COSC 290 Discrete Structures

Lecture 24: Partial orders and equivalence relations

Prof. Michael Hay

Monday, Apr. 2, 2018

Colgate University

Plan for today

1. Closures

2. Equivalence relations and partial orders

3. Hasse diagram

Closures

Closures

A closure of a relation R on A is a smallest $R' \supseteq R$ that satisfies a desired property.

- reflexive closure:

Closures

A closure of a relation R on A is a smallest $R' \supseteq R$ that satisfies a desired property.

- reflexive closure:

$$R' = R \cup \{ \langle a, a \rangle : a \in A \}$$

- symmetric closure:

Closures

A closure of a relation R on A is a smallest $R' \supseteq R$ that satisfies a desired property.

- reflexive closure:

$$R' = R \cup \{ \langle a, a \rangle : a \in A \}$$

- symmetric closure:

$$R' = R \cup R^{-1}$$

- transitive closure:

Closures

A closure of a relation R on A is a smallest $R' \supseteq R$ that satisfies a desired property.

- reflexive closure:

$$R' = R \cup \{ \langle a, a \rangle : a \in A \}$$

- symmetric closure:

$$R' = R \cup R^{-1}$$

- transitive closure:
  (*hint*: what does $R \circ R$ give you?)

Poll: towards transitive closure

Consider the *parentOf* relation on persons where $\langle p, c \rangle \in \mathit{parentOf}$ if p is the parent of c. What is $\mathit{parentOf} \circ \mathit{parentOf}$?

A) ancestorOf
B) grandParentOf
C) parentOf
D) childOf
E) grandChildOf
F) descendantOf

Bonus question for you to consider during the discussion period: what is $\mathit{parentOf} \cup (\mathit{parentOf} \circ \mathit{parentOf})$?

Computing the transitive closure

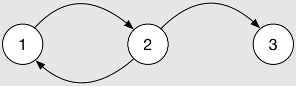
Input: Relation $R \subseteq A \times A$.

Output: smallest $R' \supseteq R$ that is *transitive*

$R' := R$
repeat
    $new := (R \circ R') - R'$
    $R' := R' \cup new$
until $|new| = 0$
return R'

Example (Applying transitive closure algorithm)

Let's apply the algorithm to this example:



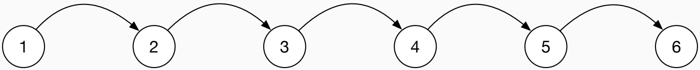
Exercise

Input: Relation $R \subseteq A \times A$.

Output: smallest $R' \supseteq R$ that is *transitive*

$R' := R$
repeat
    $new := (R \circ R') - R'$
    $R' := R' \cup new$
until $|new| = 0$
return R'

Exercise: working in groups, apply the algorithm to this graph. How many times does the loop repeat?



Closures

A closure of a relation R on A is a smallest $R' \supseteq R$ that satisfies a desired property.

- reflexive closure:

$$R' = R \cup \{ \langle a, a \rangle : a \in A \}$$

- symmetric closure:

$$R' = R \cup R^{-1}$$

- transitive closure:

$$R' = R \cup (R \circ R) \cup ((R \circ R) \circ R) \cup \cdots$$

Equivalence relations and partial orders

Recall: relation properties

For relation R on $A \times A$.

- **R** *reflexive*: for every $a \in A$, $\langle a, a \rangle \in R$.
- **IR** *irreflexive*: for every $a \in A$, $\langle a, a \rangle \notin R$.
- **S** *symmetric*: for every $a, b \in A$, if $\langle a, b \rangle \in R$, then $\langle b, a \rangle \in R$.
- **antiS** *antisymmetric*: for every $a, b \in A$, if $\langle a, b \rangle \in R$ and $\langle b, a \rangle \in R$, then $a = b$.
- **AS** *asymmetric*: for every $a, b \in A$, if $\langle a, b \rangle \in R$, then $\langle b, a \rangle \notin R$.
- **T** *transitive*: for every $a, b, c \in A$, if $\langle a, b \rangle \in R$ and $\langle b, c \rangle \in R$, then $\langle a, c \rangle \in R$.

Special relation: equivalence relation

Relation R on A is an equivalence relation if it is reflexive, symmetric, transitive.

Special relation: equivalence relation

Relation R on A is an equivalence relation if it is reflexive, symmetric, transitive.

Conventions: use $\equiv$ as the "name" of the relation (as opposed to a letter like R) and use *infix* notation: $a \equiv b$ instead of $\langle a, b \rangle \in \ \equiv$.

Special relation: equivalence relation

Relation R on A is an equivalence relation if it is reflexive, symmetric, transitive.

Conventions: use $\equiv$ as the “name” of the relation (as opposed to a letter like R) and use *infix* notation: $a \equiv b$ instead of $\langle a, b \rangle \in \; \equiv$.

Intuition: equivalence relations behave like $=$.

Equivalence classes

When R is an equivalence relation on A, the elements of A be can be partitioned into equivalence classes. (See book for formal definition.)

Equivalence classes

When R is an equivalence relation on A, the elements of A be can be partitioned into equivalence classes. (See book for formal definition.)

Example (Equivalence classes)

Let R denote the equivalence relation on $\{0, 1, 2, \ldots, 10\}$ where $\langle a, b \rangle \in R$ if $(a \bmod 2) = (b \bmod 2)$

The equivalence classes are:

Equivalence classes

When R is an equivalence relation on A, the elements of A be can be partitioned into equivalence classes. (See book for formal definition.)

Example (Equivalence classes)

Let R denote the equivalence relation on $\{0, 1, 2, \ldots, 10\}$ where $\langle a, b\rangle \in R$ if $(a \bmod 2) = (b \bmod 2)$

The equivalence classes are:

- $\{0, 2, 4, 6, 8, 10\}$
- $\{1, 3, 5, 7, 9\}$

Exercise

Let $S := \{0, 1\}^3$ be the set of length 3 bitstrings. Consider the two binary relations R_1 and R_2 on S defined as follows:

1. $(x, y) \in R_1$ if x and y are identical or reverses of each other. For example, if $x = b_1 b_2 \ldots b_n$, we say that $\text{reverse}(x) = b_n b_{n-1} \ldots b_1$. Then, $(x, y) \in R_1$ iff $x = y$ or $x = \text{reverse}(y)$.
2. $(x, y) \in R_2$ if x and y are rearrangements/permutations of each other. For example, if $x = b_1 b_2 \ldots b_n$, then $(x, y) \in R_2$ iff there exists some bijection $p : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that $y = b_{p(1)} b_{p(2)} \ldots b_{p(n)}$.

Exercise

Let $S := \{0, 1\}^3$ be the set of length 3 bitstrings. Consider the two binary relations R_1 and R_2 on S defined as follows:

1. $(x, y) \in R_1$ if x and y are identical or reverses of each other. For example, if $x = b_1 b_2 \dots b_n$, we say that $\text{reverse}(x) = b_n b_{n-1} \dots b_1$. Then, $(x, y) \in R_1$ iff $x = y$ or $x = \text{reverse}(y)$.
2. $(x, y) \in R_2$ if x and y are rearrangements/permutations of each other. For example, if $x = b_1 b_2 \dots b_n$, then $(x, y) \in R_2$ iff there exists some bijection $p : \{1, \dots, n\} \to \{1, \dots, n\}$ such that $y = b_{p(1)} b_{p(2)} \dots b_{p(n)}$.

Working in small groups, write out the equivalence classes for R_1 and R_2.

Special relation: partial order

Relation R on A is a partial order if it is reflexive, antisymmetric, transitive.

Special relation: partial order

Relation R on A is a partial order if it is reflexive, antisymmetric, transitive.

Conventions: use $\preceq$ as the "name" of the relation (as opposed to a letter like R) and use *infix* notation: $a \preceq b$ instead of $\langle a, b \rangle \in \ \preceq$.

Special relation: partial order

Relation R on A is a partial order if it is reflexive, antisymmetric, transitive.

Conventions: use $\preceq$ as the “name” of the relation (as opposed to a letter like R) and use *infix* notation: $a \preceq b$ instead of $\langle a, b \rangle \in \ \preceq$.

Intuition: partial order relations behave like $\leq$ except that some pairs may be *incomparable*.

Special relation: partial order

Relation R on A is a partial order if it is reflexive, antisymmetric, transitive.

Conventions: use $\preceq$ as the “name” of the relation (as opposed to a letter like R) and use *infix* notation: $a \preceq b$ instead of $\langle a, b \rangle \in \ \preceq$.

Intuition: partial order relations behave like $\leq$ except that some pairs may be *incomparable*.

Example (Partial order)

The *prefixOf* relation is a partial order:

- “a” $\preceq$ “aa”
- “aa” $\preceq$ “aardvark”

Note: not all pairs comparable: “a” $\not\preceq$ “b” and “b” $\not\preceq$ “a”

Special relation: strict partial order

Relation R on A is a strict partial order if it is irreflexive, (antisymmetric), transitive.

Special relation: strict partial order

Relation R on A is a strict partial order if it is irreflexive, (antisymmetric), transitive.

Conventions: use $\prec$ as the "name" of the relation (as opposed to a letter like R) and use *infix* notation: $a \prec b$ instead of $\langle a, b \rangle \in \prec$.

Special relation: strict partial order

Relation R on A is a strict partial order if it is irreflexive, (antisymmetric), transitive.

Conventions: use $\prec$ as the "name" of the relation (as opposed to a letter like R) and use *infix* notation: $a \prec b$ instead of $\langle a, b \rangle \in \prec$.

Intuition: strict partial order relations behave like $<$ except that some pairs may be *incomparable*.

Special relation: strict partial order

Relation R on A is a strict partial order if it is irreflexive, (antisymmetric), transitive.

Conventions: use $\prec$ as the "name" of the relation (as opposed to a letter like R) and use *infix* notation: $a \prec b$ instead of $\langle a, b\rangle \in\ \prec$.

Intuition: strict partial order relations behave like $<$ except that some pairs may be *incomparable*.

Example (Strict partial order)

The *ancestorOf* relation (ancestor is parent or (recursively) parent of ancestor):

- "DT" $\prec$ "Don Jr"
- "Hanns Drumpf" $\prec$ "DT" (#makedonalddrumpfagain)
- not all pairs comparable: "Harry Potter" $\not\prec$ "Aunt Petunia" and "Aunt Petunia" $\not\prec$ "Harry Potter"

Poll: partial order

Relation $\preceq$ is a partial order if it is reflexive, antisymmetric, transitive.

Consider two relations on a set of track runners:

- $a \preceq_1 b$ if the number of races in which a competed is no more than the number in which b competed.
- $a \preceq_2 b$ if the total amount of time (measured in nanoseconds with laser precision so that ties are impossible) that a ran is no more than the total amount of time that b ran.

Is $\preceq_1$ a partial order? Is $\preceq_2$ a partial order?

A) Yes, Yes
B) Yes, No
C) No, Yes
D) No, No

Hasse diagram

Hasse diagram

A partial order $\preceq$ on A can be drawn using a Hasse diagram.

- Draw nodes: one node for each A
- Draw edges: edge from a to b if $a \preceq b$, except...
- ... *omit* edges that can be inferred by reflexivity
- ... *omit* edges that can be inferred by transitivity
- ... and *layout* nodes “by level“ if $a \preceq b$ for $a \neq b$, then a is placed *lower* than b

Example: isSubstringOf relation on the strings $\{ a, b, c, ab, bc, abc, cd \}$.

Exercise: draw Hasse diagram

Complete the following exercise: on a piece of paper, draw a Hasse diagram for the relation on $A := \{1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60\}$ for the relation $R \subseteq A \times A$ where

$$R := \{\langle x, y\rangle \in A \times A : y \bmod x = 0\}$$

- Draw nodes: one node for each A
- Draw edges: edge from a to b if $a \preceq b$, except...
- ... *omit* edges that can be inferred by reflexivity
- ... *omit* edges that can be inferred by transitivity
- ... and *layout* nodes “by level“ if $a \preceq b$ for $a \neq b$, then a is placed *lower* than b

Example partial order

A to do list,

$$[attendClass, sleep, borrowBook, eat, brushTeeth, study]$$

with constraints:

- $borrowBook \preceq study$
- $study \preceq attendClass$
- $sleep \preceq attendClass$
- $eat \preceq brushTeeth$
- $brushTeeth \preceq sleep$

What should you do *first*? Brush teeth? Eat? Borrow book?

Total order

Relation R is a total order if it is a partial order where every pair is comparable (either $\langle a, b \rangle \in R$ or $\langle b, a \rangle \in R$).

A total order can be written succinctly as an ordered list.

Is previous example a total order?

Topological ordering

Given a partial order $\preceq$, a topological ordering is a total order $\preceq_{total}$ that is *consistent* with $\preceq$.

(See book for formal definition of consistent; see earlier lectures for algorithms for topological sort.)