Process Algebra
Uitwerkingen opgaven practicum 1

Hieronder staan uitwerkingen van de volgende opgaven:

1.2.11: 1 en 6;

1.3.15: 1, 2 en 3;

1.4.8: 3, 4 en 5.

Terms and Equations

Exercise 1.2.11.1 Show that $E_1 \vdash m(s(s(0)), s(s(0))) = a(s(s(0)), s(s(0)))$.

Answer 1.2.11.1 The equational specification E_1 is given by the signature

$$\Sigma_1 = \{0, s(\_), a(\_,\_), m(\_,\_)\}$$

and the equations

$$\begin{array}{lll} E_1 = \{ & a(x, 0) = x & A1 \\ , & a(x, s(y)) = s(a(x, y)) & A2 \\ , & m(x, 0) = 0 & A3 \\ , & m(x, s(y)) = a(m(x, y), x) & A4 \\ \} & & \end{array}$$

Then a possible derivation would look like:

$$\begin{array}{lll} & \underline{m(s(s(0)), s(s(0)))} & A4 \\ = & a(\underline{m(s(s(0)), s(0))}, s(s(0))) & A4 \\ = & a(a(\underline{m(s(s(0)), 0)}, s(s(0))), s(s(0))) & A3 \\ = & a(\underline{a(0, s(s(0)))}, s(s(0))) & A2 \\ = & a(s(\underline{a(0, s(0))}), s(s(0))) & A2 \\ = & a(s(s(\underline{a(0, 0)})), s(s(0))) & A1 \\ = & a(s(s(0)), s(s(0))) & \end{array}$$

Please note that we have used the "standard rules" for derivation without explicitly mentioning them.

Exercise 1.2.11.6 Let $E_3 = \{F(F(F(x))) = x(A1), F(F(F(F(F(x))))) = x(A2)\}$. Prove that $E_3 \vdash F(x) = x$.

Answer 1.2.11.6 This equational specification can only alter the number of F symbols of a given term. Application of A1 decreases/increases the number of F symbols with 3. Application of A2 decreases/increases the number of F symbols with 5. Thus we are looking for a combination of decreasing and increasing the number of F symbols from 1 to 0 F symbols. Note that it is not allowed to have an intermediate result with a negative number of F symbols. This simple analysis of the equational specification suggests that we first add 5 F symbols and then twice subtract three F symbols. Then we obtain the following proof

$$\begin{array}{lll} & F(\underline{x}) & A2 \\ = & F(F(F(\underline{F(F(F(x)))}))) & A1 \\ = & \underline{F(F(F(x)))} & A1 \\ = & x. & \end{array}$$

Of course there are also other possibilities to obtain the desired result.

Algebras

Exercise 1.3.15.1 Show that $\mathbb{B} = (\mathbf{B}, \otimes, \wedge, \neg, 0)$ is indeed a model of (Σ_1, E_1).

$\otimes$	0	1
0	0	1
1	1	0

$\wedge$	0	1
0	0	0
1	0	1

$\neg$	
0	1
1	0

Answer 1.3.15.1 Let us first consider some definitions. The algebra $\mathbb{B}$ is a model of (Σ_1, E_1), notation $\mathbb{B} \models E_1$ if $\mathbb{B}$ satisfies all equations of E_1.

To prove this we have to define an interpretation such that for every equation $l = r \in E_1$, we have $\mathbb{B} \models I(l) = I(r)$. Note that the variables that appear in $I(l) = I(r)$ are supposed to be universally quantified.

Consider the following interpretation I: $I(a) = xor$, $I(m) = and$, $I(s) = not$, and $I(0) = 0$. Thus, we have to prove

1. $\mathbb{B} \models x \otimes 0 = x$;
2. $\mathbb{B} \models x \otimes \neg y = \neg(x \otimes y)$;
3. $\mathbb{B} \models x \wedge 0 = 0$;
4. $\mathbb{B} \models x \wedge \neg y = (x \wedge y) \otimes x$.

We can easily show that these equalities hold by constructing the corresponding truth tables.

x	$x \otimes 0$	x
0	0	0
1	1	1

x	y	$x \otimes \neg y$	$\neg(x \otimes y)$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	1	1

x	$x \wedge 0$	0
0	0	0
1	0	0

x	y	$x \wedge \neg y$	$(x \wedge y) \otimes x$
0	0	0	0
0	1	0	0
1	0	1	1
1	1	0	0

Exercise 1.3.15.2 Find a model of (Σ_1, E_1) in which $a(x, y) = a(y, x)$ does not hold.

Answer 1.3.15.2 An example of such a model is the following $\mathbb{B} = (\mathbf{B}, \pi_1, \pi_2, Id, 0)$ where

$$\begin{array}{lcl} \mathbf{B} & = & \{0, 1\} \\ \pi_1(x, y) & = & x \\ \pi_2(x, y) & = & y \\ Id(x) & = & x \end{array}$$

and the following interpretation I:

$$\begin{array}{lcl} I(a) & = & \pi_1 \\ I(m) & = & \pi_2 \\ I(s) & = & Id \\ I(0) & = & 0 \end{array}$$

First we have to prove that under this interpretation I, $\mathbb{B}$ is indeed a model of (Σ_1, E_1). Thereto we must prove

1. $I(a(x, 0)) = I(x)$, i.e., $\mathbb{B} \models \pi_1(x, 0) = x$.
2. $I(a(x, s(y))) = I(s(a(x, y)))$, i.e., $\mathbb{B} \models \pi_1(x, Id(y)) = Id(\pi_1(x, y))$.
3. $I(m(x, 0)) = I(0)$, i.e., $\mathbb{B} \models \pi_2(x, 0) = 0$.
4. $I(m(x, s(y))) = I(a(m(x, y), x))$, i.e., $\mathbb{B} \models \pi_2(x, Id(y)) = \pi_1(\pi_2(x, y), x)$.

All of these are easily proven. Furthermore we must show that the equation $a(x, y) = a(y, x)$ after interpretation does not hold. Interpreting this equation gives the equation $\pi_1(x, y) = \pi_1(y, x)$. We only have to give one instantiation of x and y for which $\mathbb{B} \not\models \pi_1(x, y) = \pi_1(y, x)$. Since $\pi_1(x, y) = x$ and $\pi_1(y, x) = y$, we need to choose x and y such that $x \neq y$. Take $x = 0$ and $y = 1$. Clearly $\mathbb{B} \not\models \pi_1(0, 1) = \pi_1(1, 0)$.

Exercise 1.3.15.3 Let $\mathbb{B}$ and I as in Exercise 1.3.15.1. Verify that the equation $s(m(x,y)) = a(a(s(x), s(y)), m(s(x), s(y)))$ under interpretation I does hold in $\mathbb{B}$. Does this formula also hold in $(\mathbf{N}, +, \cdot, s, 0)$ with the interpretation $I(a) = +$, $I(m) = \cdot$, $I(s) = s$, and $I(0) = 0$?

Answer 1.3.15.3 After interpetation the equation is

$$\neg(x \wedge y) = (\neg x \otimes \neg y) \otimes (\neg x \wedge \neg y).$$

We can verify this by generating truth tables for both the left hand side and the right hand side, as follows:

x	y	$x \wedge y$	$\neg(x \wedge y)$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

x	y	$\neg x$	$\neg y$	$\neg x \otimes \neg y$	$\neg x \wedge \neg y$	$(\neg x \otimes \neg y) \otimes (\neg x \wedge \neg y)$
0	0	1	1	0	1	1
0	1	1	0	1	0	1
1	0	0	1	1	0	1
1	1	0	0	0	0	0

Since the last two columns are equal the equation holds: $\mathbb{B} \models \neg(x \wedge y) = (\neg x \otimes \neg y) \otimes (\neg x \wedge \neg y)$.

Another (more intelligent) way is a formal derivation. Remember from predicate calculus (first year computer science) that the $\otimes$ is the 'differs from' operation which is sometimes written as $\not\equiv$, that is, $x \otimes y \equiv (x \not\equiv y)$.

$$\begin{array}{ll}
 & (\neg x \not\equiv \neg y) \not\equiv \neg x \wedge \neg y \\
\equiv & \{associativity\} \\
 & \neg x \not\equiv \neg y \not\equiv \neg x \wedge \neg y \\
\equiv & \{negation\} \\
 & \neg x \equiv \neg y \equiv \neg x \wedge \neg y \\
\equiv & \{goldenrule\} \\
 & \neg x \vee \neg y \\
\equiv & \{deMorgan\} \\
 & \neg(x \wedge y)
\end{array}$$

As for the second part of the exercise, we note that interpreted in the model $\mathbb{N}$, the equation is

$$(x \cdot y) + 1 = ((x+1) + (y+1)) + ((x+1) \cdot (y+1))$$

Does it hold for all $x, y \in \mathbf{N}$? No clearly not, take $x = 0$ and $y = 0$. Then $(x \cdot y) + 1 = 1$ and $((x+1) + (y+1)) + ((x+1) \cdot (y+1)) = 3$ and $\mathbb{N} \not\models 1 = 3$.

Term Rewriting Systems

Exercise 1.4.8.3 We consider a TRS with a unary function $\neg$, and binary functions $\wedge$, $\vee$. The reduction rules (in infix notation, leaving out many brackets) are the following

$$\begin{array}{rcll} \neg\neg x & \rightarrow & x & (1) \\ \neg(x \vee y) & \rightarrow & \neg x \wedge \neg y & (2) \\ \neg(x \wedge y) & \rightarrow & \neg x \vee \neg y & (3) \\ x \vee (y \wedge z) & \rightarrow & (x \vee y) \wedge (x \vee z) & (4) \\ (x \wedge y) \vee z & \rightarrow & (x \vee z) \wedge (y \vee z) & (5) \end{array}$$

Reduce the term $\neg(x \wedge \neg y) \wedge z$ to normal form. (Note: the normal forms of this TRS are known in propositional logic as conjunctive normal forms.)

Answer 1.4.8.3

$$\underline{\neg(x \wedge \neg y)} \wedge z \quad \rightarrow_{(3)} \quad (\neg x \vee \underline{\neg\neg y}) \wedge z \quad \rightarrow_{(1)} \quad (\neg x \vee y) \wedge z$$

Exercise 1.4.8.4 Prove that the TRS with the one rule

$$ff(x)) \rightarrow f(g(f(x)))$$

is strongly normalizing.

Answer 1.4.8.4 Since the description of the exercise is a bit vague, we assume the TRS has terms of the forms c (nullary constants), $f(x)$ (applications of f), and $g(x)$ (applications of g). Note that the syntax of the nullary constant symbols is not given!

We define a function ϑ on terms that counts the number of subterms of the form $f(f(x))$:

$$\begin{array}{lcl} \vartheta(c) & = & 0 \\ \vartheta(f(c)) & = & 0 \\ \vartheta(f(f(x))) & = & 1 + \vartheta(f(x)) \\ \vartheta(g(x)) & = & \vartheta(x) \\ \vartheta(f(g(x))) & = & \vartheta(x) \end{array}$$

Note that for all terms x we have $\vartheta(x) \geq 0$.

Next, we examine the relative values of $\vartheta(x)$ and $\vartheta(x')$ if x is rewritten to x' by the rule of the TRS. There are three cases:

1. $x \equiv f(f(c))$ and $x' \equiv f(g(f(c)))$, for some nullary constant c. So, $\vartheta(x) = 1$ and $\vartheta(x') = 0$.

2. $x \equiv f(f(f(y)))$ and $x' \equiv f(g(f(f(y))))$ or $x' \equiv f(f(g(f(y))))$ for some term y. Now, we have $\vartheta(x) = 1 + 1 + \vartheta(y)$ and, in both cases for x', $\vartheta(x') = 1 + \vartheta(y)$.

3. $x \equiv f(f(g(y)))$ and $x' \equiv f(g(f(g(y))))$ for some term y. Now, we have $\vartheta(x) = 1+\vartheta(y)$ and $\vartheta(x') = \vartheta(y)$.

Therefore, after each application of the rewrite rule, the value of $\vartheta(x)$ decreases. Since we have $\vartheta(x) \geq 0$ for all terms x, the rewrite rule cannot be applicable infinitely often. Therefore, the TRS is strongly normalizing.

Exercise 1.4.8.5 Is the TRS with the one rule $f(g(x, y)) \to g(g(f(f(x)), y), y)$ strongly normalizing?

Answer 1.4.8.5 The answer is no! Consider the following reduction

$$\begin{array}{rcl} f(g(g(x,y),y)) & \to & g(g(f(f(g(x,y))),y),y) \\ & \to & g(g(f(g(g(f(f(x)),y),y)),y),y) \end{array}$$

But how do you reach this answer? The answer is in having a good understanding of the ways in which this TRS manipulates terms. Looking at the rewriting rule you can conclude that you definitely need a subterm of the form $f(g(\Box, \Box))$ since otherwise no reduction step is possible.

But we can conclude more. In order to obtain an infinitary reduction the rewriting rule must be applied over and over again. If we apply the rewriting rule once to the artificial term $f(g(\Box, \Box))$ we obtain a term of the form $g(g(f(f(\Box)), \Box), \Box)$. And we can clearly see that the first occurrence of $\Box$ must be headed by an occurrence of g.

So as a next approximation of the solution we have the "term" $f(g(g(\Box, \Box), \Box))$. Then we have the following reductions:

$$\begin{array}{rcl} f(g(g(\Box,\Box),\Box)) & \to & g(g(f(f(g(\Box,\Box))),\Box),\Box) \\ & \to & g(g(f(g(g(f(f(\Box)),\Box),\Box)),\Box),\Box) \end{array}$$

Now one can clearly observe that we have a reduction of the form

$$f(g(g(\Box,\Box),\Box)) \to C[f(g(g(\Box,\Box),\Box))]$$

which indicates an infinitary reduction.