Chapter 3

Syntax and simulation

In this chapter we define a syntactical framework for reasoning about solutions of (possibly recursive) equations. We define a syntactic domain whose elements are called schemes (definition 3.1.1), and represent solutions of equations. We are investigating ordering-statements about these solutions, and such statements are represented by pairs of schemes, called formulae (definition 3.1.2). On the syntactic domain we define a preorder (a subset of formulae) called simulation (definition 3.2.1). In the following chapters we will show that simulation is the solution of the ordering problem.

In section 3.1 we define schemes and formulae. A scheme consists of a coalgebra (a set with a certain function) together with an element of the set of the coalgebra, and a formula consists of two schemes. In section 3.2 we define the simulation preorder on schemes, which can be seen as a “containment of coalgebra-structures”. This preorder is defined with the use of relators, and is closely related to ordered $F$-bisimulation as defined in [Rut94]. Simulation determines a subset of the set of formulae, and in chapter 5 (soundness) and chapter 6 (completeness) below we will see that this set exactly consists of the universally true formulae. In section 3.3 we define the dual of semi-homs (see definition 1.4.8), called left- and right-cohoms. Cohoms are extensively used in the completeness and soundness proof. The two kinds of cohomoms are both instances of one definition, but we prefer to define them both explicitly.

From this chapter till chapter 6, let an endofunctor $F$ on $\text{Set}$ and an extension $G$ of $F$ be given, such that the trivial extension $F$ and the extension $G$ are both monotonic. Almost every result in these chapters holds for every monotonic extension $G$, and does not depend on the monotonicity of the trivial extension. However, if we substitute $F$ for $G$ in one of those results, then monotonicity of $F$ is necessary. Because in the sequel this substitution occurs frequently, we do not want to add the premise “$F$ is monotonic” every time we do so. From corollary 2.2.1.1 it follows that the extension $G^\sim$ is also monotonic, and so in every result we can substitute $G^\sim$ for $G$.
3.1 Syntax

A scheme consists of three components, where the first two together form a coalgebra and the third is an element of the set of the coalgebra.

Definition 3.1.1 An $F$-scheme is a triple $(D, \phi, d)$, where $(D, \phi)$ is an $F$-coalgebra and $d \in D$.

We often omit the first component of a scheme, and write $(\phi, d)$. Corresponding to example 1.4.2.2, for a coalgebra $(D, \phi)$ we call $D$ the set of states, and for a scheme $(D, \phi, d)$ we call $d \in D$ the start state.

In example 1.4.2 we have seen that the set of productions of a grammar, and the transition function of a finite state machine together with a set of final states, are both examples of coalgebras. In the first case, the elements of the state set of the coalgebra are the non-terminals of the grammar, and in the second case the states of the machine. We see that if in both cases we add a start state to the coalgebra (to get a scheme) then we get a grammar and finite state machine, respectively. So finite state machines and grammars are examples of schemes. In formal language theory (see [HU79]), to grammars and machines a semantic is assigned in terms of sets of strings: the set of “generated strings” and “accepted strings”, respectively. In chapter 10 below we prove that these two semantics are both instances of one general definition, in which a semantic is assigned to schemes.

In case the signature $Q$ only contains unary and constant operators ($\eta, p$ equals 0 or 1 for every $p \in Op$), then we call $Q$-schemes machines. So finite state machines are machines with one constant operator and finite state set.

In general, we see a coalgebra as a set of variables (the state set), together with a set of equations (the coalgebra function) over these variables. A scheme represents the abstract solution, for one specified variable (the start state), of the set of equations corresponding to the coalgebra of the scheme. When we define the semantics of schemes in the next chapter, this view becomes more concrete.

Almost all the examples of schemes that we present are $Q$-schemes, because $O$-schemes and $P$-schemes are too simple. Furthermore, we introduce two other ways of representing $Q$-schemes (next to the formal one), which could cause confusion if they also applied to $P$- and $O$-schemes. These two extra notations are explained by means of an example. One of them is pictorial, and the other a kind of BNF notation.

Example 3.1.1. Assume $\rho \equiv \omega$, $Op \equiv \{se, S, p\}$, and $\eta \equiv (se, S, p) \rightarrow (2, 1, 0)$. Define the $Q$-scheme $([0, 1, 2], \phi, 0)$ by

- $\phi.0 \equiv \{(S,(2)), (se, (0, 1))\}$
- $\phi.1 \equiv \{(S,(1)), (p, ())\}$
- $\phi.2 \equiv \emptyset$

In BNF notation this becomes (with start state 0):

- $0 \rightarrow S(2) | se(0, 1)$
- $1 \rightarrow S(1) | p(())$
3.2 Simulation

In this section we define a preorder on schemes, called simulation, which depends on the monotonic extension $G$. If we substitute the trivial extension $F$ for $G$ then simulation becomes an equivalence, and this equivalence equals $F$-bisimulation on the category $\text{Set}$, as defined in [AM89].

Let two schemes be given. We first define a subset of relations between the two state sets, called simulations. These simulations are relations that satisfy a certain monotonicity property with respect to the two coalgebra functions. Now the two given schemes are related by the simulation preorder iff the greatest simulation (with respect to containment) relates their start states.

**Definition 3.2.1** Let $(D, \phi, d_0), (E, \psi, e_0)$ be $F$-schemes.
1. A relation \( R \subseteq D \times E \) is called a \( G \)-simulation from \( \phi \) to \( \psi \), notation \( R : \phi \overset{G}{\rightarrow} \psi \), iff for every \( d \in D \) and \( e \in E \) we have
\[
dRe \Rightarrow (\phi, d)R^G (\psi, e).
\]

2. The scheme \((\phi, d_0)\) is said to \( G \)-simulate the scheme \((\psi, e_0)\), notation
\[
(\phi, d_0) \preceq_G (\psi, e_0), \text{ iff there exists } R \subseteq D \times E \text{ such that } d_0 R e_0 \text{ and } R : \phi \overset{G}{\rightarrow} \psi.
\]

We know that the minimal relator corresponds to the trivial extension \( F \), and from theorem 2.1.2 we see that, for a relation \( R \), we have \( R^F \subseteq R^G \). Together with the above definition of simulation we see that \( R : \phi \overset{F}{\rightarrow} \psi \) implies \( R : \phi \overset{G}{\rightarrow} \psi \), and we get \( \preceq_F \subseteq \preceq_G \).

In words: \( F \)-simulation implies \( G \)-simulation.

Because finite state machines are \( Q \)-schemes, we can also interpret simulation for machines. We first give explicit expressions for some concrete functors.

**Example 3.2.1.**

1. Let \((D, \phi), (E, \psi)\) be \( O \)-coalgebras and \( R \subseteq D \times E \). Then \( R : \phi \overset{O}{\rightarrow} \psi \) iff, for every \( d \in D \) and \( e \in E \) we have
\[
dRe \land \phi, d \equiv (p, I) \land \psi, e \equiv (q, J) \Rightarrow (p \equiv q) \land IRJ.
\]

2. Let \((D, \phi), (E, \psi)\) be \( P \)-coalgebras and \( R \subseteq D \times E \). Then
   (a) \( R : \phi \overset{P}{\rightarrow} \psi \) iff, for every \( d \in D \) and \( e \in E \) we have
   \[
dRe \Rightarrow \forall d' \in \phi, d \exists e' \in \psi, e (d' R e').
   \]
   (b) \( R : \phi \overset{P}{\rightarrow} \psi \) iff \( R : \phi \overset{P}{\rightarrow} \psi \) and \( R^r : \psi \overset{P}{\rightarrow} \phi \).

3. Let \((D, \phi), (E, \psi)\) be \( Q \)-coalgebras and \( R \subseteq D \times E \). Then
   (a) \( R : \phi \overset{Q}{\rightarrow} \psi \) iff, for every \( d \in D \) and \( e \in E \) we have
   \[
dRe \Rightarrow \forall (p, I) \in \phi, d \exists (p, J) \in \psi, e \land IRJ.
   \]
   (b) \( R : \phi \overset{Q}{\rightarrow} \psi \) iff \( R : \phi \overset{Q}{\rightarrow} \psi \) and \( R^r : \psi \overset{Q}{\rightarrow} \phi \).

**Proof.** Parts 1 and 2.a follow immediately from definition 3.2.1 and parts 1 and 2 of theorem 2.3.2. Part 2.b follows from definition 3.2.1 and theorem 2.3.1. Finally, part 3 follows definition 3.2.1 and corollary 2.3.1.

(End of example)

Assume that \( Op, \eta \) and \( \rho \) are as in example 1.4.2.2. In this example we showed that \( Q \)-schemes are finite state machines, and so \( \preceq_Q \) and \( \preceq_G \) are relations on finite state machines. Let \( l \equiv (D, \phi, d_0) \) and \( l' \equiv (E, \psi, e_0) \) be \( Q \)-schemes and \( R \subseteq D \times E \) a relation that relates the start states \((d_0 R e_0)\) holds. From the above example we see that \( R : \phi \overset{Q}{\rightarrow} \psi \) holds iff for every state \( d \in E \) and \( e \in E \) we have

- \( dRe \land (t, d') \in \phi, d \Rightarrow \exists e' ((t, e') \in \psi, e \land d Re'). \)
Proof. Let \( dR = (st, ()) \in \phi.d \Rightarrow (st, ()) \in \psi.e \).

If we interpret this in “machine terms”, then we see that it is equivalent to the following. For every state \( d \in D \) and \( e \in E \) we have

- If states \( d \) and \( e \) are related by \( R \) and the machine \( l \) can move from state \( d \) to state \( d' \) on input symbol \( t \in T \), then there exists a state \( e' \) such that \( d' \) is related to \( e' \) by \( R \) and machine \( l' \) can move to from state \( e' \) to state \( e' \) on input symbol \( t \).

- If states \( d \) and \( e \) are related by \( R \) and \( d \) is a final state then \( e \) is also a final state.

If we regard a calculation of a finite state machine as a sequence of steps that starts in the start state and ends in a final state, then intuitively the above describes “simulation of machines” in the sense that any calculation in machine \( l \) can be stepwise copied in machine \( l' \). Furthermore, from the above example it follows that \( R : \phi \xrightarrow{G} \psi \) holds iff the above condition on \( R \) holds in both directions.

For the substitutions for the extension \( G \) in the above example, it easily follows that \( \preceq_G \) is a preorder. That this holds for any monotonic extension, follows from the axioms for monotonic relators.

**Theorem 3.2.1** The relation \( \preceq_G \) on schemes is a preorder.

Proof. We have to prove that \( \preceq_G \) is (1) reflexive and (2) transitive.

1. Let \( (D, \phi) \) be an \( F \)-coalgebra. From definition 3.2.1 we see that it suffices to prove that \( 1_D : \phi \xrightarrow{G} \phi \), that is, for \( d \in D \) we have to prove \( (\phi.d)(1_D)G(\phi.d) \). This follows from \( (1_D)G \equiv \triangleleft \) and the fact that \( \triangleleft \) is a preorder.

2. Let \( (\phi, d_0), (\psi, e_0), (\chi, j_0) \) be \( F \)-schemes such that \( (\phi, d_0) \preceq_G (\psi, e_0) \) and \( (\psi, e_0) \preceq_G (\chi, j_0) \). From definition 3.2.1.2 we get relations \( R \) and \( S \), such that \( R : \phi \xrightarrow{G} \psi \), \( S : \psi \xrightarrow{G} \chi \), \( d_0R_1e_0 \) and \( e_0R_2j_0 \). With lemma 3.2.1 below and the definition of relation composition we get \( R \circ S : \phi \xrightarrow{G} \chi \) and \( d_0(R \circ S)j_0 \). Then from definition 3.2.1.2 we get \( (\phi, d_0) \preceq_G (\chi, j_0) \).

Transitivity of \( \preceq_G \) follows from the following lemma, which states that the composition of two simulations is again a simulation.

**Lemma 3.2.1** Let \( (D, \phi), (E, \psi), \) and \( (J, \chi) \) be \( F \)-coalgebras. If \( R \subseteq D \times E \) and \( S \subseteq E \times J \) are such that \( R : \phi \xrightarrow{G} \psi \) and \( S : \psi \xrightarrow{G} \chi \) then \( R \circ S : \phi \xrightarrow{G} \chi \).

Proof. Let \( d \in D \) and \( j \in J \). According to definition 3.2.1.1 we have to prove the following.

\[
\begin{align*}
d[R \circ S]j \\
\Leftrightarrow & \quad \{\text{calculus}\} \\
\exists e & \in E(dRe \land eSj) \\
\Rightarrow & \quad \{ R : \phi \xrightarrow{G} \psi \text{ and } S : \psi \xrightarrow{G} \chi \}
\end{align*}
\]
The preorder \( \preceq_{G} \) induces the equivalence \( \preceq_{G} \cap \preceq_{G} \) on schemes, as usual denoted by \( \simeq_{G} \). If two schemes are related by \( \simeq_{G} \), then we say that they are \( G \)-similar. Immediately under definition 3.2.1 we saw that \( F \)-simulation implies \( G \)-simulation. This implies that \( F \)-similarity \( (l \simeq_{F} l') \) implies \( l \simeq_{G} l' \). This fact is used frequently, and we assume the reader to memorize it.

The extension \( G \) is by assumption monotonic, and from theorem 2.1.1.1 it follows that \( G^{-} \) is also monotonic. So we can also consider \( G^{-} \)-simulation, and we prove that it is the transposed of \( G \)-simulation. An immediate consequence is that the relation \( F \)-simulation is symmetric, and thus coincides with \( F \)-similarity.

**Theorem 3.2.2**

1. \( \preceq_{G^{-}} \equiv \preceq_{G} \) and \( \simeq_{G^{-}} \equiv \simeq_{G} \).
2. The relation \( \preceq_{F} \) is symmetric.

**Proof.**

1. The second conjunct follows from the first and some calculus, and from definition 3.2.1.2 we see that part 1 follows from lemma 3.2.2 below.
2. Because \( F^{-} \equiv F \), from part 1 we see that \( \preceq_{F} \equiv \preceq_{F}^{-} \), which exactly states symmetry.

From the above theorem we see that \( \simeq_{F} \equiv \preceq_{F} \equiv \preceq_{F}^{-} \). For this reason we do not use the notation “\( \preceq_{F}^{-} \)” or “\( \simeq_{F}^{-} \)” anymore, and always use \( \simeq_{F} \).

The above connection between \( G^{-} \)-simulation and \( G \)-simulation is a consequence of a similar connection between the by \( G^{-} \) induced relator and the by \( G \) induced relator. In terms of simulations, this connection is as follows.

**Lemma 3.2.2** Let \((D, \phi), (E, \psi)\) be \( F \)-coalgebras and \( R \subseteq D \times E \). Then

\[
R : \phi \xrightarrow{G} \psi \iff R^{-} : \psi \xrightarrow{G^{-}} \phi.
\]

**Proof.** Because \( (G^{-})^{-} \equiv G \), from symmetry we see that it suffices to prove the “\( \Rightarrow \)-part”. Assume that \( R : \phi \xrightarrow{G} \psi \). According to definition 3.2.1.1 we have to prove that, for \((e, d) \in R^{-}\), we have \((\psi.e)(R^{-})^{(G^{-})}(\phi.d)\). Let \( e \in E \) and \( d \in D \) be such that \( eR^{-}d \), that is, \( dRe \) holds. Because \( R : \phi \xrightarrow{G} \psi \) holds, from definition 3.2.1.1 we get \((\phi.d)R^{G}(\psi.e)\), and with corollary 2.2.1.1 this implies \((\psi.e)(R^{-})^{(G^{-})}(\phi.d)\).
We know that $F$-similarity implies $G$-similarity, and thus $Q$-similarity implies $Qc$-similarity. From example 3.2.1.3 one might expect that the reverse implication ($Qc$-similarity implies $Q$-similarity) also holds, but this is not the case and in example 3.2.2 below we present two schemes that are $Qc$-similar but not $Q$-similar. $Q$-similarity still is weaker than might be expected: in example 3.2.3 we present three $Q$-similar schemes that (pictorially) look differently. The fact that $Q$-similar schemes, and thus $Qc$-similar schemes, have syntactically different representations, requires some care when we introduce operations on $\simeq_Q$-equivalence classes and $\simeq_{Qc}$-equivalence classes.

**Example 3.2.2.** We present two schemes that are $Qc$-similar, but not $Q$-similar. Assume $\rho \equiv \omega, Op \equiv \{S, p, q\}$, and $\eta \equiv (S, p, q) \triangleright (1, 0, 0)$. Define the $Q$-scheme ($\{0, 1\}, \phi, 0$) by

- $0 \rightarrow S \ 1$
- $1 \rightarrow p \ | \ q$

Define the $Q$-scheme ($\{0, 1, 2\}, \psi, 0$) by

- $0 \rightarrow S \ | \ S \ 2$
- $1 \rightarrow p$
- $2 \rightarrow p \ | \ q$

A pictorial representation of both schemes ($(\phi, 0)$ on top) is given in figure 3.3. It is easily proved that $\{(0, 0), (1, 2)\} : \phi \overset{Qc}{\rightarrow} \psi$ and $\{(0, 0), (1, 1), (2, 1)\} : \psi \overset{Qc}{\rightarrow} \phi$. So
(ϕ, 0) \simeq_\mathcal{Q} (ψ, 0). That the two schemes are not \mathcal{Q}-similar, is proved by means of models in example 5.2.2 below. \hfill (End of example)

**Example 3.2.3.** We present three schemes that are \mathcal{Q}-similar. Assume \rho \equiv \omega, \ Op \equiv \{S, p\}, and \eta \equiv (S, p) \rightarrow (1, 0). Define the \mathcal{Q}\text{-scheme} (\{0\}, \phi, 0) by

- 0 \rightarrow S 0 | p

Define the \mathcal{Q}\text{-scheme} (\{0, 1\}, \psi, 0) by

- 0 \rightarrow S 1 | p
- 1 \rightarrow S 0 | p

Define the \mathcal{Q}\text{-scheme} (\{0, 1\}, \chi, 0), with \chi defined by

- 0 \rightarrow S 1 | p
- 1 \rightarrow S 1 | p

In figure 3.2 we see the pictorial representation of these schemes. It can be shown that the three schemes (ϕ, 0), (ψ, 0) and (χ, 0) are \mathcal{Q}\text{-similar}. For example, use relation \{(0, 0), (1, 0)\} : \phi \overset{\psi}. The remaining verification is simple. \hfill (End of example)

**Example 3.2.4.** We present two schemes that are \mathcal{Q}\text{-similar}, and which contain a binary operator. Assume \rho \equiv \omega, \ Op \equiv \{\text{se}, p\}, and \eta \equiv (\text{se}, p) \rightarrow (2, 0). Define the \mathcal{Q}\text{-scheme} (\{0, 1\}, \phi, 0) by

- 0 \rightarrow \text{se}(0, 1) | p
- 1 \rightarrow \text{se}(0, 1) | p

Define the \mathcal{Q}\text{-scheme} (\{0\}, \psi, 0) by

- 0 \rightarrow \text{se}(0, 0) | p
The pictorial representation is shown in figure 3.4. It is easily checked that \( \{0, 1\} \times \{0\} : \phi \rightarrow_\psi \psi \), and so \( (\phi, 0) \sim_\mathcal{Q} (\psi, 0) \).

*End of example*

In the following example one might expect \( \mathcal{Q} \)-similarity, or at least \( \mathcal{Q}_c \)-similarity, of the two schemes, but only \( \mathcal{Q}_c \)-simulation (so only one direction) is true.

**Example 3.2.5.** Assume \( \rho \equiv \omega \), \( \Omega_p \equiv (\{5, p, q\}, \) and \( \eta \equiv (5, p, q) \mapsto (1, 0, 0) \). Define the \( \mathcal{Q} \)-scheme \( (\{0, 1, 2\}, \phi, 0) \) by

- \( 0 \rightarrow S \ 1 \mid S \ 2 \)
- \( 1 \rightarrow p \)
- \( 2 \rightarrow q \)

Define the \( \mathcal{Q} \)-scheme \( (\{0, 1\}, \psi, 0) \) by

- \( 0 \rightarrow S \ 1 \)
- \( 1 \rightarrow p \mid q \)

The pictorial representations of the two is shown in figure 3.5. It is easily proved that \( \{(0, 0), (1, 1), (2, 1)\} : \phi \rightarrow_\psi \psi \), so from definition 4.2.2.2 it follows that \( (\phi, 0) \not\preceq_{\mathcal{Q}_c} (\psi, 0) \).

If one tries to find a relation \( S \) such that \( S : \psi \rightarrow_\mathcal{Q} \phi \) and \( (0, 0) \in S \), a contradiction arises. In example 5.2.1 we give an indirect proof of \( (\psi, 0) \not\preceq_{\mathcal{Q}_c} (\phi, 0) \).

*End of example*
3.3 Left- and right-cohoms

Dually to semi-homs on algebras, we define left- and right-cohoms on coalgebras. Contrary to semi-homs, the definition of left- and right-cohoms is dependent on the extension $G$. In lemma 3.3.1 below we prove that the definition of a left-cohom can be given in terms of a right-cohom, and vice versa. This is done by changing the given extension to the transposed one. Because left- and right-cohoms both are used for the given extension, we give the two definitions explicitly.

In this section let $(D, \phi), (E, \psi)$ be $F$-coalgebras, $\leq$ a preorder on $E$, and $r$ a function from $D$ to $E$.

**Definition 3.3.1**

1. $r$ is a $G$-left-cohom from $(D, \phi)$ to $(E, \leq, \psi)$ iff $(F \circ \phi) \leq^G \psi \circ r$.
2. $r$ is a $G$-right-cohom from $(D, \phi)$ to $(E, \leq, \psi)$ iff $\psi \circ r \leq^G (F \circ \phi)$.

As in the case of semi-homs, we also could have equipped the domain object with a preorder. If to the definition of a cohom we also add the condition that the cohom is a monotonic function, then the composition of two cohoms is again a cohom (which we do not prove). Because composing cohoms does not occur and the above definition has other advantages, we prefer the given definition.

If in the above definition the preorder $\leq$ equals the identity relation on $E$, then we omit the preorder in the notation, and write “$r$ is a left- or right-cohom from $(D, \phi)$ to $(E, \psi)$”. Because $1^G \equiv \triangleleft_G$, we have

- $r$ is a $G$-left-cohom from $(D, \phi)$ to $(E, \psi)$ iff $(F \circ \phi) \triangleleft \psi \circ r$.
- $r$ is a $G$-right-cohom from $(D, \phi)$ to $(E, \psi)$ iff $\psi \circ r \triangleleft (F \circ \phi)$.

Because $\triangleleft_F \equiv 1$, we see that in case the preorder $\leq$ equals the identity relation, then the notions $F$-left-cohom and $F$-right-cohom coincide, and from definition 1.4.7.2 we see that they are equivalent to the notion of $F$-cohom.

We give the explicit definitions of the left and right cohoms from $(D, \phi)$ to $(E, \psi)$ in the cases that the extension $G$ equals $Qc$ and $Q$. 

![Figure 3.5: Only Qc-simulation between schemes](image)
Example 3.3.1.

1. \( r \) is a \( Qc \)-left-cohom from \((D, \phi)\) to \((E, \psi)\) iff, for every \( d \in D \), we have
   \[
   (p, I) \in \phi. d \quad \Rightarrow \quad (p, r \circ I) \in \psi.(r.d).
   \]

2. \( r \) is a \( Qc \)-right-cohom from \((D, \phi)\) to \((E, \psi)\) iff, for every \( d \in D \), we have
   \[
   (p, J) \in \psi.(r.d) \quad \Rightarrow \quad \exists I ((p, I) \in \phi.d \land J \equiv r \circ I).
   \]

3. \( r \) is a \( Q \)-cohom from \((D, \phi)\) to \((E, \psi)\) iff \( r \) is a \( Qc \)-left-cohom and a \( Qc \)-right-cohom.

Proof. Parts 1 and 2 follow from definition 3.3.1, the fact that \( \leq_{Qc} \equiv \subseteq \), and \( Q,r.(\phi.d) \equiv \{(p, r \circ I) | (p, I) \in \phi.d\} \) (use definition 1.4.4.2). Part 3 is trivial. (End of example)

We now introduce a notational convention, which is used frequently in this thesis. Let \( (D, \phi) \) be an \( F \)-coalgebra and \( r \) a function from a set \( X \) to \( D \). Then \( (\phi, r) \) (or \( (D, \phi, r) \)) denotes the function \( x \mapsto (\phi, r.x) \). The image of this function consists of schemes, but we postpone the definition of such a set of schemes until we really need it (in chapter 6). We state two often used consequences of this notation.

- Let \( s : Y \rightarrow X \). Then \((\phi, r) \circ s \equiv (\phi, r \circ s)\).
- Let \((E, \psi)\) be an \( F \)-coalgebra and \( s : X \rightarrow D \). Then
  \[
  (\phi, r) \preceq_G (\psi, s) \iff \forall x \in X ((\phi, r.x) \preceq_G (\psi, s.x)).
  \]

A function can be seen as a relation between the domain and codomain. We prove that the relation corresponding to a left- or right-cohom is a simulation (in opposite directions).

Theorem 3.3.1 Assume that \( \leq : \psi \xrightarrow{\subseteq} \psi \).

1. If \( r \) is a \( G \)-left-cohom from \((D, \phi)\) to \((E, \leq, \psi)\) then \((\phi, \text{id}_D) \preceq_G (\psi, r)\).
2. If \( r \) is a \( G \)-right-cohom from \((D, \phi)\) to \((E, \leq, \psi)\) then \((\psi, r) \preceq_G (\phi, \text{id}_D)\).

Proof.

1. From definition 3.2.1.2 we see that it suffices to give \( R \subseteq D \times E \) such that \( dR(r.d) \) for \( d \in D \), and \( R : \phi \xrightarrow{\subseteq} \psi \). Define \( R \) by the following. For \( d \in D \) and \( e \in E \) we have
   \[
   dR e \iff r.d \leq e.
   \]

Because \( \leq \) is reflexive, we see that \( dR(r.d) \) of for \( d \in D \), and so it suffices to prove that \( R : \phi \xrightarrow{\subseteq} \psi \). Put \( S \subseteq E^2 \) equal to \( (\text{id}_D \times r)[1_D] \). It is easily seen that \( R \equiv S \circ \leq \). Furthermore, from definition 2.1.1.2.d we get \((F_2 \text{id}_D) \times (F_r)[1^D_2] \subseteq S^G \), and so for \( u \in FD \) we have
   \[
   uS^G (F.r.u) \quad (\ast).
   \]
Let $d \in D$ and $e \in E$. According to definition 3.2.1.1, it suffices prove the following.

$$(\phi, d) R^G (\psi, e)$$

$$\Leftarrow \quad \{ \text{definition 2.1.1.2.c} (R \equiv S \circ \leq), \text{calculus} \}$$

$$(\phi, d) S^G (F. r (\phi, d)) \land F. r (\phi, d) \leq^G \psi, e$$

$$\Leftarrow \quad \{ (\ast), \text{calculus} \}$$

$$F. r (\phi, d) \leq^G \psi_r (r, d) \land \psi_r (r, d) \leq^G \psi, e$$

$$\Leftarrow \quad \{ \text{definition 3.3.1.1}, \leq : \psi \xrightarrow{C_l} \psi \}$$

$$r. d \leq e$$

$$\Leftrightarrow \quad \{ \text{definition of } R \}$$

$dRe$

2. From lemma 3.3.1 we see that if $r$ is a $G$-right-cohom from $(D, \phi)$ to $(E, \leq, \psi)$ then $r$ is a $G^\sim$-left-cohom from $(D, \phi)$ to $(E, \geq, \psi)$ . With theorem 3.2.2.1 and part 1 we see that it suffices to prove $\geq : \psi \xrightarrow{C_g} \psi$ . This follows from lemma 3.2.2 and the assumption $\leq : \psi \xrightarrow{C_l} \psi$ .

If in the above theorem we have $\leq \equiv 1_E$ then the premise $\leq : \psi \xrightarrow{C_l} \psi$ becomes $1_E : \psi \xrightarrow{C_g} \psi$ . Because $1^G \equiv \leq^G$, from definition 3.2.1.1 we see that this premise is equivalent to: for every $e \in E$ we have $(\psi, e) \leq^G (\psi, e)$, which is true. This proves parts 1 and 2 of the following result.

**Corollary 3.3.1**

1. If $r$ is a $G$-left-cohom from $(D, \phi)$ to $(E, \psi)$ then $(\phi, id_D) \preceq^G (\psi, r)$ .

2. If $r$ is a $G$-right-cohom from $(D, \phi)$ to $(E, \psi)$ then $(\psi, r) \preceq^G (\phi, id_D)$ .

3. If $r$ is an $F$-cohom from $(D, \phi)$ to $(E, \psi)$ then $(\psi, r) \simeq_F (\phi, id_D)$ .

Proof. For the proof of part 3, note that an $F$-cohom is an $F$-left-cohom (as well as an $F$-right-cohom), and that $\preceq^G \equiv \simeq_F$ .

Let $(D, \phi)$ be an $F$-coalgebra and let $D_0 \subseteq D$ be such that $\phi : D_0 \rightarrow F.D_0$, that is, $(D_0, \phi)$ is an $F$-coalgebra. The inclusion is an $F$-cohom from $(D_0, \phi)$ to $(D, \phi)$, and from part 3 of the above theorem it follows that $(D_0, \phi, d) \simeq_F (D, \phi, d)$ for $d \in D_0$ . We use this simple fact without referring to the above theorem.

We prove that the property of being a right-cohom can be expressed in terms of being a left-cohom.

**Lemma 3.3.1** $r$ is a $G$-right-cohom from $(D, \phi)$ to $(E, \leq, \psi)$ iff $r$ is a $G^\sim$-left-cohom from $(D, \phi)$ to $(E, \geq, \psi)$ .

Proof. From corollary 2.2.1.1, we see that, for a preorder $\leq$, we have $\leq^G \equiv (\geq^G)^\sim$ . Then use definition 3.3.1.
Let a cohom be given. Under the condition that the preorder $\leq$ is a simulation, we prove that a function that is equivalent to the given cohom is again a cohom. This result might seem trivial (even without the assumption on $\leq$), but certainly is not (recall that $=$ is the equivalence defined as $\leq \cap \geq$).

**Lemma 3.3.2** Assume that $\leq : \psi \xrightarrow{G} \psi$. Let $s : D \to E$ be such that $r = s$. Then

1. $r$ is a $G$-left-cohom iff $s$ is a $G$-left-cohom (both from $(D, \phi)$ to $(E, \leq, \psi)$).
2. $r$ is a $G$-right-cohom iff $s$ is a $G$-right-cohom (both from $(D, \phi)$ to $(E, \leq, \psi)$).

Proof. Part 2 follows immediately from lemma 3.3.1, lemma 3.2.2, and part 1. So part 1 remains. Obviously, it suffices to prove one direction. The premise $r = s$ implies $s \leq r$, which with lemma 2.1.4 implies $F.s \leq^G F.r$. So we have $(1) (F.s) \circ \phi \leq^G (F.r) \circ \phi$. The premise $r = s$ also implies $r \leq s$, which with the assumption $\leq : \psi \xrightarrow{G} \psi$ and definition 3.2.1.1, implies $(2) \psi \circ r \leq^G \psi \circ s$. Let $r$ be a $G$-left-cohom, that is, we have $(F.r) \circ \phi \leq^G \psi \circ r$. Together with (1) and (2) this implies $(F.s) \circ \phi \leq^G \psi \circ s$, that is, $s$ is a $G$-left-cohom. $\square$