq$ relations on sets.

Comparable	ניתנים להשוואה/ברי השוואה
Total order	יחס סדר מלא

Example: < Relation on $\mathbb{Z}$



Non-Example: Subset Relation



Minimum, Minimal

Definition: Let be a partial order on a set A. An element a ∈ A is **minimum** iff aRb for every other element b ∈ A

Definition: Let be a partial order on a set A. An element a ∈ A is **minimal** iff ¬(bRa) for every other element b ∈ A.

Maximum and maximal are defined analogously.

Note:

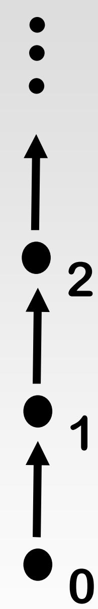
1. In a total order minimum and minimal are the same thing.
2. A partial order, however, may not have a minimum element and many minimal elements.

Totally ordered set קבוצה סדורה היטב

Example: < Relation on $\mathbb{N}$

<u>Q</u>: Is there a minimum?

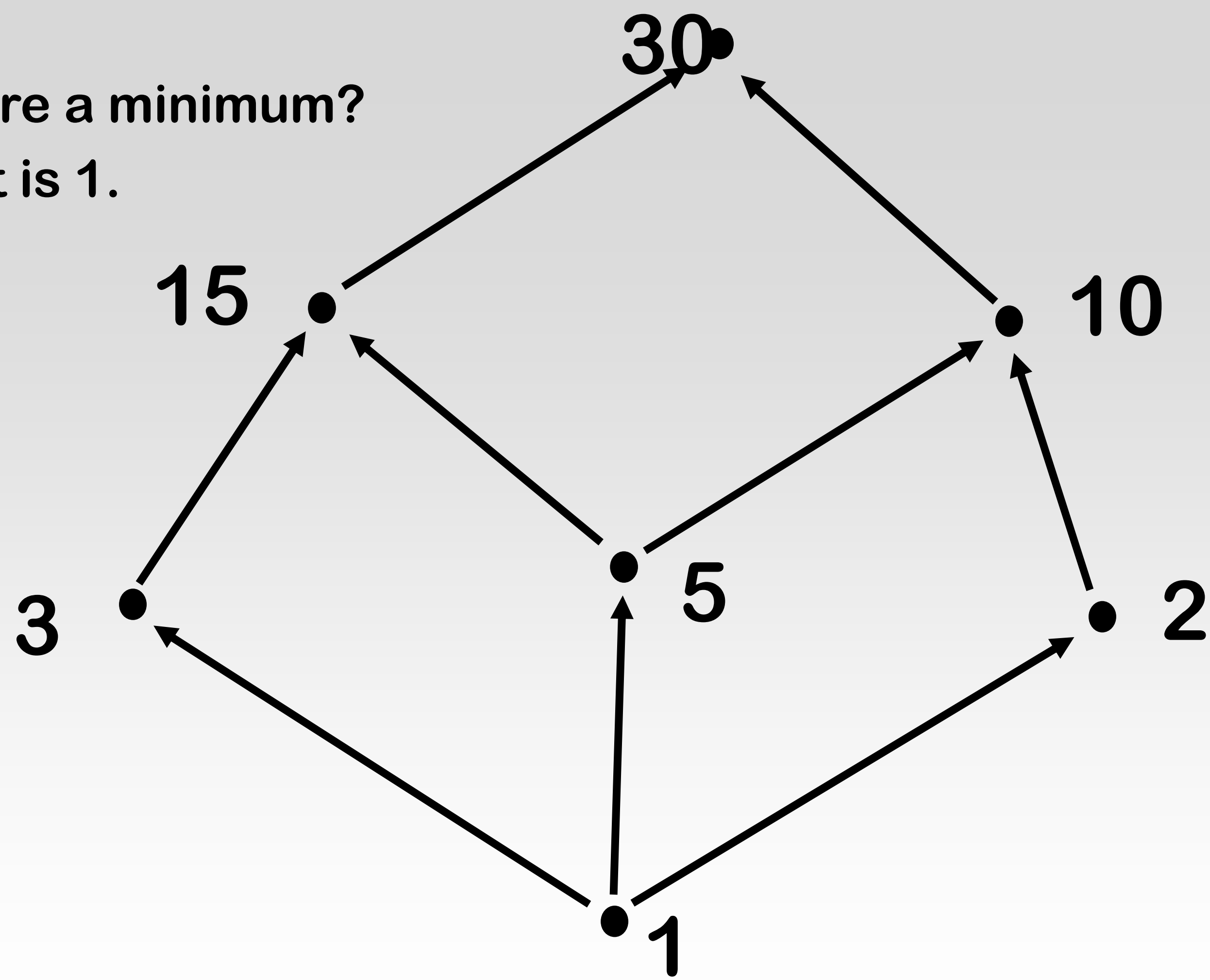
<u>A</u>: Yes. It is 0.



Example: Divides Relation

Q: Is there a minimum?

A: Yes. It is 1.



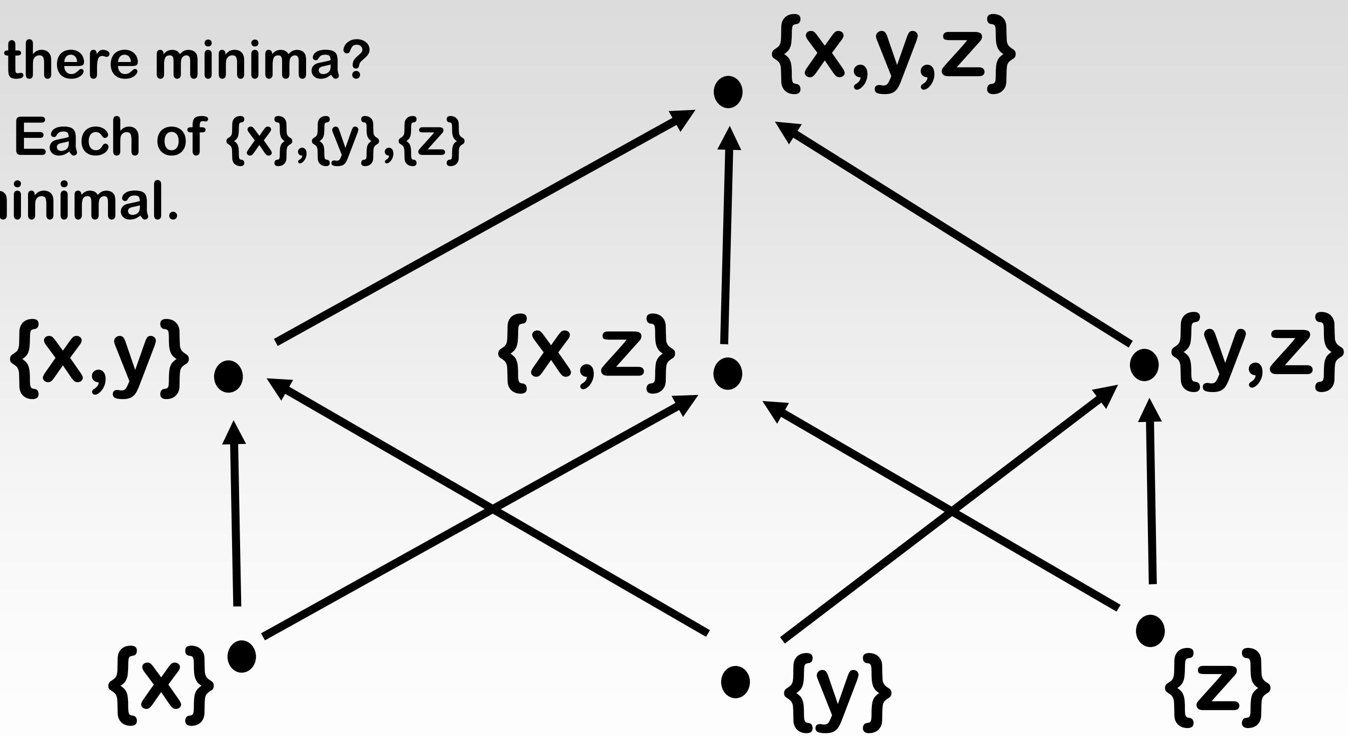
Example: Subset Relation on P({x,y,z})\ ∅

<u>Q1</u>: Is there a minimum?

<u>A1</u>: No.

<u>Q2</u>: Are there minima?

<u>A2</u>: Yes. Each of {x},{y},{z} is minimal.



Representing Partial Orders by Set Containment

Every weak partial order can be represented by the subset relation. Let

$$R\{a\} := \{x \in A \mid xRa\}$$

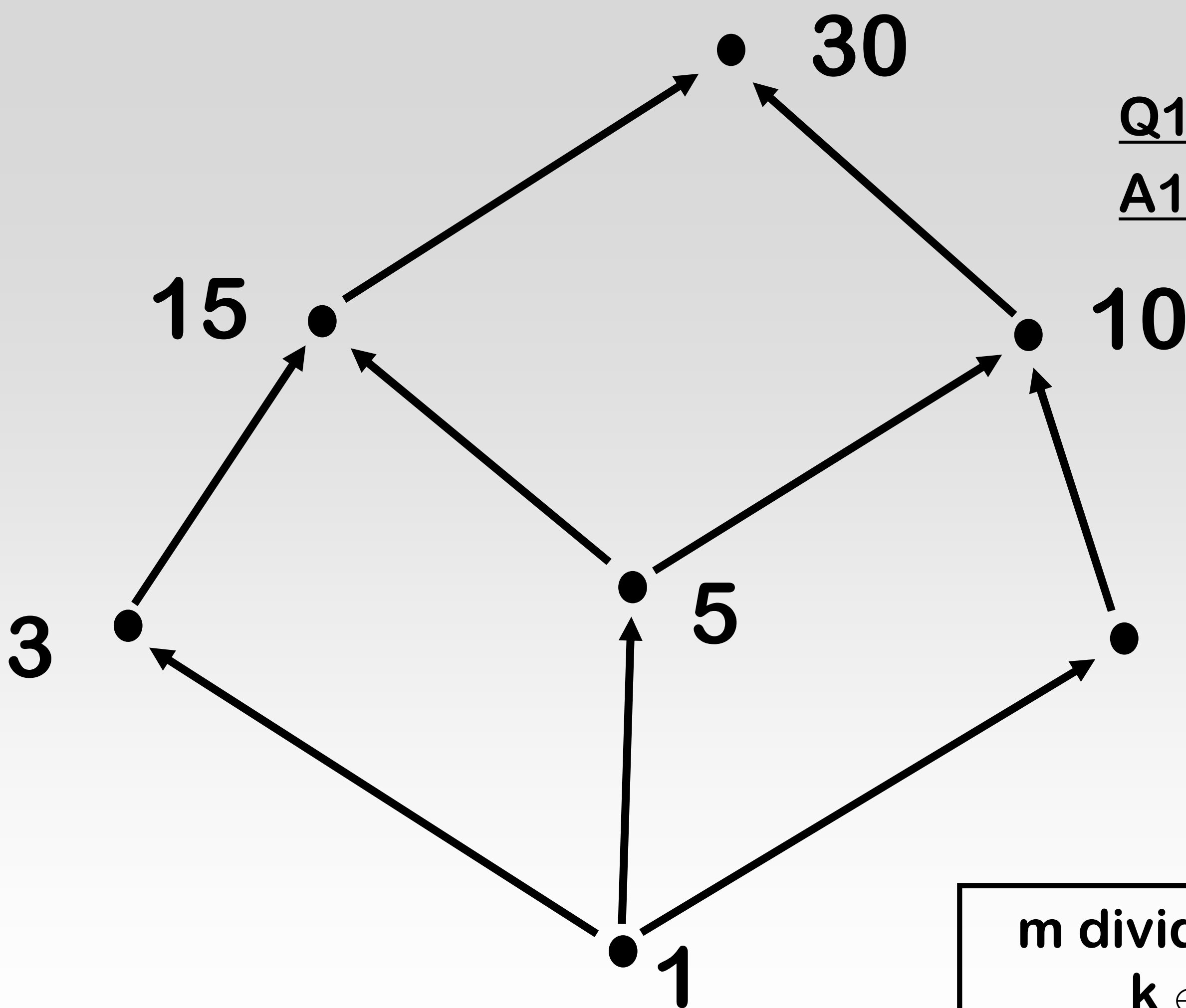
$$R\{b\} := \{x \in A \mid xRb\}$$

Then,

aRb if and only if $R\{a\} \subseteq R\{b\}$

Same applies for strict partial order and $\subset$

Example: Divides Relation

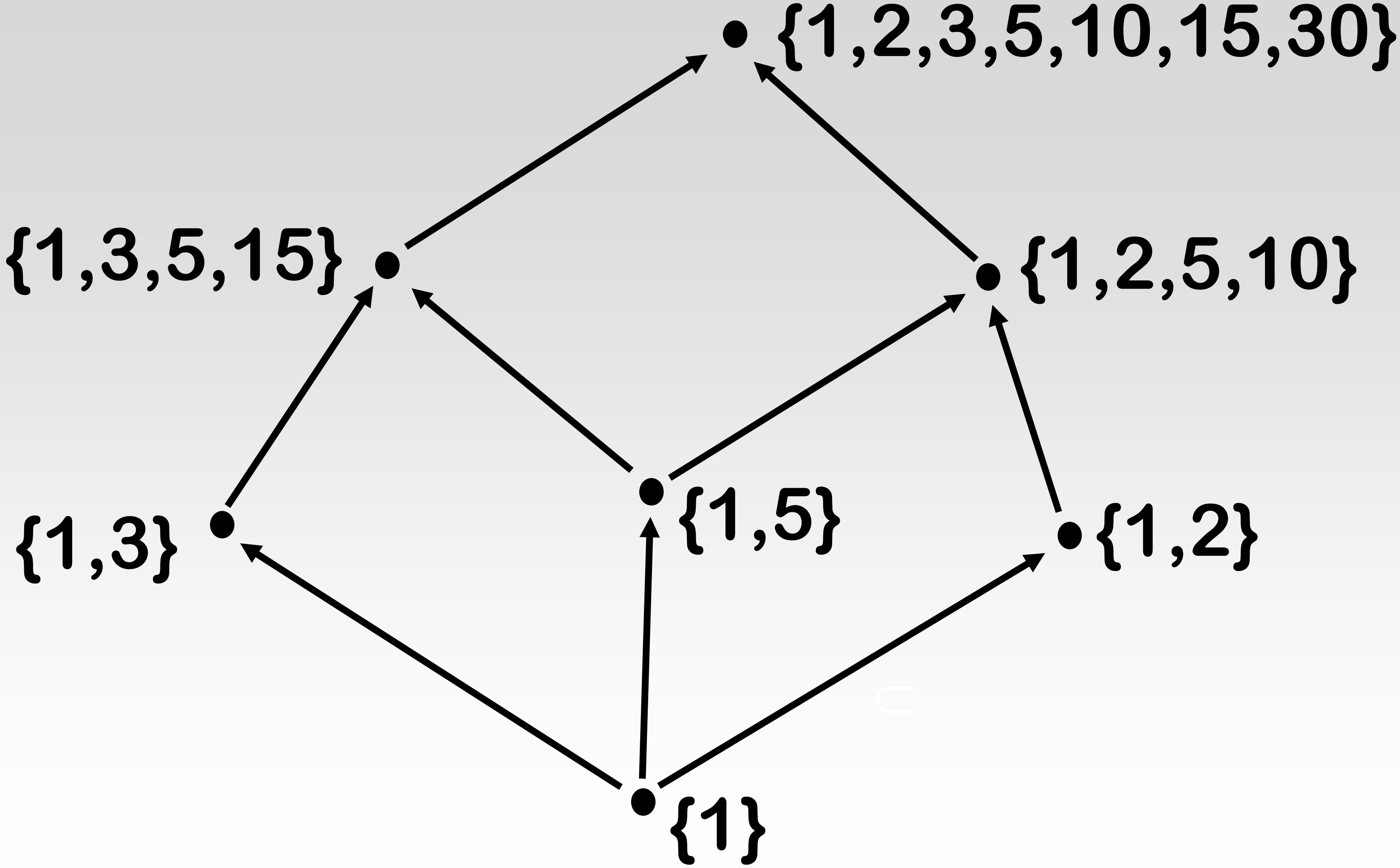


Q1: What is R{10}?

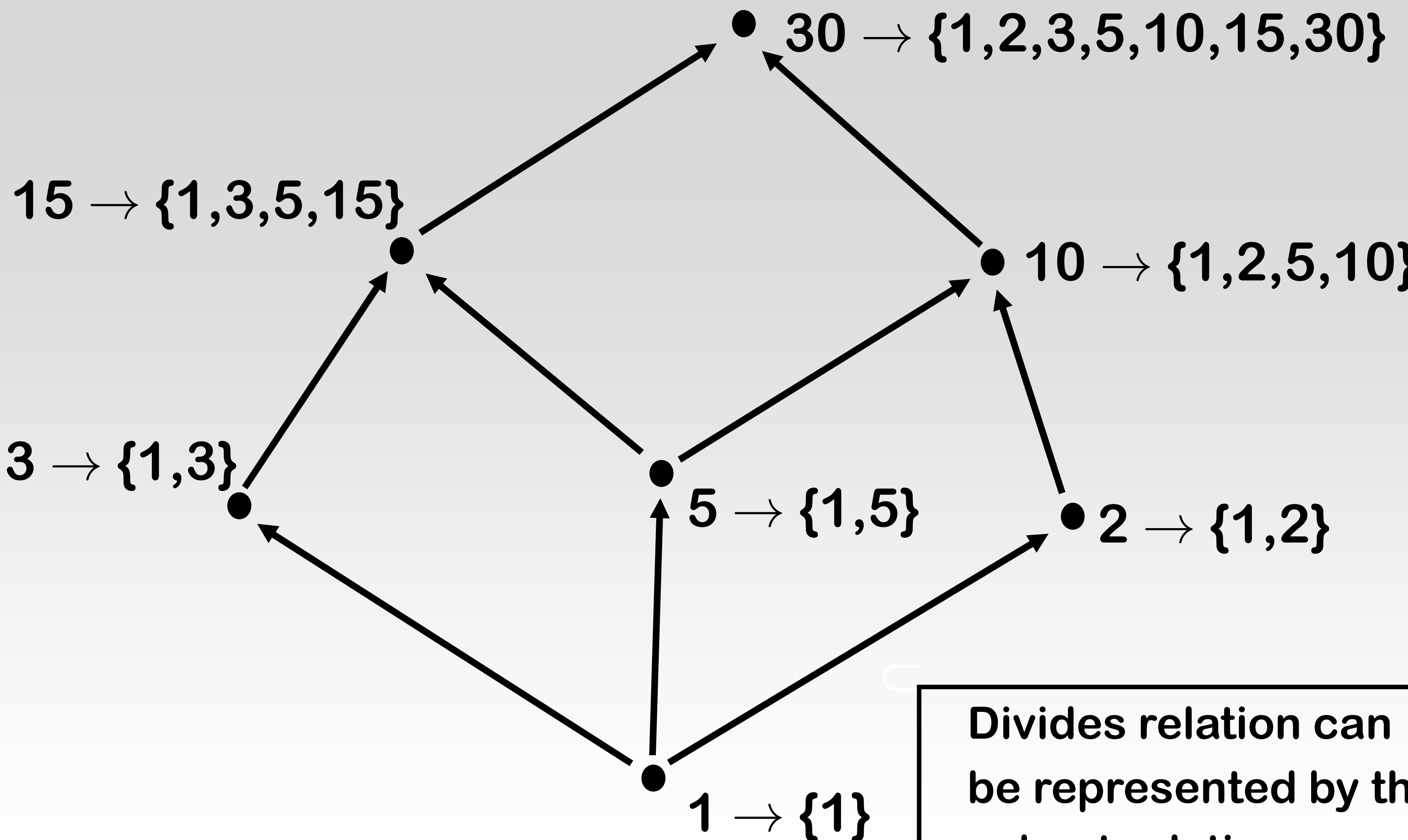
A1: {1,2,5,10}

m divides n if there exists $k \in N$ so that n=km

Strict Subset Relation



Subsets From Divides



Divides relation can be represented by the subset relation

An Application: Scheduling Problems

1. Transitive Closure

2. Topological sorting

3. Chains/antichains

Constructing a Term Schedule

Prerequisite (דרישות קדם)	Course
אינפי 1	אינפי 2
אלגברה 1	אלגברה 2
מבוא מד' מחשב	מבני נתונים
מבני נתונים	אלגוריתמים
מבוא מד' מחשב, מבני נתונים	ארכ' מחשבים
מבני נתונים, ארכ'	מע' הפעלה

Prerequisite Relation

אינפי 1 $\longrightarrow$ אינפי 2

אלגברה 1 $\longrightarrow$ אלגברה 2

מבוא מד' מחשב $\longrightarrow$ מבני נתונים

מבני נתונים $\longrightarrow$ אלגוריתמים

מבוא מד' מחשב, מבני נתונים $\longrightarrow$ ארכ' מחשבים

מבני נתונים, ארכ' $\longrightarrow$ מע' הפעלה

Quick Question

Prerequisite (דרישות קדם)	Course
אינפי 1	אינפי 2
אלגברה 1	אלגברה 2
מבוא מד' מחשב	מבני נתונים
מבני נתונים	אלגוריתמים
מבוא מד' מחשב, מבני נתונים	ארכ' מחשבים
מבני נתונים, ארכ',שפ' תכנות	מע' הפעלה

Q: Is the prerequisite relation total?

A: Not total. אינפי 1 **and** אלגברה 1 **are incomparable.**

Direct vs. Indirect Prerequisites

Direct Prerequisites:

מבוא מד' מחשב → מבני נתונים → ארכ' מחשבים → מע' הפעלה

Indirect prerequisites:

מבוא מד' מחשב → מבני נתונים → ארכ' מחשבים → מע' הפעלה

Transitivity: if a → b and b → c, then a → c.

(→ is transitive closure of →)

Transitive Closure

<u>Definition</u>: The transitive closure of a a binary relation R on a set A is the intersection of all transitive relations that contain R.

<u>Notation</u>: the transitive closure if R is denoted R*.

$$R^* = \bigcap_{\substack{R \subseteq S \\ S \text{ is transitive}}} S$$

<u>R* is well defined</u>:

1. There always exists a transitive relation that contains R (AxA).
2. The intersection of two transitive relations is also transitive.

Transitive Closure

<u>Alternative definition</u>: The <u>transitive closure</u> of a a binary relation R on a set A is the smallest transitive relation R* on A that contains R.

<u>Examples</u>:

1. R = “is parent of.” R* = “is ancestor of.”
2. R = “there is a bus from x to y.” R* = “it is possible to travel from x to y by one or more buses.”
3. R = “x is a direct prerequisite of y.” R* = “x is a prerequisite of y.”

<u>Claim</u>: aR*b if and only if there exist a finite sequence of elements $a_1,a_2,\ldots,a_k$ so that

$$aRa_1,\ a_1Ra_2,\ldots,\ a_{k-1}Ra_k,\ a_kRb$$

First Year Subjects

אינפי 1	אלגברה 1	מבוא מד' מחשב

Subjects with no prerequisites:

d is a first year subject if <*nothing*> $\rightarrow$ d

d is <u>minim*al*</u>

Recall: Minim*al* not Minim*um*

Minim*um* means "smallest"

a prerequisite for every subject

no minimum in this example.

Constructing a Term Schedule

אינפי 1 → אינפי 2

אלגברה 1 → אלגברה 2

מבוא מד' מחשב → מבני נתונים

מבני נתונים → אלגוריתמים

מבוא מד' מחשב, מבני נתונים → ארכ' מחשבים

מבני נתונים, ארכ', שפ' תכנות → מע' הפעלה

Identify minim*al* elements

Constructing a Term Schedule II

אינפי 1	אלגברה 1	מבוא מד' מחשב

Start schedule with minimal elements

Constructing a Term Schedule III

~~אינפי 1~~ → אינפי 2

~~אלגברה 1~~ → אלגברה 2

~~מבוא מד' מחשב~~ → מבני נתונים

מבני נתונים → אלגוריתמים

~~מבוא מד' מחשב~~, מבני נתונים → ארכ' מחשבים

מבני נתונים, ארכ', שפ' תכנות → מע' הפעלה

Remove <u>minim*al*</u> elements

Constructing a Term Schedule IV

אינפי 2 →

אלגברה 2 →

מבני נתונים →

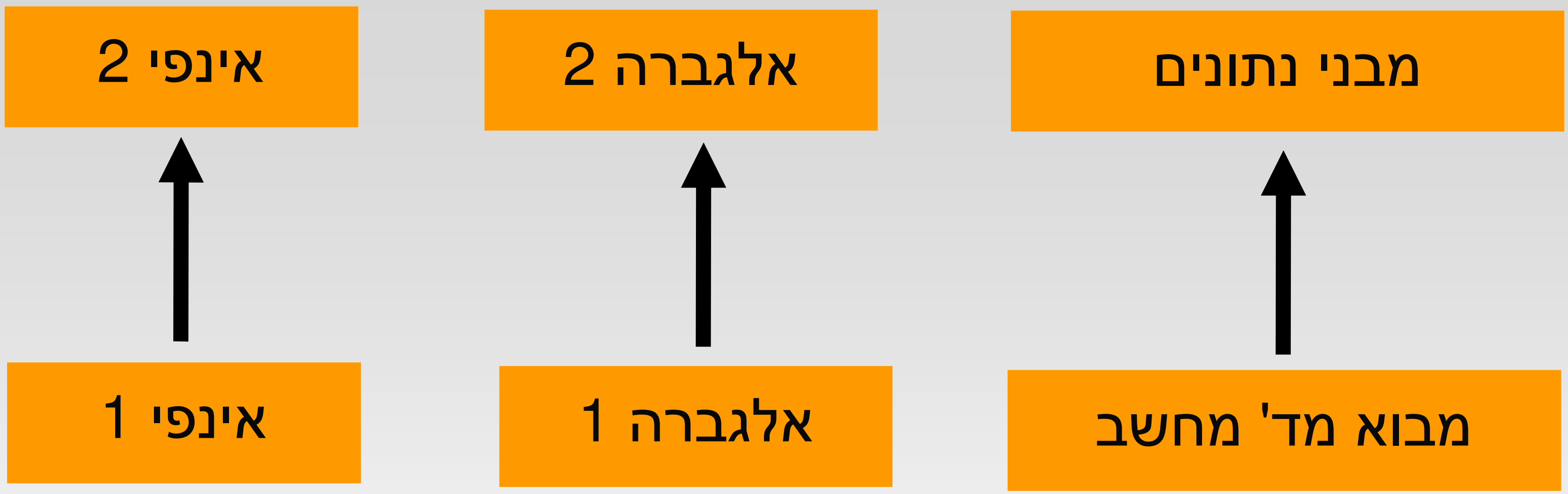
אלגוריתמים → מבני נתונים

ארכ' מחשבים → מבני נתונים

מע' הפעלה → מבני נתונים, ארכ',שפ' תכנות

Identify new <u>minim*al*</u> elements

Constructing a Term Schedule V



Schedule them next, and so on…

No Loops

Will not work if there are "loops."

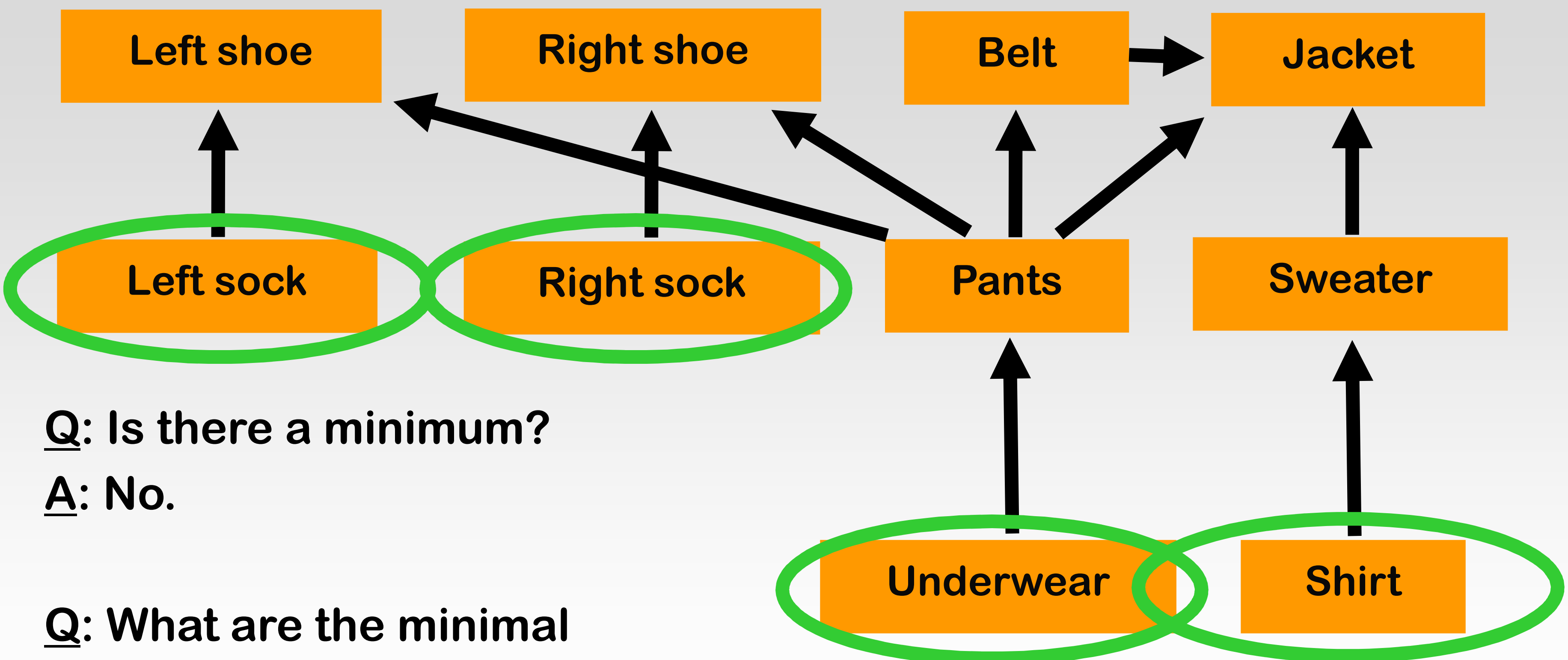
For example: if מע' הפעלה $\longrightarrow$ ארכ' מחשבים**, then**

מבוא מד' מחשב $\longrightarrow$ מבני נתונים $\longrightarrow$ ארכ' מחשבים $\longrightarrow$ מע' הפעלה

Asymmetry: a $\rightarrow$ b implies $\neg$(b $\rightarrow$ a).

Asymmetry guarantees that there are no loops.

Another Example: Getting Dressed



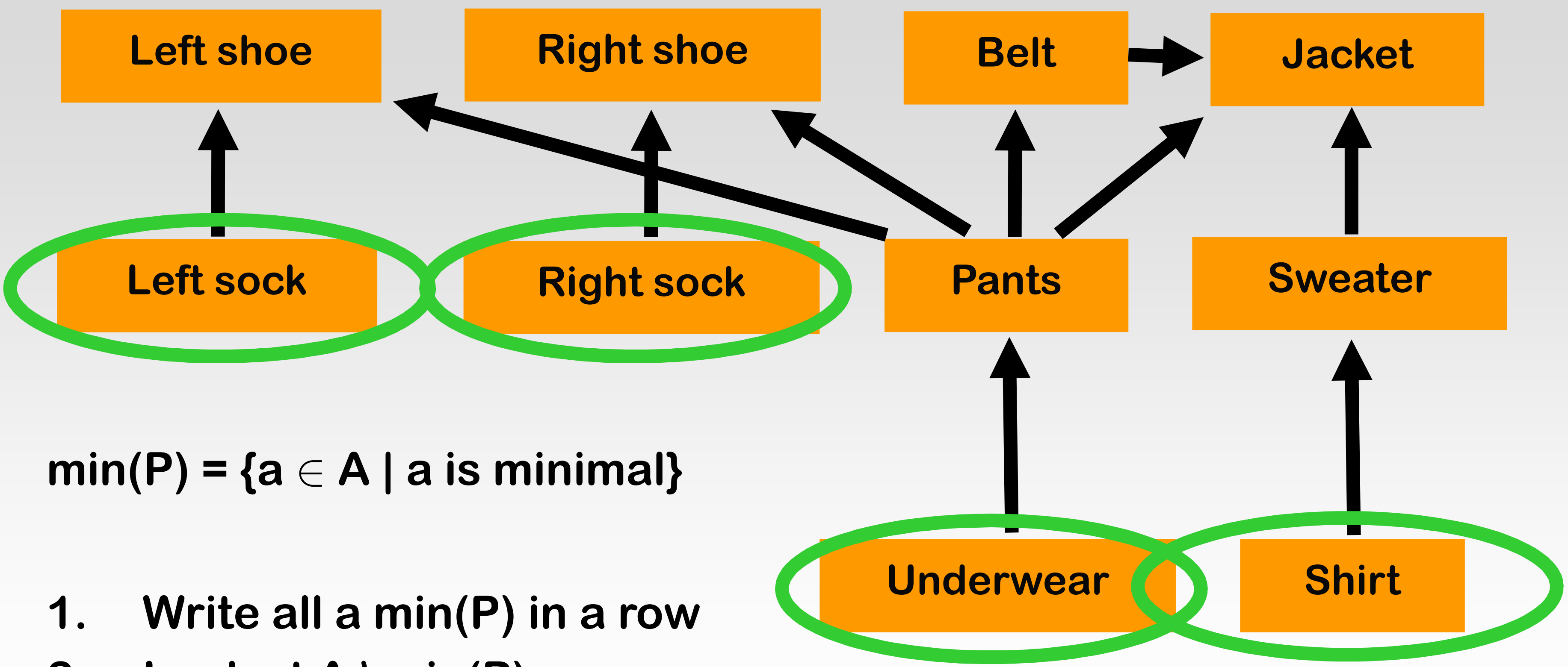
Q: Is there a minimum?

A: No.

Q: What are the minimal elements?

A: Left sock, right sock, underwear, shirt

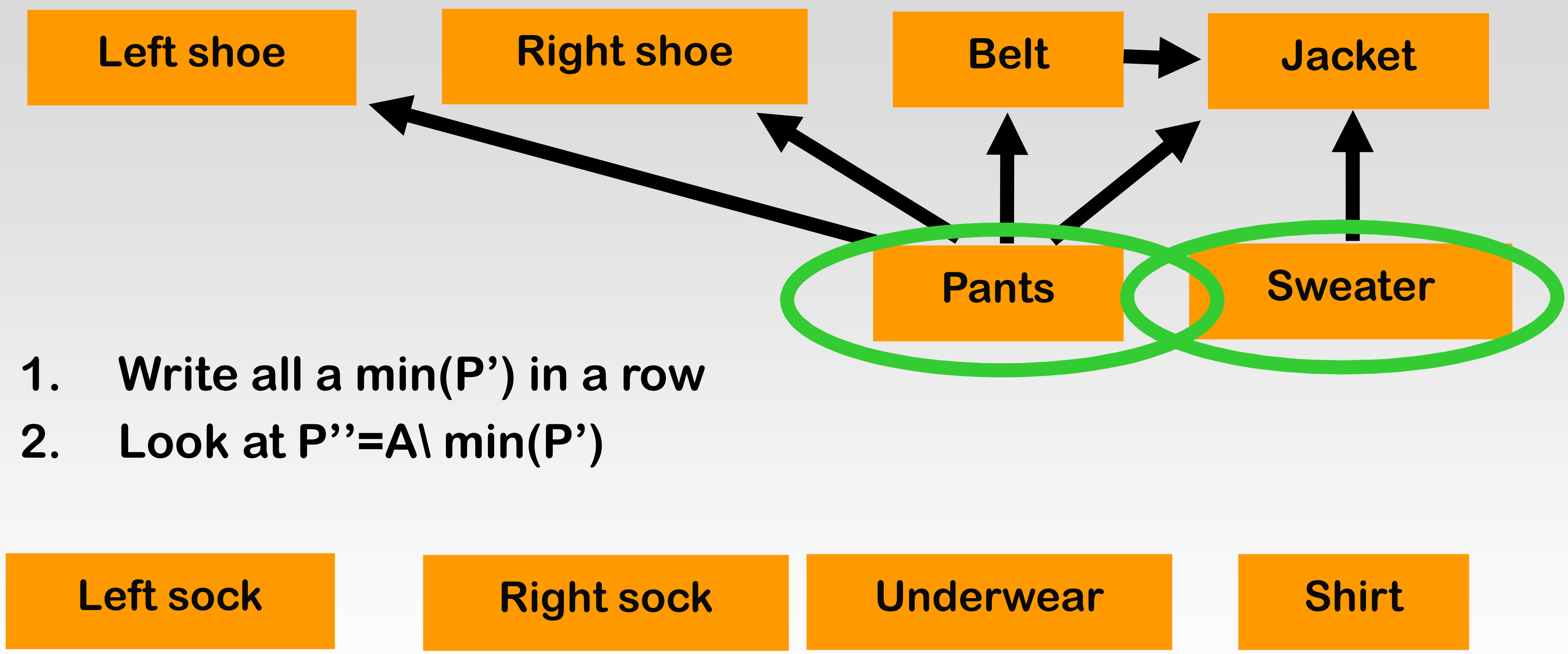
Hasse Diagram



min(P) = {a $\in$ A | a is minimal}

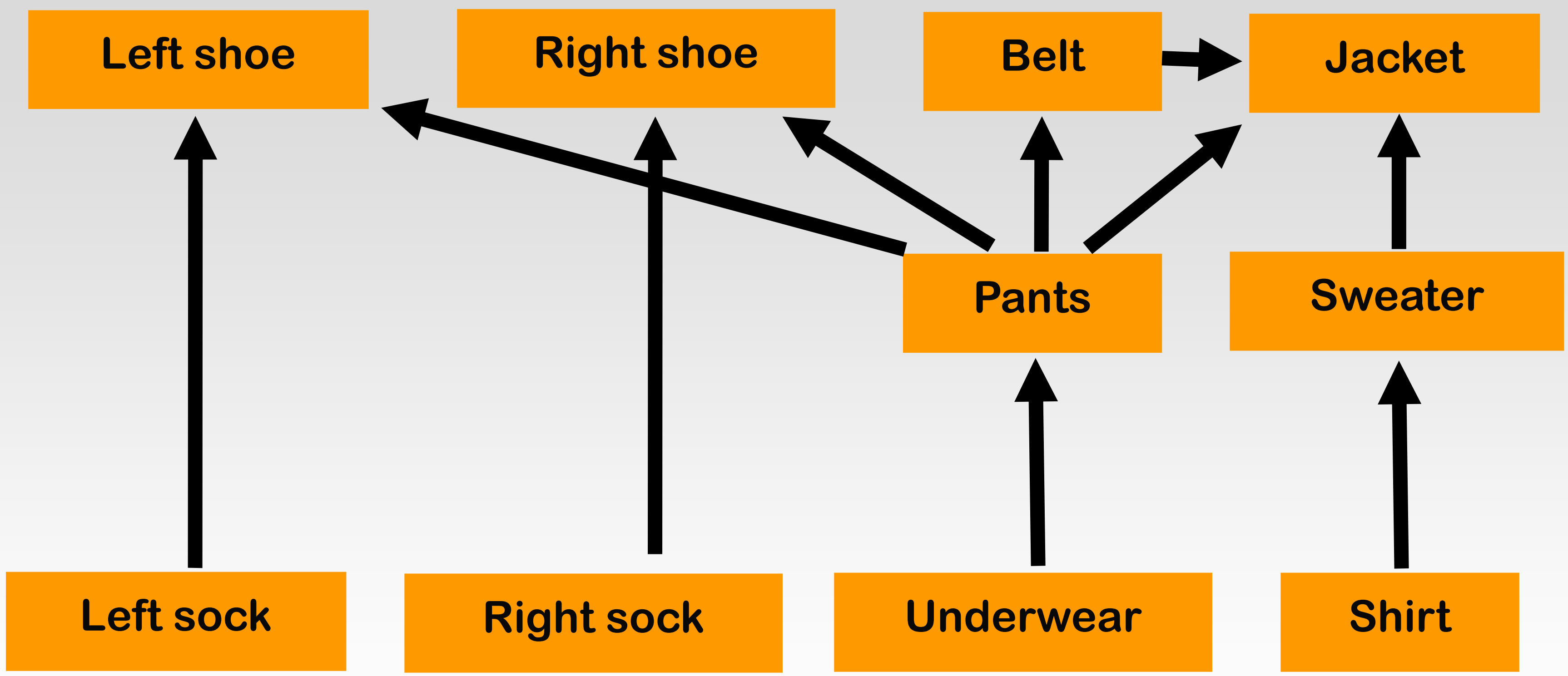
1. Write all a min(P) in a row
2. Look at A \ min(P)

Hasse Diagram II



1. Write all a min(P’) in a row
2. Look at P’’=A\ min(P’)

Hasse Diagram III



Connect related elements that do not have an element “between” them.

Topological Sorting

Consider a partial order of tasks to be perfomed. For example:

1. Term schedule.
2. Getting dressed.

We would like to have a specific order in which to perform the tasks, one at a time.

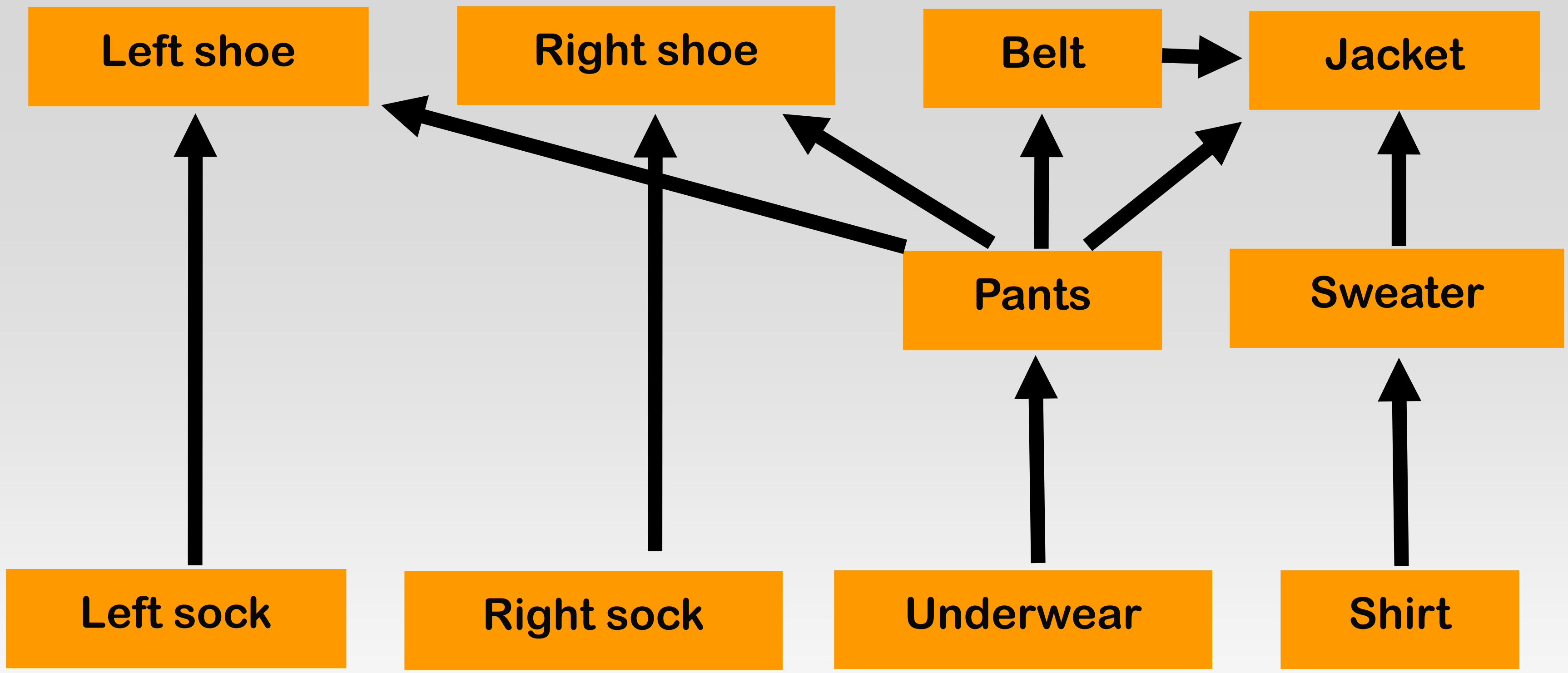
Can do this by finding a total order that is consistent with the partial order. This is what we call a topological sort.

Definition: A topological sort of a partial order R on a set A is a total ordering, S , on A such that

aRb implies aSb

Topological sort מיון טופולוגי

Topological Sort: Example



shirt S sweater S u-wear S lsock S rsock S pants S lshoe S rshoe S belt S jacket

Constructing a Topological Sort

<u>Theorem</u>: Every partial order on a finite set has a topological sort.

<u>Remark</u>: True also for infinite sets (but we focus on finite ones).

<u>Proof</u>: We already saw how to construct a topological sort:

1. Pick a minimal element
2. Pick a minimal element among the remaining ones.
3. And so on…

Need to make sure that:

1. There is always a minimal element to pick from.
2. What we are constructing is a total order:
   a. Asymmetric
   b. Transitive
   c. Any two elements are comparable

Constructing a Topological Sort II

There is always a minimal element:

1. Sounds sort of obvious for finite sets.
2. But doesn't always hold for infinite – for example (Z,<).

Lemma 1: Every partial order on a nonempty finite set has a minimal element.

Lemma 2: Construction generates a total order.

Proofs: on the board.

Parallel Task Scheduling

Topological sorting - tasks are executed sequentially.

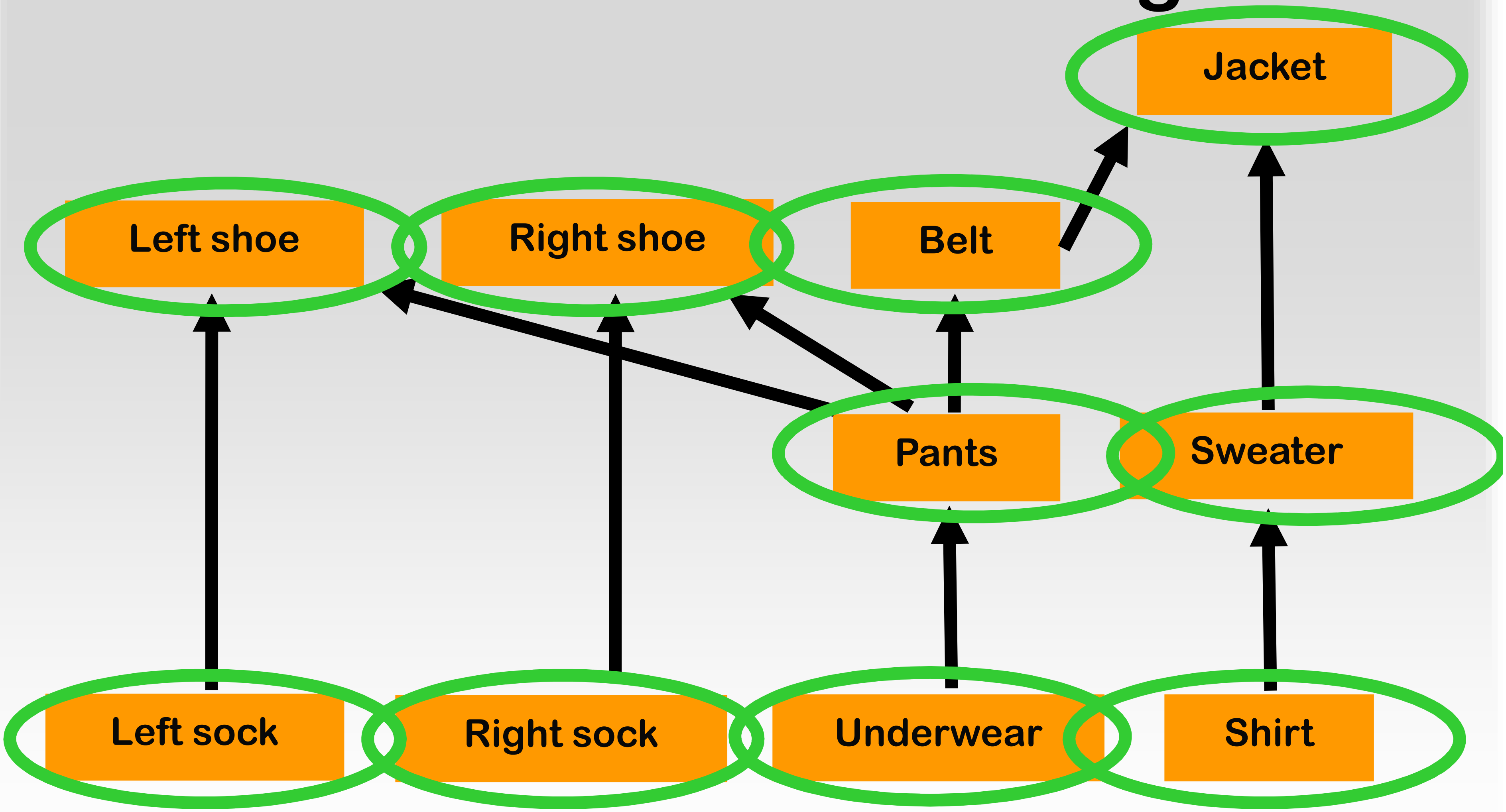
What if we can execute more than one task at a time?

For example, suppose we have a parallel machine.

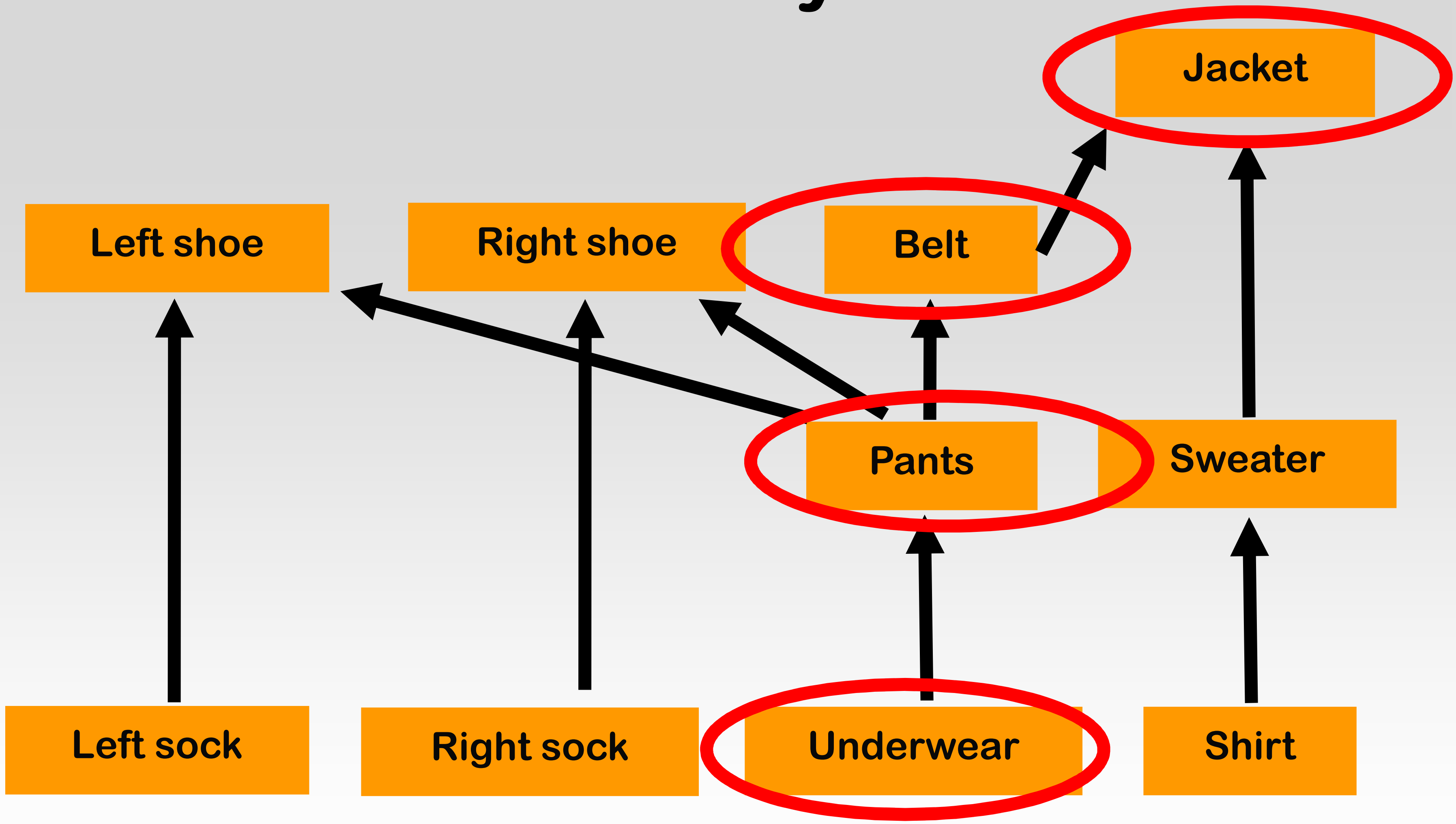
Want to minimize the total time to complete the tasks.

Sequentially	סדרתית
In parallel	במקביל

Parallel Task Scheduling II



Can we do any better?



Chain

<u>Definition</u>: A <u>chain</u> in a partial order is a set of elements such that any two elements in the set are comparable

<u>Terminology</u>: A largest chain is also known as a <u>critical path</u>.

$$\text{min parallel time} \geq \text{max chain size}$$

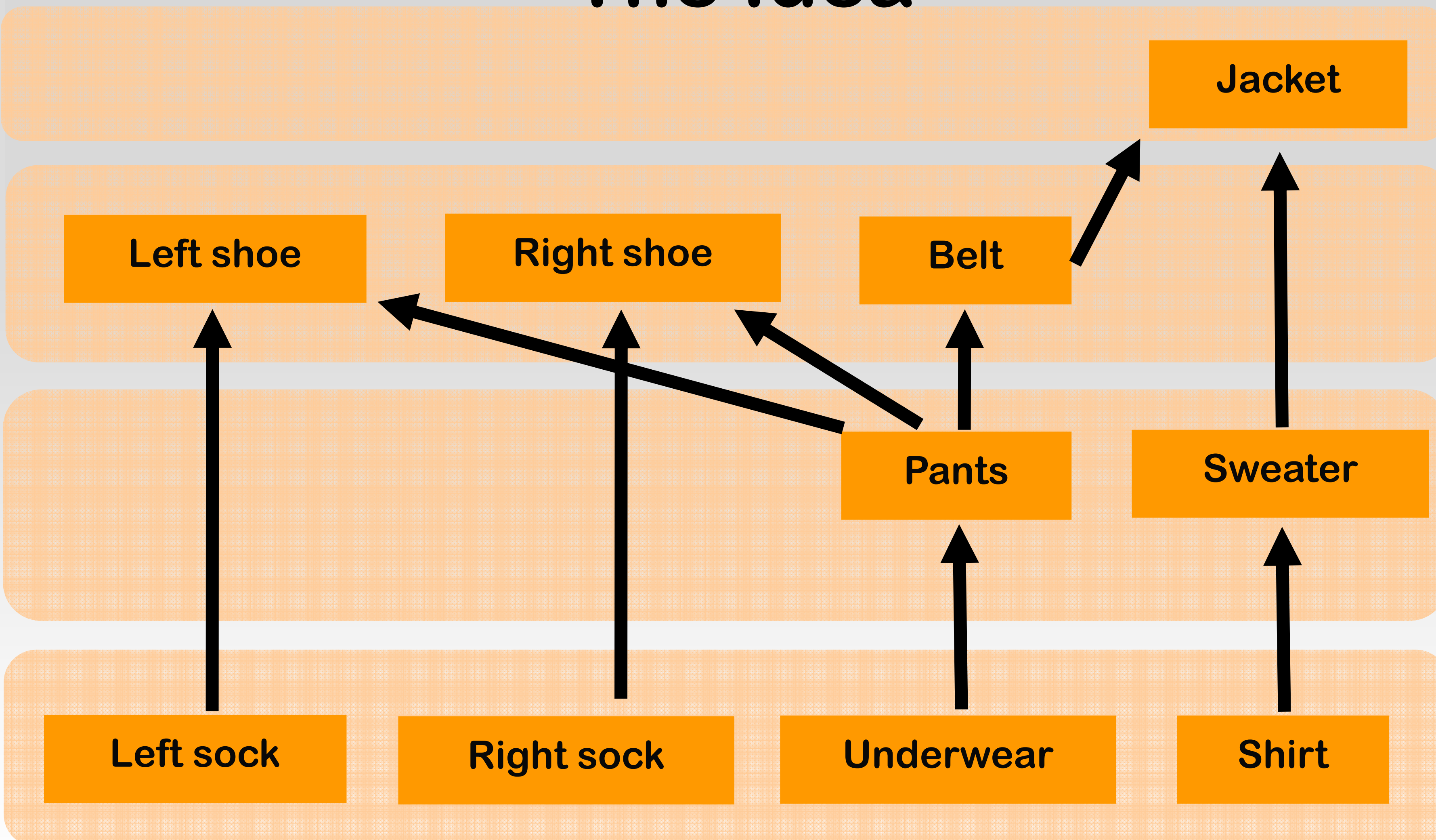
Can also show that

$$\text{min parallel time} \leq \text{max chain size}$$

<u>Corollary</u>: min parallel time = max chain size.

Chain שרשרת

The Idea



Partition into *successive* blocks of incomparable elements

Antichain

<u>Definition</u>: An <u>antichain</u> in a partial order is a set of elements such that any two elements in the set are *in*comparable

Corresponds to a “block.”

If the largest chain is of size t, then the domain can be partitioned into t antichains.

<u>Dilworth’s Lemma</u>: For all t > 0, every poset with n elements must have either:

1. A chain of size > t, OR
2. An antichain of size ≥ n/t.

Antichain אנטי-שרשרת

Dilworth’s Lemma: Example

<u>Dilworth’s Lemma</u>: For all $t > 0$, every poset with n elements must have either:

1. A chain of size $> t$, OR
2. An antichain of size $\geq n/t$.

In the “getting dressed” poset, set $n = 10$

1. For $t = 3$, there is a chain of size 4.
2. For $t = 4$, there is no chain of size 5,
   but there is an antichain of size $4 \geq 10/4$.