1. Unions of Theories have the Pushout Property

Suppose T_1 and T_2 have signatures $\Sigma_1 = sig(T_1)$ and $\Sigma_2 = sig(T_2)$ that are nicely interceptable as $\Sigma_0 = \Sigma_1 \cap \Sigma_2$. Let $T_0 = (\Sigma_0, \emptyset)$. We then have,

<u>Theorem</u> (<u>Pushout Property</u>) Let T_i, $0 \leq i \leq 2$ be as above, and let $H_i : T_i \to T$, $1 \leq i \leq 2$ be theory interpretations such that $H_1|_{T_0} = H_2|_{T_0}$, i.e., such that $J_1 ; H_1 = J_2 ; H_2$ in the diagram below. Then $H_1 \cup H_2$ is a theory interpretation $H_1 \cup H_2 : T_1 \cup T_2 \to T$, and is the only theory interpretation such that $H_1 = J'_2 ; (H_1 \cup H_2)$, and $H_2 = J'_1 ; (H_1 \cup H_2)$.

$$\begin{tikzcd}
 & & T \\
T_1 \arrow[urr, "H_1"] \arrow[r, hook, "J'_2"'] & T_1 \cup T_2 \arrow[ur, "H_1 \cup H_2"] & \\
T_0 \arrow[u, hook, "J_1"] \arrow[r, hook, "J_2"'] & T_2 \arrow[u, hook, "J'_1"'] \arrow[uur, "H_2"'] &
\end{tikzcd}$$

<u>Proof</u>. Uniqueness of $H_1 \cup H_2$ is guaranteed by the pushout property for $\Sigma_1 \cup \Sigma_2$ for the signature maps $H_i : \Sigma_i \to sig(T)$, $1 \leq i \leq n$. We just need to show that $H_1 \cup H_2$ <u>is</u> a theory interpretation, that is, that

$$mod(T)|_{H_1 \cup H_2} \subseteq mod(T_1 \cup T_2).$$

But this follows immediately from the commutative diagram:

$$
\begin{tikzcd}
 & & \mathrm{mod}(T) \arrow[dll, "-|_{H_1}"'] \arrow[dl, "-|_{H_1 \cup H_2}"] \arrow[ddr, "-|_{H_2}"] & \\
\mathrm{mod}(T_1) \arrow[dd, "-|_{\Sigma_0}"'] & \mathrm{mod}(T_1 \cup T_2) \arrow[l, "-|_{\Sigma_1}"] \arrow[dd, "-|_{\Sigma_2}"] & & \\
 & & & \\
\mathrm{mod}(T_0) & \mathrm{mod}(T_2) \arrow[l, "|_{\Sigma_0}"] & &
\end{tikzcd}
$$

(where in fact $-|_{H_1}$ and $-|_{H_2}$ are well-defined functions, since
$\mathrm{mod}(T)|_{H_i} \subseteq \mathrm{mod}(T_i)$, $1 \leq i \leq 2$ by H_i theory interpretation
$1 \leq i \leq 2$), thanks to the fact that $\mathrm{mod}(T_1 \cup T_2) = \mathrm{mod}(T_1) \times_{\Sigma_0} \mathrm{mod}(T_2)$
is a pullback in the category of sets. □

Corollary. Let $H_i : T_i \to T'_i$, $1 \leq i \leq 2$ be theory interpretations
such that $\Sigma_i := sig(T_i)$ $1 \leq i \leq 2$ are nicely intersectable as
$\Sigma_0 = \Sigma_1 \cap \Sigma_2$, and $\Sigma'_i = sig(T'_i)$, $1 \leq i \leq 2$ are likewise nicely
interceptable as $\Sigma'_0 = \Sigma'_1 \cap \Sigma_2$, and $H_1|_{\Sigma_0} = H_2|_{\Sigma_0} = H_0$,
so that H_1 and H_2 have a common restriction to a signature
map $H_0 : \Sigma_0 \to \Sigma'_0$. Then $H_1 \cup H_2 : T_1 \cup T_2 \to T'_1 \cup T'_2$ is a
theory interpretation, and the only one commuting the diagram:

$$
\begin{tikzcd}
 & & T'_1 \arrow[r, hook] & T'_1 \cup T'_2 \\
T_1 \arrow[r, hook] \arrow[urr, "H_1"] & T_1 \cup T_2 \arrow[urr, dashed, "H_1 \cup H_2"'] & & \\
 & & T'_0 \arrow[uu, hook] \arrow[r, hook] & T'_2 \arrow[uu, hook] \\
T_0 \arrow[uu, hook] \arrow[r, hook] \arrow[urr, "H_0"] & T_2 \arrow[uu, hook] \arrow[urr, "H_2"'] & &
\end{tikzcd}
$$

2. Trivial and Expansive Ascent Maps are Closed Under Unions

To increase extensibility of SMT solving ascent maps should be composable out of smaller pieces as much as possible. The fact that Ascent is a category already gives us a compositional way of obtaining new ascent maps from simpler ones by sequential composition (we exploited, for example this fact in Lecture 23, pg. 14, last two compositions). Is there, furthermore, some way of composing ascent maps in parallel? The answer is yes! (under some conditions):

Lemma Let $(H_i, \alpha_i) : (T_i, \mathcal{F}_i) \longrightarrow (T_i', \mathcal{G}_i)$ $1 \leq i \leq 2$ be trivial ascent maps such that $H_i : T_i \to T_i$, $1 \leq i \leq 2$ are expansive and satisfy the conditions in the Corollary of the Pushout Property Theorem (pg. 2 above). Suppose that Σ_0 (and also Σ_0') has no operations (only predicates), and $H_0 : \Sigma_0 \to \Sigma_0'$ is surjective. Then, defining $\mathcal{F}_1 \wedge \mathcal{F}_2 = \{\varphi_1 \wedge \varphi_2 \mid \varphi_1 \in \mathcal{F}_1 \wedge \varphi_2 \in \mathcal{F}_2\}$, and likewise $\mathcal{G}_1 \wedge \mathcal{G}_2$,

$$(H_1 \cup H_2, H_1 \cup H_2) : (T_1 \cup T_2, \mathcal{F}_1 \wedge \mathcal{F}_2) \to (T_1' \cup T_2', \mathcal{G}_1 \wedge \mathcal{G}_2)$$

is an ascent map, called the parallel composition of the two original ascent maps.

Proof. $H_1 \cup H_2 : T_1 \cup T_2 \longrightarrow T_1' \cup T_2'$ is well-defined by the Corollary; and $H_1 \cup H_2 : \mathcal{F}_1 \wedge \mathcal{F}_2 \longrightarrow \mathcal{G}_1 \wedge \mathcal{G}_2$ is a well-defined recursive function, since, by H_1 and H_2 agreeing on Σ_0 we have $H_1 \cup H_2 (\varphi_1 \wedge \varphi_2) = H_1(\varphi_1) \wedge H_2(\varphi_2)$. We just need to show that: (i) $H_1 \cup H_2 : T_1 \cup T_2 \longrightarrow T_1' \cup T_2'$ is expansive.

Let $[\mathcal{M}_1, \mathcal{M}_2] \in mod(T_1 \cup T_2)$, so that $\mathcal{M}_1|_{\Sigma_0} = \mathcal{M}_2|_{\Sigma_0}$.

By the H_i expansive, $1 \leq i \leq 2$, we have $\mathcal{M}_i' \in mod(T_i')$,

$1 \leq i \leq 2$ such that $\mathcal{M}_i'|_{H_i} = \mathcal{M}_i$, $1 \leq i \leq 2$.

We will be done if we show that $\mathcal{M}_1'|_{\Sigma_0'} = \mathcal{M}_2'|_{\Sigma_0'}$,

since then $[\mathcal{M}_1', \mathcal{M}_2']|_{H_1 \cup H_2} = [\mathcal{M}_1, \mathcal{M}_2]$. But since

Σ_0' has no operations this amounts to showing that for

each sort s_0' in Σ_0' we have $M'_{1,s_0'} = M'_{2,s_0'}$.

But by the assumption that $H_0 : \Sigma_0 \to \Sigma_0'$ is surjective,

the is a sort s_0 in Σ_0 such that $s_0' = H(s_0)$ and we

have:

$$M'_{1,H(s_0)} = M_{1,s_0} = M_{2,s_0} = M'_{2,H(s_0)}$$

as desired. □

Of course, since we have just shown that $H_1 \cup H_2 : T_1 \cup T_2' \to T_1' \cup T_2'$

is expansive under the conditions in the lema, we have trivial

ascent maps $(H_1 \cup H_2, H_1 \cup H_2) : (T_1 \cup T_2, K) \to (T_1' \cup T_2', K')$

for any $K \subseteq FO(sig(T_1 \cup T_2))$ such that $(H_1 \cup H_2)(K) \subseteq K'$. What

is the point of restricting to $\mathcal{F}_1 \wedge \mathcal{F}_2$? The point is, of course,

that we want to use ascent maps for satisfiability purposes.

Suppose, then, that we have decision procedures for T_i'-satisfiability of $\mathcal{G}_i$ formulas, $1 \leq i \leq 2$. Then, under reasonable

assumption on T_1' and T_2', the Nelson-Oppen combination

procedure (more on this in upcoming lectures) will give us a decision procedure on formulas of the form $H_1(\varphi_1) \wedge H_2(\varphi_2)$ by invokig the decision procedure for T'_1 with $H_1(\varphi_1)$, and that of T'_2 with $H_2(\varphi_2)$.

3. Useful Formula Transformations as Generalized Ascent Maps

Regarding the just-discussed notion of parallel composition $(H_1 \cup H_2, H_1 \cup H_2) : (T_1 \cup T_2, \mathcal{F}_1 \wedge \mathcal{F}_2) \longrightarrow (T'_1 \cup T'_2, \mathcal{G}_1 \wedge \mathcal{G}_2)$ an obvious question to ask is: aren't the sets of formulas $\mathcal{F}_1 \wedge \mathcal{F}_2$ and $\mathcal{G}_1 \wedge \mathcal{G}_2$ too particular (and therefore the parallel composition no so useful)? The answer is that it is <u>not</u>, provided we see it within the broader ecosystem of other ascent maps performing various useful <u>formula transformations</u>. Before discussing several of those transformations, let me generalize the notion of an ascent map (H, α) so that α is generalized from a recursive <u>function</u> to a recursive <u>relation</u>:

<u>Definition</u>. An <u>ascent map</u> $(H, \alpha) : (T, \mathcal{F}) \rightarrow (T', \mathcal{G})$, with $H : T \rightarrow T'$ a theory interpretation and $\mathcal{F} \subseteq FO(sig(T))$, $\mathcal{G} \subseteq FO(sig(T'))$ recursive sets of formulas has a second component α a recursive function

$$\alpha : \mathcal{F} \longrightarrow \mathcal{P}_{fin \neg \emptyset}(\mathcal{G})$$

where $\mathcal{P}_{fin \neg \emptyset}(\mathcal{G})$ denotes the recursive data type of <u>non-empty finite</u> sets of formulas in $\mathcal{G}$. Furthermore, the following formula equi-satisfiability property is satisfied:

$(\forall \varphi \in \mathcal{F})$ φ is T-satisfiable $\Leftrightarrow \exists \psi \in \alpha(\varphi)$ s.t. ψ is T'-satisfiable.

(H, α) is called <u>effective</u> if given φ, $\psi \in \alpha(\varphi)$, $\mathcal{M}' \in mod(T')$, and $\tilde{a} \in [fvars(\psi) \to M']$ s.t. $\mathcal{M}', \tilde{a} \models \psi$ there is an effective construction of a model $\mathcal{M} \in mod(T)$ and $a \in [fvars(\varphi) \to M]$ s.t. $\mathcal{M}, a \models \varphi$.

Note that we recover the more restrictive notion of ascent map in Lecture 23 by viewing a recursive function $\alpha : \mathcal{F} \to \mathcal{G}$ as a recursive function $\hat{\alpha} : \mathcal{F} \ni \varphi \mapsto \{\alpha(\varphi)\} \in \mathcal{P}_{fin\neq\emptyset}(\mathcal{G})$.

Note also that given ascent maps

$$(T, \mathcal{F}) \xrightarrow{(H,\alpha)} (T', \mathcal{G}) \xrightarrow{(G,\beta)} (T'', K)$$

their composition $(H, \alpha); (G, \beta)$ is defined as the ascent map $(H;G, \alpha ; \mathcal{P}_{fin\neq\emptyset}(\beta))$, where

$$\mathcal{P}_{fin\neq\emptyset}(\beta) : \mathcal{P}_{fin\neq\emptyset}(\mathcal{G}) \ni \{\psi_1, \ldots, \psi_n\} \mapsto \beta(\psi_1) \cup \cdots \cup \beta(\psi_n) \in \mathcal{P}_{fin\neq\emptyset}(K).$$

From now on, <u>Ascent</u> will denote the category of ascent maps in the more general sense just defined.

Here are some useful ascent maps, all of which are of the form: $(1_\Sigma, \alpha) : (T, \mathcal{F}) \to (T, \mathcal{G})$. That is T does not change, and 1_Σ is the identity theory inclusion $T \subseteq T$ viewed as the identity signature map 1_Σ, where $\Sigma = sig(T)$. Therefore, all the action in such maps resides in α, which is some kind of <u>satisfiability-preserving</u> <u>formula</u>

transformation. Here are several useful examples very heavily used in SMT solving. For some of them I leave the proof that they are effective ascent maps as an exercise.

1. $(1_\Sigma, dnf) : (T, QF(\Sigma)) \longrightarrow (T, DNF(\Sigma))$, where $\Sigma = sig(T)$, $DNF(\Sigma) \subseteq QF(\Sigma)$ are the formulas in disjunctive normal form, and $dnf(\varphi)$ is "the" disjunctive normal form of φ, which can be determined either by assuming $\wedge$ and $\vee$ associative-commutative or by some total order on formulas.

2. $(1_\Sigma, cnjs) : (T, DNF(\Sigma)) \longrightarrow (T, \wedge LIT(\Sigma))$, where $\wedge Lit(\Sigma)$ are the conjunction of literals of Σ, and $cnjs : DNF(\Sigma) \to \mathcal{P}_{fin \neq \emptyset}(\wedge Lit(\Sigma))$ maps each $\varphi = \bigvee_{1 \le i \le n} C_i$ to $\{C_1, .., C_n\}$.

3. $(1_\Sigma, vabs) : (T, \wedge Lit(\Sigma)) \longrightarrow (T, S\wedge Lit(\Sigma))$, where $S\wedge Lit(\Sigma) \subseteq \wedge Lit(\Sigma)$ are called simple conjunctions, where all literals $u = v$ or $u \neq v$ in the conjunction are such that either u or v are variables, and all $p(u_1, .., u_n)$ or $\neg p(u_1, .., u_n)$ are of the form $p(x_1, .., x_n)$ or $\neg p(x_1, .., x_n)$ with $x_1, .., x_n$ not necessarily distinct variables. The mapping vabs is called variable abstraction and is defined by exhaustive application of the following rewrite rules, rewriting conjunctions where $\_\wedge\_$ is treated as an associative-commutative symbol with T as its identity element, and where the symbols $\_=\_$ and $\_\neq\_$ are considered commutative (so $u = v$ is the same equation as $v = u$), so this is $ACU + C$-rewriting:

(0) $f(x_1,\dots,x_n)=g(y_1,\dots,y_n)\wedge C \longrightarrow x{:}s=f(x_1,\dots,x_n)\wedge x{:}s=g(y_1\dots y_n)\wedge (C)!_{\{f(x_1\dots x_n)\to x:s\}}$

(1) $f(u_1,\dots,u_i,\dots,u_n)=v\wedge C \longrightarrow x{:}s=u_i\wedge \big(f(u_1,\dots,x{:}s,\dots,u_n)=v\wedge C\big)!_{\{u_i\to x:s\}}$ $\quad u_i\notin X$

(2) $f(u_1,\dots,u_i,\dots,u_n)\neq v\wedge C \rightarrow x{:}s=u_i\wedge \big(f(u_1,\dots,x{:}s,\dots,u_n)\neq v\wedge C\big)!_{\{u_i\to x:s\}}$ $\quad u_i\notin X$

(3) $p(u_1,\dots,u_i,\dots,u_n)\wedge C \rightarrow x{:}s=u_i\wedge \big(p(u_1,\dots,x{:}s,\dots,u_n)\wedge C\big)!_{\{u_i\to x:s\}}$ $\quad u_i\notin X$

(4) $\neg p(u_1,\dots,u_i,\dots,u_n)\wedge C \rightarrow x{:}s=u_i\wedge \big(\neg p(u_1,\dots,x{:}s,\dots,u_n)\wedge C\big)!_{\{u_i\to x:s\}}$ $\quad u_i\notin X$

resp. $s=ls(f(x_1,\dots,x_n))$ resp. $f(x_1\dots x_n)$

where $u_i\notin X$ is a non-variable subterm, $s=ls(u_i)$ is the least sort of u_i, $x{:}s$ is a fresh variable not appearing anywhere in the conjunction, and $(C')!_{\{u_i\to x:s\}}$ denotes the canonical form obtained by exhaustively rewriting a conjunction C' by replacing each occurrence of the term u_i by the variable $x{:}s$ (note that this is happening at the metalevel, so that $u_i\to x{:}s$ is a ground rewrite rule). The process $(C')!_{\{u_i\to x:s\}}$ is not strictly required for correctness, but can greatly improve efficiency by yielding formulas with considerably fewer fresh variables.

That this is an effective ascent map can be shown by showing that:

(a) each step of rewriting (before the $!_{\{u_i\to x:s\}}$ optimization) with rules (0)-(4), and

(b) each step of rewriting with the ground rewrite rule $u_i\to x{:}s$ the righthand sides of rules (0)-(4)

preserves T-satisfiability and is effective. To illustrate the general idea, let us focus on the rule

$$f(u_1,\dots,u_i,\dots,u_n)\neq v\wedge C \longrightarrow x{:}s\wedge f(u_1,\dots,x{:}s,\dots u_n)\neq v\wedge C$$

Let $Y = vars(f(u_1,..,u_i,..,u_n) \neq v \wedge C)$, and let $\mathcal{M} = (M, -_{\mathcal{M}}) \in mod(T)$,
and $a \in [Y \to M]$ be such that $\mathcal{M}, a \models f(u_1,..,u_i,..,u_n) \neq v \wedge C$,
i.e., $\mathcal{M}, a \models C$ and $f_{\mathcal{M}}(u_1 a,..,u_i a,..,u_n a) \neq va$.
let $\tilde{a} = a \cup \{(x:s, u_i a)\}$, then ⊛ $\mathcal{M}, \tilde{a} \models x:s = u_i \wedge f(u_1,..x:s,..u_n)$.
Conversely, if for some $\tilde{a} \in [Y \cup \{x:s\}]$ ⊛ holds,
then defining $a = \tilde{a} \setminus \{(x:s, \tilde{a}(x:s))\}$ we get
$\mathcal{M}, a \models f(u_1,..,u_i,..,u_n) = v$, and, of course, $\mathcal{M}, \tilde{a} \models C$
iff $\mathcal{M}, a \models C$.

4. Let $sig(T) = \Sigma_1 \cup \Sigma_2$, with Σ_1, Σ_2 nicely intersectable and $\Sigma_0 = \Sigma_1 \cap \Sigma_2$ consisting only of sorts (no shared function or predicate symbols). Let
$(1_{\Sigma_1 \cup \Sigma_2}, pur) : (T, QF(\Sigma_1 \cup \Sigma_2)) \to (T, \bigwedge Lit(\Sigma_1) \wedge \bigwedge Lit(\Sigma_2))$
where $\bigwedge Lit(\Sigma_1) \wedge \bigwedge Lit(\Sigma_2) = \{ C_1 \wedge C_2 \mid C_i \in \bigwedge Lit(\Sigma_i), 1 \leq i \leq 2 \}$,
and pur is the formula purification transformation defined by first applying exhaustively the rules:

(1) $u[v]_p = w \wedge C \to (u[x:s] = w \wedge C)!_{\{v \to x:s\}}$

(2) $u[v]_p \neq w \wedge C \to (x:s = v \wedge u[x:s] \neq w \wedge C)!_{\{v \to x:s\}}$

(3) $p(u_1,..,u_i,..u_n) \wedge C \to (x:s = u_i \wedge p(u_1,..,x:s,..,u_n) \wedge C)!_{\{u_i \to x:s\}}$ if $u_i \notin X$

(4) $\neg p(u_1..,u_i,..,u_n) \wedge C \to (x:s = u_i \wedge \neg p(u_1,..,x:s,..,u_n) \wedge C)!_{\{u_i \to x:s\}}$ if $u_i \notin X$

where in rules (1)-(2) $p = q \cdot i \in pos_{\Sigma}(u)$ is such that for each decomposition $q = q_1 \cdot q_2$, including $q_1 = \varepsilon$ and $q_2 = q$, $top(u|_{q_1}) \in fun(\Sigma_1)$ (resp. $fun(\Sigma_2)$), and $top(u|_p) \in fun(\Sigma_2)$ (resp. $fun(\Sigma_1)$), where $top(t)$ returns the top function symbol of a non-variable term, and where if $[ls(u|_p)]_1 = [ls(u|_p)]_2$ the sort s of the <u>fresh</u> variable $x:s$ is just $ls(u|_p)$, and otherwise in the unique sort s_0 such that $[s_1]_1 \cap [ls(u|_p)]_2 = \{s_0\}$ for some $[s_1]_1 \in S_1/\equiv_{\leq_1}$ (resp. $[s_2]_2 \cap [ls(u|_p)]_1 = \{s_0\}$ for some $[s_2]_2 \in S_2/\equiv_{\leq_2}$); and for rules (3)-(4), if $top(u_i) \in fun(\Sigma_j)$ $1 \leq j \leq 2$ and $p : s_1, .., s_i, .. s_n$ in Σ_j then $s = ls(u_i)$ in the <u>fresh</u> variable $x:s$. Otherwise, if $top(u_i) \in fun(\Sigma_{j'})$ with $\{j, j'\} = \{1, 2\}$, if $[s_i]_1 = [s_i]_2$ we choose again $s = ls(u_i)$ in the fresh variable $x : s$. Finally, if $\{s_0\} = [s_i]_j \cap [s_{j'}]_{j'}$ with $\{j, j'\} = \{1, 2\}$, then we choose $s = s_0$ in the fresh variable $x : s$.

After a canonical form has been reached with rules (1)-(4) we perform a second process of rewriting to canonical form with the rule:

$$(5) \quad u = v \wedge C \longrightarrow x:s = u \wedge x:s = v \wedge (C) \big|_{\{u \to x:s \text{ if } s = ls(u)\}} \quad \text{if } u \in T_{\Sigma_1}(X) - X \wedge v \in T_{\Sigma_2}(X) - X$$

where is (5) the sort s of the fresh variable $x{:}s$ is chosen as follows:

(i) $[l_s(u)]_1 \neq [l_s(v)]_2$ we choose $s = l_s(u)$,

and (ii) otherwise, $\{s_0\} = [l_s(u)]_1 \cap [l_s(v)]_2$

and we chose $s = s_0$. Note that (C)! is conditional on $s = l_s(u)$.

The proof that $(1_\Sigma, pur)$ is an effective ascent map is quite similar to that for $(1_\Sigma, vabs)$ and is left as an exercise.

We can now come back to the question of whether the definition of $(H_1 \cup H_2, H_2 \cup H_2) : (T_1 \cup T_2, \mathcal{F}_1 \wedge \mathcal{F}_2) \to (T'_1 \cup T'_2, \mathcal{G}_1 \wedge \mathcal{G}_2)$ was or not too restrictive. It is not in the following ecosystem of ascent maps:

$$(T_1 \cup T_2, QF(\Sigma_1 \cup \Sigma_2)) \xrightarrow{(1_{\Sigma_1 \cup \Sigma_2}, dnf)} (T_1 \cup T_2, DNF(\Sigma_1 \cup \Sigma_2)) \xrightarrow{(1_{\Sigma_1 \cup \Sigma_2}, conjs)} (T_1 \cup T_2, \bigwedge Lit(\Sigma_1 \cup \Sigma_2))$$

$$\xrightarrow{(1_{\Sigma_1 \cup \Sigma_2}, pur)} (T_1 \cup T_2, \bigwedge Lit(\Sigma_1) \wedge Lit(\Sigma_2)) \xrightarrow{(H_1 \cup H_2, H_1 \cup H_2)} (T'_1 \cup T'_2, \bigwedge Lit(\Sigma'_1) \wedge \bigwedge Lit(\Sigma'_2))$$

where, recall that H_1, H_2 must be expansive for the whole composition to be an effective ascent map.

4. Descent Maps

Intuitively, an ascent map $(H, \alpha) : (T, \mathcal{F}) \to (T', \mathcal{G})$ is used to reduce the T-satisfiability of $\mathcal{F}$-formulas to that of the T'-satisfiability of $\mathcal{G}$-formulas, for which we hopefully have (or can get by further composition with additional ascent maps) a decision procedure. We do so by "climbing up" using $H : T \to T'$ from an, in principle simpler, theory T to an, in principle, more complex theory T'. I say, "in principle", because in the case of $(u,u) : ((\Sigma,\emptyset)\, QF(\Sigma)) \to ((\Sigma^u,\emptyset)\, QF(\Sigma^u))$ used to reduce order-sorted congruence closure for $(\Sigma,\emptyset)$ (or (Σ, AC_Δ)) to unsorted congruence closure for $(\Sigma^u,\emptyset)$ (or $(\Sigma^u, AC_{\Delta^u})$) the theory $(\Sigma^u,\emptyset)$ is actually simpler!

But there are many other cases (we have seen many in the <u>VarSat</u> paper) where what we want to do is not to climb up from T to T' but to <u>descend</u> from T' to T, for example when $T' \supseteq T$ is an extension. Therefore, there is also a very useful notion of <u>descent map</u> as follows:

<u>Definition</u> Let $H : T \to T'$ be a theory interpretation, and let $\mathcal{F} \subseteq FO(sig(T))$, $\mathcal{G} \subseteq FO(sig(T'))$ be recursive sets of formulas. A <u>descent map</u> from $(T', \mathcal{G})$ to $(T, \mathcal{F})$ is then a pair (H, δ), written $(H,\delta) : (T', \mathcal{G}) \to (T, \mathcal{F})$ where δ is a recursive function $\delta : \mathcal{F} \to \mathcal{P}_{fin \neq \emptyset}(\mathcal{G})$ such that for each $\varphi \in \mathcal{G}$, φ is T'-satisfiable iff ψ is T-satisfiable for some $\psi \in \delta(\varphi)$. (H,δ) is called effective if given $\varphi \in \mathcal{G}$, $\psi \in \delta(\varphi)$, $\mathcal{M} \in mod(T)$, and $a \in [fvars(\psi) \to M]$ s.t. $\mathcal{M}, a \models \psi$ there is an effective construction of an $\mathcal{M}' \in mod(T')$, $\tilde{a} \in [fvars(\varphi) \to M']$ s.t. $\mathcal{M}', \tilde{a} \models \varphi$.