Topic 3: Consistency[1]
(Version of 10th September 2012)

Pierre Flener

ASTRA Research Group on CP
Uppsala University
Sweden

Course 1DL440:
Constraint Programming

[1] Based on an early version by Christian Schulte (2010)

Outline

Outline

Constraint Problems

Definition (Constraint problems)

A constraint satisfaction problem (CSP) is $\langle V, D, C \rangle$ where:

- $V = [v_1, \ldots, v_m]$ is a finite sequence of variables.
- $D = [D_1, \ldots, D_m]$ is a finite sequence of domains (sets of possible values) for the variables.
- $C = \{c_1, \ldots, c_p\}$ is a finite set of constraints on the variables, a constraint $c(v_{i_1}, \ldots, v_{i_q})$ having arity q.

A constrained optimisation problem (COP) is $\langle V, D, C, f \rangle$:

- The triple $\langle V, D, C \rangle$ is a CSP.
- f is a function from $D_1 \times \cdots \times D_m$ to $\mathbb{R}$ (or $\mathbb{N}$), called the objective function, which is here (without loss of generality) to be minimised.

UPPSALA UNIVERSITET

Definitions

Value Consistency

Domain Consistency

Bounds Consistency

Backtracking and Consistency

Reminders on Discrete Mathematics

Constraint Problems

More on problems

Without loss of generality, we often simplify notation by requiring that all variables initially have the same domain U, called the universe: $D_1 = \cdots = D_m = U$. We then refer to a triple $\langle V, U, C\rangle$ as a CSP, and to $\langle V, U, C, f\rangle$ as a COP.

In this course, we focus on finite domains, and thus also refer to a CSP or COP as a combinatorial problem.

We distinguish a problem from its instances, defined by instance data. Ex: n-Queens vs 8-Queens (for $n = 8$). Some problems have only one instance: grocery problem.

Sometimes, we refer to a single constraint as a CSP.

Constraint Stores

Definition (Constraint store)

The (constraint) store is a function mapping each decision variable of a CSP or COP to its current domain.

Example ($\{x \mapsto \{1,2\}, y \mapsto \{2,3\}\}$ is a store)

Definition (Assigned)

A decision variable x is assigned (or fixed) under store s iff its domain under s is a singleton set: $|s(x)| = 1$.

Notation: dom($\cdot$)

When the name s of the current store is irrelevant, we denote the domain $s(x)$ of a decision variable x by $\text{dom}(x)$.

UPPSALA UNIVERSITET

Definitions

Value Consistency

Domain Consistency

Bounds Consistency

Backtracking and Consistency

Reminders on Discrete Mathematics

Stores and Solutions

Definition (Solution store)

A store s is a solution store to a constraint c iff all domains have single values constituting a solution to c: $s(x_i) = \{d_i\}$ for all $i \in [1, n]$, and $\langle d_1, \ldots, d_n \rangle$ is solution to $c(x_1, \ldots, x_n)$.

Example ($\{x \mapsto \{3\}, y \mapsto \{4\}\}$ solution store to $x \leq y$)

Definition (Solution membership in a store)

A solution $\langle d_1, \ldots, d_n \rangle$ to a constraint $c(x_1, \ldots, x_n)$ is in ($\in$) a store s iff every value belongs to the domain of the corresponding variable: $d_i \in s(x_i)$, for all $i \in [1, n]$.

Example (The solution $\langle 3, 4 \rangle$ to the constraint $x \leq y$ is in the store $\{x \mapsto \{1, 3\}, y \mapsto \{2, 4\}, z \mapsto \{5, 6\}\}$)

Outline

Value Consistency

Example (Value consistency for DISTINCT)

If a variable is assigned, then its value does not appear in the domains of all the other variables of the constraint.
Consider DISTINCT($\{x, y, z\}$) after propagation:

- Store $s = \{x, y \mapsto \{1, 2\}, z \mapsto \{3\}\}$ is value consistent.
- Store $s = \{x, y, z \mapsto \{1, 2\}\}$ is value consistent, hence search is needed to show that there is no solution in s.
- Store $s = \{x, y \mapsto \{1, 2\}, z \mapsto \{1, 2, 3\}\}$ is value consistent, hence search is needed to show that there are two solutions in s, both with $z = 3$.

Enforcing value consistency on DISTINCT($\{x_1, \ldots, x_n\}$) is known as naïve DISTINCT, and takes $\mathcal{O}(n)$ time:

- Store $s = \{w, x, y, z \mapsto \{1, 2, 3\}\}$ is contracted upon $w = 3$ into store $s' = \{w \mapsto \{3\}, x, y, z \mapsto \{1, 2\}\}$.

Value Consistency

Enforcing value consistency

To enforce value consistency for a constraint *c*: whenever a decision variable is assigned a value, any conflicting values according to *c* are removed from the domains of the remaining decision variables.

More about value consistency

In the literature, value consistency (VC) is also known as forward-checking consistency (FCC).

Consistency

Generalities about consistency

We will now study other levels of consistency.

The enforcing (or achieving) of some level of consistency is called propagation and is performed by an algorithm called a propagator: ☞ to be discussed in depth in Topic 4.

Constraints are often equipped with multiple levels of consistency, one being the default, each having different cost of propagation. Typically (but not always), a propagator takes time polynomial in the arity of its constraint.

The modeller must for each constraint (experiment and) choose a suitable level of consistency for the problem at hand and typical instances thereof.

Outline

UPPSALA UNIVERSITET

Domain Consistency

Definition (Domain consistency)

A constraint store s is domain consistent for a constraint c iff for every decision variable x and every value in the domain $s(x)$, there exist values in the domains of the other variables such that all these values form a solution to c.

Example (Domain consistency for DISTINCT($\{x, y, z\}$))

- Store $s = \{x, y, z \mapsto \{1, 2\}\}$ is domain inconsistent, but store $s' = \{x, y, z \mapsto \emptyset\}$ is domain consistent, hence no search is needed to show that there is no solution in s'.
- $\{x, y \mapsto \{1, 2\}, z \mapsto \{1, 2, 3\}\}$ is domain inconsistent, but $\{x, y \mapsto \{1, 2\}, z \mapsto \{3\}\}$ is domain consistent, so no search is needed to show that $z = 3$ in all solutions.

☞ Topic 10: Enforcing domain consistency for DISTINCT.

Domain Consistency

Example (Domain consistency for $x \neq y, y \neq z, z \neq x$)

- $\{x, y, z \mapsto \{1, 2\}\}$ is domain consistent, hence search is needed to show that there is no solution in this store.
- $\{x, y \mapsto \{1, 2\}, z \mapsto \{1, 2, 3\}\}$ is domain consistent, hence search is needed to show $z = 3$ in all solutions.

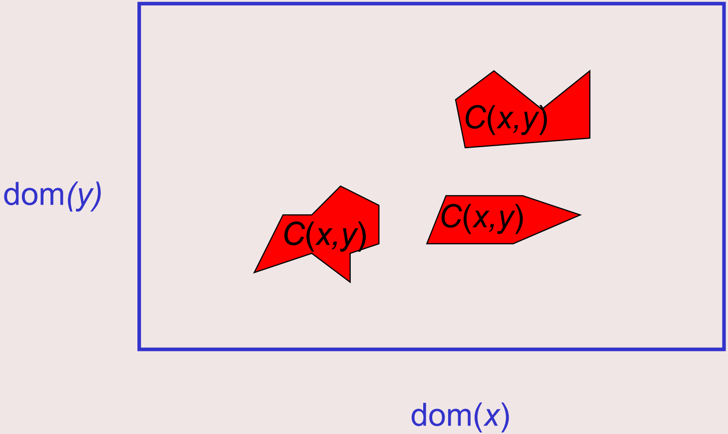
Decomposing $\text{DISTINCT}(\{x_1, \ldots, x_n\})$ into $\frac{n \cdot (n-1)}{2}$ constraints $x_i \neq x_j$ $(1 \leq i < j \leq n)$ yields VC for DISTINCT and requires $\mathcal{O}(n^2)$ space. ☞ Global constraints (Topic 6)!

Example (Domain consistency for $x = 3 \cdot y + 5 \cdot z$)

- $s = \{x \mapsto \{2, \ldots, 7\}, y \mapsto \{0, 1, 2\}, z \mapsto \{-1, \ldots, 2\}\}$ contains the solutions $\langle 3, 1, 0 \rangle$, $\langle 5, 0, 1 \rangle$, and $\langle 6, 2, 0 \rangle$.
- Hence $s' = \{x \mapsto \{3, 5, 6\}, y \mapsto \{0, 1, 2\}, z \mapsto \{0, 1\}\}$ is domain consistent.

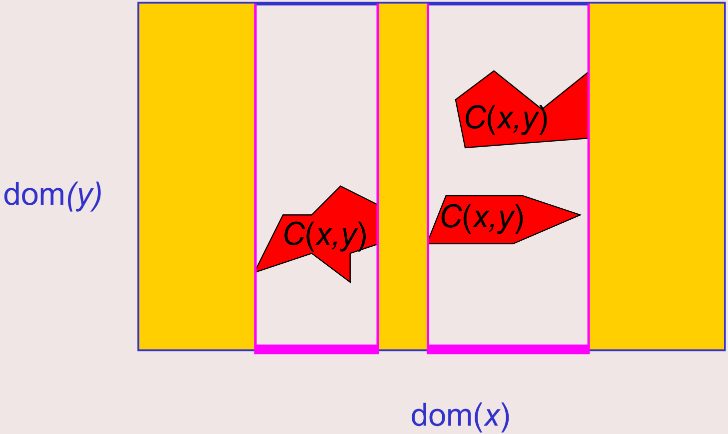
Domain Consistency

Geometric intuition (pictures: © Yves Deville)



Domain Consistency

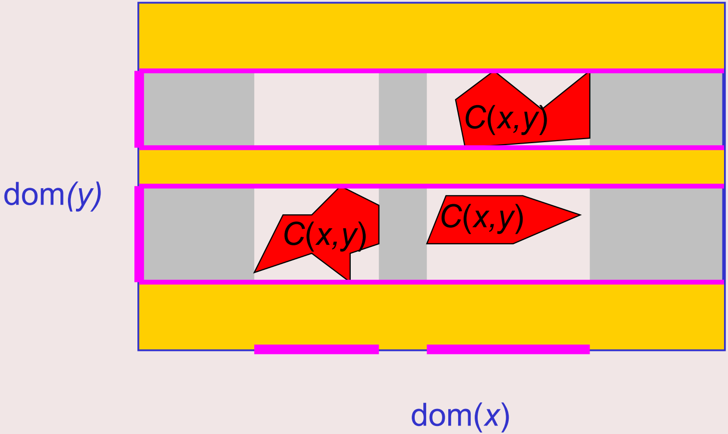
Geometric intuition (pictures: © Yves Deville)



Contracting the domain of *x*

Domain Consistency

Geometric intuition (pictures: © Yves Deville)



Contracting the domain of y

Domain Consistency

More about domain consistency

In the literature, domain consistency (DC) is also known as hyper-arc consistency (HAC) or generalised arc consistency (GAC), and as arc consistency (AC) in the case of binary (arity 2) constraints.

DC is the highest level of consistency (and thus implies VC, for instance), but enforcing it is sometimes prohibitively expensive (for instance on linear arithmetic constraints).

A naïve way to enforce DC for a constraint is to compute its solutions and to project them onto each variable: this is usually impractical! ☞ It is often possible to exploit the combinatorial structure of a constraint in order to enforce DC much faster: global constraints (Topics 6, 10, and 12).

Outline

Bounds Consistency

Example (Consistency for $2 \cdot x = y$)

Consider $s = \{x \mapsto \{1,2,3\}, y \mapsto \{1,2,3,4\}\}$:

- Enforcing DC contracts s to $\{x \mapsto \{1,2\}, y \mapsto \{2,4\}\}$.
- But *Gecode* contracts s to $\{x \mapsto \{1,2\}, y \mapsto \{2,3,4\}\}$!

Definition (Bounds($\mathbb{Z}$) and bounds($\mathbb{R}$) consistencies)

A constraint store s is bounds($\mathbb{Z}$) consistent for a constraint c iff for every decision variable x and the lower and upper bounds of the domain $s(x)$, there exist values between the bounds of the domains of the other variables such that all these values form an integer solution to c.

Similarly for a store being bounds($\mathbb{R}$) consistent.

Bounds Consistency

Definition (Bounds(D) consistency)

A constraint store s is bounds(D) consistent for a constraint c iff for every decision variable x and the lower and upper bounds of the domain $s(x)$, there exist values in the domains of the other variables such that all these values form a solution to c.

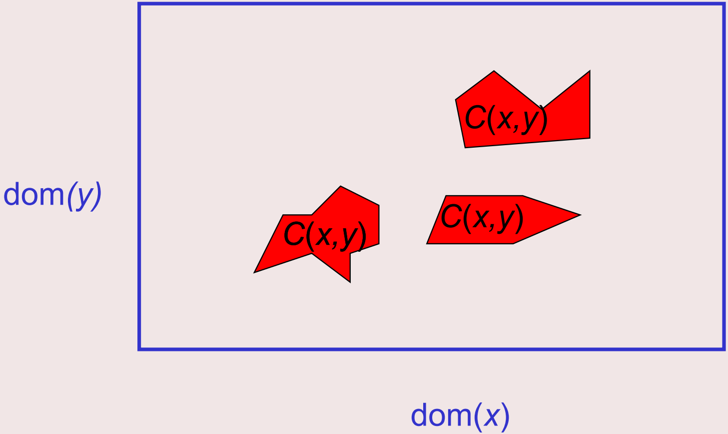
Example (Bounds consistencies for $\max(x, y) = z$)

Consider $s = \{x \mapsto \{2, 3, 5\}, y \mapsto \{3, 4, 6\}, z \mapsto \{4, 6\}\}$:

- Enforcing bounds($\mathbb{Z}$) or bounds($\mathbb{R}$) consistency leaves s unchanged.
- Enforcing bounds(D) consistency contracts s to $\{x \mapsto \{2, 3, 5\}, y, z \mapsto \{4, 6\}\}$.

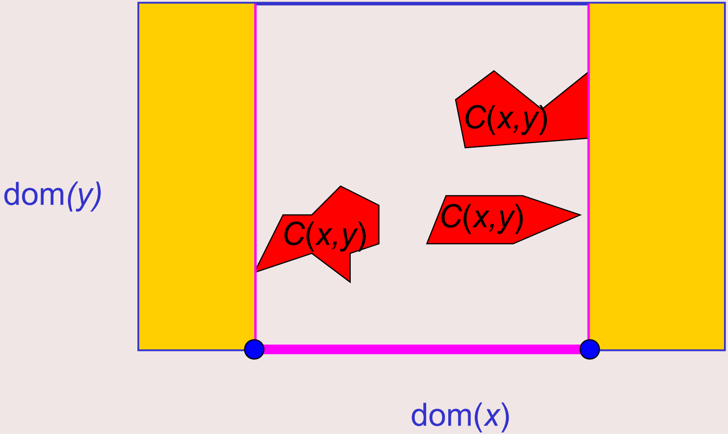
Bounds Consistency

Geometric intuition (pictures: © Yves Deville)



Bounds Consistency

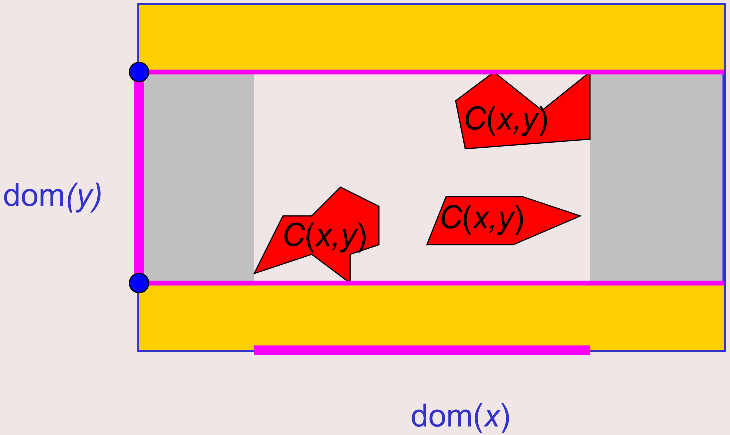
Geometric intuition (pictures: © Yves Deville)



Contracting the domain of *x*

Bounds Consistency

Geometric intuition (pictures: © Yves Deville)



Contracting the domain of *y*

Bounds Consistency

More about bounds consistencies

In the literature, bounds($\mathbb{R}$) consistency, denoted by BC($\mathbb{R}$), is also known as interval consistency. By default, *Gecode* enforces BC($\mathbb{R}$) on arithmetic constraints.

$$\text{DC} \Rightarrow \text{BC}(\text{D}) \Rightarrow \text{VC}$$

$$\text{BC}(\text{D}) \Rightarrow \text{BC}(\mathbb{Z}) \Rightarrow \text{BC}(\mathbb{R})$$

Example (Consistency for SEND + MORE = MONEY)

Enforcing DC on both DISTINCT and the linear equality suffices to solve the problem, without search!
However, this is not faster than search interleaved with enforcing DC on DISTINCT and BC($\mathbb{R}$) on the linear equality, as the problem instance is too small.

Outline

More about Consistency

Terminology

The existentially quantified values in the definitions of DC and BC are called supports. If at least one support exists for a considered value d of a universally quantified decision variable x in those definitions, then d is said to be supported, otherwise d is said to be unsupported.

Other consistencies

Not all propagators enforce VC, BC, or DC (which have simple definitions): there are many useful but unnamed consistency levels that can be enforced.

A pragmatic approach is often taken, contracting domains as much as possible at reasonable cost.

Cost of Consistency Levels

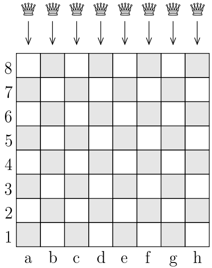
Example (DISTINCT($\{x_1, \ldots, x_n\}$))

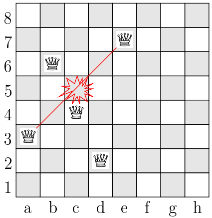
- Value consistency: $\mathcal{O}(n)$ time
- Bounds consistency: $\mathcal{O}(n \cdot \lg n)$ time; often $\mathcal{O}(n)$ time
- Domain consistency: $\mathcal{O}\left(n^{2.5}\right)$ time

Example (Arithmetic on n decision variables)

- Value consistency (useless): $\mathcal{O}(n)$ time
- Bounds consistency: $\mathcal{O}(n)$ time
- Domain consistency: exponential time (NP-hard)

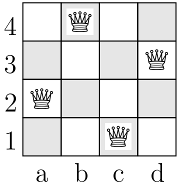
n-Queens Revisited (pics: © Ch. Lecoutre)

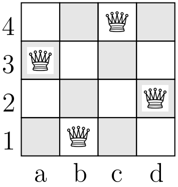




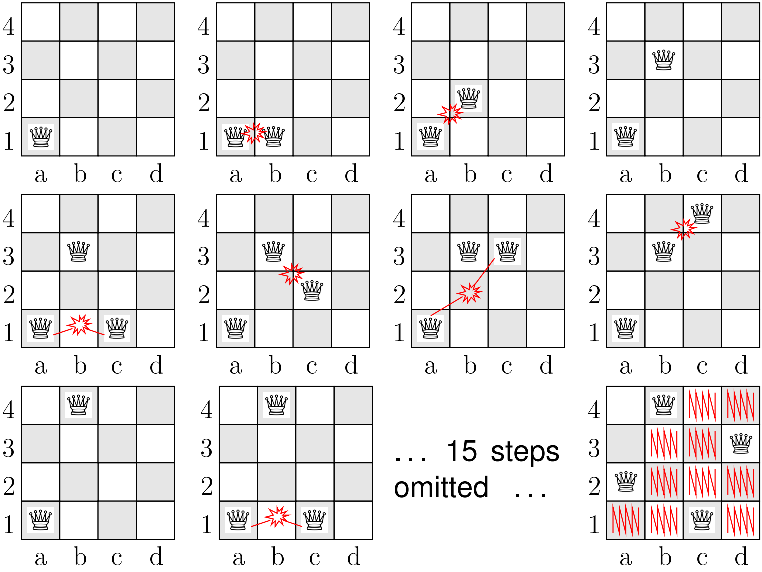
$\textsc{Distinct}(\{q_a, q_b, .., q_h\}), \textsc{Distinct}(\{|q_a-1|, |q_b-2|, .., |q_h-8|\})$

The two solutions to the 4-queens instance:

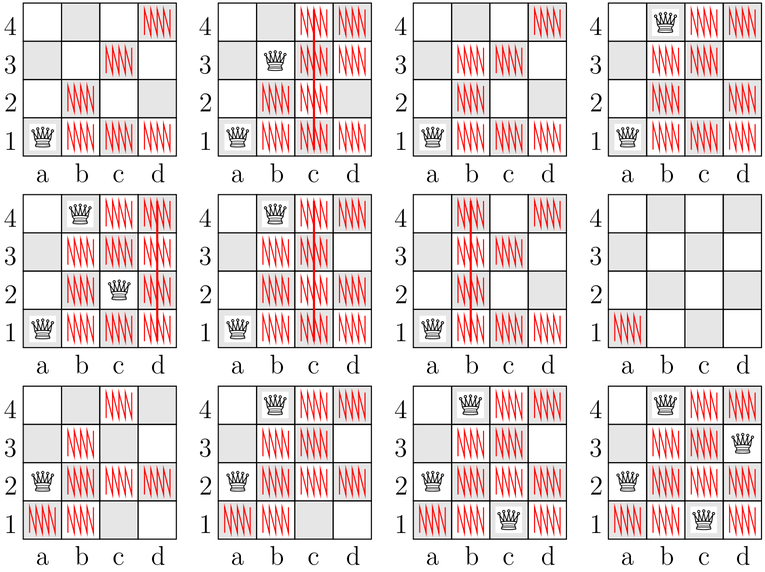




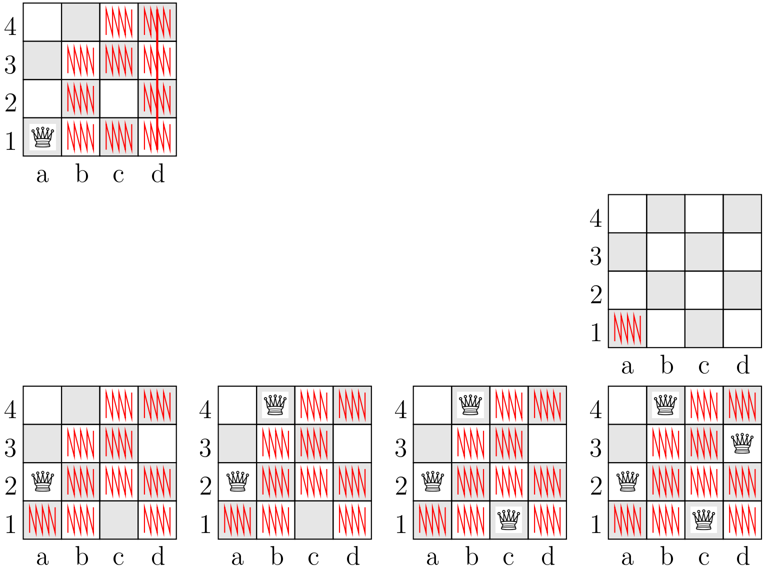
4-Queens: Backtracking Search (BT)



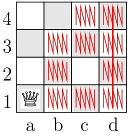
4-Queens: BT + Value Consistency (VC)

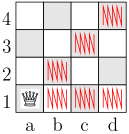


4-Queens: BT + Domain Consistency (DC)



4-Queens: BT + DC (versus BT + VC)

Why 

under DC (versus 

under VC)?

Assume the decision $q_a = 1$ is tried:

1. The DISTINCT($\{q_a, q_b, q_c, q_d\}$) row constraint propagates to $\{q_a \mapsto \{1\}, q_b, q_c, q_d \mapsto \{2, 3, 4\}\}$.
2. The DISTINCT($\{|q_a - 1|, |q_b - 2|, |q_c - 3|, |q_d - 4|\}$) diagonal constraint first propagates (like VC) to $\{q_a \mapsto \{1\}, q_b \mapsto \{3, 4\}, q_c \mapsto \{2, 4\}, q_d \mapsto \{2, 3\}\}$.
3. The previous propagator also notices that q_b cannot be 3 as the domain of q_c would then be wiped out, etc. (This would not happen with two diagonal constraints!)

VC only detects the conflicts between the just assigned variable and the remaining variables, but DC also detects the conflicts between the remaining variables.

Outline

UPPSALA UNIVERSITET

Orders

Definition (Strict partial order)

A strict partial order is a pair $\langle X, \prec \rangle$, where X is a set over which the binary relation $\prec$ is irreflexive ($\forall x \in X : x \not\prec x$) and transitive ($\forall x, y, z \in X : x \prec y \wedge y \prec z \Rightarrow x \prec z$).

Definition (Well-founded order)

A well-founded order is a strict partial order $\langle X, \prec \rangle$ in which there is no infinite decreasing sequence $\cdots \prec x_3 \prec x_2 \prec x_1$.

Definition (Lexicographic order)

Given two well-founded orders $\langle X, \prec_X \rangle$ and $\langle Y, \prec_Y \rangle$, the lexicographic order $\langle X \times Y, \prec_{\text{lex}} \rangle$ is well-founded, where $\langle x_1, y_1 \rangle \prec_{\text{lex}} \langle x_2, y_2 \rangle$ iff either $x_1 \prec_X x_2$ or $x_1 = x_2 \wedge y_1 \prec_Y y_2$. (Similarly for composing more than two (identical) orders.)

Definitions

Value Consistency

Domain Consistency

Bounds Consistency

Backtracking and Consistency

Reminders on Discrete Mathematics

Functions

Definition (Fixpoint)

A fixpoint of a function $f\colon X \to X$ is an element $x \in X$ that does not change under f, that is: $f(x) = x$.

Idempotent functions compute fixpoints:

Definition (Idempotency)

A function f is idempotent iff it is equal to its composition with itself: $\forall x : f(f(x)) = f(x)$.