Introduktion til Domæneteori

1995

Mads Rosendahl
Datalogisk Institut
Københavns Universitet

Disse noter er skrevet til Introduktionskurset i Semantik afholdt første gang i efteråret 1992. Noterne er siden udvidet og visse dele er omarbejdet. Formålet med noterne er at give en introduktion til domæneteori og fixpunktsemantik for derved at give de nødvendige forudsætninger til kursets dele om denotationssemantik og abstrakt fortolkning. I disse noter dækkes fixpunktsætningen, definition af semantik ved brug af fixpunkter, fixpunktinduktion og løsning af rekursive domæneligninger. Noterne er stadig under revision så jeg vil bede læserne om råd til hvordan jeg kan forbedre teksten. Hvis du mener der er noget der er forkert eller uklart så fortæl mig det.

Contents

1 Domæneteori

Fundamentet for vor gennemgang af abstrakt fortolkning, programanalyse, fixpunktsemantik, mm. er en ganske lille teori kaldet *domæneteori*. Vi skal kun bruge to definitioner og en sætning. Udover dette skal vi bruge lidt mængdelære, så man skal helst kunne huske hvad en fællesmængde og en foreningsmængde er for noget. Vi skal faktisk ikke bruge mere, men hvis man ikke har set et matematisk bevis før vil teksten nok være lidt tung i starten.

Det centrale i denne teori er en slagkraftig sætning: *fixpunktsætningen*. Det er det værktøj vi skal bruge for at give et matematisk fundament for fixpunktsemantik og abstrakt fortolkning. Først skal vi dog lige definere to begreber, nemlig hvad vi forstår ved et domæne og hvad det vil sige at en funktion er kontinuert.

1.1 Domæner

Definition. Et par $(\mathbf{D}, \sqsubseteq)$ kaldes et *domæne* hvis og kun hvis

- $\mathbf{D}$ er en mængde.
- $\sqsubseteq$ er en *partiel ordning* over $\mathbf{D}$. Det betyder at det skal være en *refleksiv*, *transitiv* og *antisymmetrisk* relation.

$$
\begin{array}{lll}
\forall \mathbf{x} \in \mathbf{D}. & \mathbf{x} \sqsubseteq \mathbf{x} & \text{—refleksiv} \\
\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{D}. & \mathbf{x} \sqsubseteq \mathbf{y} \wedge \mathbf{y} \sqsubseteq \mathbf{z} \Rightarrow \mathbf{x} \sqsubseteq \mathbf{z} & \text{—transitiv} \\
\forall \mathbf{x}, \mathbf{y} \in \mathbf{D}. & \mathbf{x} \sqsubseteq \mathbf{y} \wedge \mathbf{y} \sqsubseteq \mathbf{x} \Rightarrow \mathbf{x} = \mathbf{y} & \text{—antisymmetrisk}
\end{array}
$$

- $\mathbf{D}$ har et *mindste element* kaldet $\bot_{\mathbf{D}}$. Det betyder at relationen skal opfylde følgende betingelse.

$$\forall \mathbf{x} \in \mathbf{D}.\ \bot_{\mathbf{D}} \sqsubseteq \mathbf{x}$$

- Alle kæder $\mathbf{x}_1 \sqsubseteq \mathbf{x}_2 \sqsubseteq \ldots \sqsubseteq \mathbf{x_i} \sqsubseteq \ldots$ i $\mathbf{D}$ har en *mindste overgrænse* $\bigsqcup_{\mathbf{i}} \mathbf{x_i}$ i $\mathbf{D}$

$$
\begin{array}{ll}
\forall \mathbf{n}.\ \mathbf{x_n} \sqsubseteq \bigsqcup_{\mathbf{i}} \mathbf{x_i} & \text{—overgrænse} \\
\forall \mathbf{d} \in \mathbf{D}.\ (\forall \mathbf{n}.\ \mathbf{x_n} \sqsubseteq \mathbf{d}) \Rightarrow \bigsqcup_{\mathbf{i}} \mathbf{x_i} \sqsubseteq \mathbf{d} & \text{—den mindste}
\end{array}
$$

Definition slut.

Et domæne er altså en ordnet mængde, der nedadtil har et mindste element og opadtil har mindste overgrænser for alle voksende kæder.

På engelsk kaldes et domæne for *domain*, *cpo* eller *pointed cpo*. Forkortelsen *cpo* står for *complete partial order* (eller *chain-complete partial order*). En mindste overgrænse kaldes på engelsk for *least upper bound* eller blot *lub*. Vi vil også omtale overgrænsen som *grænseværdien* af en kæde. Det kan også kaldes *limit* eller *supremum* for kæden. Mindsteelementet kaldes på dansk også for *bundelementet* eller *bund*. På engelsk kaldes det for *bottom*.

Vi skal bruge kæder igen og igen, så lad os straks indføre en notation for dem. For et domæne $\mathbf{D}$ er $\mathsf{kæder}(\mathbf{D})$ mængden af kæder i $\mathbf{D}$ og vi skriver $(\mathbf{x_i}) \in \mathsf{kæder}(\mathbf{D})$ for en kæde $\mathbf{x}_1 \sqsubseteq \mathbf{x}_2 \sqsubseteq \cdots$ i $\mathbf{D}$. En *kæde* er altså en voksende følge af elementer fra et domæne $\mathbf{D}$.

Løsagtig notation. Vi vil bruge symbolet $\perp$ om det mindste element i en række domæner. Hvis der er mulighed for tvivl om hvilket, vil vi bruge den underlæggende mængde som index. Tilsvarende vil $\sqsubseteq$ blive brugt om ordningen i en række domæner når der ikke er tvivl om hvilket domæne vi taler om. Vi vil ofte referere til domænet alene ved navnet på den underlæggende mængde når der ikke er tvivl om hvilken ordning vi antager.

Relationer. Definitionen omtaler relationer som måske ikke tydeligt er en del af mængdelæren fra folkeskolen. Det er dog rimeligt nemt at rette op på. En relation $\mathbf{R}$ mellem elementer i to mængder $\mathbf{A}$ og $\mathbf{B}$ kan opfattes som en delmængde $\mathbf{R} \subseteq (\mathbf{A} \times \mathbf{B})$. Som en notation skriver vi så $\mathbf{a}\ \mathbf{R}\ \mathbf{b}$ i stedet for $\langle \mathbf{a}, \mathbf{b} \rangle \in \mathbf{R}$. Denne infix notation giver bedre en fornemmelse af at det er en relation. Kravene til en partiel ordning om at være en refleksiv, transitiv og antisymmetrisk relation er så bare ekstra betingelser til denne delmængde.

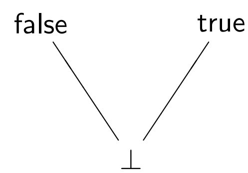
Eksempler. Etpunktsmængden $\mathbb{U} = \{\perp\}$ med ordningen $\perp \sqsubseteq_{\mathbb{U}} \perp$ er et domæne. Et domæne kan ikke være tomt da det skal have et mindste element.

Enhver mængde $\mathbf{S}$ kan udvides til et domæne ved at tilføje et mindste element $\perp$ og benytte en såkaldt "flad" ordning $\sqsubseteq$ således at

$$\forall \mathbf{x}, \mathbf{y} \in \mathbf{S} \cup \{\perp\}.\ \mathbf{x} \sqsubseteq \mathbf{y} \Leftrightarrow \mathbf{x} = \perp \vee \mathbf{x} = \mathbf{y}$$

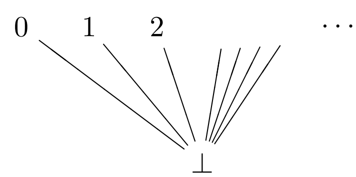
Vi bruger ofte symbolet $\mathbf{S}_\perp$ for domænet med mængden $\mathbf{S} \cup \{\perp\}$ og den flade ordning. Vi siger at et domæne er fladt hvis det har en flad ordning (altså en ordning som opfylder $\forall \mathbf{x}, \mathbf{y}.\ \mathbf{x} \sqsubseteq \mathbf{y} \Leftrightarrow \mathbf{x} = \perp \vee \mathbf{x} = \mathbf{y}$).

Domænet $\mathbb{B}_\perp$ af boolske værdier består af mængden $\{\perp, \mathsf{true}, \mathsf{false}\}$ med en "flad" ordning $\sqsubseteq$ således at $\perp \sqsubseteq \mathsf{false}$ og $\perp \sqsubseteq \mathsf{true}$. Ordningen kan illustreres som en træstruktur



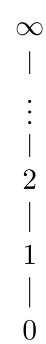
Domænet $\mathbb{B}_\perp$

Domænet $\mathbb{N}_\perp$ består af de naturlige tal $\{0, 1, 2, \ldots\}$ udvidet med et mindste element $\perp$ og en "flad" ordning.



Domænet $\mathbb{N}_\perp$

Parret $(\mathbb{N}, \leq)$ af de naturlige tal med den sædvanlige heltalsordning $0 \leq 1 \leq 2 \leq \cdots$ er *ikke* et domæne selv om det har et mindste element 0. Der findes voksende kæder af naturlige tal, som ikke har en overgrænse. Udvides de naturlige tal med et særligt symbol for uendelig (∞) som er større end alle værdier i $\mathbb{N}$ vil $(\mathbb{N}^\infty, \leq)$ være et domæne.



Domænet $\mathbb{N}^\infty$

1.2 Kontinuitet

Definition. Lad $(\mathbf{D}, \sqsubseteq)$ og $(\mathbf{E}, \sqsubseteq)$ være domæner. En funktion $\mathbf{f} : \mathbf{D} \to \mathbf{E}$ siges at være *kontinuert* hvis og kun hvis

- $\mathbf{f}$ er *monoton*:

$$\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbf{D}.\ \ \mathbf{x}_1 \sqsubseteq \mathbf{x}_2 \Rightarrow \mathbf{f}(\mathbf{x}_1) \sqsubseteq \mathbf{f}(\mathbf{x}_2)$$

- $\mathbf{f}$ er *grænseværdibevarende*

$$\forall (\mathbf{x_i}) \in \mathsf{kæder}(\mathbf{D}).\ \ \mathbf{f}(\bigsqcup_{\mathbf{i}} \mathbf{x_i}) = \bigsqcup_{\mathbf{i}} \mathbf{f}(\mathbf{x_i})$$

Definition slut.

Monotonicitet sikrer at den sidste grænseværdi er veldefineret, idet følgen

$$\mathbf{f}(\mathbf{x}_1) \sqsubseteq \mathbf{f}(\mathbf{x}_2) \sqsubseteq \ldots \sqsubseteq \mathbf{f}(\mathbf{x_i}) \sqsubseteq \ldots$$

er en kæde i $\mathbf{E}$.

Eksempler. Det er rimeligt nemt at finde funktioner, der ikke er monotone. Som et lille eksempel kan vi betragte en funktion $\mathbf{f} : \mathbb{B}_\perp \to \mathbb{B}_\perp$ defineret ved

$$\mathbf{f}(\mathbf{x}) = \textbf{if } \mathbf{x} = \perp \textbf{ then } \mathsf{false} \textbf{ else } \mathsf{true}$$

Det er straks vanskeligere at finde monotone funktioner, der ikke er kontinuerte. For at lave et sådant eksempel kræves det at definitionsmængden har kæder som ikke indeholder deres egen mindste overgrænse. Betragt funktionen $\mathbf{f} : \mathbb{N}^\infty \to \mathbb{B}_\perp$ defineret ved

$$\mathbf{f}(\mathbf{x}) = \textbf{if } \mathbf{x} = \infty \textbf{ then } \mathsf{true} \textbf{ else } \perp$$

Kæden af værdier $0 \leq 1 \leq 2 \leq \cdots$ har grænseværdi ∞ som med $\mathbf{f}$ afbildes til true. Elementerne i kæden afbildes derimod alle til værdien $\perp$.

Hvis $\mathbf{f} : \mathbb{N} \to \mathbb{N}$ er en funktion over de naturlige tal (uden det ekstra mindste element) kan den let udvides til en kontinuert funktion $\mathbf{f'} : (\mathbb{N}_\perp, \sqsubseteq) \to (\mathbb{N}_\perp, \sqsubseteq)$.

$$\mathbf{f'}(\mathbf{x}) = \textbf{if } \mathbf{x} = \perp \textbf{ then } \perp \textbf{ else } \mathbf{f}(\mathbf{x})$$

For at overbevise sig om det skal man blot konstatere at der kun er to mulige typer kæder i $(\mathbb{N}_\perp,\ \sqsubseteq)$; enten er alle elementerne mindste elementet $\perp$ eller også vil kæden fra et trin antage en værdi i $\mathbb{N}$ og derefter være konstant.

En *partiel funktion* $\mathbf{f} : \mathbb{N} \to \mathbb{N}$, dvs. en funktion der ikke er defineret for alle argumenter kan udvides på samme måde til en total kontinuert funktion $\mathbf{f'} : (\mathbb{N}_\perp,\ \sqsubseteq) \to (\mathbb{N}_\perp,\ \sqsubseteq)$ ved at lade udefineret repræsentere ved bundelementet $\perp$.

Notation. Fra den lidt mere filosofiske side kan man opfatte mængder som noget der lever i den matematiske fantasiverden. Når vi taler om $\mathbb{B}_\perp$ er det hverken blækket på papiret eller navnet, der er mængden. $\mathbb{B}_\perp$ er et navn for en given mængde, vi tror på eksisterer i denne fantasiverden (mere formelt tror vi på Zermelo-Fraenkels aksiomssystem for mængdelæren).

Tilsvarende kan man spørge sig selv hvad en funktion er. Ligesom med relationer kan funktioner udtrykkes i mængdelæren som delmængder. En funktion $\mathbf{f} : \mathbf{A} \to \mathbf{B}$ fra en mængde $\mathbf{A}$ til en mængde $\mathbf{B}$ er en delmængde $\mathbf{f} \subseteq (\mathbf{A} \times \mathbf{B})$ som opfylder

$$\forall \mathbf{x} \in \mathbf{A}.\ \exists \langle \mathbf{z}, \mathbf{v} \rangle \in \mathbf{f}.\ \mathbf{x} = \mathbf{z}$$
$$\forall \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbf{f}.\ \forall \langle \mathbf{z}, \mathbf{v} \rangle \in \mathbf{f}.\ \mathbf{z} = \mathbf{x} \Rightarrow \mathbf{v} = \mathbf{y}$$

Med andre ord, funktionen skal være total og have en entydig værdi for alle argumenter. Som med relationer bruger vi en lidt pudsig notation (i forhold til mængdelæren) når vi skriver $\mathbf{f}(\mathbf{x}) = \mathbf{y}$ i stedet for $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbf{f}$.

Når vi beskriver funktioner som

$$\mathbf{f}(\mathbf{x}) = \mathbf{if}\ \mathbf{x} = \infty\ \mathbf{then}\ \mathsf{true}\ \mathbf{else}\ \perp$$

er det som med mængder hverken blækket eller teksten der er funktionen, men det er navnet på en funktion vi også kunne have beskrevet ved

$$\mathbf{f} = \{\langle \infty, \mathsf{true} \rangle\} \cup \{\langle \mathbf{x}, \perp \rangle \mid \mathbf{x} \in \mathbb{N}\}$$

Begge definitioner er navne på den samme funktion (delmængde) i mængdelæren. Vi vil ikke holde os tilbage med forskellige måder at definere funktioner på, så længe vi altid kan gå tilbage og kontrollere fundamentet. Når vi skal definere højere-ordens funktioner er det ofte nemmest at bruge lambda-notation og visse funktioner som "+" og "∗" antager vi blot alle ved hvad er.

Scott-topologi. Kontinuitet, som defineret her, minder måske ikke så meget om kontinuitet nogle måske kender fra metriske rum, men der er en sammenhæng.

Lad $\mathbf{D}$ være et domæne, og definer for en delmængde $\mathbf{A} \subseteq \mathbf{D}$, at

$$\begin{aligned} \mathbf{A} \text{ er åben } \Leftrightarrow\ & (\forall \mathbf{x}, \mathbf{y} \in \mathbf{D}.\ \mathbf{x} \in \mathbf{A} \wedge \mathbf{x} \sqsubseteq \mathbf{y} \Rightarrow \mathbf{y} \in \mathbf{A}) \wedge \\ & (\forall (\mathbf{x_i}) \in \mathsf{k\ae der}(\mathbf{D}).\ \bigsqcup_{\mathbf{i}} \mathbf{x_i} \in \mathbf{A} \Rightarrow \{\mathbf{x_i}\} \cap \mathbf{A} \neq \varnothing \end{aligned}$$

Man kan vise, at $\mathbf{U} = \{\mathbf{A} \subseteq \mathbf{D} \mid \mathbf{A} \text{ er åben}\}$ da er et system af åbne mængder, og at det for to domæner $\mathbf{D}$ og $\mathbf{E}$ gælder, at de kontinuerte funktioner fra $\mathbf{D}$ til $\mathbf{E}$ i domæneteoretisk forstand er de samme som de kontinuerte funktioner i topologisk forstand.

1.3 Fixpunktsætning

Et *fixpunkt* for en funktion $\mathbf{f} : \mathbf{D} \to \mathbf{D}$ er et element $\mathbf{d} \in \mathbf{D}$ så $\mathbf{f}(\mathbf{d}) = \mathbf{d}$. Den følgende sætning er årsagen til at vi overhovedet beskæftiger os med domæneteori.

Fixpunktsætning. En kontinuert funktion over et domæne $(\mathbf{D}, \sqsubseteq)$

$$\mathbf{f} : (\mathbf{D}, \sqsubseteq) \to (\mathbf{D}, \sqsubseteq)$$

har et *mindste fixpunkt* der kan findes som

$$\bigsqcup_{\mathbf{i}} \mathbf{f}^{\mathbf{i}}(\perp)$$

Bevis. Beviset har tre dele

- Veldefineret: $\perp \sqsubseteq \mathbf{f}(\perp) \sqsubseteq \mathbf{f}^2(\perp) \sqsubseteq \ldots$
- Et fixpunkt: $\mathbf{f}(\bigsqcup_{\mathbf{i}} \mathbf{f}^{\mathbf{i}}(\perp)) = \bigsqcup_{\mathbf{i}} \mathbf{f}^{\mathbf{i}}(\perp)$
- Det mindste: $\forall \mathbf{d} \in \mathbf{D}.\ \mathbf{f}(\mathbf{d}) = \mathbf{d} \Rightarrow \bigsqcup_{\mathbf{i}} \mathbf{f}^{\mathbf{i}}(\perp) \sqsubseteq \mathbf{d}$

<u>Veldefineret.</u> Da $\perp$ er mindste element er $\perp \sqsubseteq \mathbf{f}(\perp)$. Da $\mathbf{f}$ er monoton er $\mathbf{f}(\perp) \sqsubseteq \mathbf{f}^2(\perp)$. Resten følger ved induktion.

<u>Fixpunkt.</u> Vi har at $\mathbf{f}(\bigsqcup_{\mathbf{i}} \mathbf{f}^{\mathbf{i}}(\perp)) = \bigsqcup_{\mathbf{i}} \mathbf{f}(\mathbf{f}^{\mathbf{i}}(\perp))$ da $\mathbf{f}$ er kontinuert. Ligeledes er $\bigsqcup_{\mathbf{i}} \mathbf{f}(\mathbf{f}^{\mathbf{i}}(\perp)) = \bigsqcup_{\mathbf{i}} \mathbf{f}^{\mathbf{i}}(\perp)$ da begge værdier er overgrænse for den andens følge - og derfor den mindste overgrænse. (Se også hjælpesætningen senere).

<u>Det mindste.</u> Antag $\mathbf{f}(\mathbf{d}) = \mathbf{d}$. Ved induktion følger $\mathbf{f}^{\mathbf{i}}(\perp) \sqsubseteq \mathbf{d}$ for alle $\mathbf{i}$: $\perp \sqsubseteq \mathbf{d} \Rightarrow \mathbf{f}(\perp) \sqsubseteq \mathbf{f}(\mathbf{d}) \Rightarrow \mathbf{f}(\perp) \sqsubseteq \mathbf{d}$. Værdien $\mathbf{d}$ er derfor en overgrænse for $\mathbf{f}^{\mathbf{i}}(\perp)$.

Bevis slut.

Notation. Det at finde fixpunkter for kontinuerte funktioner er så nyttig en ting at vi straks vil indføre en særlig notation. Sætningen siger at der for alle domæner $\mathbf{D}$ findes en funktion $\mathsf{fix}_{\mathbf{D}} : (\mathbf{D} \rightarrow_{\mathbf{c}} \mathbf{D}) \rightarrow \mathbf{D}$ som finder det mindste fixpunkt for en kontinuert funktion. Vi skal senere se at funktionen fix også selv er kontinuert.

En funktion kan godt have flere fixpunkter, men sætningen siger altså at mængden af fixpunkter vil have et mindste element. Det er ikke givet at en funktion også har et største fixpunkt, men hvis den har kaldes det ofte for $\mathsf{gfp}(\mathbf{f})$ (for *greatest fixed point*) i modsætning til $\mathsf{fix}(\mathbf{f}) = \mathsf{lfp}(\mathbf{f})$ (for *least fixed point*).

Kæden $\bot \sqsubseteq \mathbf{f}(\bot) \sqsubseteq \mathbf{f}^2(\bot) \sqsubseteq \ldots$ kaldes på engelsk for *the ascending Kleene chain* (eller *AKC*).

Eksempler. Måske er det mest overraskende ved sætningen at alle kontinuerte funktioner har et fixpunkt, altså en værdi der afbildes til sig selv. Man kan let spørge sig selv om der ikke findes modeksempler hvor alle værdier afbildes til noget forskelligt. For eksempel kunne man tage efterfølgerfunktionen over $\mathbb{N}_\bot$ som afbilder $\bot$ til 0, 0 til 1 osv. Denne funktion er imidlertid ikke monoton, men selvfølgelig burde beviset også afskrække en fra at stille sådanne spørgsmål.

1.4 Blandede sætninger

For at vænne os lidt til hvordan man kan jonglere rundt med kæder og fixpunkter vil vi lige vise et par småsætninger.

Sætning. For kæder $(\mathbf{a_i}) \in \mathsf{kæder}(\mathbf{D})$ og $(\mathbf{b_i}) \in \mathsf{kæder}(\mathbf{D})$ gælder

$$(\forall \mathbf{i}.\ \exists \mathbf{j}.\ \mathbf{a_i} \sqsubseteq \mathbf{b_j}) \Rightarrow \bigsqcup_{\mathbf{i}} \mathbf{a_i} \sqsubseteq \bigsqcup_{\mathbf{j}} \mathbf{b_j}$$

Bevis. Antag at der for kæder $(\mathbf{a_i})$ og $(\mathbf{b_j})$ gælder $\forall \mathbf{i}.\ \exists \mathbf{j}.\ \mathbf{a_i} \sqsubseteq \mathbf{b_j}$, da gælder også $\forall \mathbf{i}.\ \mathbf{a_i} \sqsubseteq \bigsqcup_{\mathbf{j}} \mathbf{b_j}$ eftersom $\bigsqcup_{\mathbf{j}} \mathbf{b_j}$ er en overgrænse for all $\mathbf{b_j}$ værdier. Dette betyder at $\bigsqcup_{\mathbf{j}} \mathbf{b_j}$ er en overgrænse for $\mathbf{a_i}$ værdierne og derfor større eller lig den mindste overgrænse.

Bevis slut.

Sætning. Lad $(\mathbf{a_{ij}})$ være en*dobbeltindiceret kæde* i et domæne $\mathbf{D}$ således at

$$\mathbf{i} \leq \mathbf{i'} \wedge \mathbf{j} \leq \mathbf{j'} \Rightarrow \mathbf{a_{ij}} \sqsubseteq \mathbf{a_{i'j'}}$$

Der gælder da

$$\bigsqcup_{\mathbf{i}} \mathbf{a_{ii}} = \bigsqcup_{\mathbf{i}} \bigsqcup_{\mathbf{j}} \mathbf{a_{ij}} = \bigsqcup_{\mathbf{j}} \bigsqcup_{\mathbf{i}} \mathbf{a_{ij}}$$

Bevis. Beviset har to dele:

<u>Grænseværdierne er veldefinerede.</u> Vi skal sikre os at $\bigsqcup_{\mathbf{j}} \mathbf{a_{ij}}$ er en kæde i den variable $\mathbf{i}$, altså at for alle $\mathbf{i}$ at $\bigsqcup_{\mathbf{j}} \mathbf{a_{ij}} \sqsubseteq \bigsqcup_{\mathbf{j}} \mathbf{a_{(i+1)j}}$. Hertil kan vi bruge sætningen ovenfor idet $\mathbf{a_{ij}} \sqsubseteq \mathbf{a_{(i+1)j}}$ for alle $\mathbf{j}$.

<u>Ækvivalens.</u> Vi skal vise at $\bigsqcup_{\mathbf{i}} \mathbf{a_{ii}} = \bigsqcup_{\mathbf{i}} \bigsqcup_{\mathbf{j}} \mathbf{a_{ij}}$. Beviset har to dele idet vi vil udnytte antisymmetrien af ækvivalensen.
"$\sqsubseteq$": Vi skal bruge at for alle $\mathbf{i}$ er $\mathbf{a_{ii}} \sqsubseteq \bigsqcup_{\mathbf{j}} \mathbf{a_{ij}}$. Dette følger af at $\mathbf{a_{ij}}$ med fast $\mathbf{i}$ er en kæde.
"$\sqsupseteq$": Vi skal bruge at der for ethvert $\mathbf{k}$ gælder at $\bigsqcup_{\mathbf{i}} \mathbf{a_{ii}} \sqsupseteq \bigsqcup_{\mathbf{j}} \mathbf{a_{kj}}$. Givet et $\mathbf{m}$, da har vi at $\mathbf{a_{km}} \sqsubseteq \mathbf{a_{max(k,\,m)max(k,\,m)}} \sqsubseteq \bigsqcup_{\mathbf{i}} \mathbf{a_{ii}}$. Det betyder at $\bigsqcup_{\mathbf{i}} \mathbf{a_{ii}}$ er en overgrænse for kæden $\bigsqcup_{\mathbf{j}} \mathbf{a_{kj}}$ med variabel $\mathbf{k}$ og derfor større eller lig den mindste overgrænse.

Bevis slut

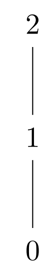
Kommentar. En sætning som den sidste giver et indtryk af hvor stabil domæneteorien er. Man kan bytte rundt på grænseværdioperationer og flytte dem ind og ud af kontinuerte funktioner uden at det ændrer værdien. Man skal faktisk anstrenge sig en hel del før noget går galt.

Historiske noter. Domæneteorien er en forholdsvis ung teori inden for matematikken. Den er startet som en del af gitterteorien (*eng. lattice theory*), der især er blevet behandlet af Garrett Brikhoff omkring 1940erne. Fixpunktsætningen for gitre blev udledt af Tarski og publiceret i 1955. Beviset med brug af en kæde skyldes Kleene (1952). Brugen af gitre til at beskrive semantik af programmeringssprog skyldes Dana Scott og Christopher Strachey med en række arbejder fra 1970 og 1971. I arbejder fra 70erne blev ordet "domæne" ofte brugt om gitre, men fra 80erne er det blevet almindeligt kun at kræve eksistens af grænseværdier for kæder. Resultatet om løsning af rekursive domæneligninger skyldes Dana Scott (1970).

Gitre. Et gitter er en partielt ordnet mængde hvor hvert par af værdier har en mindste overgrænse og en største undergrænse. I et komplet gitter har enhver delmængde en mindste overgrænse og en største undergrænse. Tarskis fixpunktsætning siger at mængden af fixpunkter for en monoton funktion på et komplet gitter udgør et komplet delgitter, og altså har et mindste element.

1.5 Opgaver

Opgave 1.1. Domænet 3 består af elementerne 0, 1 og 2 med en ordning, der kan illustreres således.



Domænet 3

Angiv de mulige kontinuerte funktioner fra 3 til 3. For hver funktion angiv desuden dens mindste fixpunkt.

Opgave 1.2. Højden af et domæne er længden af den længste skarpt voksende følge i domænet. (Den samme værdi forekommer ikke to gange i følgen). Vis at en monoton funktion $\mathbf{A} \rightarrow_{\mathbf{m}} \mathbf{B}$ hvor $\mathbf{A}$ har endelig højde også vil være kontinuert.

Opgave 1.3. Konstruer et eksempel på et domæne som ikke har endelig højde men hvor ingen kæde kan have uendeligt mange forskellige elementer.

Opgave 1.4. En funktion $\mathbf{f} : \mathbf{D} \rightarrow \mathbf{D}$ siges at være ekstensiv hvis den opfylder $\forall \mathbf{d} \in \mathbf{D}.\ \mathbf{d} \sqsubseteq \mathbf{f}(\mathbf{d})$. Vis at en ekstensiv funktion på et domæne $\mathbf{D}$ har et fixpunkt. $\mathbf{D}$ kan antages at være tællelig.

2 Domænekonstruktioner

Vi så i sidste afsnit at domæneteorien giver stor fleksibilitet i brugen af grænseværdioperationer. Vi skal nu se at det samme gælder når vi vil danne sammensatte domæner. Domæneegenskaben synes at være forbløffende stabil ved en lang række mængdekonstruktioner.

2.1 Cartesisk produkt

Definition. For to domæner $(\mathbf{A}, \sqsubseteq_\mathbf{A})$ og $(\mathbf{B}, \sqsubseteq_\mathbf{B})$ kan vi definere det *cartesiske produkt* $(\mathbf{A} \times \mathbf{B}, \sqsubseteq_{\mathbf{A}\times\mathbf{B}})$ som domænet af tupler (par) af værdier fra mængderne $\mathbf{A}$ og $\mathbf{B}$.

$$\mathbf{A} \times \mathbf{B} = \{\langle \mathbf{a}, \mathbf{b} \rangle \mid \mathbf{a} \in \mathbf{A} \wedge \mathbf{b} \in \mathbf{B}\}$$
$$\langle \mathbf{a}_1, \mathbf{b}_1 \rangle \sqsubseteq_{\mathbf{A}\times\mathbf{B}} \langle \mathbf{a}_2, \mathbf{b}_2 \rangle \Leftrightarrow \mathbf{a}_1 \sqsubseteq_\mathbf{A} \mathbf{a}_2 \wedge \mathbf{b}_1 \sqsubseteq_\mathbf{B} \mathbf{b}_2$$

Ordningen kaldes ofte en *elementvis ordning* af produktet.

Sætning. Det cartesiske produkt af to domæner er et domæne.

Bevis. Vi skal vise følgende punkter

- Relationen $\sqsubseteq_{\mathbf{A}\times\mathbf{B}}$ er en partiel ordning.
- Der findes et mindste element i $\mathbf{A} \times \mathbf{B}$.
- Det er muligt at finde grænseværdier for kæder i $\mathbf{A} \times \mathbf{B}$.

De fleste af disse punkter er ganske trivielle.

<u>Ordning.</u> $\sqsubseteq_{\mathbf{A}\times\mathbf{B}}$ refleksiv: $\langle \mathbf{a}, \mathbf{b} \rangle \sqsubseteq \langle \mathbf{a}, \mathbf{b} \rangle$ fordi $\mathbf{a} \sqsubseteq \mathbf{a} \wedge \mathbf{b} \sqsubseteq \mathbf{b}$.

$\sqsubseteq_{\mathbf{A}\times\mathbf{B}}$ transitiv: hvis $\langle \mathbf{a}_1, \mathbf{b}_1 \rangle \sqsubseteq \langle \mathbf{a}_2, \mathbf{b}_2 \rangle \wedge \langle \mathbf{a}_2, \mathbf{b}_2 \rangle \sqsubseteq \langle \mathbf{a}_3, \mathbf{b}_3 \rangle$ er $\mathbf{a}_1 \sqsubseteq \mathbf{a}_2 \wedge \mathbf{b}_1 \sqsubseteq \mathbf{b}_2 \wedge \mathbf{a}_2 \sqsubseteq \mathbf{a}_3 \wedge \mathbf{b}_2 \sqsubseteq \mathbf{b}_3$ så $\mathbf{a}_1 \sqsubseteq \mathbf{a}_3 \wedge \mathbf{b}_1 \sqsubseteq \mathbf{b}_3$ hvilket medfører $\langle \mathbf{a}_1, \mathbf{b}_1 \rangle \sqsubseteq \langle \mathbf{a}_3, \mathbf{b}_3 \rangle$.

$\sqsubseteq_{\mathbf{A}\times\mathbf{B}}$ antisymmetrisk: hvis $\langle \mathbf{a}_1, \mathbf{b}_1 \rangle \sqsubseteq \langle \mathbf{a}_2, \mathbf{b}_2 \rangle \wedge \langle \mathbf{a}_2, \mathbf{b}_2 \rangle \sqsubseteq \langle \mathbf{a}_1, \mathbf{b}_1 \rangle$ er $\mathbf{a}_1 \sqsubseteq \mathbf{a}_2 \wedge \mathbf{b}_1 \sqsubseteq \mathbf{b}_2 \wedge \mathbf{a}_2 \sqsubseteq \mathbf{a}_1 \wedge \mathbf{b}_2 \sqsubseteq \mathbf{b}_1$ så $\mathbf{a}_1 = \mathbf{a}_2 \wedge \mathbf{b}_1 = \mathbf{b}_2$ og $\langle \mathbf{a}_1, \mathbf{b}_1 \rangle = \langle \mathbf{a}_2, \mathbf{b}_2 \rangle$.

<u>Mindste element.</u> For alle $\langle \mathbf{a}, \mathbf{b} \rangle$ gælder $\perp_\mathbf{A} \sqsubseteq \mathbf{a} \wedge \perp_\mathbf{B} \sqsubseteq \mathbf{b}$ så $\langle \perp_\mathbf{A}, \perp_\mathbf{B} \rangle \sqsubseteq \langle \mathbf{a}, \mathbf{b} \rangle$.

<u>Mindste overgrænse.</u> For en kæde $(\langle \mathbf{a_i}, \mathbf{b_i} \rangle) \in \mathsf{kæder}(\mathbf{A} \times \mathbf{B})$ kan vi konstruere den mindste overgrænse $\langle \bigsqcup_\mathbf{i} \mathbf{a_i}, \bigsqcup_\mathbf{i} \mathbf{b_i} \rangle$ da $\langle \mathbf{a_i}, \mathbf{b_i} \rangle \sqsubseteq \langle \bigsqcup_\mathbf{i} \mathbf{a_i}, \bigsqcup_\mathbf{i} \mathbf{b_i} \rangle$ og

hvis $\langle \mathbf{a_i}, \mathbf{b_i} \rangle \sqsubseteq \langle \mathbf{c}, \mathbf{d} \rangle$ for alle i er $\bigsqcup_\mathbf{i} \mathbf{a_i} \sqsubseteq \mathbf{c}$ og $\bigsqcup_\mathbf{i} \mathbf{b_i} \sqsubseteq \mathbf{d}$, så $\langle \bigsqcup_\mathbf{i} \mathbf{a_i}, \bigsqcup_\mathbf{i} \mathbf{b_i} \rangle \sqsubseteq \langle \mathbf{c}, \mathbf{d} \rangle$.

Bevis slut.

Notation. Projektionerne $\mathsf{fst} : \mathbf{A} \times \mathbf{B} \to \mathbf{A}$ og $\mathsf{snd} : \mathbf{A} \times \mathbf{B} \to \mathbf{B}$ er for alle $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbf{A} \times \mathbf{B}$ defineret ved,

$$\begin{aligned} \mathsf{fst}(\langle \mathbf{x}, \mathbf{y} \rangle) &= \mathbf{x} \\ \mathsf{snd}(\langle \mathbf{x}, \mathbf{y} \rangle) &= \mathbf{y} \end{aligned}$$

Vi vil også ofte bruge en nedadrettet pil til at betyde udvælgelse af elementer fra tupler. Dette betyder at

$$\langle \mathbf{x}, \mathbf{y} \rangle \downarrow 1 = \mathbf{x} \quad \text{og} \quad \langle \mathbf{x}, \mathbf{y} \rangle \downarrow 2 = \mathbf{y}$$

For funktioner fra et cartesisk produkt udelader vi ofte de kantede parenteser ved funktionskald. Vi kunne således også skrive $\mathsf{fst}(\mathbf{x}, \mathbf{y}) = \mathbf{x}$.

Sætning. En funktion $\mathbf{A} \times \mathbf{B} \to \mathbf{C}$ i to argumenter er kontinuert hvis og kun hvis den er kontinuert i hvert argument for sig.

Bevis. "Kun hvis" delen følger let ved at betragte kæder hvor den ene komponent er konstant.
"Hvis". Lad $\mathbf{f} : \mathbf{A} \times \mathbf{B} \to \mathbf{C}$ være kontinuert i hvert argument for sig og lad $(\langle \mathbf{x_i}, \mathbf{y_i} \rangle) \in \mathsf{kæder}(\mathbf{A} \times \mathbf{B})$. Da er $(\mathbf{x_i})$ en kæde i A og $(\mathbf{y_i})$ en kæde i B Af monotoniciteten af $\mathbf{f}$ følger da at $(\mathbf{f}(\mathbf{x_i}, \mathbf{y_j}))$ er en dobbeltindiceret kæde i $\mathbf{C}$ hvorfor vi kan bruge hjælpesætningen fra sidste afsnit. En udregning giver da

$$\begin{aligned} \mathbf{f}(\sqcup_\mathbf{i} \langle \mathbf{x_i}, \mathbf{y_i} \rangle) &= \mathbf{f}(\sqcup_\mathbf{i} \mathbf{x_i},\ \sqcup_\mathbf{j} \mathbf{y_j}) \\ &= \sqcup_\mathbf{i} \mathbf{f}(\mathbf{x_i},\ \sqcup_\mathbf{j} \mathbf{y_j}) \\ &= \sqcup_\mathbf{i} \sqcup_\mathbf{j} \mathbf{f}(\mathbf{x_i}, \mathbf{y_j}) \\ &= \sqcup_\mathbf{i} \mathbf{f}(\mathbf{x_i}, \mathbf{y_i}) \end{aligned}$$

Bevis slut.

Kommentar. Den tilsvarende egenskab gælder ikke kontinuitet i metriske rum. Som et modeksempel betragt

$$\mathbf{p}(\mathbf{x}, \mathbf{y}) = \mathbf{if}\ \mathbf{x} = \mathbf{y} = 0\ \mathbf{then}\ 0\ \mathbf{else}\ (\mathbf{x} * \mathbf{y})/(\mathbf{x}^2 + \mathbf{y}^2)$$

I domæneteorien er det sjældent det store problem at overbevise sig om at en funktion er kontinuert.

Strikt produkt. Det cartesiske produkt kaldes af og til for det dovne produkt (*lazy product*) i modsætning til det strikse produkt (*strict product* eller *smashed product*) $\mathbf{A} \otimes \mathbf{B} = \{\langle \perp_{\mathbf{A}}, \perp_{\mathbf{B}} \rangle\} \cup (\mathbf{A} \setminus \{\perp_{\mathbf{A}}\}) \times (\mathbf{B} \setminus \{\perp_{\mathbf{B}}\})$. Også dette er et domæne med elementvis ordning.

2.2 Sumdomæne

For to ordnede mængder $(\mathbf{A}, \sqsubseteq_{\mathbf{A}})$ og $(\mathbf{B}, \sqsubseteq_{\mathbf{B}})$ kan vi konstruere summen (*disjoint union*) $(\mathbf{A}+\mathbf{B}, \sqsubseteq_{\mathbf{A}+\mathbf{B}})$ som

$$
\begin{array}{ll}
\mathbf{A}+\mathbf{B} = \{\perp_{\mathbf{A}+\mathbf{B}}\} \cup \{\langle 1, \mathbf{a}\rangle \mid \mathbf{a} \in \mathbf{A}\} \cup \{\langle 2, \mathbf{b}\rangle \mid \mathbf{b} \in \mathbf{B}\} & \\
\perp_{\mathbf{A}+\mathbf{B}} \sqsubseteq_{\mathbf{A}+\mathbf{B}} \mathbf{x} & \forall \mathbf{x} \in \mathbf{A}+\mathbf{B} \\
\langle 1, \mathbf{a}_1\rangle \sqsubseteq_{\mathbf{A}+\mathbf{B}} \langle 1, \mathbf{a}_2\rangle & \Leftrightarrow \mathbf{a}_1 \sqsubseteq_{\mathbf{A}} \mathbf{a}_2 \\
\langle 2, \mathbf{b}_1\rangle \sqsubseteq_{\mathbf{A}+\mathbf{B}} \langle 2, \mathbf{b}_2\rangle & \Leftrightarrow \mathbf{b}_1 \sqsubseteq_{\mathbf{B}} \mathbf{b}_2
\end{array}
$$

Sætning. Summen af to domæner er også et domæne.

Bevis. Atter består beviset af en række dele. Denne gang overlader vi dog dele af beviset som en opgave til læseren.

<u>Ordning.</u> Se opgaverne.

<u>Mindste element.</u> Følger af definitionen.

<u>Grænseværdi.</u> Der er tre muligheder for kæder i $\mathbf{A}+\mathbf{B}$.

$$
\begin{array}{l}
\perp_{\mathbf{A}+\mathbf{B}} \sqsubseteq \perp_{\mathbf{A}+\mathbf{B}} \sqsubseteq \cdots \\
\perp_{\mathbf{A}+\mathbf{B}} \sqsubseteq \cdots \sqsubseteq \perp_{\mathbf{A}+\mathbf{B}} \sqsubseteq \langle 1, \mathbf{a}_1\rangle \sqsubseteq \langle 1, \mathbf{a}_2\rangle \ldots \\
\perp_{\mathbf{A}+\mathbf{B}} \sqsubseteq \cdots \sqsubseteq \perp_{\mathbf{A}+\mathbf{B}} \sqsubseteq \langle 2, \mathbf{b}_1\rangle \sqsubseteq \langle 2, \mathbf{b}_2\rangle \ldots
\end{array}
$$

I det første tilfælde er $\perp_{\mathbf{A}+\mathbf{B}}$ en overgrænse for kæden. I de andre tilfælde udgør andenkomponenten fra et vist trin en kæde i henholdsvis $\mathbf{A}$ og $\mathbf{B}$. Da $\mathbf{A}$ og $\mathbf{B}$ er domæner har de grænseværdiegenskaben og vi kan konstruere grænseværdien af kæden i $\mathbf{A}+\mathbf{B}$ som et par hvor andenkomponenten er grænseværdien af andenkomponenterne fra kæden. Vi skal så sikre os at denne grænseværdi er den mindste overgrænse, men det følger af den tilsvarende egenskab i $\mathbf{A}$ eller $\mathbf{B}$.

Bevis slut

Notation. For summen af to domæner $\mathbf{A} + \mathbf{B}$ kan vi definere en række funktioner.

$$\begin{aligned}
&\mathsf{isl}, \mathsf{isr} : \mathbf{A} + \mathbf{B} \to \mathbb{B}_\perp \\
&\mathsf{inl} : \mathbf{A} \to \mathbf{A} + \mathbf{B} \\
&\mathsf{inr} : \mathbf{B} \to \mathbf{A} + \mathbf{B} \\
&\mathsf{outl} : \mathbf{A} + \mathbf{B} \to \mathbf{A} \\
&\mathsf{outr} : \mathbf{A} + \mathbf{B} \to \mathbf{B}
\end{aligned}$$

hvor

$$\begin{array}{lll}
\mathsf{isl}(\perp) = \perp & \mathsf{isr}(\perp) = \perp & \\
\mathsf{isl}(\langle 1, \mathbf{x}\rangle) = \mathsf{true} & \mathsf{isr}(\langle 1, \mathbf{x}\rangle) = \mathsf{false} & \mathbf{x} \in \mathbf{A} \\
\mathsf{isl}(\langle 2, \mathbf{x}\rangle) = \mathsf{false} & \mathsf{isr}(\langle 2, \mathbf{x}\rangle) = \mathsf{true} & \mathbf{x} \in \mathbf{B} \\
\mathsf{inl}(\mathbf{x}) = \langle 1, \mathbf{x}\rangle & \mathsf{inr}(\mathbf{x}) = \langle 2, \mathbf{x}\rangle & \\
\mathsf{outl}(\perp) = \perp & \mathsf{outr}(\perp) = \perp & \\
\mathsf{outl}(\langle 1, \mathbf{x}\rangle) = \mathbf{x} & \mathsf{outr}(\langle 1, \mathbf{x}\rangle) = \perp & \mathbf{x} \in \mathbf{A} \\
\mathsf{outl}(\langle 2, \mathbf{x}\rangle) = \perp & \mathsf{outr}(\langle 2, \mathbf{x}\rangle) = \mathbf{x} & \mathbf{x} \in \mathbf{B}
\end{array}$$

Strikt sum. Sumdomænet $\mathbf{A} + \mathbf{B}$ af to domæner kaldes også for en *separat sum* (*separate sum*). For to domæner $(\mathbf{A},\ \sqsubseteq_\mathbf{A})$ og $(\mathbf{B},\ \sqsubseteq_\mathbf{B})$ er den *strikse sum* (*coalesced sum*) $\mathbf{A} \oplus \mathbf{B}$ defineret som $(\mathbf{A} \setminus \{\perp_\mathbf{A}\}) + (\mathbf{B} \setminus \{\perp_\mathbf{B}\})$, hvor vi bruger den samme ordning som for sumdomænet. Også den strikse sum af to domæner er et domæne.

2.3 Funktiondomæne

For to domæner $(\mathbf{A},\ \sqsubseteq_\mathbf{A})$ og $(\mathbf{B},\ \sqsubseteq_\mathbf{B})$ kan vi konstruere *funktionsdomænet* $(\mathbf{A} \to \mathbf{B},\ \sqsubseteq_{\mathbf{A}\to\mathbf{B}})$ af kontinuerte funktioner fra $\mathbf{A}$ til $\mathbf{B}$ med ordningen

$$\mathbf{f} \sqsubseteq_{\mathbf{A}\to\mathbf{B}} \mathbf{g} \Leftrightarrow \forall \mathbf{x} \in \mathbf{A}.\ \mathbf{f}(\mathbf{x}) \sqsubseteq_\mathbf{B} \mathbf{g}(\mathbf{x})$$

Bevis.

Ordning. Ordningsegenskaberne arves direkte fra domænet $\mathbf{B}$. For vilkårlige $\mathbf{f}$ og $\mathbf{g}$ har vi:

$$\begin{aligned}
\mathbf{f} \sqsubseteq \mathbf{f} &\Leftrightarrow \forall \mathbf{x}.\ \mathbf{f}(\mathbf{x}) \sqsubseteq \mathbf{f}(\mathbf{x}) \\
\mathbf{f} \sqsubseteq \mathbf{g} \wedge \mathbf{g} \sqsubseteq \mathbf{f} &\Rightarrow \forall \mathbf{x}.\ \mathbf{f}(\mathbf{x}) \sqsubseteq \mathbf{g}(\mathbf{x}) \wedge \mathbf{g}(\mathbf{x}) \sqsubseteq \mathbf{f}(\mathbf{x}) \\
&\Rightarrow \forall \mathbf{x}.\ \mathbf{f}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) \\
&\Rightarrow \mathbf{f} = \mathbf{g} \\
\mathbf{f} \sqsubseteq \mathbf{g} \wedge \mathbf{g} \sqsubseteq \mathbf{h} &\Rightarrow \forall \mathbf{x}.\ \mathbf{f}(\mathbf{x}) \sqsubseteq \mathbf{g}(\mathbf{x}) \wedge \mathbf{g}(\mathbf{x}) \sqsubseteq \mathbf{h}(\mathbf{x}) \\
&\Rightarrow \forall \mathbf{x}.\ \mathbf{f}(\mathbf{x}) \sqsubseteq \mathbf{h}(\mathbf{x}) \\
&\Rightarrow \mathbf{f} \sqsubseteq \mathbf{h}
\end{aligned}$$

Mindste element. Mindste elementet i domænet $\mathbf{A} \to \mathbf{B}$ er funktionen $\lambda \mathbf{x}.\ \perp_{\mathbf{B}}$ idet $\perp_{\mathbf{B}} \sqsubseteq \mathbf{g}(\mathbf{x})$ for alle $\mathbf{x} \in \mathbf{A}$ og $\mathbf{g} \in \mathbf{A} \to \mathbf{B}$.

Grænseværdi. For en kæde af kontinuerte funktioner $(\mathbf{f_i})$ i $\mathbf{A} \to \mathbf{B}$ konstruerer vi grænseværdien som $\lambda \mathbf{x}.\ \bigsqcup_{\mathbf{i}} \mathbf{f_i}(\mathbf{x})$. Grænseværdien er veldefineret da $(\mathbf{f_i}(\mathbf{x}))$ for ethvert $\mathbf{x}$ udgør en kæde i $\mathbf{B}$. At det er en overgrænse og den mindste overgrænse følger let af $\mathbf{B}$'s domæneegenskaber. Vi mangler blot at vise at grænseværdien er kontinuert og altså ligger i domænet $\mathbf{A} \to \mathbf{B}$. Dette overlades som en opgave.

Bevis slut.

Funktionsdomæne. Mængden af funktioner fra en ikke tom mængde $\mathbf{A}$ til et domæne $\mathbf{B}$ er et domæne. Dette følger af at vi kun bruger ordningen af $\mathbf{A}$ i beviset for at grænseværdier er kontinuerte.

Notation. Som det ses kan man let komme ud for at bruge pil symbolet om forskellige slags funktioner. Hvis mængderne $\mathbf{A}$ og $\mathbf{B}$ er ordnede som domæner vil $\mathbf{A} \to \mathbf{B}$ normalt betyde funktionsdomænet (og altså kun mængden af kontinuerte funktioner fra $\mathbf{A}$ til $\mathbf{B}$). Hvis man gerne vil understrege at vi her hentyder til de kontinuerte funktioner fra $\mathbf{A}$ til $\mathbf{B}$ skriver man gerne $\mathbf{A} \to_{\mathbf{c}} \mathbf{B}$. Hvis det er de monotone funktioner vi er interesserede i skriver man $\mathbf{A} \to_{\mathbf{m}} \mathbf{B}$. Endelig ser man i litteraturen også symbolet $\mathbf{A} \to_{\mathbf{p}} \mathbf{B}$ for de partielle funktioner fra $\mathbf{A}$ til $\mathbf{B}$.

Sætning. Fixpunktoperatoren $\mathsf{fix} : (\mathbf{D} \to_{\mathbf{c}} \mathbf{D}) \to \mathbf{D}$ er kontinuert.

Bevis.

<u>Monotonicitet.</u> For $\mathbf{f} \sqsubseteq \mathbf{g}$ følger ved induktion at $\mathbf{f}^{\mathbf{i}}(\perp) \sqsubseteq \mathbf{g}^{\mathbf{i}}(\perp)$ hvorfor

$$\mathsf{fix}(\mathbf{f}) = \bigsqcup_{\mathbf{i}} \mathbf{f}^{\mathbf{i}}(\perp) \sqsubseteq \bigsqcup_{\mathbf{i}} \mathbf{g}^{\mathbf{i}}(\perp) = \mathsf{fix}(\mathbf{g})$$

<u>Grænseværdibevarende.</u> Lad $(\mathbf{f_i})$ være en kæde af kontinuerte funktioner i domænet $\mathbf{D} \rightarrow_{\mathbf{c}} \mathbf{D}$. Vi skal så vise at

$$\mathsf{fix}(\bigsqcup_{\mathbf{i}} \mathbf{f_i}) = \bigsqcup_{\mathbf{i}} \mathsf{fix}(\mathbf{f_i})$$

Ved induktion følger at $(\bigsqcup_{\mathbf{i}} \mathbf{f_i})^{\mathbf{j}} = (\bigsqcup_{\mathbf{i}} \mathbf{f_i^j})$ for alle $\mathbf{j}$. Dette følger af at grænseværdien for en kæde af funktioner er defineret elementvis. Herefter udleder vi:

$$\begin{aligned}
\mathsf{fix}(\bigsqcup_{\mathbf{i}} \mathbf{f_i}) &= \bigsqcup_{\mathbf{j}}((\bigsqcup_{\mathbf{i}} \mathbf{f_i})^{\mathbf{j}}(\perp)) \\
&= \bigsqcup_{\mathbf{j}} \bigsqcup_{\mathbf{i}} \mathbf{f_i^j}(\perp) \\
&= \bigsqcup_{\mathbf{i}} \bigsqcup_{\mathbf{j}} \mathbf{f_i^j}(\perp) \\
&= \bigsqcup_{\mathbf{i}} \mathsf{fix}(\mathbf{f_i})
\end{aligned}$$

da $(\mathbf{f_i^j}(\perp))$ er en dobbeltindiceret kæde i $\mathbf{D}$.

Bevis slut.

2.4 Potensmængde

For en mængde $\mathbf{S}$ kan man danne *potensmængden* $\mathcal{P}(\mathbf{S})$. Dette er et domæne med delmængdeordningen og den tomme mængde som mindste element:

Den tomme mængde $\varnothing$ opfylder klart $\forall \mathbf{X} \subseteq \mathbf{S}.\ \varnothing \subseteq \mathbf{X}$.

At delmængderelationen er en partiel ordning skulle gerne være et af de resultater vi er bekendt med på forhånd.

For en kæde af delmængder $\mathbf{X}_1 \subseteq \mathbf{X}_2 \subseteq \cdots$ kan vi konstruere grænseværdien som

$$\bigsqcup_{\mathbf{i}} \mathbf{X_i} = \bigcup_{\mathbf{i}} \mathbf{X_i}$$

At man kan tillade sig at tage foreningsmængden af en tællelig klasse af mængder er en af de spilleregler mængdelæren giver os. Den sikrer os endvidere at man herved danner den mindste mængde som indeholder alle $\mathbf{X_i}$ mængderne.

Begrænsning. Vi er imidlertid nu rede til den måske væsentligste restriktion i domæneteorien. For et domæne $\mathbf{D}$ er funktionen

$$\begin{aligned}&\mathsf{singleton} : \mathbf{D} \to \mathcal{P}(\mathbf{D})\\&\mathsf{singleton}(\mathbf{d}) = \{\mathbf{d}\}\end{aligned}$$

ikke nødvendigvis kontinuert.

Dette kan måske synes ganske uskyldigt, men der er anvendelser hvor det er vigtigt at have en kontinuert afbildning ind i potensmængden. Det gælder især når man vil modellere ikke-determinisme i programmeringssprog. Vejen udenom er at bruge en anden ordning end delmængdeordningen, og indføre en restriktion på hvilke delmængder vi vil betragte. Herved definerer man et såkaldt *potensdomæne* (eng. *power domain*). Vi skal se tre forskellige måder at gøre det på.

Egli-Milner ordningen. På en potensmængde $\mathcal{P}(\mathbf{D})$ over et domæne $\mathbf{D}$ indfører vi relationen $\sqsubseteq_{\mathbf{EM}}$ som følger.

$$\mathbf{X} \sqsubseteq_{\mathbf{EM}} \mathbf{Y} \Leftrightarrow (\forall \mathbf{x} \in \mathbf{X}.\ \exists \mathbf{y} \in \mathbf{Y}.\ \mathbf{x} \sqsubseteq \mathbf{y}) \wedge (\forall \mathbf{y} \in \mathbf{Y}.\ \exists \mathbf{x} \in \mathbf{X}.\ \mathbf{x} \sqsubseteq \mathbf{y})$$

Relationen kan læses som at alle dele af $\mathbf{X}$ skal ligge under noget af $\mathbf{Y}$ (i domænet $\mathbf{D}$), og at alle dele af $\mathbf{Y}$ skal ligge over noget af $\mathbf{X}$.

Plotkins potensdomæne. Relationen $\sqsubseteq_{\mathbf{EM}}$ er ikke en partiel ordning på potensmængden da den mangler antisymmetri egenskaben. Dette kan vi råde bod på ved alene at betragte de ikke-tomme konvekse delmængder af $\mathbf{D}$. En mængde $\mathbf{X}$ er konveks hvis den opfylder

$$\forall \mathbf{x}_1, \mathbf{x}_3 \in \mathbf{X}.\ \forall \mathbf{x}_2 \in \mathbf{D}.\ \mathbf{x}_1 \sqsubseteq \mathbf{x}_2 \sqsubseteq \mathbf{x}_3 \Rightarrow \mathbf{x}_2 \in \mathbf{X}$$

Dette potensdomæne skrives normalt som $(\wp_{\mathbf{P}}(\mathbf{D}),\ \sqsubseteq_{\mathbf{EM}})$.

Hoares potensdomæne. Der er to andre konstruktioner man ser brugt. I Hoares potensdomæne bruges alene de nedadafsluttede delmængder. En mængde $\mathbf{X}$ er nedadafsluttet hvis

$$\forall \mathbf{x}, \mathbf{y} \in \mathbf{D}.\ \mathbf{y} \in \mathbf{X} \wedge \mathbf{x} \sqsubseteq \mathbf{y} \Rightarrow \mathbf{x} \in \mathbf{X}$$

En nedadafsluttet mængde er et specialtilfælde af en konveks mængde. Dette potensdomæne skrives normalt som $(\wp_{\mathbf{H}}(\mathbf{D}),\ \sqsubseteq_{\mathbf{EM}})$. I Hoares potensdomæne er der kun en etpunktsmængde: $\{\perp\}$. Alle andre delmængder skal indeholde nedadafslutningen. For at konstruere disse nedadafsluttede mængder indfører man funktionen LC:

$$\begin{aligned}
&\mathsf{LC} : \mathcal{P}(\mathbf{D}) \to \wp_{\mathbf{H}}(\mathbf{D}) \\
&\mathsf{LC}(\mathbf{X}) = \{\mathbf{y} \in \mathbf{D} \mid \exists \mathbf{x} \in \mathbf{X}.\ \mathbf{y} \sqsubseteq \mathbf{x}\}
\end{aligned}$$

Funktionen kaldes *left closure* eller *lower closure*.

Smyths potensdomæne. I Smyths potensdomæne bruges alene de opadafsluttede delmængder. Dette potensdomæne skrives som $(\wp_{\mathbf{S}}(\mathbf{D}),\ \sqsubseteq_{\mathbf{EM}})$.

2.5 Opgaver

Opgave 2.1. Vis at ordningen for sumdomænet er en partiel ordning.

Opgave 2.2. Med funktionen $\mathsf{cond} : \mathbb{B}_\perp \times \mathbf{X} \times \mathbf{X} \to \mathbf{X}$ defineret for ethvert domæne $\mathbf{X}$ ved

$$\begin{aligned}
\mathsf{cond}(\perp, \mathbf{x}, \mathbf{y}) &= \perp_{\mathbf{X}} \\
\mathsf{cond}(\mathsf{true}, \mathbf{x}, \mathbf{y}) &= \mathbf{x} \\
\mathsf{cond}(\mathsf{false}, \mathbf{x}, \mathbf{y}) &= \mathbf{y}
\end{aligned}$$

vis at

$$\forall \mathbf{x} \in \mathbf{A} + \mathbf{B}.\ \mathsf{cond}(\mathsf{isl}(\mathbf{x}), \mathsf{inl}(\mathsf{outl}(\mathbf{x})), \mathsf{inr}(\mathsf{outr}(\mathbf{x}))) = \mathbf{x}$$

Opgave 2.3. For et domæne $\mathbf{A}$ med mindste element $\perp_{\mathbf{A}}$ kan man danne det løftede domæne $\mathbf{A}_\perp$ (*lifted cpo*) $(\mathbf{A}_\perp,\ \sqsubseteq_\perp)$ som værende $\mathbf{A}$ udvidet med et nyt mindste element $\perp$ således at

$$\begin{aligned}
&\mathbf{A}_\perp = \{\perp\} \cup \{\langle 1, \mathbf{x}\rangle \mid \mathbf{x} \in \mathbf{A}\} \\
&\perp \sqsubseteq_\perp \mathbf{x} && \forall \mathbf{x} \in \mathbf{A}_\perp \\
&\langle 1, \mathbf{x}\rangle \sqsubseteq_\perp \langle 1, \mathbf{y}\rangle && \Leftrightarrow \mathbf{x} \sqsubseteq \mathbf{y}
\end{aligned}$$

Vis at dette faktisk er et domæne.

Notation. Til et løftet domæne kan vi definere følgende funktioner.

$$\begin{aligned}
\mathsf{def} &: \mathbf{A}_\perp \to \mathbb{B}_\perp \\
\mathsf{up} &: \mathbf{A} \to \mathbf{A}_\perp \\
\mathsf{down} &: \mathbf{A}_\perp \to \mathbf{A}
\end{aligned}$$

hvor

$$\begin{aligned}
\mathsf{def}(\perp) &= \perp \\
\mathsf{def}(\langle 1, \mathbf{x}\rangle) &= \mathsf{true} \qquad \mathbf{x} \in \mathbf{A} \\
\mathsf{up}(\mathbf{x}) &= \langle 1, \mathbf{x}\rangle \\
\mathsf{down}(\perp) &= \perp_{\mathbf{A}} \\
\mathsf{down}(\langle 1, \mathbf{x}\rangle) &= \mathbf{x}
\end{aligned}$$

Disse funktioner opfylder:

$$\forall \mathbf{x} \in \mathbf{A}_\perp.\ \mathsf{cond}(\mathsf{def}(\mathbf{x}), \mathsf{up}(\mathsf{down}(\mathbf{x})), \perp) = \mathbf{x}$$

Opgave 2.4. Vis at grænseværdien af en kæde af kontinuerte funktioner i funktionsdomænet selv er kontinuert.

Opgave 2.5. For Hoares og Smyths potensdomæner kan Egli-Milner ordningen forenkles noget. Forklar hvordan.

Opgave 2.6. Der er et modstykke til sætningen om funktioner af to argumenter. Vis at for en funktion

$$\mathbf{f} : \mathbf{A} \to \mathbf{B} \times \mathbf{C}$$

at $\mathbf{f}$ er kontinuert hvis og kun hvis $\mathsf{fst} \circ \mathbf{f}$ og $\mathsf{snd} \circ \mathbf{f}$ er kontinuerte.

3 Rekursivt definerede funktioner

Vi er allerede stødt på en række funktionsdefinitioner i disse noter. Indtil nu har de haft en form som

$$\mathbf{p}(\mathbf{x}) = 3 + 2 * \mathbf{x} + \mathbf{x}^2$$

hvor højresiden er et simpelt udtryk i argumenterne. Som tidligere nævnt har sådanne funktioner et fundament i mængdelæren. Det betyder at en funktion i virkeligheden (eller rettere i matematikken) opfattes som en mængde.

$$\mathbf{f} : \mathbf{D} \rightarrow \mathbf{E}$$
$$\mathbf{f}(\mathbf{x}) = \mathbf{e} \Leftrightarrow \mathbf{f} = \{\langle \mathbf{x}, \mathbf{r} \rangle \mid \mathbf{x} \in \mathbf{D} \wedge \mathbf{r} = \mathbf{e}\}$$

Funktionen ovenfor bliver således til

$$\mathbf{p} = \{\langle \mathbf{x}, \mathbf{r} \rangle \mid \mathbf{x} \in \mathbb{N} \wedge \mathbf{r} = 3 + 2 * \mathbf{x} + \mathbf{x}^2\}$$

I dette afsnit skal vi se på rekursive funktionsdefinitioner og hvordan vi kan bruge fixpunktsætningen til at give dem mening.

3.1 Paradokser

Hvis vi bruger oversættelsen af funktioner til mængder på en typisk rekursiv funktion som fakultetsfunktionen:

$$\mathbf{f}(\mathbf{x}) = \mathbf{if}\ \mathbf{x} = 0\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{x} * \mathbf{f}(\mathbf{x} - 1)$$

ja, så får vi følgende mængde:

$$\mathbf{f} = \{\langle \mathbf{x}, \mathbf{r} \rangle \mid \mathbf{x} \in \mathbb{N} \wedge ((\mathbf{x} = 0 \wedge \mathbf{r} = 1) \vee (\mathbf{x} \neq 0 \wedge \langle \mathbf{x} - 1, \mathbf{s} \rangle \in \mathbf{f} \wedge \mathbf{r} = \mathbf{x} * \mathbf{s}))\}$$

Nu er vi ude på et skråplan. Når vi forsøger at definere mængder ved hjælp af sig selv åbner vi op for masser af mulige paradokser. Vi kan for eksempel spørge om hvilke elementer mængden **g** har:

$$\mathbf{g} = \{\mathbf{x} \mid \mathbf{x} \notin \mathbf{g}\}$$

I almindelige ord svarer det til et af de klassiske paradokser i mængdelæren: Hvem barberer barberen, der barberer alle de folk i byen, som ikke barberer sig selv?

Løsningen i mængdelæren er at forbyde sådanne definitioner, men det er ikke særlig attraktivt for os, så vi må søge nye veje.

Løsninger. Intuitivt er det klart at $\mathbf{f}$ gerne skal være fakultetsfunktionen. Vi kan da også hurtigt overbevise os om at følgende mængde vil opfylde ligningen.

$$\mathbf{f} = \{\langle \mathbf{x}, \mathbf{y} \rangle \mid \mathbf{x} \geq 0 \wedge \mathbf{y} = \mathbf{x}!\}$$

Det er dog ikke alle sådanne ligninger der har løsninger og nogle ligninger kan have flere løsninger. Følgende fixpunktligning

$$\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x})$$

svarer til mængden

$$\mathbf{g} = \{\langle \mathbf{x}, \mathbf{r} \rangle \mid \langle \mathbf{x}, \mathbf{r} \rangle \in \mathbf{g}\}$$

og det siger sig selv at et hvilket som helst cartesisk produkt opfylder ligningen. Modsat vil ligningen

$$\mathbf{h}(\mathbf{x}) = \mathbf{h}(\mathbf{x}) + 1$$

ikke have nogle løsninger. Målet må nu være at finde en metode til at sikre løsninger, og at sortere ud blandt løsninger.

3.2 Fixpunkter

Det kommer næppe som en kæmpe overraskelse at fixpunktsætningen kan hjælpe os til at finde løsninger til rekursive funktionsdefinitioner. For en funktionsdefinition

$$\mathbf{f}(\mathbf{x}) = \mathbf{e}$$

hvor udtrykket $\mathbf{e}$ indeholder $\mathbf{f}$ kan vi definere en funktion

$$\mathbf{F}(\mathbf{f}) = \lambda\mathbf{x}.\ \mathbf{e}$$

hvilket også kan skrives som

$$\mathbf{F} = \lambda\mathbf{f}.\ \lambda\mathbf{x}.\ \mathbf{e}$$

Denne funktional (højere-ordens funktion) er ikke rekursiv og giver god mening i mængdelæren. Udtrykket $\mathbf{e}$ er jo et udtryk i argumenterne $\mathbf{x}$ og $\mathbf{f}$, men det afhænger ikke af $\mathbf{F}$. Kravet til en løsning på ligningen er da at det skal være et fixpunkt for funktionalen $\mathbf{F}$.

$$\mathbf{f} = \mathbf{F}(\mathbf{f})$$

For eksempel kan vi nu omformulere “definitionen” af fakultetsfunktionen til

$$\mathbf{F} = \lambda\mathbf{f}.\ \lambda\mathbf{x}.\ \textbf{if } \mathbf{x} = 0 \textbf{ then } 1 \textbf{ else } \mathbf{x} * \mathbf{f}(\mathbf{x} - 1)$$

og herefter vise at

$$\mathbf{F}(\lambda\mathbf{x}.\ \mathbf{x}!) = \lambda\mathbf{x}.\ \mathbf{x}!$$

Vi har således omformuleret spørgsmålet til at finde et fixpunkt for en funktion $\mathbf{F}$. Det er så fristende at spørge om funktionen $\mathbf{F}$ er kontinuert for så ved vi at den har fixpunkter og vi kender en metode til at finde det mindste.

Kontinuitet. Umiddelbart er funktionen $\mathbf{F}$ ikke kontinuert. Vi har ikke angivet hvilke mængder funktionen er defineret for og når der ikke er nogen ordning kan vi altså heller ikke tale om kontinuitet.

Som tidligere nævnt er $\mathbb{N} \to \mathbb{N}_\perp$ et domæne og vi vil nu forsøge at definere $\mathbf{F}$ som en kontinuert funktion i domænet

$$(\mathbb{N} \to \mathbb{N}_\perp) \to (\mathbb{N} \to \mathbb{N}_\perp)$$

Dette kræver dog at multiplikation udvides til domænet $\mathbb{N}_\perp$. En enkel måde at gøre det på er at kræve at $\mathbf{x} * \mathbf{y} = \perp \Leftrightarrow \mathbf{x} = \perp \vee \mathbf{y} = \perp$, altså at multiplikation opfører sig normalt bortset fra på den specielle $\perp$-værdi. Det er ikke svært at overbevise sig om at denne funktion er kontinuert. Det kan af og til være nyttigt at kunne skelne mellem de to slags multiplikationer og vi bruger da symbolet $*_\perp$.

$$\mathbf{x} *_\perp \mathbf{y} = \textbf{if } \mathbf{x} = \perp \vee \mathbf{y} = \perp \textbf{ then } \perp \textbf{ else } \mathbf{x} * \mathbf{y}$$

Vi kan herefter definere $\mathbf{f}$ til at være det mindste fixpunkt for $\mathbf{F}$

$$\mathbf{f} = \mathsf{fix}(\mathbf{F})$$
$$\mathbf{F} = \lambda\mathbf{f}.\ \lambda\mathbf{x}.\ \textbf{if } \mathbf{x} = 0 \textbf{ then } 1 \textbf{ else } \mathbf{x} *_\perp \mathbf{f}(\mathbf{x} - 1)$$

Spørgsmålet er så om denne definition svarer til hvad vi intuitivt havde forventet. Dette er selvfølgelig svært at svare på hvis man ikke er sikker på hvad man forventer, men vi kan da altid undersøge hvilken funktion, der rent faktisk er defineret ved dette fixpunkt.

Ikke-termination. Modsat funktioner kan et program risikere ikke at stoppe. Når vi gerne vil forklare hvad en rekursiv funktionsdefinition burde betyde vil vi normalt tænke på hvad et tilsvarende program vil gøre i et programmeringssprog med funktioner. Hvad er så parallellen til ikke-termination for rekursive funktioner? Som de fleste nok allerede har gættet er det mindsteelementet $\perp$. Hvis man kalder fakultetsfunktionen med et negativt tal vil man forvente at det tilsvarende program ikke standsede, og når vi undersøger funktionen finder vi at den vil returnere værdien $\perp$. Mere formelt kan vi konstatere at funktionen

$$\mathbf{f} = \lambda \mathbf{x}.\ \mathbf{if}\ \mathbf{x} < 0\ \mathbf{then}\ \perp\ \mathbf{else}\ \mathbf{x}!$$

er et fixpunkt for funktionalen. Bemærk dog at funktionen

$$\mathbf{g} = \lambda \mathbf{x}.\ \mathbf{if}\ \mathbf{x} < 0\ \mathbf{then}\ 0\ \mathbf{else}\ \mathbf{x}!$$

også er et fixpunkt. Med ordningen i funktionsdomænet har vi imidlertid at $\mathbf{f} \sqsubseteq \mathbf{g}$ og at $\mathbf{f}$ er det mindste fixpunkt. At det er det mindste fixpunkt der svarer til hvordan et program vil opføre sig passer med de fleste programmørers erfaring: Hvis et program kan slippe afsted med ikke at stoppe så vil det også gøre det.

Det er fristende at omtale $\perp$ som ikke-termination eller ikke-defineret men man skal huske at $\perp$ blot er symbolet for et bestemt element i et domæne. Elementet $\perp$ lever i den matematiske verden og er særdeles veldefineret; det er faktisk det eneste element vi er sikker på at have i et domæne. Det er når vi bruger domæneteorien med fixpunkter at $\perp$ bruges til at beskrive ikke-termination. Når vi kommer til abstrakt fortolkning skal vi se semantikker hvor der ikke er spor "udefineret" ved $\perp$.

3.3 Løsningen

Vi har defineret funktionen $\mathbf{f}$ som grænseværdien af en kæde af kontinuerte funktioner. For at undersøge hvordan funktionen opfører sig kan vi forsøge at udregne de første approximationer. Fixpunktet for $\mathbf{F}$ er defineret som grænseværdien af kæden $\perp \sqsubseteq \mathbf{F}(\perp) \sqsubseteq \mathbf{F}^2(\perp) \sqsubseteq \cdots$. Lad os nu undersøge disse approximationer til fixpunktet.

$$\begin{aligned}
\perp_{\mathbb{N}\to\mathbb{N}_\perp} &= \lambda \mathbf{x}.\ \perp_{\mathbb{N}} \\
\mathbf{F}(\perp_{\mathbb{N}\to\mathbb{N}_\perp}) &= \lambda \mathbf{x}.\ \mathbf{if}\ \mathbf{x} = 0\ \mathbf{then}\ 1\ \mathbf{else}\ \perp_{\mathbb{N}} \\
\mathbf{F}^2(\perp_{\mathbb{N}\to\mathbb{N}_\perp}) &= \lambda \mathbf{x}.\ \mathbf{if}\ \mathbf{x} = 0\ \mathbf{then}\ 1\ \mathbf{else} \\
&\qquad\quad \mathbf{x} *_\perp (\mathbf{if}\ \mathbf{x} - 1 = 0\ \mathbf{then}\ 1\ \mathbf{else}\ \perp_{\mathbb{N}}) \\
&= \lambda \mathbf{x}.\ \mathbf{if}\ \mathbf{x} = 0\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{if}\ \mathbf{x} = 1\ \mathbf{then}\ 1\ \mathbf{else}\ \perp_{\mathbb{N}}
\end{aligned}$$

$$\begin{aligned}\mathbf{F}^3(\bot_{\mathbb{N}\to\mathbb{N}_\bot}) = \lambda\mathbf{x}.\ & \mathbf{if\ x} = 0\ \mathbf{then}\ 1\ \mathbf{else} \\ & \mathbf{x} *_\bot (\mathbf{if\ x} - 1 = 0\ \mathbf{then}\ 1\ \mathbf{else\ if\ x} - 1 = 1\ \mathbf{then}\ 1\ \mathbf{else}\ \bot_{\mathbb{N}}) \\ = \lambda\mathbf{x}.\ & \mathbf{if\ x} = 0\ \mathbf{then}\ 1\ \mathbf{else} \\ & \mathbf{if\ x} = 1\ \mathbf{then}\ 1\ \mathbf{else} \\ & \mathbf{if\ x} = 2\ \mathbf{then}\ 2\ \mathbf{else}\ \bot_{\mathbb{N}})\end{aligned}$$

Hver ny approximation vil altså have flere og flere værdier beregnet. Således vil $\mathbf{F}^{\mathbf{i}}(\bot)$ have beregnet de værdier, der kan beregnes med $\mathbf{i} - 1$ rekursive kald.

Såvidt approximationerne, men hvad med grænseværdien? Kan vi være sikker på at fixpunktet for **F** ikke pludselig har en anden værdi? Ja, det sikrer ordningen på domænet $\mathbb{N}_\bot$ af funktioner. Hvis den **i**de approximation for et argument **x** har en værdi forskellig fra bundelementet $\bot$ så vil alle senere approximationer og grænseværdien have samme værdi. Tilsvarende ved vi at hvis alle approximationerne har bundelementet som værdi for et givet argument så vil grænseværdien have samme værdi.

Eksempel. Kikker vi på samme måde på funktionen

$$\mathbf{h}(\mathbf{x}) = \mathbf{h}(\mathbf{x})$$

med funktionalen

$$\mathbf{H} = \lambda\mathbf{h}.\ \lambda\mathbf{x}.\ \mathbf{h}(\mathbf{x})$$

får vi

$$\begin{aligned}\bot_{\mathbb{N}\to\mathbb{N}_\bot} &= \lambda\mathbf{x}.\ \bot_{\mathbb{N}} \\ \mathbf{H}(\bot_{\mathbb{N}\to\mathbb{N}_\bot}) &= \lambda\mathbf{x}.\ \bot_{\mathbb{N}}\end{aligned}$$

og altså at $\mathbf{H}(\bot_{\mathbb{N}\to\mathbb{N}_\bot}) = \bot_{\mathbb{N}\to\mathbb{N}_\bot}$. Det mindste fixpunkt for funktionalen **H** er altså den konstante funktion $\lambda\mathbf{x}.\ \bot_{\mathbb{N}}$.

3.4 Beregnelige funktioner

Det er ikke alle rekursive definitioner vi kan give mening på denne måde. For eksempel vil ligningen

$$\mathbf{q}(\mathbf{x}) = \mathbf{if\ q}(\mathbf{x}) = \bot\ \mathbf{then}\ 1\ \mathbf{else}\ \bot$$

ikke svare til en kontinuert funktional. Det er heller ikke oplagt hvilken betydning man skal tillægge en sådan definition, den svarer ihvertfald ikke til funktioner vi normalt kan forestille os at implementere i et programmeringssprog. I et forsøg på at sige noget om hvad vi kan og ikke kan vil vi vove et par påstande. Det er blot uformelle udsagn idet begreberne *beregning* og *beregnelig* ikke er formaliserede.

Påstand 1: *Alle beregnelige funktioner er kontinuerte.*

Påstanden er mere præcist at hvis man formulerer en beregnelig funktion som et fixpunkt af en funktional vil funktionalen være kontinuert. Dette betyder så også at vi påstår at alle de konstruktioner vi ser i programmeringssprog svarer til kontinuerte funktioner.

Påstand 2: *Værdien $\perp$ som funktionsværdi betyder at beregningen af funktionen ikke vil terminere for det givne argument.*

Dette betyder at et program svarende til funktionen $\mathbf{h}$ defineret ved $\mathbf{h}(\mathbf{x}) = \mathbf{h}(\mathbf{x})$ ikke vil terminere for noget argument. Det er denne påstand der ligger bag når man af og til (uformelt) omtaler mindsteelementet som "ikke-termination" eller "udefineret". Ret beset er det blot et matematisk objekt, der på denne måde kan bruges til at beskrive "ikke-termination".

Konklusion. Fixpunktsætningen kan bruges til at give rekursive funktionsdefinitioner mening. Kravet er blot at definitionen, opfattet som funktional, er kontinuert. Vi har uformelt argumenteret for, at dette ikke er et stort problem sålænge vi snakker om funktioner, der svarer til hvad vi ser i programmeringssprog. Vi har ligeledes argumenteret for, at den løsning fixpunktsætningen giver, svarer til vor normale opfattelse af beregning, med mindste elementet $\perp$ som betydende ikke-termination.

Det centrale i denne diskussion er at give en fornemmelse af hvordan funktioner opfører sig, når de er defineret som fixpunkter. Der er ikke nogle oplagte muligheder for at bevise at de funktioner vi definerer svarer til hvad vi intuitivt ville forvente. Det synes heller ikke muligt at bevise at alle "intuitivt beregnelige" funktioner kan defineres enten direkte i mængdelæren eller ved at benytte fixpunkter. For at fortsætte læsningen af disse noter er det ikke noget krav at man tror på disse påstande. Der er stadig en mulighed for at der findes et magisk sprog hvori man kan udtrykke noget der ellers ikke vil være beregneligt. I resten af disse noter skal vi dog arbejde med nogle mere jordnære programmeringssprog.

Scott's theorem. Påstand 1 refereres ofte til som Scott's theorem. Som det fremgår går påstanden kun den ene vej. Det skyldes at klassen af kontinuerte funktioner er større en mængden af beregnelige funktioner. Den sidste klassse er oplagt tællelig (man kan nummerere alle programmer ved at betragte programtekster som tal i 256-talsystemet). Derimod er der ingen begrænsninger på tællelighed af kontinuerte funktioner. Der findes andre definitioner af domæneteori, hvor man indfører flere begrænsninger for at bringe domænerne tættere på det de skal beskrive. Til dette hører såkaldte *algebraiske domæner* eller ω-algebraiske domæner, hvor man kun tillader tælleligt mange endelige elementer (et element er endeligt hvis det ikke kan opnås af en kæde der ikke indeholder sig selv). Sådanne definitioner er ikke helt så stabile over for domænekonstruktioner og kan heller ikke fjerne alle uberegnelige funktioner. I det hele taget findes der mange forskellige definitioner af domæner i et omfang så nogle folk opfatter domæneteorien som stor og kompliceret. Kapitel 1 skulle gerne have givet et andet indtryk.

3.5 Opgaver

Opgave 3.1. Med $\mathbf{D}$ og $\mathbf{E}$ som domæner, vis at funktionsanvendelse er kontinuert for kontinuerte funktioner. Dvs. vis at

$$\begin{aligned}&\mathsf{apply} : ((\mathbf{D} \to_{\mathbf{c}} \mathbf{E}) \times \mathbf{D}) \to \mathbf{E}\\ &\mathsf{apply}(\mathbf{f}, \mathbf{x}) = \mathbf{f}(\mathbf{x})\end{aligned}$$

er kontinuert. Vink: Vis at funktionen er kontinuert i hvert argument for sig.

Opgave 3.2. Med $\mathbf{C}$, $\mathbf{D}$ og $\mathbf{E}$ som domæner, vis at funktionssammensætning er kontinuert. Dvs. vis at

$$\begin{aligned}&\mathsf{compose} : ((\mathbf{C} \to_{\mathbf{c}} \mathbf{D}) \times (\mathbf{D} \to_{\mathbf{c}} \mathbf{E})) \to (\mathbf{C} \to_{\mathbf{c}} \mathbf{E})\\ &\mathsf{compose}(\mathbf{f}, \mathbf{g}) = \lambda \mathbf{x}.\ \mathbf{g}(\mathbf{f}(\mathbf{x}))\end{aligned}$$

er kontinuert.

Opgave 3.3. I påstand 1 går argumentet kun den ene vej. Konstruer en kontinuert, men intuitivt ikke-beregnelig funktion.

Opgave 3.4. I forrige afsnit definerede vi en funktion for betingede udtryk $\mathsf{cond} : \mathbb{B}_\perp \times \mathbf{X} \times \mathbf{X} \to \mathbf{X}$ ved

$$\begin{aligned}\mathsf{cond}(\perp, \mathbf{x}, \mathbf{y}) &= \perp_{\mathbf{X}}\\ \mathsf{cond}(\mathsf{true}, \mathbf{x}, \mathbf{y}) &= \mathbf{x}\\ \mathsf{cond}(\mathsf{false}, \mathbf{x}, \mathbf{y}) &= \mathbf{y}\end{aligned}$$

Vis at funktionen

$$\begin{aligned}\mathsf{pcond}(\perp, \mathbf{x}, \mathbf{y}) &= \perp_{\mathbf{X}} && \mathbf{if}\ \mathbf{x} \neq \mathbf{y}\\ \mathsf{pcond}(\perp, \mathbf{x}, \mathbf{y}) &= \mathbf{x} && \mathbf{if}\ \mathbf{x} = \mathbf{y}\\ \mathsf{pcond}(\mathsf{true}, \mathbf{x}, \mathbf{y}) &= \mathbf{x}\\ \mathsf{pcond}(\mathsf{false}, \mathbf{x}, \mathbf{y}) &= \mathbf{y}\end{aligned}$$

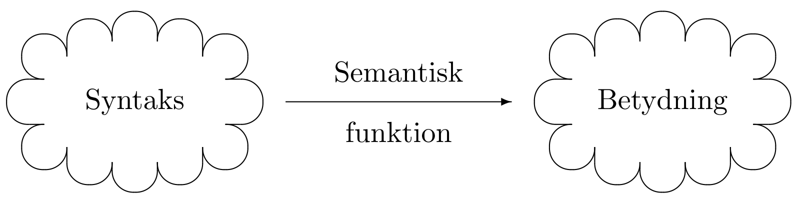
er kontinuert når $\mathbf{X}$ er et fladt domæne. Hvordan kan funktionen implementeres så $\perp$ netop svarer til ikke-termination?

4 Fixpunktsemantik

Vi har nu set hvordan fixpunktsætningen kan bruges til at lave rekursive funktionsdefinitioner. Vort næste punkt er så at bruge det mere generelt som et værktøj til at definere semantik. I dette afsnit vil vi opstille spillereglerne for denne metode til at beskrive semantik og derefter vise med nogle eksempler hvordan den kan bruges.

4.1 Semantiske funktioner

En fixpunktsemantik er en semantik hvor betydningen af programmer beskrives som et fixpunkt for en kontinuert funktion på et domæne. Der er forholdsvis få, enkle regler for hvordan dette må gøres og disse skal gerne sikre os at det vi gør giver mening. Den semantiske beskrivelse er en funktion der afbilder programmer til deres betydning i det valgte domæne.



Det væsentligste krav til semantikken er at betydningen af udtryk beskrives alene ved hjælp af dets bestanddele. Dette kaldes ofte *den denotationelle antagelse*, eller man siger at semantikken skal være *kompositionel*. Man kan også opfatte det som at semantikken skal være en makro, der direkte kan udfoldes og efterlade et udtryk, som giver mening i domæneteorien.

Syntaks. Udgangspunktet for en semantisk beskrivelse af et sprog er syntaksen. Vi må starte med at beskrive hvordan de mulige programmer kan se ud. Dette gøres normalt ved at præsentere en grammatik for sproget. Vi kan gøre det ved en entydig kontekstfri grammatik, men da det nu er semantikken vi er interesseret i, udelades ofte alle de finere detaljer som præcedensregler, mm.

For eksempel kan vi definere et lille funktionssprog ved følgende lille grammatik.

$$
\begin{array}{lll}
\mathbf{e} ::= & \mathbf{c_i} & \text{konstanter} \\
\mid & \mathbf{x_i} & \text{parametre} \\
\mid & \mathbf{a_i}(\mathbf{e}_1,\ \ldots,\mathbf{e}_\ell) & \text{standardfunktioner} \\
\mid & \textbf{if } \mathbf{e}_1 \textbf{ then } \mathbf{e}_2 \textbf{ else } \mathbf{e}_3 & \text{betinget udtryk} \\
\mid & \mathbf{f_i}(\mathbf{e}_1,\ \ldots,\mathbf{e_k}) & \text{funktionskald}
\end{array}
$$

hvor programmer har en fast form med $\mathbf{n}$ funktioner, hver med $\mathbf{k}$ argumenter.

$$
\begin{aligned}
\mathbf{p} ::= \mathbf{f}_1(\mathbf{x}_1,\ \ldots,\mathbf{x_k}) &= \mathbf{e}_1 \\
&\vdots \\
\mathbf{f_n}(\mathbf{x}_1,\ \ldots,\mathbf{x_k}) &= \mathbf{e_n}
\end{aligned}
$$

Grammatikken lever ikke helt op til de regler man kræver i en parsergenerator, men det er alt vi skal bruge for at kunne diskutere semantik. Påstanden er, at det nok skulle være rimeligt enkelt at generalisere grammatikken til at definere et mere brugervenligt sprog. Det kunne da være rart hvis alle funktioner ikke skulle have samme antal argumenter og at de ikke nødvendigvis skulle kaldes de ret intetsigende navne $\mathbf{f}_1,\ \ldots,\mathbf{f_n}$. Problemet er blot, at hvis vi hænger os for meget i de spørgsmål, vi har en glimrende forståelse af, ja så når vi aldrig længere.

Grammatikken ovenfor indeholder to nonterminaler $\mathbf{e}$ og $\mathbf{p}$. Nonterminalerne i en sådan grammatik omtales ofte som *syntaktiske kategorier* idet man opfatter dem som mængder af de tegnstrenge de kan afledes til. Det kan være nyttigt at tænke på det på denne måde, for semantikken er så en afbildning fra de syntaktiske kategorier til nogle passende domæner.

Semantiske funktioner. En fixpunktsemantik konstrueres så ved at definere en særlig semantisk funktion for hver nonterminal i grammatikken. Til hver nonterminal må man så også angive domænet hvori udtrykkenes betydning skal beskrives. For eksemplet ovenfor skal vi altså definere en semantisk funktion E for udtryk og en semantisk funktion P for programmer. Betydningen af udtryk beskrives i et domæne $\mathbf{D}_1$ og betydningen af programmer i et domæne $\mathbf{D}_2$. De semantiske funktioner definerer så betydningen af udtrykkene alene ved udtrykkenes direkte bestanddele. I dette tilfælde får vi altså en række definitioner som

$$
\begin{aligned}
\mathsf{E}[\![\mathbf{c_i}]\!] &= \mathbf{m}_1(\text{“}\mathbf{c_i}\text{”}) \\
\mathsf{E}[\![\mathbf{x_i}]\!] &= \mathbf{m}_2(\text{“}\mathbf{x_i}\text{”}) \\
\mathsf{E}[\![\mathbf{a_i}(\mathbf{e}_1,\ \ldots,\mathbf{e}_\ell)]\!] &= \mathbf{m}_3(\text{“}\mathbf{a_i}\text{”},\mathsf{E}[\![\mathbf{e}_1]\!],\ \ldots,\mathsf{E}[\![\mathbf{e}_\ell]\!]) \\
\mathsf{E}[\![\textbf{if } \mathbf{e}_1 \textbf{ then } \mathbf{e}_2 \textbf{ else } \mathbf{e}_3]\!] &= \mathbf{m}_4(\mathsf{E}[\![\mathbf{e}_1]\!],\mathsf{E}[\![\mathbf{e}_2]\!],\mathsf{E}[\![\mathbf{e}_3]\!])
\end{aligned}
$$

$$
\begin{aligned}
&\mathsf{E}[\![\mathbf{f_i}(\mathbf{e}_1,\ \ldots,\mathbf{e_k})]\!] && = \mathbf{m}_5(\text{“}\mathbf{f_i}\text{”}, \mathsf{E}[\![\mathbf{e}_1]\!],\ \ldots, \mathsf{E}[\![\mathbf{e_k}]\!])\\
&\mathsf{P}[\![\ \mathbf{f}_1(\mathbf{x}_1,\ \ldots,\mathbf{x_k}) = \mathbf{e}_1 \\
&\qquad\qquad\vdots \\
&\quad \mathbf{f_n}(\mathbf{x}_1,\ \ldots,\mathbf{x_k}) = \mathbf{e_n}]\!] && = \mathbf{m}_6(\text{“}\mathbf{f}_1\text{”},\ \ldots,\ \text{“}\mathbf{f_n}\text{”},\ \text{“}\mathbf{x}_1\text{”},\ \ldots,\ \text{“}\mathbf{x_k}\text{”}, \mathsf{E}[\![\mathbf{e}_1]\!],\ \ldots, \mathsf{E}[\![\mathbf{e_n}]\!])
\end{aligned}
$$

hvor symbolerne $\mathbf{m}_1,\ \ldots,\ \mathbf{m}_6$ er navne på givne funktioner af rette type. Der er ikke noget krav om at disse funktioner skal være kontinuerte. Kravet er, at hvis de bruges i fixpunkt udtryk skal de være kontinuerte. De syntaktiske udtryk til de semantiske funktioner omgives med de særlige *semantiske parenteser* $[\![\cdot]\!]$ for at gøre det klart, at det ikke er udtryk, der skal udregnes.

Sammenfattende kan spillereglerne for at angive en fixpunktsemantik opstilles som:

- De semantiske funktioner må alene beskrive meningen af udtryk ved betydningen af deres direkte bestanddele.
- Betydningen af programmer beskrives som elementer i domæner.
- Argumenter til fixpunktoperationer skal være kontinuerte funktioner.

Bemærk at der i disse regler ikke kræves at der optræder fixpunkter i en fixpunktsemantik. Det vil imidlertid være et ret trivielt programmeringssprog, hvis man kan klare sig foruden.

Eksempler. I resten af dette afsnit vil vi præsentere tre eksempler på fixpunktsemantikker. Vi starter med det lille funktionssprog brugt ovenfor. Derefter vil vi beskrive et lille imperativt sprog og et logik sprog. Vi skulle dermed gerne havde dækket de tre væsentligste retninger inden for programmeringssprog. For hvert sprog gælder at den semantik vi giver bestemt ikke er den eneste mulige. Vi skal senere se andre fixpunktsemantikker for sprogene, men de skal gerne i en eller anden forstand være ækvivalente.

4.2 Funktionssprog

Vi vil først give en fixpunktsemantik for funktionssproget vi brugte i eksemplet ovenfor. Sproget blev beskrevet ved en grammatik med to nonterminaler **e** og **p**. Til hver af disse hører så en semantisk funktion: E og P. Vi vil først uformelt diskutere hvilken betydning vi gerne vil tillægge sproget og derefter formalisere det som en fixpunktsemantik.

Betydning. Syntaksen i sig selv kan naturligvis kun antyde hvilken betydning vi har i tankerne med et sprog. Der er derfor en række afgørelser vi skal tage før semantikken formaliseres.

I dette sprog vil vi for det første antage en ivrig (dvs. *call-by-value*) udregning af funktionsargumenter. Vi vil desuden antage at alle udregnede værdier tilhører en mængde $\mathbf{V}$. Denne mængde skal som minimum indeholde de boolske værdier true og false. Til alle konstanter $\mathbf{c_i}$ i sproget skal der være en tilsvarende værdi $\mathbf{const_i} \in \mathbf{V}$ og for standard operationerne $\mathbf{a_i}$ må der være funktioner $\mathbf{basic_i} : \mathbf{V}^\ell \to \mathbf{V}_\perp$.

Semantiske domæner.

$$
\begin{array}{lll}
\mathbf{D} & = \mathbf{V}_\perp & \text{—værdier} \\
\Phi & = (\mathbf{V^k} \to \mathbf{D})^\mathbf{n} & \text{—funktionsbetydninger}
\end{array}
$$

Semantiske funktioner.

$$
\begin{array}{ll}
\mathsf{E}[\![\mathbf{exp}]\!] & : \Phi \to \mathbf{V^k} \to \mathbf{D} \\
\mathsf{P}[\![\mathbf{prog}]\!] & : \Phi
\end{array}
$$

Definition.

$$
\begin{array}{ll}
\mathsf{E}[\![\mathbf{c_i}]\!]\phi\nu & = \mathbf{const_i} \\
\mathsf{E}[\![\mathbf{x_i}]\!]\phi\nu & = \nu_\mathbf{i} \\
\mathsf{E}[\![\mathbf{a_i}(\mathbf{e}_1, \ldots, \mathbf{e}_\ell)]\!]\phi\nu & = \mathbf{let}\ \mathbf{v_j} = \mathsf{E}[\![\mathbf{e_j}]\!]\phi\nu, \mathbf{j} = 1, \ldots \ell\ \mathbf{in} \\
 & \quad \mathbf{if\ some}\ \mathbf{v_j} = \perp\ \mathbf{then}\ \perp\ \mathbf{else} \\
 & \quad \mathbf{basic_i}(\mathbf{v}_1, \ldots, \mathbf{v}_\ell) \\
\mathsf{E}[\![\mathbf{if}\ \mathbf{e}_1\ \mathbf{then}\ \mathbf{e}_2\ \mathbf{else}\ \mathbf{e}_3]\!]\phi\nu & = \mathbf{let}\ \mathbf{v}_1 = \mathsf{E}[\![\mathbf{e}_1]\!]\phi\nu\ \mathbf{in} \\
 & \quad \mathbf{if}\ \mathbf{v}_1 = \mathsf{true}\ \mathbf{then}\ \mathsf{E}[\![\mathbf{e}_2]\!]\phi\nu\ \mathbf{else} \\
 & \quad \mathbf{if}\ \mathbf{v}_1 = \mathsf{false}\,\mathbf{then}\ \mathsf{E}[\![\mathbf{e}_3]\!]\phi\nu\ \mathbf{else} \\
 & \quad \perp \\
\mathsf{E}[\![\mathbf{f_i}(\mathbf{e}_1, \ldots, \mathbf{e_k})]\!]\phi\nu & = \mathbf{let}\ \mathbf{v_j} = \mathsf{E}[\![\mathbf{e_j}]\!]\phi\nu, \mathbf{j} = 1, \ldots \mathbf{k}\ \mathbf{in} \\
 & \quad \mathbf{if\ some}\ \mathbf{v_j} = \perp\ \mathbf{then}\ \perp\ \mathbf{else} \\
 & \quad \phi_\mathbf{i}(\mathbf{v}_1, \ldots, \mathbf{v_k})
\end{array}
$$

og

$$
\begin{array}{l}
\mathsf{P}[\![\ \mathbf{f}_1(\mathbf{x}_1, \ldots, \mathbf{x_k}) = \mathbf{e}_1 \\
\qquad\qquad\vdots \\
\quad \mathbf{f_n}(\mathbf{x}_1, \ldots, \mathbf{x_k}) = \mathbf{e_n}]\!] = \mathsf{fix}\lambda\phi.\ \langle \mathsf{E}[\![\mathbf{e}_1]\!]\phi, \ldots, \mathsf{E}[\![\mathbf{e_n}]\!]\phi \rangle
\end{array}
$$

Semantikken er defineret som det mindste fixpunkt for en funktional $\mathbf{F}$ med typen $\Phi \to \Phi$ og defineret som

$$\mathbf{F}(\phi) = \langle \mathsf{E}[\![\mathbf{e}_1]\!]\phi,\ \ldots,\ \mathsf{E}[\![\mathbf{e_n}]\!]\phi \rangle$$

Dette fixpunkt er veldefineret da funktionalen er kontinuert.

Vi kunne også have udtrykt den semantiske funktion E noget kortere ved at bruge et par hjælpefunktioner. Vi kunne så have skrevet:

$$\begin{aligned}
\mathsf{E}[\![\mathbf{c_i}]\!]\phi\nu &= \mathbf{const_i} \\
\mathsf{E}[\![\mathbf{x_i}]\!]\phi\nu &= \nu_\mathbf{i} \\
\mathsf{E}[\![\mathbf{a_i}(\mathbf{e}_1,\ \ldots,\ \mathbf{e}_\ell)]\!]\phi\nu &= \mathsf{strict}\ \mathbf{basic_i}\langle \mathsf{E}[\![\mathbf{e}_1]\!]\phi\nu,\ \ldots,\ \mathsf{E}[\![\mathbf{e}_\ell]\!]\phi\nu \rangle \\
\mathsf{E}[\![\mathbf{if}\ \mathbf{e}_1\ \mathbf{then}\ \mathbf{e}_2\ \mathbf{else}\ \mathbf{e}_3]\!]\phi\nu &= \mathsf{cond}(\mathsf{E}[\![\mathbf{e}_1]\!]\phi\nu, \mathsf{E}[\![\mathbf{e}_2]\!]\phi\nu, \mathsf{E}[\![\mathbf{e}_3]\!]\phi\nu) \\
\mathsf{E}[\![\mathbf{f_i}(\mathbf{e}_1,\ \ldots,\ \mathbf{e_k})]\!]\phi\nu &= \mathsf{strict}\ \phi_\mathbf{i}\langle \mathsf{E}[\![\mathbf{e}_1]\!]\phi\nu,\ \ldots,\ \mathsf{E}[\![\mathbf{e_k}]\!]\phi\nu \rangle
\end{aligned}$$

med

$$\mathsf{strict}\ \mathbf{f}\langle \mathbf{v}_1,\ \ldots,\ \mathbf{v_k} \rangle = \mathbf{if}\ \mathbf{v}_1 = \bot \vee \cdots \vee \mathbf{v_k} = \bot\ \mathbf{then}\ \bot\ \mathbf{else}\ \mathbf{f}(\mathbf{v}_1,\ \ldots,\ \mathbf{v_k})$$

og hvor cond blev defineret i afsnit 1.

Diskussion. En semantik kan bruges som en definition af hvad betydningen af et program skal være. Mere almindeligt er det dog at vi har en intuitiv forståelse af hvad programmer er, og vi skal så argumentere for at den matematiske beskrivelse svarer hertil. For funktionssprog har vi allerede været igennem denne diskussion i sidste afsnit. Den eneste forskel er at vi her tillader flere gensidigt rekursive funktioner, men dette giver i sig selv ikke ekstra kraft til sproget idet funktionsnummeret kan opfattes som en ekstra parameter til funktionsomgivelserne ϕ. For funktionssproget er semantikken her defineret som grænseværdien af en følge af funktionsomgivelser, hvor den første approximation definerer alle de funktioner, der kan udregnes uden funktionskald; den anden approximation ved også at tillade kald til således definerede funktioner; osv. Det der så skal argumenteres for er at denne udregningsmodel svarer til hvad vi forventer af funktionssproget.

Eksempel. Betragt følgende program

$$\mathbf{f}(\mathbf{x}) = \mathbf{if}\ \mathbf{x} = 0\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{x} * \mathbf{f}(\mathbf{x} - 1)$$

For dette program er $\mathbf{k} = 1$ og $\mathbf{n} = 1$ så $\phi \in \mathbb{N} \to \mathbb{N}_\perp$ og $\nu \in \mathbb{N}$. En direkte udfoldning af de semantiske funktioner giver da

$$\mathsf{fix}\lambda\phi.\ \lambda\nu.\ \mathsf{cond}(\mathsf{strict}\ \mathsf{eq}(\nu, 0), 1, \mathsf{strict}\ \mathsf{mul}(\nu, \mathsf{strict}\ \phi(\mathsf{strict}\ \mathsf{sub}(\nu, 1))))$$

Dette kan simplificeres noget, da kun argumenterne til standardfunktionen mul kan være bundelementet $\perp$.

$$\begin{aligned}&\mathsf{P}[\![\mathbf{f}(\mathbf{x}) = \mathbf{if}\ \mathbf{x} = 0\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{x} * \mathbf{f}(\mathbf{x} - 1)]\!] = \\ &\qquad \mathsf{fix}\lambda\phi.\ \lambda\nu.\ \mathsf{cond}(\nu = 0, 1, \nu *_\perp \phi(\nu - 1))\end{aligned}$$

Dette ser måske ikke umiddelbart særligt imponerende ud. Oversættelsen er meget direkte og viser os måske ikke så meget nyt om programmet. Domæneteorien er i bund og grund funktionel så at give semantikken for et lille funktionssprog er derfor meget enkelt. Når vi går videre til andre sprog er denne process lidt mindre triviel.

4.3 Imperativt sprog

Det næste sprog vi vil betragte er et lille imperativt sprog. Syntaksen er fastlagt ved nogle få produktioner

$$\begin{aligned}\mathbf{p} ::= {}& \mathbf{null} \\ \mid\ \ & \mathbf{x_i} := \mathbf{e} \\ \mid\ \ & \mathbf{p}_1; \mathbf{p}_2 \\ \mid\ \ & \mathbf{if\ b\ then\ p_1\ else\ p_2} \\ \mid\ \ & \mathbf{while\ b\ do\ p_1} \\ \mathbf{e} ::= {}& \mathbf{x_i} \\ \mid\ \ & \mathbf{c_i} \\ \mid\ \ & \mathbf{e}_1\ \mathbf{iop_j}\ \mathbf{e}_2 \\ \mathbf{b} ::= {}& \mathbf{true} \mid \mathbf{false} \\ \mid\ \ & \mathbf{e}_1\ \mathbf{bop_j}\ \mathbf{e}_2\end{aligned}$$

Semantiske domæner. Vi antager at der er et fast antal $\mathbf{m}$ af variabler $\mathbf{x_i}$. Vi vil desuden antage at værdier af variabler er naturlige tal $\mathbb{N}$. Lageret på et givet tidspunkt kan derfor beskrives ved et element i $\mathbb{N}^{\mathbf{m}}$. Betydningen af et program er en afbildning af lagerindhold før start til lagerindhold efter udførelsen. Dette kan beskrives som en funktion af type $\mathbb{N}^{\mathbf{m}} \to (\mathbb{N}^{\mathbf{m}})_\perp$ hvor bundelementet betyder at programmet ikke terminerer eller at der har været en køretidsfejl.

I udtryk kan der forekomme to slags standardoperationer, $\mathbf{iop_j}$ for heltalsoperationer og $\mathbf{bop_j}$ for boolske operationer (operationer på heltal der returnerer boolske værdier). Til hver af disse forventer vi givet en betydning $\underline{\mathbf{iop}}_\mathbf{j}$ og $\underline{\mathbf{bop}}_\mathbf{j}$

$$\underline{\mathbf{iop}}_\mathbf{j} : \mathbb{N}_\perp \times \mathbb{N}_\perp \to \mathbb{N}_\perp$$
$$\underline{\mathbf{bop}}_\mathbf{j} : \mathbb{N}_\perp \times \mathbb{N}_\perp \to \mathbb{B}_\perp$$

til konstanter $\mathbf{c_i}$ skal der være givet elementer $\mathbf{const_i} \in \mathbb{N}$.

Semantiske funktioner.

$$\mathsf{P}[\![\mathbf{p}]\!] : \mathbb{N}^{\mathbf{m}} \to (\mathbb{N}^{\mathbf{m}})_\perp$$
$$\mathsf{E}[\![\mathbf{e}]\!] : \mathbb{N}^{\mathbf{m}} \to \mathbb{N}_\perp$$
$$\mathsf{B}[\![\mathbf{b}]\!] : \mathbb{N}^{\mathbf{m}} \to \mathbb{B}_\perp$$

Definition.

$$
\begin{aligned}
&\mathsf{P}[\![\mathbf{null}]\!]\sigma && = \sigma\\
&\mathsf{P}[\![\mathbf{x_i} := \mathbf{e}]\!]\sigma && = \mathsf{update}(\mathbf{i}, \mathsf{E}[\![\mathbf{e}]\!]\sigma, \sigma)\\
&\mathsf{P}[\![\mathbf{p}_1; \mathbf{p}_2]\!]\sigma && = \mathsf{appl}(\mathsf{P}[\![\mathbf{p}_2]\!], \mathsf{P}[\![\mathbf{p}_1]\!]\sigma)\\
&\mathsf{P}[\![\mathbf{if\ b\ then\ p}_1\ \mathbf{else\ p}_2]\!]\sigma && = \mathsf{cond}(\mathsf{B}[\![\mathbf{b}]\!]\sigma, \mathsf{P}[\![\mathbf{p}_1]\!]\sigma, \mathsf{P}[\![\mathbf{p}_2]\!]\sigma)\\
&\mathsf{P}[\![\mathbf{while\ b\ do\ p}_1]\!]\sigma && = \mathsf{repeat}(\mathsf{B}[\![\mathbf{b}]\!], \mathsf{P}[\![\mathbf{p}_1]\!], \sigma)\\
&\mathsf{E}[\![\mathbf{x_i}]\!]\sigma && = \sigma_\mathbf{i}\\
&\mathsf{E}[\![\mathbf{c_i}]\!]\sigma && = \mathbf{const_i}\\
&\mathsf{E}[\![\mathbf{e}_1\ \mathbf{iop_j}\ \mathbf{e}_2]\!]\sigma && = \underline{\mathbf{iop}}_\mathbf{j}(\mathsf{E}[\![\mathbf{e}_1]\!]\sigma, \mathsf{E}[\![\mathbf{e}_2]\!]\sigma)\\
&\mathsf{B}[\![\mathbf{true}]\!]\sigma && = \mathsf{true}\\
&\mathsf{B}[\![\mathbf{false}]\!]\sigma && = \mathsf{false}\\
&\mathsf{B}[\![\mathbf{e}_1\ \mathbf{bop_j}\ \mathbf{e}_2]\!]\sigma && = \underline{\mathbf{bop}}_\mathbf{j}(\mathsf{E}[\![\mathbf{e}_1]\!]\sigma, \mathsf{E}[\![\mathbf{e}_2]\!]\sigma)
\end{aligned}
$$

med

$$\begin{aligned}
&\mathsf{update}(\mathbf{i}, \mathbf{d}, \sigma) = \mathbf{if}\ \mathbf{d} = \bot\ \mathbf{then}\ \bot\ \mathbf{else}\ \langle \sigma_1,\ \ldots, \sigma_{\mathbf{i}-1}, \mathbf{d}, \sigma_{\mathbf{i}+1},\ \ldots, \sigma_{\mathbf{m}} \rangle \\
&\mathsf{appl}(\mathbf{f}, \sigma) = \mathsf{cond}(\sigma = \bot, \bot, \mathbf{f}(\sigma)) \\
&\mathsf{repeat}(\mathbf{b}, \mathbf{f}, \sigma) = \mathsf{cond}(\mathsf{appl}(\mathbf{b}, \sigma), \mathsf{repeat}(\mathbf{b}, \mathbf{f}, \mathsf{appl}(\mathbf{f}, \sigma)), \sigma)
\end{aligned}$$

Hvor repeat her er udtrykt rekursivt. Vi burde derfor vise at funktionen kan defineres som et fixpunkt.

$$\mathsf{repeat} = \mathsf{fix}\lambda\mathbf{F}.\ \lambda\langle \mathbf{b}, \mathbf{f}, \sigma\rangle.\ \mathsf{cond}(\mathsf{appl}(\mathbf{b}, \sigma), \mathbf{F}(\mathbf{b}, \mathbf{f}, \mathsf{appl}(\mathbf{f}, \sigma)), \sigma)$$

Diskussion. Semantikken er her ikke beskrevet ved et globalt fixpunkt, men ved fixpunkter for hver **while** sætning i programmet. De endelige approximationer til beskrivelsen af disse sætninger følger mønsteret fra funktionssprog: I den første approximation beskrives **while** sætninger hvor betingelsen er falsk, i den anden approximation sætninger hvor betingelsen bliver falsk før første eller andet gennemløb, osv. Herved får man altså givet mening til **while** sætninger hvor betingelsen bliver falsk efter et endeligt antal gennemløb. Sætninger, hvor betingelsen altid er sand, beskrives ved en afbildning til bundelementet blandt lagerindhold.

Eksempel. Betragt programmet

$$\begin{aligned}
&\mathbf{x}_2 := 1; \\
&\mathbf{while}\ \mathbf{x}_1 > 0\ \mathbf{do} \\
&\quad \mathbf{x}_2 := \mathbf{x}_2 * \mathbf{x}_1; \\
&\quad \mathbf{x}_1 := \mathbf{x}_1 - 1;
\end{aligned}$$

En direkte udfoldning af de semantiske funktioner giver her

$$\begin{aligned}
\lambda\sigma.\ \mathsf{appl}(\ &\lambda\sigma^1.\ \mathsf{repeat}(\ \lambda\sigma^2.\ \sigma^2_1 > 0, \\
&\qquad \lambda\sigma^3.\ \mathsf{appl}(\lambda\sigma^4.\ \mathsf{update}(1, \sigma^4_1 - 1, \sigma^4), \mathsf{update}(2, \sigma^3_2 * \sigma^3_1, \sigma^3)), \\
&\qquad \sigma^1), \\
&\mathsf{update}(2, 1, \sigma))
\end{aligned}$$

Dette kan forenkles noget til

$$\lambda\sigma.\ \mathsf{repeat}(\ \lambda\sigma^2.\ \sigma^2_1 > 0, \lambda\sigma^3.\ \langle \sigma^3_1 - 1, \sigma^3_2 * \sigma^3_1\rangle, \mathsf{update}(2, 1, \sigma))$$

Et videre forsøg på at vise at denne funktion er ækvivalent med fakultetsfunktionen må vente til næste afsnit.

4.4 Logiksprog

Vort sidste eksempel er et lille sprog af logikprogrammer. Sammenlignet med de sidste to semantikker er dette eksempel nok interessant idet vi her for alvor har bevæget os væk fra en denotationel semantik med en beregnelig model. Vi præsenterer en semantik, der giver god mening i domæneteorien, men som ikke direkte kan opfattes som en fortolker for sproget.

Et program består af en række klausuler, som enten er betingede prædikater eller fakta. Modsat Prolog er rækkefølgen af klausuler ikke afgørende i dette sprog.

$$
\begin{array}{lll}
\mathbf{p} & ::= & \mathbf{cl}_1 \cdots \mathbf{cl_j} \\
\mathbf{cl} & ::= & \mathbf{a}_0 \text{ :- } \mathbf{a}_1,\ \ldots,\ \mathbf{a_n}. \\
 & \mid & \mathbf{a}_0. \\
\mathbf{a} & ::= & \mathbf{P_j}(\mathbf{t}_1,\ \ldots,\ \mathbf{t}_\ell) \\
\mathbf{t} & ::= & \mathbf{x_i} \\
 & \mid & \mathbf{c_i} \\
 & \mid & \mathbf{f_j}(\mathbf{t}_1,\ \ldots,\ \mathbf{t_k})
\end{array}
$$

Semantiske domæner. Basis for semantikken er begrebet *grundtermer*. En term siges at være en grundterm hvis den ikke indeholder frie variabler. Tilsvarende er grundprædikaterne de prædikater, der ikke indeholder frie variabler. Mængderne af grundtermer og grundprædikater kan defineres som

$$
\begin{array}{lll}
\mathbf{GT} & ::= & \underline{\mathbf{f}}_{\mathbf{j}}(\mathbf{GT}_1,\ \ldots,\ \mathbf{GT_k}) \\
 & \mid & \underline{\mathbf{c}}_{\mathbf{i}} \\
\mathbf{GP} & ::= & \underline{\mathbf{P}}_{\mathbf{j}}(\mathbf{GT}_1,\ \ldots,\ \mathbf{GT}_\ell)
\end{array}
$$

En substitution er en afbildning af variabler til grundtermer. Lad **Var** være mængden af variabler

$$\mathbf{Var} = \{\mathbf{x}_1,\ \ldots,\ \mathbf{x_m}\}$$

En substitution er derfor en funktion i $\mathbf{Var} \to \mathbf{GT}$. I semantikken skal vi bruge mængden af substitutioner og definerer derfor

$$\Theta = \mathbf{Var} \to \mathbf{GT}$$

Betydningen af et program vil være mængden af de grundprædikater, der kan afledes fra programmet ved gentagen udfoldning af højresider og brug af substitutioner.

Semantiske funktioner.

$$\begin{array}{ll} \mathsf{P}[\![\mathbf{p}]\!] & : \mathcal{P}(\mathbf{GP}) \\ \mathsf{C}[\![\mathbf{cl}]\!] & : \mathcal{P}(\mathbf{GP}) \to \mathcal{P}(\mathbf{GP}) \\ \mathsf{A}[\![\mathbf{a}]\!] & : (\mathbf{Var} \to \mathbf{GT}) \to \mathbf{GP} \\ \mathsf{T}[\![\mathbf{t}]\!] & : (\mathbf{Var} \to \mathbf{GT}) \to \mathbf{GT} \end{array}$$

Definition.

$$\begin{array}{ll} \mathsf{P}[\![\mathbf{cl}_1 \cdots \mathbf{cl_j}]\!] & = \mathsf{fix}\lambda\mathbf{s}.\ (\mathsf{C}[\![\mathbf{cl}_1]\!]\mathbf{s} \cup \cdots \cup \mathsf{C}[\![\mathbf{cl_j}]\!]\mathbf{s}) \\ \mathsf{C}[\![\mathbf{a}_0 \text{ :- } \mathbf{a}_1,\ \ldots,\ \mathbf{a_n}.\ ]\!]\mathbf{s} & = \{\mathsf{A}[\![\mathbf{a}_0]\!]\theta \mid \theta \in \Theta \wedge \forall \mathbf{j}.\ \mathsf{A}[\![\mathbf{a_j}]\!]\theta \in \mathbf{s}\} \\ \mathsf{C}[\![\mathbf{a}_0.\ ]\!]\mathbf{s} & = \{\mathsf{A}[\![\mathbf{a}_0]\!]\theta \mid \theta \in \Theta\} \\ \mathsf{A}[\![\mathbf{P_j}(\mathbf{t}_1,\ \ldots,\ \mathbf{t}_\ell)]\!]\theta & = \underline{\mathbf{P}}_\mathbf{j}(\mathsf{T}[\![\mathbf{t}_1]\!]\theta,\ \ldots,\ \mathsf{T}[\![\mathbf{t}_\ell]\!]\theta) \\ \mathsf{T}[\![\mathbf{x_i}]\!]\theta & = \theta(\mathbf{x_i}) \\ \mathsf{T}[\![\mathbf{c_i}]\!]\theta & = \underline{\mathbf{c}}_\mathbf{i} \\ \mathsf{T}[\![\mathbf{f_j}(\mathbf{t}_1,\ \ldots,\ \mathbf{t_k})]\!]\theta & = \underline{\mathbf{f}}_\mathbf{j}(\mathsf{T}[\![\mathbf{t}_1]\!]\theta,\ \ldots,\ \mathsf{T}[\![\mathbf{t_k}]\!]\theta) \end{array}$$

Diskussion. Modsat de to tidligere semantikker beskrives programmer her ikke ved funktioner, men ved en mængde. Når vi skal undersøge fixpunktets egenskaber gælder der atter at det er tilstrækkeligt at undersøge approximationerne. For et givent program og et givet grundprædikat gælder nemlig at det er med i fixpunktet hvis og kun hvis det er med i approximationerne fra et vist trin.

De semantiske funktioner **A** og **T** omformer blot en term eller et prædikat til en grundterm eller et grundprædikat ved at anvende en substitution på udtrykket. Hvis vi skulle have gjort det lidt simplere kunne vi blot have skrevet $\theta(\mathbf{a_i})$ i stedet for $\mathsf{A}[\![\mathbf{a_i}]\!]\theta$ og så udeladt **A** og **T**. Det ser man da også ofte gjort i litteraturen, men ret beset er det en lidt ufin blanding af syntaks og semantik.

Den interessante semantiske funktion er **C**. For fakta, dvs. klausuler uden en højreside, beskrives betydningen ved at erstatte alle variabler i prædikatet med grundtermer. Det væsentlige er her at hvis den samme variabel optræder flere gange skal hver forekomst erstattes med samme grundterm. Betydningen er mængden af de mulige grundprædikater der fremkommer ved sådanne substitutioner. For klausuler med en højreside følger semantikken samme mønster idet man dog der kun medtager de substitutioner, hvor alle prædikaterne på højresiden er vist at være mulige afledninger. Man siger også ofte at man beskriver betydningen ved succes-mængden for prædikatet.

Eksempel. Betragt programmet

$$\mathbf{append}(\mathbf{nil}, \mathbf{x}, \mathbf{x}).$$
$$\mathbf{append}(\mathbf{cons}(\mathbf{x}, \mathbf{y}), \mathbf{w}, \mathbf{cons}(\mathbf{x}, \mathbf{v})) \text{ :- } \mathbf{append}(\mathbf{y}, \mathbf{w}, \mathbf{v}).$$

En udfoldning af de semantiske funktioner giver

$$\begin{aligned}\mathsf{fix}\lambda\mathbf{s}.\ (\ &\{\underline{\mathbf{append}}(\underline{\mathbf{nil}}, \mathbf{t}, \mathbf{t}) \mid \mathbf{t} \in \mathbf{GT}\} \cup \\ &\{\underline{\mathbf{append}}(\underline{\mathbf{cons}}(\mathbf{t}, \mathbf{t}_1), \mathbf{t}_2, \underline{\mathbf{cons}}(\mathbf{t}, \mathbf{t}_3)) \mid \mathbf{t} \in \mathbf{GT} \wedge \underline{\mathbf{append}}(\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3) \in \mathbf{s}\})\end{aligned}$$

Dette udtryk kan forenkles yderligere ved for eksempel at definere den karakteristiske funktion for mængden (funktionen, der tester om et givet element ligger i mængden).

4.5 Opgaver

Opgave 4.1. Omskriv funktionsprogrammet nedenfor til et domæneteoretisk udtryk og simplificer det så vidt muligt.

$$\mathbf{fib}(\mathbf{x}) = \mathbf{if}\ \mathbf{x} < 2\ \mathbf{then}\ 1\ \mathbf{else}\ \mathbf{fib}(\mathbf{x} - 1) + \mathbf{fib}(\mathbf{x} - 2)$$

Opgave 4.2. Omskriv det imperative program nedenfor til et domæneteoretisk udtryk og simplificer det så vidt muligt.

$$\begin{aligned}&\mathbf{x}_3 := 0;\\ &\mathbf{while}\ \mathbf{x}_1 > \mathbf{x}_2\ \mathbf{do}\\ &\quad \mathbf{x}_1 := \mathbf{x}_1 - \mathbf{x}_2;\\ &\quad \mathbf{x}_3 := \mathbf{x}_3 + 1;\end{aligned}$$

Opgave 4.3. Vis at

$$\mathsf{P}[\![\mathbf{while\ true\ do\ null}]\!] = \lambda\sigma.\ \bot$$

Opgave 4.4. Udvid det imperative sprog med en **repeat** sætning:

repeat p until b

og definer dens semantik.

Opgave 4.5. Omskriv logikprogrammet nedenfor til et domæneteoretisk udtryk og simplificer det så vidt muligt.

$$\mathbf{member}(\mathbf{x}, \mathbf{cons}(\mathbf{x}, \mathbf{y})).$$
$$\mathbf{member}(\mathbf{x}, \mathbf{cons}(\mathbf{z}, \mathbf{y})) \text{ :- } \mathbf{member}(\mathbf{x}, \mathbf{y}).$$

5 Fixpunktinduktion

Vi har nu set hvordan fixpunktsætningen kan bruges til at definere rekursive funktioner, og vi har set hvordan den kan bruges som basis for at konstruere en matematisk beskrivelse af programmeringssprog. Vi har hver gang uformelt argumenteret for de herved definerede objekters egenskaber. Vi har således undersøgt de endelige approximationer til fixpunkter og derfra sluttet os til egenskaber ved fixpunktet. Sådanne argumenter ligner sædvanlig induktion, men der er en forskel. Ved alene at undersøge approximationerne kan vi godt nok slutte os til at alle approximationerne har en given egenskab, men det i sig selv siger jo ikke noget om fixpunktet. Vi har brug for en stærkere sætning, og den kommer her.

5.1 Fixpunktinduktion

Definition. Et prædikat $\mathbf{p} : \mathbf{D} \to \mathbb{B}$ på et domæne $\mathbf{D}$ siges at være *induktivt* hvis og kun hvis der for alle kæder $(\mathbf{x_i}) \in \mathsf{kæder}(\mathbf{D})$ gælder

$$\mathbf{p}(\bot)$$

og

$$(\forall \mathbf{i}.\ \mathbf{p}(\mathbf{x_i})) \Rightarrow \mathbf{p}(\bigsqcup_{\mathbf{i}} \mathbf{x_i})$$

Kært barn har mange navne: Induktive prædikater kaldes på engelsk også for *admissible*, *inductive*, *inclusive* eller ω*-complete*.

Sætning. Lad $\mathbf{f}$ være en kontinuert funktion $\mathbf{f} : \mathbf{D} \to \mathbf{D}$ og $\mathbf{p} : \mathbf{D} \to \mathbb{B}$ et induktivt prædikat med egenskaben $\forall \mathbf{d} \in \mathbf{D}.\ \mathbf{p}(\mathbf{d}) \Rightarrow \mathbf{p}(\mathbf{f}(\mathbf{d}))$, da gælder $\mathbf{p}(\mathsf{fix}(\mathbf{f}))$.

Bevis. Fixpunktet for en funktion $\mathbf{f} : \mathbf{D} \to \mathbf{D}$ findes netop som en grænseværdi af en kæde i $\mathbf{D}$. Ved induktion følger at alle kædens elementer opfylder prædikatet. Da prædikatet er induktivt vil grænseværdien også opfylde prædikatet.

Eksempler. Vi kan hurtigt komme med nogle eksempler på induktive prædikater:

Lad $\mathbf{D}$ være et domæne og $\mathbf{c} \in \mathbf{D}$ et vilkårligt element i $\mathbf{D}$. Prædikatet

$$\mathbf{p}(\mathbf{x}) = (\mathbf{x} \sqsubseteq \mathbf{c})$$

er da induktivt.

Modeksempler. Det er nok også på plads at give et par eksempler på prædikater, der ikke er induktive.

Det er ganske let at konstruere et ikke induktivt prædikat ved blot at sørge for det er falsk for mindsteelementet:

$$\mathbf{p}(\mathbf{x}) = \mathbf{falsk}$$

eller

$$\mathbf{p}(\mathbf{x}) = (\mathbf{x} \neq \perp)$$

Det er straks vanskeligere at finde prædikater, som er sande for mindsteelementet, men ikke induktive. Et eksempel er dog

$$\mathbf{p} : \mathbb{N}^{\infty} \rightarrow \mathbb{B}$$
$$\mathbf{p}(\mathbf{x}) = (\mathbf{x} \neq \infty)$$

hvor $\mathbb{N}^{\infty} = \mathbb{N} \cup \{\infty\}$ er ordnet med heltalsordningen hvor 0 er mindste element og ∞ er største element. I domænet kan vi betragte kæden

$$0 \leq 1 \leq 2 \leq \cdots$$

hvis elementer alle opfylder prædikatet. Grænseværdien af kæden, derimod, er ∞ og opfylder ikke prædikatet. Konstruktionen her minder iøvrigt om eksemplet på en monoton, men ikke kontinuert funktion.

5.2 Eksempel

I sidste afsnit beskrev vi fakultetsfunktionen i både funktionssproget og i det imperative sprog. Forskellen var ikke stor, men den er der alligevel. Essensen af det imperative program er en fixpunktdefinition.

$$\mathbf{G} : (\mathbb{N}^2 \rightarrow (\mathbb{N}^2)_\perp) \rightarrow (\mathbb{N}^2 \rightarrow (\mathbb{N}^2)_\perp)$$
$$\mathbf{G} = \lambda\mathbf{g}.\ \lambda\langle\mathbf{x}, \mathbf{y}\rangle.\ \mathsf{cond}(\mathbf{x} > 0, \mathbf{g}(\mathbf{x} - 1, \mathbf{y} * \mathbf{x}), \langle\mathbf{x}, \mathbf{y}\rangle)$$
$$\mathbf{f} = \mathsf{fix}\mathbf{G}$$

Med denne definition vil vi forvente at

$$\mathbf{f}(\mathbf{x}, 1) = \langle\mathbf{x}', \mathbf{y}'\rangle \Rightarrow \mathbf{y}' = \mathbf{x}!$$

Det er nok ikke så overraskende da funktionen $\mathbf{f}$ ligner den funktionelle definition af fakultetsfunktionen til forveksling. De ligner nok hinanden, men der er dog en forskel idet multiplikationerne her sker i den modsatte rækkefølge. Det betyder at vi ikke kan vise en ækvivalens ved blot at lave nogle simple omskrivninger. Bemærk for eksempel at i funktionalen $\mathbf{G}$ bruger vi almindelig multiplikation og ikke den *løftede* multiplikation $*_\perp$ som for den funktionelle definition. Det skyldes at $\mathbf{x}$ og $\mathbf{y}$ begge er elementer i $\mathbb{N}$ og de kan derfor ikke være udefinerede ($\perp$).

Ækvivalens. Betragt følgende prædikat for funktioner $\mathbf{g}$.

$$\forall \mathbf{x}, \mathbf{y}, \mathbf{x}', \mathbf{y}' \in \mathbb{N}.\ \mathbf{g}(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}', \mathbf{y}' \rangle \Rightarrow \mathbf{y}' = \mathbf{y} * \mathbf{x}!$$

I ord vil det sige, at for alle $\langle \mathbf{x}, \mathbf{y} \rangle$ vil $\mathbf{g}(\mathbf{x}, \mathbf{y})$ enten være mindsteelementet $\perp$ eller også vil anden komponenten være produktet af $\mathbf{y}$ og $\mathbf{x}$ fakultet. Påstanden er at dette prædikat er opfyldt for $\mathsf{fix}(\mathbf{G})$.

Bevis. Beviset har to dele: Først skal vi vise at prædikatet er induktivt, dernæst skal vi vise at alle approximationerne til $\mathbf{f}$ (dvs. $\mathbf{G}^{\mathbf{i}}(\lambda\langle \mathbf{x}, \mathbf{y} \rangle.\ \perp)$) opfylder prædikatet.

At prædikatet er induktivt følger af at grænseværdien af en kæde af funktioner er defineret elementvis. For en kæde $(\mathbf{g}_\mathbf{i})$ gælder nemlig

$$(\bigsqcup_{\mathbf{i}} \mathbf{g}_\mathbf{i})\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{x}', \mathbf{y}' \rangle \Rightarrow \exists \mathbf{j}.\ \mathbf{g}_\mathbf{j}\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{x}', \mathbf{y}' \rangle$$

Det følger heraf at hvis alle approximationerne opfylder prædikatet vil også grænseværdien opfylde det. Bemærk iøvrigt at et prædikat som $\mathbf{q}(\mathbf{h}) = \exists\langle \mathbf{x}, \mathbf{y} \rangle.\ \mathbf{h}(\mathbf{x}, \mathbf{y}) = \perp$ ikke er induktivt på dette domæne.

Beviset for at alle approximationerne til $\mathbf{f}$ opfylder prædikatet benytter sædvanlig induktion. Det er let nok at vise at dette er opfyldt for bundelementet. Mindsteelementet i domænet er funktionen der afbilder alle par af tal til mindsteelementet $\perp$.

Vi skal nu antage at betingelsen er opfyldt for en funktion $\mathbf{g}$ og vise at det så også er opfyldt for $\mathbf{G}(\mathbf{g})$. Ved at bruge induktionsantagelsen får vi

$$\begin{aligned}\mathsf{snd}((\mathbf{G}(\mathbf{g}))\langle \mathbf{x}, \mathbf{y} \rangle) &= \mathsf{cond}(\mathbf{x} > 0, \mathsf{snd}(\mathbf{g}(\mathbf{x} - 1, \mathbf{y} * \mathbf{x})), \mathbf{y}) \\ &= \mathsf{cond}(\mathbf{x} > 0, (\mathbf{y} * \mathbf{x}) * (\mathbf{x} - 1)!, \mathbf{y})\end{aligned}$$

og videre simplifikation giver

$$\begin{aligned}&\mathsf{cond}(\mathbf{x} > 0, (\mathbf{y} * \mathbf{x}) * (\mathbf{x} - 1)!, \mathbf{y}) \\ =\ &\mathsf{cond}(\mathbf{x} > 0, \mathbf{y} * \mathbf{x}!, \mathbf{y}) \\ =\ &\mathbf{y} * \mathbf{x}!\end{aligned}$$

hvilket var hvad vi skulle vise.

Bevis slut

5.3 Induktive relationer

Man kan lave en del variationer over princippet for fixpunktinduktion. En hyppigt forekommende situation er at man gerne vil vise en relation mellem to rekursivt definerede funktioner.

Definition. Lad $\mathbf{D}$ og $\mathbf{E}$ være domæner og $\mathbf{R}$ en relation mellem elementer i $\mathbf{D}$ og $\mathbf{E}$. Relationen siges at være induktiv hvis og kun hvis der for alle voksende kæder $(\langle \mathbf{x_i}, \mathbf{y_i} \rangle) \in \mathsf{kæder}(\mathbf{D} \times \mathbf{E})$ gælder

$$\perp_{\mathbf{D}} \mathbf{R} \perp_{\mathbf{E}}$$

og

$$(\forall \mathbf{i}.\ \mathbf{x_i}\ \mathbf{R}\ \mathbf{y_i}) \Rightarrow (\bigsqcup_{\mathbf{i}} \mathbf{x_i})\ \mathbf{R}\ (\bigsqcup_{\mathbf{i}} \mathbf{y_i})$$

Fixpunktinduktion. Lad $\mathbf{R}$ være en induktiv relation mellem elementer i domænerne $\mathbf{D}$ og $\mathbf{E}$ og lad $\mathbf{f} : \mathbf{D} \to \mathbf{D}$ og $\mathbf{g} : \mathbf{E} \to \mathbf{E}$ være kontinuerte funktioner med egenskaberne:

$$\forall \mathbf{a} \in \mathbf{D}, \mathbf{b} \in \mathbf{E}.\ \mathbf{a}\ \mathbf{R}\ \mathbf{b} \Rightarrow \mathbf{f}(\mathbf{a})\ \mathbf{R}\ \mathbf{g}(\mathbf{b})$$

Fixpunkterne for $\mathbf{f}$ og $\mathbf{g}$ vil da også være relaterede med $\mathbf{R}$: $\mathsf{fix}(\mathbf{f})\ \mathbf{R}\ \mathsf{fix}(\mathbf{g})$.

Lemma. Lad $\mathbf{R}$ og $\mathbf{S}$ være induktive relationer mellem elementer i domænerne $\mathbf{D}$ og $\mathbf{E}$. Da gælder det samme for relationerne

$$\mathbf{x}\ \mathbf{R}{\cup}\mathbf{S}\ \mathbf{y} \Leftrightarrow \mathbf{x}\ \mathbf{R}\ \mathbf{y} \vee \mathbf{x}\ \mathbf{S}\ \mathbf{y}$$
$$\mathbf{x}\ \mathbf{R}{\cap}\mathbf{S}\ \mathbf{y} \Leftrightarrow \mathbf{x}\ \mathbf{R}\ \mathbf{y} \wedge \mathbf{x}\ \mathbf{S}\ \mathbf{y}$$

Advarsel. Lad $\mathbf{R}$ og $\mathbf{S}$ være induktive relationer mellem elementer i domænerne $\mathbf{D}$ og $\mathbf{E}$. Da er relationen

$$\mathbf{x}\ \mathbf{R}{\setminus}\mathbf{S}\ \mathbf{y} \Leftrightarrow \mathbf{x}\ \mathbf{R}\ \mathbf{y} \wedge \neg(\mathbf{x}\ \mathbf{S}\ \mathbf{y})$$

ikke nødvendigvis induktiv. Hvis $\mathbf{R}$ er en induktiv relation mellem elementer i domænerne $\mathbf{C}$ og $\mathbf{D}$, og $\mathbf{S}$ en induktiv relation mellem elementer i domænerne $\mathbf{D}$ og $\mathbf{E}$, da er sammensætningen

$$\mathbf{x}\ \mathbf{S}{\circ}\mathbf{R}\ \mathbf{y} \Leftrightarrow \exists \mathbf{z} \in \mathbf{D}.\ \mathbf{x}\ \mathbf{R}\ \mathbf{z} \wedge \mathbf{z}\ \mathbf{S}\ \mathbf{y}$$

ikke nødvendigvis induktiv.

Eksempel. Lad $\mathbf{f} : \mathbf{D} \to \mathbf{E}$ være en kontinuert funktion. Da er følgende relationer induktive

$$\mathbf{x}\ \mathbf{R}_1\ \mathbf{y} \Leftrightarrow \mathbf{f}(\mathbf{x}) \sqsupseteq \mathbf{y}$$
$$\mathbf{x}\ \mathbf{R}_2\ \mathbf{y} \Leftrightarrow \mathbf{f}(\mathbf{x}) \sqsubseteq \mathbf{y} \vee \mathbf{x} = \perp$$

Sætning. En relation $\mathbf{R}$ er induktiv hvis den er striks

$$\perp \mathbf{R} \perp$$

og har formen

$$\mathbf{x}\ \mathbf{R}\ \mathbf{y} \Leftrightarrow \langle \mathbf{IR} \rangle$$

med $\langle \mathbf{IR} \rangle$ som et udtryk i $\mathbf{x}$ og $\mathbf{y}$ der følger regler

$$\begin{aligned} \langle \mathbf{IR} \rangle &::= \langle \mathbf{IR} \rangle \wedge \langle \mathbf{IR} \rangle \,|\, \forall \mathbf{d_i} \in \mathbf{D_i} : \langle \mathbf{ER} \rangle \\ \langle \mathbf{ER} \rangle &::= \langle \mathbf{ER} \rangle \vee \langle \mathbf{ER} \rangle \,|\, \mathbf{Q_j}(\mathbf{d_i}) \,|\, \mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{d_i}) \sqsubseteq \mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{d_i}) \end{aligned}$$

hvor $\mathbf{Q_j}$ er et første-ordens prædikat og $\mathbf{f}$ og $\mathbf{g}$ er to kontinuerte funktioner.

Logiske relationer. Induktive relationer kan også konstrueres ved at løfte relationer mellem domæner til produkter, funktionsrum og sumdomæner. Lad $\mathbf{R}$ være en induktiv relation mellem elementer i domæner $\mathbf{D}_1$ og $\mathbf{D}_2$ og lad $\mathbf{S}$ være en induktiv relation mellem elementer i domæner $\mathbf{E}_1$ og $\mathbf{E}_2$. Vi kan da også danne de følgende induktive relationer

$$\begin{aligned} &\forall \langle \mathbf{d}_1, \mathbf{e}_1 \rangle \in \mathbf{D}_1 \times \mathbf{E}_1, \langle \mathbf{d}_2, \mathbf{e}_2 \rangle \in \mathbf{D}_2 \times \mathbf{E}_2. \\ &\qquad \langle \mathbf{d}_1, \mathbf{e}_1 \rangle\ \mathbf{R{\times}S}\ \langle \mathbf{d}_2, \mathbf{e}_2 \rangle \Leftrightarrow \mathbf{d}_1\ \mathbf{R}\ \mathbf{d}_2 \wedge \mathbf{e}_1\ \mathbf{S}\ \mathbf{e}_2 \end{aligned}$$

og

$$\begin{aligned} &\forall \mathbf{f} : \mathbf{D}_1 \to \mathbf{E}_1, \mathbf{g} : \mathbf{D}_2 \to \mathbf{E}_2. \\ &\qquad \mathbf{f}\ \mathbf{R{\to}S}\ \mathbf{g} \Leftrightarrow (\forall \mathbf{d}_1 \in \mathbf{D}_1, \mathbf{d}_2 \in \mathbf{D}_2.\ \mathbf{d}_1\ \mathbf{R}\ \mathbf{d}_2 \Rightarrow \mathbf{f}(\mathbf{d}_1)\ \mathbf{S}\ \mathbf{g}(\mathbf{d}_2)\ ) \end{aligned}$$

En lignende metode kan bruges for sumdomæner (se opgaverne). Relationer, der på denne måde er løftet op igennem en typestruktur, omtales ofte som *logiske relationer*. De er som sådan ikke mere logiske end så meget andet, men man sparer jo nogle definitioner og beviser når man får nogle komplicerede relationer foræret fra nogle relationer over grunddomænerne i en typestruktur. Det er da meget logisk.

5.4 Indlejring

Der er masser af andre variationer over princippet for fixpunktinduktion. Vi skal her til sidst se på en brug i forbindelse med *indlejring* af et domæne i et andet. Denne metode har traditionelt haft stor betydning for bevisteknikkerne i abstrakt fortolkning. Vi får også brug for indlejringer i næste kapitel, hvor vi skal kikke på løsninger af rekursive domæneligninger.

Indlejring. Lad os betragte to domæner $\mathbf{D}$ og $\mathbf{E}$. Vi siger at $\mathbf{D}$ kan *indlejres* i $\mathbf{E}$ hvis der findes to kontinuerte funktioner

$$\begin{aligned}\phi &: \mathbf{D} \to \mathbf{E} \\ \psi &: \mathbf{E} \to \mathbf{D}\end{aligned}$$

som opfylder

$$\begin{aligned}&\forall \mathbf{x} \in \mathbf{D} \;\; . \; \psi(\phi(\mathbf{x})) = \mathbf{x} \\ &\forall \mathbf{y} \in \mathbf{E} \;\; . \; \phi(\psi(\mathbf{y})) \sqsubseteq \mathbf{y}\end{aligned}$$

Den første betingelse siger at elementerne i $\mathbf{D}$ kan repræsenteres som en delstruktur af $\mathbf{E}$. Den anden betingelse siger at indlejringen i en eller anden forstand skal bruge så små værdier som muligt. For eksempel vil bund blive repræsenteret ved bund da $\phi(\bot) \sqsubseteq \phi(\psi(\bot)) \sqsubseteq \bot$.

Vi vil af og til bruge notationen $\mathbf{D} \underset{\phi}{\overset{\psi}{\leftrightarrows}} \mathbf{E}$ for at angive at vi har en indlejring af $\mathbf{D}$ i $\mathbf{E}$ med funktionerne ϕ og ψ. Notationen kan læses som at ved afbildningen fra $\mathbf{D}$ til $\mathbf{E}$ bevares al information hvorimod der fra $\mathbf{E}$ til $\mathbf{D}$ kan tabes information. Det følger da også at ϕ er injektiv og ψ er surjektiv.

I forbindelse med abstrakt fortolkning kaldes funktionerne ψ og ϕ for henholdsvis *abstraktionsfunktion* og *konkretiseringsfunktion*. Vi skal nok vende tilbage til dette senere.

Fixpunktinduktion. Lad $\mathbf{D}$ være indlejret i $\mathbf{E}$ med ψ og ϕ. For kontinuerte funktioner $\mathbf{f} : \mathbf{D} \to \mathbf{D}$ og $\mathbf{g} : \mathbf{E} \to \mathbf{E}$ som opfylder

$$\begin{aligned}&\forall \mathbf{x} \in \mathbf{D}.\; \psi(\mathbf{g}(\phi(\mathbf{x}))) = \mathbf{f}(\mathbf{x}) \\ &\forall \mathbf{y} \in \mathbf{E}.\; \phi(\mathbf{f}(\psi(\mathbf{y}))) \sqsupseteq \mathbf{g}(\mathbf{y})\end{aligned}$$

gælder

$$\psi(\mathsf{fix}(\mathbf{g})) = \mathsf{fix}(\mathbf{f})$$

Bevis. Vi vil her basere os på en induktiv relation: Relationen

$$\forall \mathbf{x} \in \mathbf{D}, \mathbf{y} \in \mathbf{E}.\ \mathbf{x}\ \mathbf{R}\ \mathbf{y} \Leftrightarrow \psi(\mathbf{y}) = \mathbf{x}$$

er induktiv.

At den er grænseværdibevarende følger af ψ's kontinuitet. At den er strikt følger af at $\psi(\perp) = \perp$ idet

$$\phi(\perp) = \perp \Rightarrow \psi(\phi(\perp)) = \psi(\perp) \Rightarrow \perp = \psi(\perp)$$

Vi skal nu vise at $\mathbf{x}\ \mathbf{R}\ \mathbf{y} \Rightarrow \mathbf{f}(\mathbf{x})\ \mathbf{R}\ \mathbf{g}(\mathbf{y})$. Resultatet følger derefter ved fixpunktinduktion. Der er to veje at vise idet vi bruger antisymmetrien For alle $\mathbf{x} \in \mathbf{D}$, $\mathbf{y} \in \mathbf{E}$ har vi nemlig at

$$\begin{aligned} \psi(\mathbf{y}) = \mathbf{x} &\Rightarrow \phi(\psi(\mathbf{y})) = \phi(\mathbf{x}) \\ &\Rightarrow \mathbf{g}(\mathbf{y}) \sqsupseteq \mathbf{g}(\phi(\mathbf{x})) \\ &\Rightarrow \psi(\mathbf{g}(\mathbf{y})) \sqsupseteq \psi(\mathbf{g}(\phi(\mathbf{x}))) \\ &\Rightarrow \psi(\mathbf{g}(\mathbf{y})) \sqsupseteq \mathbf{f}(\mathbf{x}) \end{aligned}$$

og

$$\begin{aligned} \psi(\mathbf{y}) = \mathbf{x} &\Rightarrow \mathbf{f}(\psi(\mathbf{y})) = \mathbf{f}(\mathbf{x}) \\ &\Rightarrow \phi(\mathbf{f}(\psi(\mathbf{y}))) = \phi(\mathbf{f}(\mathbf{x})) \\ &\Rightarrow \mathbf{g}(\mathbf{y}) \sqsubseteq \phi(\mathbf{f}(\mathbf{x})) \\ &\Rightarrow \psi(\mathbf{g}(\mathbf{y})) \sqsubseteq \mathbf{f}(\mathbf{x}) \end{aligned}$$

Bevis slut

Om det så er smart at bruge dette princip afhænger af situationen. Umiddelbart er det blot et specialtilfælde af induktive relationer, idet vi dog har to relationer at vise i stedet for en i induktionsskridtet. Det ligner jo ikke nogen forbedring, men der kan være situationer, hvor det er lettere at bruge.

Galois forbindelse. Indlejring af et domæne i et andet kan ses som et specialtilfælde af en såkaldt *Galois forbindelse*. Vi siger at der er en Galois forbindelse mellem to domæner $\mathbf{D}$ og $\mathbf{E}$ hvis der findes to funktioner

$$\begin{aligned} &\phi : \mathbf{D} \to \mathbf{E} \\ &\psi : \mathbf{E} \to \mathbf{D} \end{aligned}$$

som opfylder

$$\forall \mathbf{x} \in \mathbf{D}, \mathbf{y} \in \mathbf{E}\ .\ \psi(\mathbf{y}) \sqsubseteq \mathbf{x} \Leftrightarrow \mathbf{y} \sqsubseteq \phi(\mathbf{x})$$

Galois forbindelser har mange pæne egenskaber, men dem kan vi altså ikke komme ind på her. På engelsk kaldes de for *Galois connections* og indlejringer *Galois insertions* eller blot *embeddings*.

5.5 Mere om indlejringer

Lige som logiske relationer kan løftes op igennem en typestruktur kan indlejringer også løftes til mere komplicerede domæner.

Domæneudtryk. Et *domæneudtryk* $\mathbf{F}(\mathbf{D})$ med formel parameter $\mathbf{D}$ er et udtryk opbygget med domænekonstruktionerne: cartesisk produkt, sum, funktionsdomæne og løftet domæne. Udtrykket kan opbygges ud fra den ubekendte $\mathbf{D}$ og faste domæner som $\mathbb{N}_\perp$, $\mathbb{B}_\perp$ osv.

Sætning. For domæner $\mathbf{E}_1$ og $\mathbf{E}_2$, hvor $\mathbf{E}_1$ kan indlejres i $\mathbf{E}_2$ og et domæneudtryk $\mathbf{F}(\mathbf{D})$, så kan også $\mathbf{F}(\mathbf{E}_1)$ indlejres i $\mathbf{F}(\mathbf{E}_2)$.

Bevis. Beviset følger ved strukturel induktion over domæneudtrykket $\mathbf{F}(\mathbf{D})$'s form.

Hvis $\mathbf{F}(\mathbf{D})$ har formen $\mathbf{D}$ ved vi at $\mathbf{E}_1$ kan indlejres i $\mathbf{E}_2$. Hvis $\mathbf{F}(\mathbf{D})$ blot er et af de faste domæner ($\mathbb{N}_\perp, \ldots$) så kan $\mathbf{F}(\mathbf{D})$ indlejres i sig selv ved brug af den identiske afbildning.

Nu antager vi induktivt at for domæneudtryk $\mathbf{F}_1(\mathbf{D})$ og $\mathbf{F}_2(\mathbf{D})$ kan $\mathbf{F}_1(\mathbf{E}_1)$ indlejres i $\mathbf{F}_1(\mathbf{E}_2)$ og $\mathbf{F}_2(\mathbf{E}_1)$ indlejres i $\mathbf{F}_2(\mathbf{E}_2)$.

$$\mathbf{F}_1(\mathbf{E}_1) \underset{\phi_1}{\overset{\psi_1}{\rightleftarrows}} \mathbf{F}_1(\mathbf{E}_2) \qquad \mathbf{F}_2(\mathbf{E}_1) \underset{\phi_2}{\overset{\psi_2}{\rightleftarrows}} \mathbf{F}_2(\mathbf{E}_2)$$

For domæneudtrykket $\mathbf{F}(\mathbf{D}) = \mathbf{F}_1(\mathbf{D}) \times \mathbf{F}_2(\mathbf{D})$ gælder påstanden idet vi kan bruge følgende indlejring:

$$(\mathbf{F}_1(\mathbf{E}_1) \times \mathbf{F}_2(\mathbf{E}_1)) \underset{\phi_3}{\overset{\psi_3}{\rightleftarrows}} (\mathbf{F}_1(\mathbf{E}_2) \times \mathbf{F}_2(\mathbf{E}_2))$$
$$\phi_3(\langle \mathbf{a}_1, \mathbf{a}_2 \rangle) = \langle \phi_1(\mathbf{a}_1), \phi_2(\mathbf{a}_2) \rangle$$
$$\psi_3(\langle \mathbf{b}_1, \mathbf{b}_2 \rangle) = \langle \psi_1(\mathbf{b}_1), \psi_2(\mathbf{b}_2) \rangle$$

For funktionsdomæner kan vi bruge følgende indlejring:

$$(\mathbf{F}_1(\mathbf{E}_1) \to \mathbf{F}_2(\mathbf{E}_1)) \underset{\phi_4}{\overset{\psi_4}{\rightleftarrows}} (\mathbf{F}_1(\mathbf{E}_2) \to \mathbf{F}_2(\mathbf{E}_2))$$
$$\phi_4(\mathbf{f}) = \lambda \mathbf{b}_1.\ \phi_2(\mathbf{f}(\psi_1(\mathbf{b}_1)))$$
$$\psi_4(\mathbf{g}) = \lambda \mathbf{a}_1.\ \psi_2(\mathbf{g}(\phi_1(\mathbf{a}_1)))$$

For sum domæner kan vi benytte følgende indlejring

$$(\mathbf{F}_1(\mathbf{E}_1) + \mathbf{F}_2(\mathbf{E}_1)) \underset{\phi_5}{\overset{\psi_5}{\rightleftarrows}} (\mathbf{F}_1(\mathbf{E}_2) + \mathbf{F}_2(\mathbf{E}_2))$$
$$\phi_5(\mathbf{a}) = \mathsf{cond}(\mathsf{isl}(\mathbf{a}), \mathsf{inl}(\phi_1(\mathsf{outl}(\mathbf{a}))), \mathsf{inr}(\phi_2(\mathsf{outr}(\mathbf{a}))))$$
$$\psi_5(\mathbf{b}) = \mathsf{cond}(\mathsf{isl}(\mathbf{b}), \mathsf{inl}(\psi_1(\mathsf{outl}(\mathbf{b}))), \mathsf{inr}(\psi_2(\mathsf{outr}(\mathbf{b}))))$$

For et løftet domæne har vi

$$\mathbf{F}_1(\mathbf{E}_1)_\perp \underset{\phi_6}{\overset{\psi_6}{\rightleftarrows}} \mathbf{F}_1(\mathbf{E}_2)_\perp$$
$$\phi_6(\mathbf{a}) = \mathsf{cond}(\mathsf{def}(\mathbf{a}), \mathsf{up}(\phi_1(\mathsf{down}(\mathbf{a}))), \perp)$$
$$\psi_6(\mathbf{b}) = \mathsf{cond}(\mathsf{def}(\mathbf{b}), \mathsf{up}(\psi_1(\mathsf{down}(\mathbf{b}))), \perp)$$

Bevis slut.

Notation. For en indlejring $\mathbf{A} \underset{\phi_1}{\overset{\psi_1}{\rightleftarrows}} \mathbf{B}$ og et domæneudtryk $\mathbf{F}(\mathbf{D})$ kan vi ifølge sætningen nu konstruere en indlejring $\mathbf{F}(\mathbf{A}) \underset{\phi_2}{\overset{\psi_2}{\rightleftarrows}} \mathbf{F}(\mathbf{B})$. Vi vil bruge notationen $\underset{\phi_2}{\overset{\psi_2}{\rightleftarrows}} = \mathbf{F}(\underset{\phi_1}{\overset{\psi_1}{\rightleftarrows}})$ for at angive at parret (ϕ_2, ψ_2) kan dannes fra (ϕ_1, ψ_1).

Sætning. Betragt følgende indlejringer

$$\mathbf{A} \underset{\phi_1}{\overset{\psi_1}{\rightleftarrows}} \mathbf{B} \underset{\phi_2}{\overset{\psi_2}{\rightleftarrows}} \mathbf{C}$$

Da er $\mathbf{A} \underset{\phi_2 \circ \phi_1}{\overset{\psi_1 \circ \psi_2}{\rightleftarrows}} \mathbf{C}$ en indlejring. Vi vil bruge notationen

$$\underset{\phi_2 \circ \phi_1}{\overset{\psi_1 \circ \psi_2}{\rightleftarrows}} = \underset{\phi_1}{\overset{\psi_1}{\rightleftarrows}} \circ \underset{\phi_2}{\overset{\psi_2}{\rightleftarrows}}$$

for at angive sammensætning af indlejringer.

Bevis. oplagt.

Sætning. For indlejringer

$$\mathbf{A} \underset{\phi_1}{\overset{\psi_1}{\rightleftarrows}} \mathbf{B} \underset{\phi_2}{\overset{\psi_2}{\rightleftarrows}} \mathbf{C}$$

gælder

$$\mathbf{F}(\underset{\phi_2 \circ \phi_1}{\overset{\psi_1 \circ \psi_2}{\rightleftarrows}}) = \mathbf{F}(\underset{\phi_1}{\overset{\psi_1}{\rightleftarrows}}) \circ \mathbf{F}(\underset{\phi_2}{\overset{\psi_2}{\rightleftarrows}})$$

Bevis. Sætningen følger ved strukturel induktion over domæneudtrykkets form. For udtryk $\mathbf{D}$ og konstante domæner er sætningen oplagt.

Antag nu at sætningen er vist for deludtryk $\mathbf{F}_1(\mathbf{D})$ og $\mathbf{F}_2(\mathbf{D})$ hvor vi har indlejringer

$$\mathbf{F}_1(\mathbf{A}) \underset{\phi_1}{\overset{\psi_1}{\leftrightarrows}} \mathbf{F}_1(\mathbf{B}) \underset{\phi_2}{\overset{\psi_2}{\leftrightarrows}} \mathbf{F}_1(\mathbf{C}) \qquad \mathbf{F}_2(\mathbf{A}) \underset{\phi_3}{\overset{\psi_3}{\leftrightarrows}} \mathbf{F}_2(\mathbf{B}) \underset{\phi_4}{\overset{\psi_4}{\leftrightarrows}} \mathbf{F}_2(\mathbf{C})$$

og

$$\mathbf{F}_1(\mathbf{A}) \underset{\phi_2 \circ \phi_1}{\overset{\psi_1 \circ \psi_2}{\leftrightarrows}} \mathbf{F}_1(\mathbf{C}) \qquad \mathbf{F}_2(\mathbf{A}) \underset{\phi_4 \circ \phi_3}{\overset{\psi_3 \circ \psi_4}{\leftrightarrows}} \mathbf{F}_2(\mathbf{C})$$

For det cartesiske produkt skal vi vise at

$$\begin{aligned}
&\lambda\langle \mathbf{a}_1, \mathbf{a}_2\rangle.\ \langle(\phi_2 \circ \phi_1)(\mathbf{a}_1), (\phi_4 \circ \phi_3)(\mathbf{a}_2)\rangle \\
&\quad = (\lambda\langle \mathbf{b}_1, \mathbf{b}_2\rangle.\ \langle\phi_2(\mathbf{b}_1), \phi_4(\mathbf{b}_2)\rangle) \circ (\lambda\langle \mathbf{a}_1, \mathbf{a}_2\rangle.\ \langle\phi_1(\mathbf{a}_1), \phi_3(\mathbf{a}_2)\rangle) \\
&\lambda\langle \mathbf{c}_1, \mathbf{c}_2\rangle.\ \langle(\psi_1 \circ \psi_2)(\mathbf{c}_1), (\psi_3 \circ \psi_4)(\mathbf{c}_2)\rangle \\
&\quad = (\lambda\langle \mathbf{b}_1, \mathbf{b}_2\rangle.\ \langle\psi_1(\mathbf{b}_1), \psi_3(\mathbf{b}_2)\rangle) \circ (\lambda\langle \mathbf{c}_1, \mathbf{c}_2\rangle.\ \langle\psi_2(\mathbf{c}_1), \psi_4(\mathbf{c}_2)\rangle)
\end{aligned}$$

Resultatet følger ved simpel udregning.

Tilsvarende udregninger kan foretages for funktionsdomæner, sum og løftet domæne. Dette overlades dog som en øvelse til læseren.

Bevis slut

5.6 Opgaver

Opgave 5.1. Lad $\mathbf{f} : \mathbf{D} \to_{\mathbf{c}} \mathbf{D}$ være en kontinuert funktion på et domæne $\mathbf{D}$. Vis at følgende gælder:

$$\forall \mathbf{d} \in \mathbf{D}.\ (\mathbf{f}(\mathbf{d}) \sqsubseteq \mathbf{d} \Rightarrow \mathsf{fix}(\mathbf{f}) \sqsubseteq \mathbf{d})$$

Dette princip kaldes ofte for Park induktion

Opgave 5.2. Lad $\mathbf{f} : \mathbf{D} \to_{\mathbf{c}} \mathbf{E}$ og $\mathbf{g} : \mathbf{E} \to_{\mathbf{c}} \mathbf{D}$ mellem domæner $\mathbf{D}$ og $\mathbf{E}$. Vis at følgende gælder:

$$\mathsf{fix}(\mathbf{g} \circ \mathbf{f}) = \mathbf{g}(\mathsf{fix}(\mathbf{f} \circ \mathbf{g}))$$

Opgave 5.3. Lad $\mathbf{R}$ være en induktiv relation mellem elementer i domæner $\mathbf{D}_1$ og $\mathbf{D}_2$ og lad $\mathbf{S}$ være en induktiv relation mellem elementer i domæner $\mathbf{E}_1$ og $\mathbf{E}_2$. Vis at man kan definere en induktiv relation mellem elementerne i $\mathbf{D}_1 + \mathbf{E}_1$ og $\mathbf{D}_2 + \mathbf{E}_2$

Opgave 5.4. Vis at de *logiske* relationer $\mathbf{R}\times\mathbf{S}$ og $\mathbf{R}\to\mathbf{S}$ dannet fra induktive relationer $\mathbf{R}$ og $\mathbf{S}$ er induktive.

Opgave 5.5. Afslut beviset for sætningen om sammensætning af indlejringer:

$$\mathbf{F}\Big(\underset{\phi_2\circ\phi_1}{\overset{\psi_1\circ\psi_2}{\leftrightarrows}}\Big) = \mathbf{F}\Big(\underset{\phi_1}{\overset{\psi_1}{\leftrightarrows}}\Big) \circ \mathbf{F}\Big(\underset{\phi_2}{\overset{\psi_2}{\leftrightarrows}}\Big)$$

6 Rekursive domæneligninger

Vi har i disse noter set hvordan domæneteori kan bruges til at beskrive semantikken af programmeringssprog. Vi har benyttet at fixpunktsætningen kan bruges til at beskrive de rekursive afhængigheder, der normalt optræder i programmer. En semantik der på denne måde direkte er funderet i domæneteorien har vi kaldt for en fixpunktsemantik. Der er imidlertid et andet aspekt af semantikken vi endnu ikke har beskæftiget os med, og det er betydningen af typer.

Vi viste i starten af disse noter hvordan man kan danne nye domæner som produkter, sum og funktionsrum af andre domæner. Alt ialt er domæneteorien overraskende stabil over for sådanne konstruktioner. Vi skal her til slut opridse et af hovedresultaterne inden for domæneteorien; nemlig at rekursive domæneligninger opbygget af sædvanlige domænekonstruktorer altid har en løsning.

Beviset er ganske omfattende så dette afsnit vil nok ikke ligefrem være let læsning. Hovedformålet med afsnittet er at man kan forstå konstruktionen så man er fortrolig med hvordan løsningerne til domæneligninger ser ud. Derimod er det mindre væsentligt om man har overbevist sig om at alle de mængder, der bliver brugt, er domæner, at alle funktionerne er kontinuerte og at alle følgerne er kæder. Disse beviser er i sig selv ganske enkle, men der er bare så mange af dem at man let mister overblikket.

6.1 Rekursive domænedefinitioner

Rekursive typedefinitioner kendes fra mange programmeringssprog. I Pascal kan man således benytte følgende erklæring til hægtede lister af heltal.

```
type x = ↑ y;
     y = record
           v :integer ;
           z : x
         end
```

Også lambdakalkule kender til rekursive definitioner: Objekter er enten lambdaudtryk eller (partielle) funktioner mellem lambdaudtryk. Det kunne vi fristes til at udtrykke ved at sige at mængden af lambdaudtryk $\mathbf{D}$ skal opfylde ligningen.

$$\mathbf{D} = \mathbf{D} \to \mathbf{D}_{\perp}$$

Som med rekursive funktionsdefinitioner er det imidlertid ret uklart hvilken mening vi skal tillægge en sådan definition. Skal man finde et mængdeudtryk som kan indsættes og give det samme, tekstuelt, på begge sider? For at kunne snakke om semantikken af et sprog er det centralt at vi er sikre på, at der findes en mængde af mulige værdier for hver type der kan benyttes i sproget.

Manglende løsninger. Problemet minder om vor diskussion af rekursive funktionsdefinitioner. Der er imidlertid her flere grunde til at vi er ude på et skråplan og må træde forsigtigt.

I mængdelæren betyder lighed mellem mængder at de indeholder samme elementer. Med mindre man har meget simple ligninger som $\mathbf{D} = \mathbf{D}$ synes det svært at finde en måde at finde ud af om en given mængde kan siges at opfylde ligningen. Det bedste vi så kan gøre er da blot at kræve at de to mængder på hver side er isomorfe. I ord kan man så sige at vi om ligningen ovenfor gerne vil have en mængde så lambdaudtryk kan *opfattes* som funktioner mellem lambdaudtryk og omvendt. Ligningen skriver vi så i stedet som

$$\mathbf{D} \simeq \mathbf{D} \to \mathbf{D}_\perp$$

og vi siger, at vi ønsker en løsning op til isomorfi.

Cantors sætning siger imidlertid at på grund af kardinaliteten kan ligningen

$$\mathbf{D} \simeq \mathbf{D} \to \mathbf{D}_\perp$$

ingen løsninger have. Det viser sig imidlertid, at hvis vi begrænser os til de kontinuerte funktioner har denne ligning, og en lang række andre, løsninger som er domæner.

Det kunne være fristende at bruge konstruktionen fra de rekursive funktionsdefinitioner på disse ligninger. Klassen af domæner er imidlertid ikke en mængde — og derfor heller ikke et domæne. Vi kan derfor ikke umiddelbart bruge fixpunktsætningen.

De spilleregler mængdelæren giver os er, at vi godt kan konstruere sum, produkt, foreningsmængde og fællesmængde af en uendelig (tællelig) klasse af mængder. Til gengæld kan mængdelæren ikke direkte hjælpe os med at konstruere sådanne uendeligt dybe indlejrede strukturer som en ligning som $\mathbf{D} \simeq \mathbf{D} \to \mathbf{D}_\perp$ kræver.

6.2 Løsning op til isomorfi

Når vi blot kræver løsninger op til isomorfi skal vi hurtigt se at vi i konkrete tilfælde kan finde løsninger. Senere vil vi generalisere det til en egentlig konstruktion af løsninger til rekursive domæneligninger. Lad os starte med at formalisere hvad vi vil kræve af en løsning til en sådan domæneligning.

Isomorfe mængder. Givet en domæneligning

$$\mathbf{D} \simeq \mathbf{F}(\mathbf{D})$$

hvor $\mathbf{F}$ er et domæneudtryk i $\mathbf{D}$. Ved en løsning til ligningen forstår vi et domæne $\mathbf{D}$ sammen med to kontinuerte funktioner:

$$\Phi : \mathbf{D} \to \mathbf{F}(\mathbf{D})$$
$$\Psi : \mathbf{F}(\mathbf{D}) \to \mathbf{D}$$

således at

$$\forall \mathbf{x} \in \mathbf{D}.\ \Psi(\Phi(\mathbf{x})) = \mathbf{x}$$
$$\forall \mathbf{x} \in \mathbf{F}(\mathbf{D}).\ \Phi(\Psi(\mathbf{x})) = \mathbf{x}$$

Vi kunne også have sagt at $\mathbf{D}$ skal kunne indlejres i $\mathbf{F}(\mathbf{D})$ samtidig med at $\mathbf{F}(\mathbf{D})$ kan indlejres i $\mathbf{D}$.

Eksempel. Lad $\mathbf{A}$ være et vilkårligt domæne. Betragt ligningen

$$\mathbf{D} \simeq \mathbf{A} \times \mathbf{D}$$

En løsning til ligningen er

$$\mathbf{D} = \mathbb{N} \to \mathbf{A}$$

idet vi kan bruge de følgende funktioner som isomorfier.

$$\Phi : (\mathbb{N} \to \mathbf{A}) \to (\mathbf{A} \times (\mathbb{N} \to \mathbf{A}))$$
$$\Psi : (\mathbf{A} \times (\mathbb{N} \to \mathbf{A})) \to (\mathbb{N} \to \mathbf{A})$$
$$\Phi(\mathbf{f}) = \langle \mathbf{f}(0), \lambda\mathbf{x}.\ \mathbf{f}(\mathbf{x}+1)\rangle$$
$$\Psi(\langle \mathbf{y}, \mathbf{f}\rangle) = \lambda\mathbf{x}.\ \ \mathbf{if}\ \mathbf{x} = 0\ \mathbf{then}\ \mathbf{y}\ \mathbf{else}\ \mathbf{f}(\mathbf{x}-1)$$

Vi skal nu vise at de faktisk opfylder betingelserne for isomorfi.

$$\Phi(\Psi(\langle \mathbf{y}, \mathbf{f}\rangle)) = \langle \mathbf{y}, \lambda\mathbf{x}.\ \mathbf{f}(\mathbf{x})\rangle = \langle \mathbf{y}, \mathbf{f}\rangle$$
$$\Psi(\Phi(\mathbf{f})) = \lambda\mathbf{x}.\ \ \mathbf{if}\ \mathbf{x} = 0\ \mathbf{then}\ \mathbf{f}(0)\ \mathbf{else}\ \mathbf{f}(\mathbf{x}) = \mathbf{f}$$

Domæneligningen fastlægger at $\mathbf{D}$ skal svare til de uendelige lister af elementer fra et domæne $\mathbf{A}$. Løsningen her viser så, at en funktion fra de naturlige tal til $\mathbf{A}$ kan opfattes som en uendelig liste af elementer fra $\mathbf{A}$. Det er nok ikke særlig overraskende, men det skulle gerne vise at de løsninger vi får ud af det svarer til hvad vi ville forvente intuitivt.

Anvendelse. Det kan måske synes tungt at skulle slæbe rundt på isomorfierne når man skal bruge elementer i domænet defineret ved ligningen $\mathbf{D} \simeq \mathbf{A} \times \mathbf{D}$. I praksis er man ikke interesseret i den indre struktur af sådanne domæner når man blot har visse basisoperationer. I dette tilfælde kan man for eksempel tænke sig følgende fire operationer:

$$\begin{array}{ll}
\mathsf{nil} \in \mathbf{D} & \mathsf{nil} = \lambda\mathbf{x}.\ \bot_{\mathbf{A}} \\
\mathsf{cons} : \mathbf{A} \times \mathbf{D} \to \mathbf{D} & \mathsf{cons}(\mathbf{a}, \mathbf{d}) = \Psi(\langle \mathbf{a}, \mathbf{d} \rangle) \\
\mathsf{head} : \mathbf{D} \to \mathbf{A} & \mathsf{head}(\mathbf{d}) = \mathsf{fst}(\Phi(\mathbf{d})) \\
\mathsf{tail} : \mathbf{D} \to \mathbf{D} & \mathsf{tail}(\mathbf{d}) = \mathsf{snd}(\Phi(\mathbf{d}))
\end{array}$$

Nu kan vi arbejde med værdier i domænet uden at tænke på deres struktur. Det ligner et eksempel på dataabstraktion og modulær programmering. Når man skal bruge domænet er man så nok mest interesseret i at visse regler gælder, som for eksempel $\mathsf{head}(\mathsf{cons}(\mathbf{x}, \mathbf{y})) = \mathbf{x}$ og $\mathsf{tail}(\mathsf{cons}(\mathbf{x}, \mathbf{y})) = \mathbf{y}$. Bemærk at disse regler vil gælde uanset hvilken løsning vi havde valgt til domæneligningen.

6.3 Approximationer

Det er selvfølgelig meget godt at vi i sådanne tilfælde kan finde en løsning, men er det altid muligt? Svaret er et tøvende ja. Det er faktisk muligt sålænge vi holder os til de domæneudtryk vi har brugt indtil nu, altså funktioner, produkter, sum og løftet domæne.

Indlejring. Vi vil nu lade os inspirere af metoden til at finde fixpunkter for funktioner. Der startede vi med bundelementet og dannede en voksende følge af funktionsværdier, hvor vi kaldte funktionen med det forrige element i følgen som argument. Hvis vi skulle gøre det samme her skulle vi starte med et mindste domæne og danne domænet $\mathbf{F}(\mathbf{D})$ med det mindste domæne i stedet for $\mathbf{D}$.

Det der bedst svarer til en ordning blandt domæner er begrebet *indlejring*. Det udgør ikke nogen partiel ordning idet det ikke opfylder kravet om antisymmetri. Det betyder at hvis $\mathbf{D}_1$ og $\mathbf{D}_2$ kan indlejres i hinanden betyder det ikke at de er identiske, men kun at de er isomorfe. Det er naturligvis også isomorfe mængder vi er interesseret i her, så derfor er begrebet nyttigt til denne konstruktion.

Lad os lige minde om definitionen af indlejring. Hvis vi har to domæner $\mathbf{D}_1$ og $\mathbf{D}_2$ så kan $\mathbf{D}_1$ indlejres i $\mathbf{D}_2$ hvis der findes to kontinuerte funktioner $\phi : \mathbf{D}_1 \to \mathbf{D}_2$ og $\psi : \mathbf{D}_2 \to \mathbf{D}_1$ som opfylder

$$\forall \mathbf{x} \in \mathbf{D}_1.\ \psi(\phi(\mathbf{x})) = \mathbf{x}$$
$$\forall \mathbf{y} \in \mathbf{D}_2.\ \phi(\psi(\mathbf{y})) \sqsubseteq \mathbf{y}$$

Billedlig talt betyder det at domænet $\mathbf{D}_2$ skal have domænet $\mathbf{D}_1$ som en delstruktur. For indlejringer bruger vi notationen

$$\mathbf{D}_1 \underset{\phi}{\overset{\psi}{\leftrightarrows}} \mathbf{D}_2$$

til at vise at $\mathbf{D}_1$ kan indlejres i $\mathbf{D}_2$ med funktionerne ϕ og ψ. Vi så desuden i sidste kapitel at indlejringer kan løftes til sammensatte domæner.

"Mindste" domæne. Ifølge modellen fra fixpunktsætningen skal vi starte fra et "mindste" domæne. Med indlejring som en slags ordning kan ethvert etpunktsdomæne indlejres i alle domæner. For eksempel kan domænet

$$\mathbb{U} = \{\bot\}$$

indlejres i ethvert domæne idet vi for et domæne $\mathbf{D}$ kan bruge funktionerne

$$\begin{array}{ll} \phi : \mathbb{U} \to \mathbf{D} & \phi(\bot) = \bot_{\mathbf{D}} \\ \psi : \mathbf{D} \to \mathbb{U} & \psi(\mathbf{x}) = \bot, \quad \mathbf{x} \in \mathbf{D} \end{array}$$

Nu findes der mange forskellige etpunktsdomæne, hvor mindsteelementet hedder noget forskelligt, men de er oplagt isomorfe. For den videre konstruktion er det ikke afgørende hvilken vi vælger. Vi skal alligevel kun finde løsninger op til isomorfi.

Approximationer. I fixpunktsætningen dannede vi en kæde af værdier $\bot \sqsubseteq \mathbf{f}(\bot) \sqsubseteq \mathbf{f}^2(\bot)$. For en domæneligning

$$\mathbf{D} \simeq \mathbf{F}(\mathbf{D})$$

kan vi lave noget lignende idet vi kan konstruere domænerne

$$\mathbf{D_i} = \mathbf{F^i}(\mathbb{U})$$

Vi har allerede set at domænet $\mathbb{U}$ kan indlejres i $\mathbf{F}(\mathbb{U})$ og ved induktion følger det at $\mathbf{F^i}(\mathbb{U})$ kan indlejres i $\mathbf{F^{i+1}}(\mathbb{U})$ for alle $\mathbf{i}$. Vi har altså konstrueret en følge af approximationer $\mathbf{D}_0, \mathbf{D}_1, \ldots$ hvor $\mathbf{D_i}$ kan indlejres i $\mathbf{D_{i+1}}$ med funktioner $\phi_\mathbf{i}$ og $\psi_\mathbf{i}$.

$$\mathbf{D}_0 \underset{\phi_0}{\overset{\psi_0}{\leftrightarrows}} \mathbf{D}_1 \underset{\phi_1}{\overset{\psi_1}{\leftrightarrows}} \mathbf{D}_2 \underset{\phi_2}{\overset{\psi_2}{\leftrightarrows}} \mathbf{D}_3 \underset{\phi_4}{\overset{\psi_3}{\leftrightarrows}} \mathbf{D}_4 \cdots$$

Indlejringerne er defineret induktivt med

$$
\begin{aligned}
&\phi_0(\perp) = \perp_{\mathbf{D}_1} \\
&\psi_0(\mathbf{x}) = \perp \quad , \ \mathbf{x} \in \mathbf{D}_1 \\
&\underset{\phi_{\mathbf{i+1}}}{\overset{\psi_{\mathbf{i+1}}}{\rightleftarrows}} = \mathbf{F}(\underset{\phi_{\mathbf{i+1}}}{\overset{\psi_{\mathbf{i+1}}}{\rightleftarrows}})
\end{aligned}
$$

Eksempel. Lad os lige kikke på eksemplet med ligningen for uendelige lister.

$$\mathbf{D} \simeq \mathbb{N}_\perp \times \mathbf{D}$$

Vi kan ganske let konstruere følgen af approximationer for denne ligning.

$$
\begin{aligned}
&\mathbf{D}_0 = \mathbb{U} \\
&\mathbf{D}_1 = \mathbb{N}_\perp \times \mathbb{U} \\
&\mathbf{D}_2 = \mathbb{N}_\perp \times (\mathbb{N}_\perp \times \mathbb{U}) \\
&\mathbf{D}_3 = \ldots
\end{aligned}
$$

Domænet $\mathbf{D_n}$ er så domænet af lister af længde $\mathbf{n}$ og det skulle være klart at det kan indlejres i domænet $\mathbf{D_{n+1}}$ ved at bruge en liste af længde $\mathbf{n}+1$, hvor sidste elementet er mindsteelementet. Mere præcist har vi indlejringer

$$
\begin{aligned}
&\phi_0(\perp) = \langle \perp_{\mathbb{N}}, \perp \rangle \\
&\phi_1(\langle \mathbf{d}_1, \perp \rangle) = \langle \mathbf{d}_1, \langle \perp_{\mathbb{N}}, \perp \rangle \rangle \\
&\phi_2(\langle \mathbf{d}_1, \langle \mathbf{d}_2, \perp \rangle \rangle) = \langle \mathbf{d}_1, \langle \mathbf{d}_2, \langle \perp_{\mathbb{N}}, \perp \rangle \rangle \rangle \\
&\ldots \\
&\psi_0(\langle \mathbf{d}_1, \perp \rangle) = \perp \\
&\psi_1(\langle \mathbf{d}_1, \langle \mathbf{d}_2, \perp \rangle \rangle) = \langle \mathbf{d}_1, \perp \rangle \\
&\ldots
\end{aligned}
$$

6.4 Grænseværdi

Vi har vist hvordan man kan konstruere en følge af domæner $\mathbf{D}_0, \mathbf{D}_1, \ldots$ Vi kalder disse domæner for approximationerne til løsningen og det kunne nu godt se ud som om vi bare skulle tage grænseværdien af denne følge. Den går imidlertid ikke for nu holder parallellen til fixpunktsætningen ikke længere. Vi har ikke vist at klassen af domæner har grænseværdiegenskaben for domæner og det er derfor ikke sikkert at der faktisk findes sådanne grænseværdier.

Lister. Vi er nu klar til det centrale trick i konstruktionen. Når vi ikke umiddelbart kan finde en grænseværdi kan vi i stedet kikke på følger af approximationer

$$\mathbf{D}_0 \times \mathbf{D}_1 \times \mathbf{D}_2 \times \cdots$$

Elementer i denne mængde skriver vi som lister: $[\mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2, \ldots]$ hvor $\mathbf{d_i} \in \mathbf{D_i}$. I denne konstruktion skal man nok lige bemærke at der ikke er nogen problemer med mængdelæren i at definere en mængde af uendelige lister. Er man i tvivl kan det hurtigt ses at listerne kan opfattes som funktioner fra naturlige tal til foreningsmængden af approximationer.

Grænseværdi. Vi kan godt konstruere lister af approximationer, men blandt dem vil vi kun betragte lister, hvor alle elementer approximerer samme senere elementer i listen.

$$\mathbf{D}_\infty = \{[\mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2, \ldots] \in \mathbf{D}_0 \times \mathbf{D}_1 \times \mathbf{D}_2 \times \cdots \mid \forall \mathbf{i}.\ \mathbf{d_i} = \psi_\mathbf{i}(\mathbf{d_{i+1}})\}$$

Her er $\psi_\mathbf{i}$ funktionen der bruges i indlejringen af $\mathbf{D_i}$ i $\mathbf{D_{i+1}}$. Bemærk at et element $\mathbf{d_i}$ i en sådan liste entydigt bestemmer alle de tidligere (til venstre) elementer i listen. Man kan også opfatte elementerne i en liste som værende entydigt bestemt af en tænkt uendelig nøjagtig approximation. Det er det, der er det smarte ved konstruktionen. Problemet er nu at vise at man derved får defineret et domæne $\mathbf{D}_\infty$, der kan indlejres i $\mathbf{F}(\mathbf{D}_\infty)$.

Parallellen fra fixpunktsætningen er at vi nu har postuleret et fixpunkt og vi vil så gerne overbevise os om at det er en overgrænse for approximationerne og at vi får en løsning til ligningen.

Et domæne. Det første punkt er at overbevise os om at vi rent faktisk har fået defineret et domæne med $\mathbf{D}_\infty$. De fleste egenskaber bevares fra deldomænerne. Ordningen er elementvis så

$$[\mathbf{d}_0, \mathbf{d}_1, \ldots] \sqsubseteq [\mathbf{e}_0, \mathbf{e}_1, \ldots] \Leftrightarrow \forall \mathbf{i}.\ \mathbf{d_i} \sqsubseteq \mathbf{e_i}$$

Mindsteelementet i $\mathbf{D}_\infty$ er listen af mindsteelementer fra domænerne idet det oplagt opfylder $\mathbf{d_i} = \psi_\mathbf{i}(\mathbf{d_{i+1}})$. Grænseværdien af en kæde er defineret elementvis og tilbage er blot at vise at denne værdi opfylder betingelsen. Dette er overladt som en øvelse til læseren.

En "overgrænse". Det andet punkt er at overbevise os om at domænet $\mathbf{D}_\infty$ kan opfattes som en overgrænse til kæden af approximationer $\mathbf{D}_0$, $\mathbf{D}_1$, ... Med andre ord skal vi vise at disse domæner kan indlejres i $\mathbf{D}_\infty$. Dette kan vi gøre med funktionerne

$$\begin{aligned}
&\theta_{0\infty}(\mathbf{d}) = [\mathbf{d}, \phi_0(\mathbf{d}), \phi_1(\phi_0(\mathbf{d})), \ldots] \\
&\theta_{1\infty}(\mathbf{d}) = [\psi_0(\mathbf{d}), \mathbf{d}, \phi_1(\mathbf{d}), \phi_2(\phi_1(\mathbf{d})), \ldots] \\
&\ldots \\
&\theta_{\mathbf{i}\infty}(\mathbf{d}) = [\psi_0(\ldots\psi_{\mathbf{i}-1}(\mathbf{d})\ldots), \ldots, \psi_{\mathbf{i}-1}(\mathbf{d}), \mathbf{d}, \phi_{\mathbf{i}}(\mathbf{d}), \phi_{\mathbf{i}+1}(\phi_{\mathbf{i}}(\mathbf{d})), \ldots] \\
&\ldots
\end{aligned}$$

og

$$\theta_{\infty\mathbf{i}}([\mathbf{d}_0, \mathbf{d}_1, \ldots]) = \mathbf{d}_\mathbf{i}$$

Påstanden er at vi herved har en indlejring af $\mathbf{D}_\mathbf{i}$ i $\mathbf{D}_\infty$:

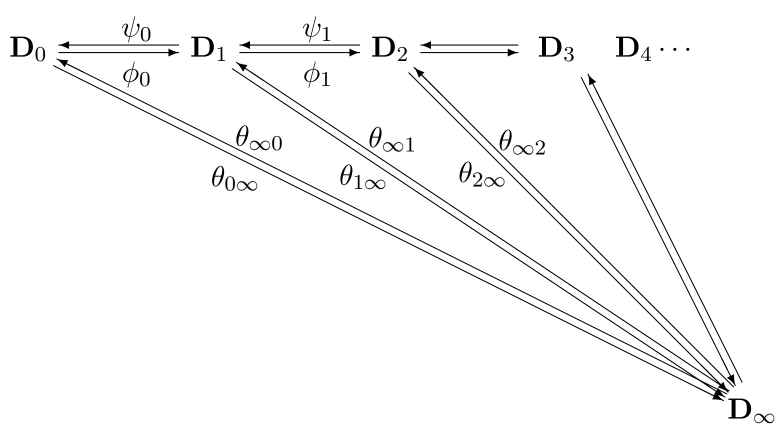
$$\mathbf{D}_\mathbf{i} \underset{\theta_{\mathbf{i}\infty}}{\overset{\theta_{\infty\mathbf{i}}}{\rightleftarrows}} \mathbf{D}_\infty$$

Ved udregning finder vi

$$\begin{aligned}
&\theta_{\infty\mathbf{i}}(\theta_{\mathbf{i}\infty}(\mathbf{d})) = \theta_{\infty\mathbf{i}}([\ldots, \psi_{\mathbf{i}-1}(\mathbf{d}), \mathbf{d}, \phi_\mathbf{i}(\mathbf{d}), \ldots]) = \mathbf{d} \\
&\theta_{\mathbf{i}\infty}(\theta_{\infty\mathbf{i}}([\mathbf{d}_0, \mathbf{d}_1, \ldots])) = [\ldots, \psi_{\mathbf{i}-1}(\mathbf{d}_\mathbf{i}), \mathbf{d}_\mathbf{i}, \phi_\mathbf{i}(\mathbf{d}_\mathbf{i}), \ldots] \sqsubseteq [\ldots, \mathbf{d}_{\mathbf{i}-1}, \mathbf{d}_\mathbf{i}, \mathbf{d}_{\mathbf{i}+1}, \ldots]
\end{aligned}$$

idet vi fra $\mathbf{d}_\mathbf{i} = \psi_\mathbf{i}(\mathbf{d}_{\mathbf{i}+1})$ slutter at $\phi_\mathbf{i}(\mathbf{d}_\mathbf{i}) = \phi_\mathbf{i}(\psi_\mathbf{i}(\mathbf{d}_{\mathbf{i}+1})) \sqsubseteq \mathbf{d}_{\mathbf{i}+1}$.

Indlejringerne illustreres ofte med en tegning:



Ved udregning får vi desuden følgende lighed

$$\underset{\theta_{\mathbf{i}\infty}}{\overset{\theta_{\infty\mathbf{i}}}{\rightleftarrows}} = \underset{\phi_\mathbf{i}}{\overset{\psi_\mathbf{i}}{\rightleftarrows}} \circ \underset{\theta_{\mathbf{i}+1\infty}}{\overset{\theta_{\infty\mathbf{i}+1}}{\rightleftarrows}}$$

hvilket betyder at

$$\forall \mathbf{d_i} \in \mathbf{D_i}.\ \theta_{\mathbf{i}\infty}(\mathbf{d_i}) = \theta_{\mathbf{i+1}\infty}(\phi_{\mathbf{i}}(\mathbf{d_i}))$$
$$\forall \mathbf{d} \in \mathbf{D}_\infty.\ \theta_{\infty\mathbf{i}}(\mathbf{d}) = \psi_{\mathbf{i}}(\theta_{\infty\mathbf{i+1}}(\mathbf{d}))$$

Indlejringer. Det næste punkt er at vise at domænerne $\mathbf{D_i}$ kan indlejres i domænet $\mathbf{F}(\mathbf{D}_\infty)$.

Vi så at domænet $\mathbf{D_i}$ kan indlejres $\mathbf{D}_\infty$ og vi har set at hvis $\mathbf{E}_1$ kan indlejres i $\mathbf{E}_2$ så kan $\mathbf{F}(\mathbf{E}_1)$ indlejres i $\mathbf{F}(\mathbf{E}_2)$. Det betyder at vi kan definere funktioner

$$\theta'_{\mathbf{i+1}\infty} : \mathbf{F}(\mathbf{D_i}) \to \mathbf{F}(\mathbf{D}_\infty)$$
$$\theta'_{\infty\mathbf{i+1}} : \mathbf{F}(\mathbf{D}_\infty) \to \mathbf{F}(\mathbf{D_i})$$

som etablerer en indlejring af $\mathbf{F}(\mathbf{D_i})$ i $\mathbf{F}(\mathbf{D}_\infty)$. Ved at bruge notationen fra sidste kapitel har vi at

$$\mathbf{D_i} \overset{\theta'_{\infty\mathbf{i}}}{\underset{\theta'_{\mathbf{i}\infty}}{\rightleftarrows}} \mathbf{F}(\mathbf{D}_\infty)$$
$$\overset{\theta'_{\infty\mathbf{i+1}}}{\underset{\theta'_{\mathbf{i+1}\infty}}{\rightleftarrows}} = \mathbf{F}(\overset{\theta_{\infty\mathbf{i}}}{\underset{\theta_{\mathbf{i}\infty}}{\rightleftarrows}})$$

Ved at bruge sætningen om sammensætning af indlejringer får vi følgende ligheder

$$\overset{\theta'_{\infty\mathbf{i}}}{\underset{\theta'_{\mathbf{i}\infty}}{\rightleftarrows}} = \mathbf{F}(\overset{\theta_{\infty\mathbf{i-1}}}{\underset{\theta_{\mathbf{i-1}\infty}}{\rightleftarrows}}) = \mathbf{F}(\overset{\psi_{\mathbf{i-1}}}{\underset{\phi_{\mathbf{i-1}}}{\rightleftarrows}} \circ \overset{\theta_{\infty\mathbf{i}}}{\underset{\theta_{\mathbf{i}\infty}}{\rightleftarrows}}) = \mathbf{F}(\overset{\psi_{\mathbf{i-1}}}{\underset{\phi_{\mathbf{i-1}}}{\rightleftarrows}}) \circ \mathbf{F}(\overset{\theta_{\infty\mathbf{i}}}{\underset{\theta_{\mathbf{i}\infty}}{\rightleftarrows}}) = \overset{\psi_{\mathbf{i}}}{\underset{\phi_{\mathbf{i}}}{\rightleftarrows}} \circ \overset{\theta'_{\infty\mathbf{i+1}}}{\underset{\theta'_{\mathbf{i+1}\infty}}{\rightleftarrows}}$$

Isomorfi. Vi er nu parate til hovedresultatet - nemlig at $\mathbf{D}_\infty$ kan indlejres i $\mathbf{F}(\mathbf{D}_\infty)$ og omvendt. Vi vil gerne konstruere isomorfierne $\Phi : \mathbf{D}_\infty \to \mathbf{F}(\mathbf{D}_\infty)$ og $\Psi : \mathbf{F}(\mathbf{D}_\infty) \to \mathbf{D}_\infty$. Til inspiration kan vi lave et lille diagram (dog ikke kommuterende) af disse funktioner sammen med Φ og Ψ.

$$\begin{tikzcd}
\mathbf{D_i} \arrow[r, shift left, "\theta_{\mathbf{i}\infty}"] \arrow[d, shift right, "\phi_{\mathbf{i}}"'] & \mathbf{D}_\infty \arrow[l, shift left, "\theta_{\infty\mathbf{i}}"] \arrow[d, shift right, "\Phi"'] \\
\mathbf{D_{i+1}} \arrow[u, shift right, "\psi_{\mathbf{i}}"'] \arrow[r, shift left, "\theta'_{\mathbf{i+1}\infty}"] & \mathbf{F}(\mathbf{D}_\infty) \arrow[l, shift left, "\theta'_{\infty\mathbf{i+1}}"] \arrow[u, shift right, "\Psi"']
\end{tikzcd}$$

Det ville være fristende som kandidat til Φ at kikke på

$$\lambda\mathbf{d}.\ \theta'_{\mathbf{i}+1\infty}(\phi_{\mathbf{i}}(\theta_{\infty\mathbf{i}}(\mathbf{d})))$$

eller blot $\theta'_{\mathbf{i}\infty} \circ \theta_{\infty\mathbf{i}}$ og

$$\lambda\mathbf{d}.\ \theta_{\mathbf{i}\infty}(\psi_{\mathbf{i}}(\theta'_{\infty\mathbf{i}+1}(\mathbf{d})))$$

eller $\theta_{\mathbf{i}\infty} \circ \theta'_{\infty\mathbf{i}}$ som kandidat til Ψ. De lever desværre ikke helt op til kravet — for ethvert $\mathbf{i}$ vil vi med disse funktioner kun have en Galois forbindelse, men altså ikke en indlejring. I stedet kan vi udnytte at funktionerne, set som følge i $\mathbf{i}$ danner en voksende kæde i domænerne $\mathbf{D}_\infty \to \mathbf{F}(\mathbf{D}_\infty)$ og $\mathbf{F}(\mathbf{D}_\infty) \to \mathbf{D}_\infty$. Det betyder at vi kan definere følgende funktioner:

$$\Phi = \lambda\mathbf{x}.\ \bigsqcup_{\mathbf{i}} \theta'_{\mathbf{i}\infty}(\theta_{\infty\mathbf{i}}(\mathbf{x}))$$
$$\Psi = \lambda\mathbf{x}.\ \bigsqcup_{\mathbf{i}} \theta_{\mathbf{i}\infty}(\theta'_{\infty\mathbf{i}}(\mathbf{x}))$$

Påstanden er at disse funktioner etablerer en isomorfi.

Bevis. Vi skal først vise at funktionerne er veldefinerede; altså at $\theta'_{\mathbf{i}\infty}(\theta_{\infty\mathbf{i}}(\mathbf{d}))$ for fast $\mathbf{d}$ er en kæde i $\mathbf{D}_\infty$. Til det benytter vi at

$$\phi_{\mathbf{i}}(\theta_{\infty\mathbf{i}}(\mathbf{d})) \sqsubseteq \theta_{\infty\mathbf{i}+1}(\mathbf{d})$$
$$\theta'_{\mathbf{i}\infty}(\psi_{\mathbf{i}}(\mathbf{d})) \sqsubseteq \theta'_{\mathbf{i}+1\infty}(\mathbf{d})$$

idet vi da har at

$$\theta'_{\mathbf{i}\infty}(\psi_{\mathbf{i}}(\phi_{\mathbf{i}}(\theta_{\infty\mathbf{i}}(\mathbf{d})))) \sqsubseteq \theta'_{\mathbf{i}+1\infty}(\theta_{\infty\mathbf{i}+1}(\mathbf{d}))$$

Vi skal derefter vise at $\Phi \circ \Psi = \Psi \circ \Phi = \mathsf{id}$. Der er fire uligheder at vise:

<u>1.</u> $\forall\mathbf{d} \in \mathbf{D}_\infty.\ \Psi(\Phi(\mathbf{d})) \sqsubseteq \mathbf{d}$. For et givet $\mathbf{d}$ får vi

$$\begin{aligned}\Psi(\Phi(\mathbf{d})) &= \bigsqcup_{\mathbf{i}} \theta_{\mathbf{i}\infty}(\theta'_{\infty\mathbf{i}}(\theta'_{\mathbf{i}\infty}(\theta_{\infty\mathbf{i}}(\mathbf{d})))) \\ &= \bigsqcup_{\mathbf{i}} \theta_{\mathbf{i}\infty}(\theta_{\infty\mathbf{i}}(\mathbf{d})) \\ &\sqsubseteq \mathbf{d}\end{aligned}$$

da $\forall\mathbf{i}.\ \theta_{\mathbf{i}\infty}(\theta_{\infty\mathbf{i}}(\mathbf{d})) \sqsubseteq \mathbf{d}$.

<u>2.</u> $\forall\mathbf{d} \in \mathbf{D}_\infty.\ \Psi(\Phi(\mathbf{d})) \sqsupseteq \mathbf{d}$. Givet et $\mathbf{d} = [\mathbf{d}_0, \mathbf{d}_1, \ldots] \in \mathbf{D}_\infty$. For ethvert $\mathbf{i}$ har vi

$$\begin{aligned}\theta_{\infty\mathbf{i}}(\Psi(\Phi([\mathbf{d}_0, \mathbf{d}_1, \ldots]))) &= \bigsqcup_{\mathbf{j}} \theta_{\infty\mathbf{i}}(\theta_{\mathbf{j}\infty}(\theta_{\infty\mathbf{j}}(\mathbf{d}))) \\ &\sqsupseteq \theta_{\infty\mathbf{i}}(\theta_{\mathbf{i}\infty}(\theta_{\infty\mathbf{i}}(\mathbf{d}))) \\ &= \theta_{\infty\mathbf{i}}(\mathbf{d}) = \mathbf{d}_{\mathbf{i}}\end{aligned}$$

3 og 4. De to andre uligheder, der skal vises er $\forall \mathbf{d} \in \mathbf{F}(\mathbf{D}_\infty).\ \Phi(\Psi(\mathbf{d})) \sqsupseteq \mathbf{d}$ og $\forall \mathbf{d} \in \mathbf{F}(\mathbf{D}_\infty).\ \Phi(\Psi(\mathbf{d})) \sqsubseteq \mathbf{d}$. Beviset følger her et tilsvarende mønster.

Bevis slut.

Terminologi. Ja det var en ordentlig omgang. Vi har altså fundet en løsning til en rekursiv domæneligning ved at konstruere produktet af en sekvens af approximationer. På engelsk kaldes hele denne konstruktion ofte for *the inverse limit construction.*

Ekstra sætning. Hvis vi skulle afslutte parallellen fra fixpunktsætningen skal vi ikke alene vise at vi har fundet et fixpunkt (op til isomorfi) for domæneligningen, men også vise at det er den mindste. Vi skal altså vise at hvis vi har en løsning så kan $\mathbf{D}_\infty$ indlejres i denne løsning. Et bevis for dette overlades som en øvelse til læseren!

Resume. Det væsentligste i denne tekst skulle gerne være at give et indtryk af hvordan man konstruerer løsninger til rekursive domæneligninger. Der er masser af små detaljer i beviset for korrektheden af løsningen, men selve konstruktionen burde være lettere at overskue.

Vi starter med en domæneligning $\mathbf{D} \simeq \mathbf{F}(\mathbf{D})$ og konstruerer approximationerne $\mathbf{D_i} = \mathbf{F^i}(\mathbb{U})$. Vi bemærker at $\mathbf{D_i}$ kan indlejres i $\mathbf{D_{i+1}}$, altså at $\mathbf{D_i}$ kan opfattes som en delstruktur af $\mathbf{D_{i+1}}$. Vi konstruerer herefter et domæne $\mathbf{D}_\infty$ af bestemte følger af elementer fra disse approximationer og viser at dette domæne opfylder domæneligningen.

Når vi benytter domæner defineret ved rekursive domæneligninger er vi ofte kun interesseret i de elementer der svarer til en endelig udfoldning af domæneligningen. I lambdakalkulen er det således normalt kun lambdatermer af endelig funktionalitet vi er interesseret i. Løsningen indeholder alle disse idet approximationerne kan indlejres i løsningen, men derudover får vi altså også elementer svarende til en uendelige udfoldning af domæneligningen.

6.5 Semantik for højere ordens funktioner

Lad os straks se en anvendelse af dette idet vi vil konstruere en semantik for et lille højere ordens sprog. Pointen er her at vi i denne semantik vil have et underliggende domæne, hvor værdier enten kan være basisværdier (tal, boolske udtryk el.lign.) eller funktioner mellem værdier.

Som basisværdier vælger vi her de naturlige tal. Domæneligningen kan så udtrykkes som

$$\mathbf{D} \simeq \mathbb{N}_{\perp} + (\mathbf{D} \to \mathbf{D})$$

Sproget vil være en form for λ-kalkule med konstanter

$$\begin{array}{lll} \mathbf{e} & ::= \mathbf{x_i} & \text{variabler} \\ & \mid \quad \lambda\mathbf{x_j}.\ \mathbf{e} & \text{funktioner} \\ & \mid \quad \mathbf{e}_1(\mathbf{e}_2) & \text{funktionskald} \end{array}$$

Semantikken defineres som en semantisk funktion

$$\begin{aligned} &\mathsf{E}[\![\mathbf{e}]\!] : (\mathbf{Var} \to \mathbf{D}) \to \mathbf{D} \\ &\mathsf{E}[\![\mathbf{x_i}]\!]\rho = \rho(\mathbf{x_i}) \\ &\mathsf{E}[\![\lambda\mathbf{x_j}.\ \mathbf{e}]\!]\rho = \Psi(\mathsf{inr}(\lambda\mathbf{y}.\ \mathsf{E}[\![\mathbf{e}]\!]\rho[\mathbf{x_j} \mapsto \mathbf{y}])) \\ &\mathsf{E}[\![\mathbf{e}_1(\mathbf{e}_2)]\!]\rho = \mathsf{outr}(\Phi(\mathsf{E}[\![\mathbf{e}_1]\!]\rho))(\mathsf{E}[\![\mathbf{e}_2]\!]\rho) \end{aligned}$$

Konstanter kan defineres i et initielt sæt omgivelser. Vi kan for eksempel tænke os **add** og **fix**:

$$\begin{aligned} &\mathbf{add} = \Psi(\mathsf{inr}(\lambda\mathbf{x}.\ \Psi(\mathsf{inr}(\lambda\mathbf{y}.\ \Psi(\mathsf{inl}(\mathsf{outl}(\Phi(\mathbf{x})) +_{\perp} \mathsf{outl}(\Phi(\mathbf{y}))))))) \\ &\mathbf{fix} = \Psi(\mathsf{inr}(\lambda\mathbf{f}.\ \mathsf{fix}(\mathsf{outr}(\Phi(\mathbf{f}))))) \end{aligned}$$

Alternativer. Vi har her defineret én semantik for sproget. Ved at ændre på domæneligningen kan man for eksempel introducere en speciel værdi til typefejl således at brug af funktioner som tal og omvendt repræsenteres som denne værdi. Semantikken her er desuden doven, så argumenter udregnes ved brug og ikke ved kald. Her kan man tænke sig en semantik, hvor argumenter ved kald udregnes til man ved om de er tal eller funktioner.

Nemme løsninger. I praksis vil man normalt stiltiende glemme isomorfier og funktionerne på sumdomænet. Gør vi det bliver den semantiske funktion til

$$\begin{aligned} &\mathsf{E}[\![\mathbf{x_i}]\!]\rho = \rho(\mathbf{x_i}) \\ &\mathsf{E}[\![\lambda\mathbf{x_j}.\ \mathbf{e}]\!]\rho = \lambda\mathbf{y}.\ \mathsf{E}[\![\mathbf{e}]\!]\rho[\mathbf{x_j} \mapsto \mathbf{y}] \\ &\mathsf{E}[\![\mathbf{e}_1(\mathbf{e}_2)]\!]\rho = \mathsf{E}[\![\mathbf{e}_1]\!]\rho(\mathsf{E}[\![\mathbf{e}_2]\!]\rho) \end{aligned}$$

og konstanterne til

$$\begin{aligned} &\mathbf{add} = \lambda\mathbf{x}.\ \lambda\mathbf{y}.\ \mathbf{x} +_{\perp} \mathbf{y} \\ &\mathbf{fix} = \lambda\mathbf{f}.\ \mathsf{fix}(\mathbf{f}) \end{aligned}$$

Det er selvfølgelig ikke helt korrekt. Den forkerte måde er lige godt noget nemmere at læse så når vi nu ved hvad vi gør er der vel ingen grund til at gå og prale med at vi ved hvordan det gøres korrekt.

6.6 Opgaver

Opgave 6.1. Hvilke af disse ligninger har etpunktsdomænet som løsning?

$$
\begin{array}{llll}
\mathbf{D} \simeq \mathbf{D} \to \mathbf{D} & \mathbf{D} \simeq \mathbf{D} \times \mathbf{D} & \mathbf{D} \simeq \mathbf{D} + \mathbf{D} & \mathbf{D} \simeq \{1\} + (\mathbf{D} \to \mathbf{D}) \\
\mathbf{D} \simeq \mathbf{D} \to (\mathbf{D}_\perp) & \mathbf{D} \simeq \mathbf{D}_\perp \times \mathbf{D} & \mathbf{D} \simeq \mathbf{D} \oplus \mathbf{D} & \mathbf{D} \simeq \{1\} \oplus (\mathbf{D} \to \mathbf{D}) \\
\mathbf{D} \simeq (\mathbf{D} \to \mathbf{D})_\perp & \mathbf{D} \simeq \mathbf{D}_\perp \times \mathbf{D}_\perp & \mathbf{D} \simeq \mathbf{D}_\perp \oplus \mathbf{D}_\perp & \mathbf{D} \simeq \{1\} + (\mathbf{D} \times \mathbf{D}) \\
\mathbf{D} \simeq \mathbf{D}_\perp \to \mathbf{D} & \mathbf{D} \simeq (\mathbf{D} \times \mathbf{D})_\perp & \mathbf{D} \simeq \mathbf{D}_\perp \otimes \mathbf{D}_\perp & \mathbf{D} \simeq \{1\}_\perp \oplus (\mathbf{D} \times \mathbf{D})
\end{array}
$$

Opgave 6.2. Angiv de 3 første approximationer for 3 af de ligninger ovenfor som ikke har etpunktsdomænet som løsning. Hvormange elementer indeholder hver af disse approximationer?

Opgave 6.3. Afslut beviset for at $\mathbf{D}_\infty$ faktisk er et domæne. Det vil sige at man skal bevise at $\mathbf{D}_\infty$ har grænseværdier af kæder.

Opgave 6.4. Find en del af konstruktionen af $\mathbf{D}_\infty$ som ikke var bevist formelt i disse noter og gennemfør et bevis for det.

Opgave 6.5. Lad $\mathbf{D}'$ være en løsning op til isomorfi for ligningen $\mathbf{D} \simeq \mathbf{F}(\mathbf{D})$. Vis at løsningen $\mathbf{D}_\infty$ kan indlejres i $\mathbf{D}'$.

Opgave 6.6. Angiv en løsning op til isomorfi for ligningen

$$\mathbf{D} \simeq (\mathbf{D} \times \mathbf{D})_\perp$$

6.7 Afsluttende bemærkning

Tak for kommentarer og rettelser fra Nils Andersen, Peter Holst Andersen, Arne Glenstrup, Dimitri Graedel, Jens Kruse, Kristian Nielsen, Henning Niss, Peter Skadhauge.

Yderligere rettelser og forslag til forbedringer af disse noter modtages med glæde.