Program Verification: Lecture 12

José Meseguer

Computer Science Department
University of Illinois at Urbana-Champaign

Evaluating Program Expressions

Q1: Can we model the evaluation of expressions in a programming language using initial algebras?

A1: We first of all need a signature Σ of operations.

For example, Σ could be a signature for integer operations, and/or Boolean operations, and/or real number operations (typically using a floating point representation).

Assume, for example, a programming language in which we only have integers and integer operations (note that we can encode true and false as, respectively, 0 and 1). In this case Σ can be unsorted and have two constants, 0 and 1, and three binary function symbols: $\_ + \_$, $\_ - \_$, and $\_ * \_$.

Evaluating Program Expressions (II)

Q2: What else do we need?

A2: We need a set X of variables appearing on our expressions. This means that we need to extend Σ to $\Sigma(X)$, so that our program expressions will be terms $t \in T_{\Sigma(X)}$.

Q3: And what else do we need if we want to evaluate such expressions?

A3: We of course need a Σ-algebra in which they will be evaluated. For integers expressions the most natural choice is the algebra $\mathbb{Z} = (\mathbb{Z}, \_\_{\mathbb{Z}})$ of the integers, with the standard interpretation $\_\_{\mathbb{Z}}$ for $+, *, -, 0, 1$.

Evaluating Program Expressions (III)

Q4: And what else do we need?

A4: Since expression evaluation depends on the memory state, we need to model mathematically memory states.

Q5: And how can we model memory states?

A5: Assuming programs with just global variables, a memory state for arithmetic expressions is just a function $m : X \to \mathbb{Z}$. This is a special instance of the general notions of an assignment of values to variables in an algebra.

Assignments

Given variables in $X = \{X_s\}$ we will often be interested in assignments (also called valuations) of data elements in a given Σ-algebra $\mathbb{A} = (A, \_\_{}_{\mathbb{A}})$ to those variables. Of course, if $x \in X_s$ then the value, say $a(x)$, assigned to x should be an element of A_s. That is, the assignments should be well-sorted. This can be made precise by defining an assignment to the variables X in a Σ-algebra $\mathbb{A} = (A, \_\_{}_{\mathbb{A}})$ to be an S-indexed family of functions, $a = \{a_s : X_s \longrightarrow A_s\}_{s \in S}$, denoted $a : X \longrightarrow A$.

Often what we want to do with such assignments is to extend them from variables to terms on such variables in the obvious, homomorphic way. This is what expression evaluation is all about.

Evaluating Program Expressions (VI)

Q6: Now that we have everything we need, how can evaluation of arithmetic expressions be precisely defined relative to a memory (state) $m : X \to \mathbb{Z}$?

A6: As a function $\_\_{(\mathbb{Z},m)} : T_{\Sigma(X)} \to \mathbb{Z}$ defined inductively by:

1. $x_{(\mathbb{Z},m)} = m(x)$ for $x \in X$
2. $0_{(\mathbb{Z},m)} = 0 \in \mathbb{Z}$, $1_{(\mathbb{Z},m)} = 1 \in \mathbb{Z}$
3. $f(t,t')_{(\mathbb{Z},m)} = f_{\mathbb{Z}}(t_{(\mathbb{Z},m)}, t'_{(\mathbb{Z},m)})$ for $f \in \{+, *, -\}$.

Evaluating Program Expressions (VII)

Q7: Conditions (2)–(3) show that $\_\_{(\mathbb{Z},m)}$ is a Σ-homomorphism. What about condition (1)?

A7: Condition (1) plus (2)–(3) show that it is a $\Sigma(X)$-homomorphism, when we extend the algebra $\mathbb{Z}$ of the integers with the additional constants X, where each $x \in X$ is interpreted in $\mathbb{Z}$ as $m(x)$. Therefore, the extension of $\mathbb{Z}$ to a $\Sigma(X)$-algebra is just $(\mathbb{Z}, \_\_{\mathbb{Z} \uplus m})$, which we abbreviate to: $(\mathbb{Z}, m)$. Then the evaluation of arithmetic expressions is the unique $\Sigma(X)$-homomorphism:

$$\_\_{(\mathbb{Z},m)} : \mathbb{T}_{\Sigma(X)} \rightarrow (\mathbb{Z}, m)$$

to the $\Sigma(X)$-algebra $(\mathbb{Z}, m)$ (extending the Σ-algebra $\mathbb{Z}$ with memory m) ensured by the initiality of $\mathbb{T}_{\Sigma(X)}$.

More on $\Sigma(X)$-Algebras

The evaluation of integer arithmetic expressions with memory $m : X \rightarrow \mathbb{Z}$ was formalized as the unique $\Sigma(X)$-homomorphism:

$$\_\__{(\mathbb{Z},m)} : \mathbb{T}_{\Sigma(X)} \rightarrow (\mathbb{Z}, m)$$

where $(\mathbb{Z}, m)$ extends the integer Σ-algebra $\mathbb{Z} = (\mathbb{Z}, \_\__{\mathbb{Z}})$ by interpreting the constants X as the memory map $m : X \rightarrow \mathbb{Z}$.

This situation is completely general: For any signature Σ and any Σ-algebra $\mathbb{A} = (A, \_\__{\mathbb{A}})$, an assignment, i.e., a "memory map," $a : X \rightarrow A$ extends $\mathbb{A}$ to the $\Sigma(X)$-algebra $(A, \_\__{\mathbb{A}} \uplus a)$, abbreviated to $(\mathbb{A}, a)$, and the evaluation of $\Sigma(X)$-expressions in $\mathbb{A}$ is the unique $\Sigma(X)$-homomorphism:

$$\_\__{(\mathbb{A},a)} : \mathbb{T}_{\Sigma(X)} \rightarrow (\mathbb{A}, a)$$

Notation: $\_\__{(\mathbb{A},a)}$ is abbreviated to $\_a : \mathbb{T}_{\Sigma(X)} \rightarrow (\mathbb{A}, a)$.

More $\Sigma(X)$-Algebras (II)

We can summarize this situation as the following:

Fact 1: Any pair $(\mathbb{A}, a)$ with $\mathbb{A} = (A, \_{\mathbb{A}})$ a Σ-algebra and $a : X \to A$ an assignment defines a $\Sigma(X)$-algebra $(\mathbb{A}, a)$.

Q: Are all $\Sigma(X)$-algebras of this form?

A: Yes! We just need to recall the definition of an order-sorted Σ-algebra in Lecture 3:

For $\Sigma = ((S, <), F, G)$ a signature, Σ-algebra $\mathbb{A} = (A, \_{\mathbb{A}})$ is just a pair with: (i) $A_s \subseteq A_{s'}$ if $s < s'$ and (ii) $\_{\mathbb{A}}$ a function $\_{\mathbb{A}} : f \mapsto f_{\mathbb{A}}$ interpreting each constant $c :\to s$ as an element $c_{\mathbb{A}} \in A_s$, and each symbol $f : w \to s$ in Σ as a function $f_{\mathbb{A}} \in [A^w \to A_s]$, so that, if f has subsort overloaded typings, the different $f_{\mathbb{A}}$ agree on common data.

More $\Sigma(X)$-Algebras (III)

If $\Sigma = ((S,<),F,G)$, then $\Sigma(X) = ((S,<),F \uplus \bigcup_{s \in S} X_s, G \uplus \overline{X})$, where $\uplus$ denotes disjoint union of function symbols $(F \uplus \bigcup_{s \in S} X_s)$ and of typings $(G \uplus \overline{X})$, where, by definition,
$\overline{X} = \{x :\to s \mid x \in X_s,\ \ s \in S\}$.

Recall from STACS that if $U \cap V = \emptyset$, any function $h : U \uplus V \to W$ decomposes uniquely as a disjoint union $h = h|_U \uplus h|_V$ of the restriction functions $h|_U : U \to W$ and $h|_V : V \to W$.

Therefore, if $\mathbb{B} = (B, \_\_{\mathbb{B}})$ is a $\Sigma(X)$-algebra, then $\_\_{\mathbb{B}}$ decomposes uniquely as a pair $(\_\_{\mathbb{B}}|_G, \_\_{\mathbb{B}}|_{\overline{X}})$. But note that $\_\_{\mathbb{B}}|_{\overline{X}} : X \to B$ is just an assignment! and $(B, \_\_{\mathbb{B}}|_G)$ is just a Σ-algebra! **Notation**: $(B, \_\_{\mathbb{B}}|_G) = \mathbb{B}|_{\Sigma}$, is called the Σ-reduct of $\mathbb{B}$.

Fact 2: $\mathbb{B} = (B, \_\_{\mathbb{B}})$ decomposes uniquely as $\mathbb{B} = (\mathbb{B}|_{\Sigma}, \_\_{\mathbb{B}}|_{\overline{X}})$.

More $\Sigma(X)$-Homomorphisms

Facts 1 and 2 tell us that any $\Sigma(X)$-algebra is exactly of the form $(\mathbb{A}, a) =_{def} (A, \_\_{}_{\mathbb{A}} \uplus a)$, with $\mathbb{A}$ a Σ-algebra and $a \in [X {\to} A]$ an assignment.

Q: What is a $\Sigma(X)$-homomorphism $h : (\mathbb{A}, a) \to (\mathbb{C}, c)$?

A: The answer is summarized in **Fact** 3 below.

Fact 3: Since h must preserve both the interpretation the F-typings G and the $\bigcup_{s \in S} X_s$-typings $\overline{X}$ of the new constants $\bigcup_{s \in S} X_s$, but $G \cap \overline{X} = \emptyset$, h is exactly:

1. a Σ-homomorphism $h : \mathbb{A} \to \mathbb{C}$ such that
2. for each $s \in S$ and $x \in X_s$, $h_s(a(x)) = c(x)$, i.e., $a; h = c$.

Example: Substitutions Revisited

Let us apply **Fact** 2 to the initial $\Sigma(X)$-algebra $\mathbb{T}_{\Sigma(X)} = (T_{\Sigma(X)}, \_\mathbb{T}_{\Sigma(X)})$. What unique decomposition do we get for $\mathbb{T}_{\Sigma(X)}$? We get a pair $(\mathbb{T}_{\Sigma(X)}|_{\Sigma}, \eta_X)$, where:

1. $\mathbb{T}_{\Sigma(X)}|_{\Sigma} = (T_{\Sigma(X)}, \_\mathbb{T}_{\Sigma(X)}|_G)$, that is, the elements $t \in T_{\Sigma(X)}$ are the same: (Σ-terms with variables in X), but only the Σ-operations are considered; and

2. $\eta_X : X \to T_{\Sigma(X)} : x \mapsto x$ is the identity assignment for each variable x in X, that is, the identity substitution.

To simplify the notation, we will denote $\mathbb{T}_{\Sigma(X)}|_{\Sigma}$ by $\mathbb{T}_{\Sigma}(X)$, and will call it the free Σ-algebra on the variables X.

Example: Substitutions Revisited (II)

Consider now another S-sorted set Y of variables and a substitution $\theta : X \to T_{\Sigma(Y)}$.

Q: how can we model the extension of θ to the map on terms $\_\theta : T_{\Sigma(X)} \to T_{\Sigma(Y)}$ defined in Lecture 3?

A: Easy! Consider the $\Sigma(X)$-algebra $(\mathbb{T}_\Sigma(Y), \theta)$. Then, $\_\theta$ is just the unique $\Sigma(X)$-homomorphism:

$$\_\theta : \mathbb{T}_{\Sigma(X)} \to (\mathbb{T}_\Sigma(Y), \theta),$$

which decomposing $\mathbb{T}_{\Sigma(X)}$ as $\mathbb{T}_{\Sigma(X)} = (\mathbb{T}_\Sigma(X), \eta_X)$, is the unique $\Sigma(X)$-homomorphism:

$$\_\theta : (\mathbb{T}_\Sigma(X), \eta_X) \to (\mathbb{T}_\Sigma(Y), \theta).$$

Example: Substitutions Revisited (III)

But by Fact 3, $\_\theta : (\mathbb{T}_\Sigma(X), \eta_X) \to (\mathbb{T}_\Sigma(Y), \theta)$ is a $\Sigma(X)$-homomorphism iff:

1. $\_\theta : \mathbb{T}_\Sigma(X) \to \mathbb{T}_\Sigma(Y)$ is a Σ-homomorphism, and

2. $\eta_X ; \_\theta = \theta$

Therefore, each substitution θ has a unique extension to a Σ-homomorphism $\_\theta$ such that the following diagram commutes:

Homomorphic Extension of Substitutions

$$\begin{tikzcd}
X \arrow[r, "\eta_X"] \arrow[dr, "\forall\theta"'] & T_{\Sigma(X)} \arrow[d, dashed, "\_\theta"] \\
 & T_{\Sigma(Y)}
\end{tikzcd}$$

$$\begin{tikzcd}
\mathbb{T}_{\Sigma}(X) \arrow[d, dashed, "\exists!\_\theta"'] \\
\mathbb{T}_{\Sigma}(Y)
\end{tikzcd}$$

$\mathbf{Set}^S$: S-Indexed Families and S-Indexed Functions $\mathbf{Alg}_\Sigma$: Σ-Algebras and Σ-Homomorphism

Freeness Theorem

The extension $\theta \mapsto \_\theta$ is an instance of the more general:

Theorem (Freeness Theorem). For each Σ-algebra $\mathbb{A} = (A, \__{\mathbb{A}})$, and assignment $a : X \longrightarrow A$ there exists a unique Σ-homomorphism $\_a : \mathbb{T}_\Sigma(X) \longrightarrow \mathbb{A}$ such that $\eta_X ; \_a = a$.

Proof: Since $(\mathbb{A}, a)$ is a $\Sigma(X)$-algebra, by the initiality of $\mathbb{T}_{\Sigma(X)}$ there is a unique $\Sigma(X)$-homomorphism

$$\_a : \mathbb{T}_{\Sigma(X)} \to (\mathbb{A}, a),$$

which decomposing $\mathbb{T}_{\Sigma(X)}$ as $\mathbb{T}_{\Sigma(X)} = (\mathbb{T}_\Sigma(X), \eta_X)$, is the same thing as a unique $\Sigma(X)$-homomorphism:

$$\_a : (\mathbb{T}_\Sigma(X), \eta_X) \to (\mathbb{A}, a),$$

which by the definition of $\Sigma(X)$-homomorphism is the same thing as a unique Σ-homomorphism

$$\_a : \mathbb{T}_{\Sigma}(X) \to \mathbb{A}$$

such that $\eta_X ; \_a = a$, as desired. q.e.d.

This theorem can be summarized in the following diagram:

$\mathbb{T}_\Sigma(X)$ as a Free Σ-Algebra on X

$$\begin{tikzcd}
X \arrow[r, "\eta_X"] \arrow[dr, "\forall a"'] & T_{\Sigma(X)} \arrow[d, dashed, "\_a"] \\
 & A
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\mathbb{T}_\Sigma(X) \arrow[d, dashed, "\exists!\_a"'] \\
\mathbb{A}
\end{tikzcd}$$

$\mathbf{Set}^S$: S-Indexed Families and S-Indexed Functions $\mathbf{Alg}_\Sigma$: Σ-Algebras and Σ-Homomorphism

Useful Corollary on Free Σ-Algebras

Corollary (Freeness Corollary). For any Σ-homomorphism $h : \mathbb{A} \to \mathbb{B}$, and assignments $a : X \to A$, $b : X \to B$ such that $a; h = b$, the following identity between Σ-homomorphisms holds:

$$\_a; h = \_b$$

Proof: $\_a; h$ is a Σ-homomorphism $\_a; h : \mathbb{T}_{\Sigma}(X) \to \mathbb{B}$. But since, by hypothesis, we have $a; h = b$, we must also have: $\eta_X ; \_a; h = a; h = b$, which by the Freeness Theorem forces $\_a; h = \_b$, as desired. q.e.d.

The corollary can be summarized in the following diagram:

Useful Corollary on Free Σ-Algebras (II)

$$\begin{tikzcd}
X \arrow[r, "a"] \arrow[dr, "b"'] & A \arrow[d, "h"] \\
 & B
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{T}_{\Sigma}(X) \arrow[r, "\_a"] \arrow[dr, "\_b"'] & \mathbb{A} \arrow[d, "h"] \\
 & \mathbb{B}
\end{tikzcd}$$

$\mathbf{Set}^{S}$: S-Indexed Families and S-Indexed Functions $\mathbf{Alg}_{\Sigma}$: Σ-Algebras and Σ-Homomorphism

What is “free” about a Free Algebra?

Clearly, the concept of a free Σ-algebra $\mathbb{T}_\Sigma(X)$ generalizes the case of an initial algebra, since when $X = \emptyset$, where $\emptyset$ here denotes the S-indexed set having all its components empty, we have $\mathbb{T}_\Sigma(\emptyset) = \mathbb{T}_\Sigma$. As in the case of initial algebras, free algebras have (provided Σ is sensible) no confusion. Therefore, the first meaning of “free” is that no equalities force terms in $\mathbb{T}_\Sigma(X)$ to become equal: they are all different, unconstrained, and in this sense “free.”

Note that if X is nonempty $\mathbb{T}_\Sigma(X)$ has junk! (even though, $\mathbb{T}_{\Sigma(X)}$, with the same data elements, doesn’t!). Which junk? Well, X, of course, and all the junk spread by X when building terms with variables. However, this “junk” is very well-behaved.

What is “free” about a Free Algebra? (II)

X is well-behaved: we can feely interpret the variables in X as data elements in any Σ-algebra $\mathbb{B}$ by any assignment $b : X \longrightarrow B$ with the guarantee that b will always extend to a unique Σ-homomorphism $\_b$. This freedom of interpreting variables and homomorphic extensibility provide the second meaning of “free.”

This freedom is not enjoyed by other algebras. Let Σ be the unsorted signature with constant 0 and unary s. $\mathbb{T}_\Sigma$ is the natural numbers in Peano notation. Define $\mathbb{T}_\Sigma \cup \{x, y, z\}$ with elements $T_\Sigma \cup \{x, y, z\}$, with 0 and s interpreted as before on the T_Σ part, and with $s(x) = y$, $s(y) = z$, and $s(z) = x$. Now the junk $X = \{x, y, z\}$ is badly behaved. Let $\mathbb{N}$ be the natural numbers in decimal notation with 0 and succesor. There is no assignment at all $b : X \to \mathbb{N}$ that can be extended to a Σ-homomorphism $\mathbb{T}_\Sigma \cup \{x, y, z\} \to \mathbb{N}$.

Satisfaction of Equations

Let $X = \{X_s\}$ be such that for each $s \in S$, X_s is a countably infinite set. Given a Σ-algebra $\mathbb{A}$, an assignment $a : X \to A$, and a Σ-equation $t = t'$ with variables in X, we define the satisfaction relation $(\mathbb{A}, a) \models t = t'$ by means of the equivalence,

$$(\mathbb{A}, a) \models t = t' \quad \Leftrightarrow \quad t\, a = t'\, a.$$

We then define the satisfaction relation $\mathbb{A} \models t = t'$ iff for all assignments $a : X \longrightarrow A$ we have $(\mathbb{A}, a) \models t = t'$.

Note that, since each $(\mathbb{A}, a)$ is a $\Sigma(X)$-algebra, we have defined the satisfaction of $\mathbb{A} \models t = t'$ as the satisfaction of the ground $\Sigma(X)$-equation $t = t'$ by each $(\mathbb{A}, a)$, denoted $(\mathbb{A}, a) \models t = t'$, for all assignments a.

Examples of Satisfaction

Consider the unsorted signature Σ with constants $0, 1$, and operations of addition $\_ + \_$, and multiplication $\_ * \_$. Then all the algebras $\mathbb{N}$, $\mathbb{N}_k$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, in Lecture 3, pgs. 3–5, satisfy the equations:

- $x + 0 = x$
- $x + y = y + x$
- $x + (y + z) = (x + y) + z$
- $x * 1 = x$
- $x * y = y * x$
- $x * (y * z) = (x * y) * z$

Examples of Satisfaction (II)

Consider the signature Σ for Boolean operations in page 6 of Lecture 3. Then the Σ-algebras $\mathbb{B}$ and $\mathbb{P}(X)$ satisfy the equations:

- $x\ and\ true = x \qquad (\forall x)\ \ x\ or\ false = x$
- $x\ and\ y = y\ and\ x \qquad (\forall x, y)\ \ x\ or\ y = y\ or\ x$
- $x\ and\ (y\ and\ z) = (x\ and\ y)\ and\ z$
- $x\ or\ (y\ or\ z) = (x\ or\ y)\ or\ z$
- $x\ and\ x = x \qquad x\ or\ x = x$

Examples of Satisfaction (III)

Consider the `NAT-LIST` signature in Lecture 2, and the two algebras for it defined in Lecture 4, pages 4–5. Show that the first algebra (where the sort `List` is interpreted as finite strings of natural numbers) satisfies all the equation in the module `NAT-LIST`.

Show also that the second algebra (where the sort `List` is interpreted as finite sets of natural numbers) does not satisfy the equation

```
eq length(N . L) = s length(L) .
```

Examples of Satisfaction (IV)

Consider all the examples 1–3 of algebras for the “vector-space-like” signature of Picture 4.1 defined in pages 5–6 of Lecture 4. Prove that, for x, y variables of sort `Scalar`, and v, v' variables of sort `Vector`, all these algebras satisfy the equations:

- $(x + y).v = (x.v) + (y.v)$
- $x.(v + v') = (x.v) + (x.v')$
- $0.v = \vec{0}$
- $1.v = v$

Examples of Satisfaction (V)

A permutation on n elements is a bijective function $\pi : [n] \longrightarrow [n]$, where $[n] = \{1, \ldots, n\}$. The set of all such permutations is denoted S_n and has function composition as a binary operation $\_ ; \_$ for which the identity permutation $1_{[n]} : [n] \longrightarrow [n]$ is an identity element. Also, for each $\pi \in S_n$ the inverse function π^{-1} is another permutation such that, $\pi ; \pi^{-1} = 1_{[n]} = \pi^{-1} ; \pi$. S_n is called the symmetric group on n elements, because it satisfies the group theory axioms,

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z \quad \text{(associativity)}$$
$$x \cdot 1 = x = 1 \cdot x \quad \text{(identity)}$$
$$x \cdot x^{-1} = 1 = x^{-1} \cdot x \quad \text{(inverse)}$$

Similarly, given a set X of elements, the set X^* of its strings with the concatenation operation is a monoid, because it satisfies the above associativity and identity axioms.

Models and Theorems of Theories

Given an order-sorted equational theory (Σ, E) and a Σ-algebra $\mathbb{A}$, we write $\mathbb{A} \models E$, iff $\mathbb{A}$ satisfies all the equations in E, i.e., $\forall (u = v) \in E, \mathbb{A} \models u = v$. We then call $\mathbb{A}$ a model of (Σ, E), or a (Σ, E)-algebra. For example, for (Σ, E) the theory groups (resp. monoids), a model of (Σ, E) is called a group (resp. a monoid).

Given a theory (Σ, E), what other equations, besides those in E, does any (Σ, E)-algebra satisfy? We call an equation $t = t'$ a theorem of (Σ, E) iff for each (Σ, E)-algebra $\mathbb{A}$ we have, $\mathbb{A} \models t = t'$. We then write $(\Sigma, E) \models t = t'$.

We have now two different relations: (i) $(\Sigma, E) \vdash t = t'$, telling us which equations we can mechanically prove, and (ii) $(\Sigma, E) \models t = t'$, telling us which equations are true. i.e., theorems.

Soundness and Completeness

There are now two obvious questions:

Soundness: Does the implication

$$(\Sigma, E) \vdash t = t' \quad \Rightarrow \quad (\Sigma, E) \models t = t$$

always hold? That is, is anything we can prove always true, i.e., always a theorem? For example, we can prove the equations $1^{-1} = 1$ and $(x \cdot y)^{-1} = y^{-1} \cdot x^{-1}$ from the theory of groups, but are they really theorems of group theory?

Completeness: Does the implication

$$(\Sigma, E) \models t = t' \quad \Rightarrow \quad (\Sigma, E) \vdash t = t$$

always hold? That is, can we prove all the equations that are theorems of (Σ, E)?

Exercises

Ex.12.1 For $\Sigma = ((S, \leq), F, G)$, $\Sigma' = ((S, \leq), F', G')$, with $\Sigma \subseteq \Sigma'$, and $\mathbb{A} = (A, \_\_{\mathbb{A}})$ a Σ'-algebra, define its Σ-reduct $\mathbb{A}|_{\Sigma}$ as the Σ-algebra $\mathbb{A}|_{\Sigma} = (A, \_\_{\mathbb{A}}|_{G})$. Prove that for any Σ-equation $u = v$ we have the equivalence:

$$\mathbb{A} \models u = v \quad \Leftrightarrow \quad \mathbb{A}|_{\Sigma} \models u = v.$$

Ex.12.2 (i) Let $h : \mathbb{A} \longrightarrow \mathbb{B}$ be a Σ-isomorphism, and $u = v$ a Σ-equation. Prove that

$$\mathbb{B} \models u = v \quad \Leftrightarrow \quad \mathbb{A} \models u = v.$$

(ii) Give an example of a bijective Σ-homomorphism h such that the above equivalence does not hold (Hint: Consider order-sorted signatures Σ that are not kind-complete).

Exercises (II)

Ex.12.3 Call a Σ-algebra $\mathbb{A}$ a subalgebra of a Σ-algebra $\mathbb{B}$ iff for each sort $s \in S$ we have $A_s \subseteq B_s$, and the S-family of inclusion functions $j = \{j_s : A_s \hookrightarrow B_s\}_{s \in S}$, with $j_s : A_s \ni a \mapsto a \in B_s$, is a Σ-homomorphism $j : \mathbb{A} \longrightarrow \mathbb{B}$. We then write: $\mathbb{A} \subseteq \mathbb{B}$. Show that if $\mathbb{A} \subseteq \mathbb{B}$, for any Σ-equation $u = v$ we have:

$$\mathbb{B} \models u = v \quad \Rightarrow \quad \mathbb{A} \models u = v$$

Give an example showing that the implication in the other direction in general does not hold.

Exercises (II)

Ex.12.4 Let $h : \mathbb{A} \longrightarrow \mathbb{B}$ be a surjective Σ-homomorphism, and $u = v$ a Σ-equation. Prove that

$$\mathbb{A} \models u = v \quad \Rightarrow \quad \mathbb{B} \models u = v$$

Show, by giving a counterexample, that the implication in the other direction in general does not hold.

Ex.12.5 Let $h : \mathbb{A} \longrightarrow \mathbb{B}$ be an injective Σ-homomorphism, and $u = v$ a Σ-equation. Prove that

$$\mathbb{B} \models u = v \quad \Rightarrow \quad \mathbb{A} \models u = v$$

Show, by giving a counterexample, that the implication in the other direction in general does not hold.