# AC-Unification of Higher-order Patterns

Alexandre Boudet
Evelyne Contejean

LRI Orsay France

## Motivations

$*$ Higher-order unification is undecidable
$*$ Unification of higher-order patterns is decidable

$*$ Combination of algebraic and functional programming paradigms
$*$ Local confluence of HRSs

$\Downarrow$

Unification of higher-order patterns
modulo equational theories

## Patterns

**Definition** A pattern is
* a term of the simply-typed $\lambda$-calculus in $\beta$-normal form
* in which the arguments of a free variable are $\eta$-equivalent to distinct bound variables.

$$\left.\begin{array}{r}\lambda xyz.f(H(x,y),H(x,z))\\ \lambda x.F(\lambda z.x(z)) =_\eta \lambda x.F(x)\end{array}\right\}\text{ are patterns}$$

$$\left.\begin{array}{r}\lambda xy.G(x,x,y)\\ \lambda xy.H(F(x),y)\end{array}\right\}\text{ are not patterns}$$

No equational theory, but $\alpha, \beta, \eta$.

**Theorem (Miller)** The unifiability of patterns is decidable and if two patterns are unifiable, there is an algorithm computing a unique most general unifier.

**Notation** $\lambda\overline{x}.F(\overline{x}^\pi)$ denotes $\lambda x_1 \ldots \lambda x_n.F(x_{\pi(1)}, \ldots, x_{\pi(n)})$, where $\pi$ is a permutation over $\{1, \ldots, n\}$.

## AC-unification of Patterns

**Definition** Let $E = \{l_1 \simeq r_1, \ldots, l_n \simeq r_n\}$ a set of *axioms* such that $l_i$ and $r_i$ of the same type, for $1 \leq i \leq n$.
The equational theory $=_E$ is
the least congruence (compatible also with application and abstraction) containing all the instances of the axioms of $E$.

**Definition** An equation $s = t$ is a pair of patterns of the same type.

A unification problem is $\begin{cases} - & \top \\ - & \bot \\ - & P \equiv s_1 = t_1 \ \wedge \ \cdots \ \wedge \ s_n = t_n. \end{cases}$

A substitution $\sigma$ is an $E$-unifier of $P$ if $\forall i$, $s_i =_{\beta\eta E} t_i$.

**Theorem (Tannen)** $\forall\, u, v \quad u =_{\beta\eta E} v \Longleftrightarrow u \updownarrow_\beta^\eta =_E v \updownarrow_\beta^\eta$.

We consider the case where $E$ is

$$\bigcup_{+\in \mathrm{AC}} \{x + y = y + x, x + (y + z) = (x + y) + z\}$$

The type of $+$ is $\alpha \rightarrow \alpha \rightarrow \alpha$, where $\alpha$ is a base type.

## Alien subterms and Purification

**Definition** $u = t|_{p \cdot i}$ is an alien subterm of $t$
$*$ if $t(p) \in$ AC and Head$(u) \notin \mathcal{FV}$,
$*$ or $t(p) \in \mathcal{FC}$ and Head$(u) \in$ AC.

**VA**
$\lambda\overline{x}.t[u]_p = \lambda\overline{x}.s \quad \rightarrow \; \lambda\overline{x}.t[H(\overline{y})]_p = \lambda\overline{x}.s \; \wedge \; \lambda\overline{y}.H(\overline{y}) = \lambda\overline{y}.u$
if $u$ is an alien subterm of $t[u]_p$ at position $p$, and $\overline{y} = \mathcal{FV}(u) \cap \overline{x}$,
where $H$ is a new variable.

A unification problem in normal form wrt **VA** has the form

$$P \equiv P_F \; \wedge \; P_0 \; \wedge \; P_1 \; \wedge \; \cdots \; \wedge \; P_n$$

$*$ $P_0$ is pure in the free theory, with no $\lambda\overline{x}.F(\overline{x}) = \lambda\overline{x}.F(\overline{x}^{\pi})$.
$*$ $P_F$ contains all the Flex-Flex equations $\lambda\overline{x}.F(\overline{x}) = \lambda\overline{x}.F(\overline{x}^{\pi})$.
$*$ $P_i$ is a pure unification problem in the $AC$ theory of $+_i$, the arguments of $+_i$ being of the form $F(\overline{x})$ where $F$ is a free variable.

## Why Frozen Equations ?

**Example (Qian & Wang)**

$$\lambda xy \cdot F(x,y) = \lambda xy \cdot F(y,x)$$

has the solutions

$$\sigma_n = \{F \mapsto \lambda xy \cdot G(H_1(x,y){+}H_1(y,x), \ldots, H_n(x,y){+}H_n(y,x))\}$$

for all $n \in \mathbb{N}$.

In addition $\sigma_{n+1}$ is strictly more general than $\sigma_n$ (nullary theory).

## Free: a subset of Nipkow's rules for pattern unification

**Dec-free**

$\lambda\overline{x}.a(s_1,\ldots,s_n) = \lambda\overline{x}.a(t_1,\ldots,t_n) \ \wedge\ P_0 \quad\rightarrow$

$\lambda\overline{x}.s_1 = \lambda\overline{x}.t_1 \ \wedge\ \cdots \ \wedge\ \lambda\overline{x}.s_n = \lambda\overline{x}.t_n \ \wedge\ P_0$

if $a$ is a free constant symbol or a bound variable of $\overline{x}$.

**FR-free**

$\lambda\overline{x}.F(\overline{y}) = \lambda\overline{x}.a(s_1,\ldots,s_m) \ \wedge\ P_0 \quad\rightarrow$

$\lambda\overline{y}.H_1(\overline{y}) = \lambda\overline{y}.s_1 \ \wedge\ \cdots \ \wedge\ \lambda\overline{y}.H_m(\overline{y}) = \lambda\overline{y}.s_m$

$\wedge\ P_0\{F \mapsto \lambda\overline{x}.a(H_1(\overline{y}),\ldots,H_m(\overline{y}))\}$

$\wedge\ F = \lambda\overline{x}.a(H_1(\overline{y}),\ldots,H_m(\overline{y}))\}$

If $F$ is a free variable, $a$ a free constant and $F \notin \mathcal{FV}(s_i)$ for $1 \leq i \leq m$, where $H_1,\ldots,H_m$ are new variables.

**FF$\neq$**

$\lambda\overline{x}.F(\overline{y}) = \lambda\overline{x}.G(\overline{z}) \ \wedge\ P_0 \quad\rightarrow$

$P_0\{F \mapsto \lambda\overline{y}.H(\overline{v}), G \mapsto \lambda\overline{z}.H(\overline{v})\}$

$\wedge\ F = \lambda\overline{y}.H(\overline{v}) \ \wedge\ G = \lambda\overline{z}.H(\overline{v})$

if $F$ and $G$ are different free variables where $\overline{v} = \overline{y} \cap \overline{z}$.

**Fail1**

$\lambda\overline{x}.a(\overline{s}) = \lambda\overline{x}.b(\overline{t}) \quad\rightarrow\ \bot$

if $a$ and $b$ are constants or bound variables and $a \neq b$.

**Fail2**

$\lambda\overline{x}.F(\overline{y}) = \lambda\overline{x}.a(\overline{s}) \quad\rightarrow\ \bot$

if $F$ is free in $\overline{s}$ or $a \in \overline{x} \setminus \overline{y}$.

**Freeze**

$P_F \wedge (\lambda\overline{x}.F(\overline{x}) = \lambda\overline{x}.F(\overline{x}^\pi) \wedge P_0) \wedge P_1 \wedge \cdots \wedge P_n \quad \to$
$(\lambda\overline{x}.F(\overline{x}) = \lambda\overline{x}.F(\overline{x}^\pi) \wedge P_F) \wedge P_0 \wedge P_1 \wedge \cdots \wedge P_n$
if $F$ is a free variable.

## Normal Forms

A unification problem (with no alien subterms) in normal form wrt (**Free** + **Freeze**) has the form

$$P_F \wedge P_0 \wedge P_1 \wedge \cdots \wedge P_n$$

where
* $P_F$ contains only $\lambda\overline{x}.F(\overline{x}) = \lambda\overline{x}.F(\overline{x}^\pi)$,
* $P_0$ is solved,
* $P_i$ is a pure problem in the $AC$-theory of $+_i$.

## Pure AC Patterns: An Example

### From AC Problems to Diophantine Problems

$$\lambda xyz.2F(x,y,z) + F(y,z,x) = \lambda xyz.2G(x,y,z)$$

Let $\sigma$ be a solution of the above equation:

$*$ $\sigma$ may introduce $t(x,y,z)$ such that $t(x,y,z) = t(y,z,x)$.

| | $\lambda xyz.2F(x,y,z)$ | $+F(y,z,x)$ | $=$ | $\lambda xyz.2G(x,y,z)$ |
|---|---|---|---|---|
| # of $t$ | $2\alpha$ | $+\alpha$ | $=$ | $2\beta$ |

$(\alpha, \beta)$ is a positive solution of $3n = 2m$.

$*$ $\sigma$ may introduce $t(x,y,z)$ such that $t(x,y,z) \neq t(y,z,x)$.

| | $\lambda xyz.2F(x,y,z)$ | $+F(y,z,x)$ | $=$ | $\lambda xyz.2G(x,y,z)$ |
|---|---|---|---|---|
| # of $t(x,y,z)$ | $2\alpha_1$ | $+\alpha_3$ | $=$ | $2\beta_1$ |
| # of $t(y,z,x)$ | $2\alpha_2$ | $+\alpha_1$ | $=$ | $2\beta_2$ |
| # of $t(z,x,y)$ | $2\alpha_3$ | $+\alpha_2$ | $=$ | $2\beta_3$ |

$(\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3)$ are positive solutions of the system

$$\begin{aligned} 2n_1 + n_3 &= 2m_1 \\ 2n_2 + n_1 &= 2m_2 \\ 2n_3 + n_2 &= 2m_3 \end{aligned}$$

## Pure AC Patterns: An Example (Continued)

### Recombination + Constraints

The set of minimal solutions of $3n = 2m$ is $\{(2,3)\}$
The set of minimal solutions of

$$\begin{aligned} 2n_1 + n_3 &= 2m_1 \\ 2n_2 + n_1 &= 2m_2 \\ 2n_3 + n_2 &= 2m_3 \end{aligned}$$

is $\{(2,0,0,2,1,0), (0,2,0,0,2,1), (0,0,2,1,0,2)\}$.

| | F(x,y,z) | | | G(x,y,z) | | |
|---|---|---|---|---|---|---|
| $L_1$ | 2 | | | 3 | | |
| $L_2$ | 2 | 0 | 0 | 2 | 1 | 0 |
| $L_3$ | 0 | 2 | 0 | 0 | 2 | 1 |
| $L_4$ | 0 | 0 | 2 | 1 | 0 | 2 |
| | (x,y,z) | (y,z,x) | (z,x,y) | (x,y,z) | (y,z,x) | (z,x,y) |

## Pure AC Patterns: An Example (En

A solution of the original problem is

$$\{ \quad F \mapsto \lambda xyz. \quad 2L_1(x,y,z)\theta + 2L_2(x,y,z) + 2L_3(y,z,x) + 2L$$
$$G \mapsto \lambda xyz. \quad 3L_1(x,y,z)\theta + 2L_2(x,y,z) + L_2(y,z,x) + 2L$$
$$L_4(x,y,z) + 2L_4(z,x,y) \quad \}$$

provided that $L_1(x,y,z)\theta = L_1(y,z,x)\theta = L_1(z,x,y)\theta$.

## Pure AC Patterns: Another Example

$$\lambda xyz.G(x,y,z) = \lambda xyz.G(y,z,x) \wedge$$
$$\lambda xyz.2F(x,y,z) + F(y,z,x) = \lambda xyz.2G(x,y,z)$$

Let $\sigma$ be a solution of the above problem:

$\sigma$ may introduce $t(x,y,z)$ such that $t(x,y,z) \neq t(y,z,x)$.

| | | | | |
|---|---|---|---|---|
| | $\lambda xyz.2F(x,y,z)$ | $+F(y,z,x)$ | $=$ | $\lambda xyz.2G(x,y,z)$ |
| # of $t(x,y,z)$ | $2\alpha_1$ | $+\alpha_3$ | $=$ | $2\beta_1$ |
| # of $t(y,z,x)$ | $2\alpha_2$ | $+\alpha_1$ | $=$ | $2\beta_2$ |
| # of $t(z,x,y)$ | $2\alpha_3$ | $+\alpha_2$ | $=$ | $2\beta_3$ |
| | $\lambda xyz.G(x,y,z)$ | | $=$ | $\lambda xyz.G(x,y,z)$ |
| # of $t(x,y,z)$ | $\beta_1$ | | $=$ | $\beta_3$ |
| # of $t(y,z,x)$ | $\beta_2$ | | $=$ | $\beta_1$ |
| # of $t(z,x,y)$ | $\beta_3$ | | $=$ | $\beta_2$ |

$(\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3)$ are positive solutions of the system

$$\begin{aligned} 2n_1 + n_3 &= 2m_1 \\ 2n_2 + n_1 &= 2m_2 \\ 2n_3 + n_2 &= 2m_3 \\ m_1 &= m_3 \\ m_2 &= m_1 \\ m_3 &= m_2 \end{aligned}$$

## Pure AC Patterns: Another Example (End)

The set of minimal solutions of $3n = 2m$ is $\{(2,3)\}$
The set of minimal solutions of $P_{Dioph}$ is $\{(2,2,2,3,3,3)\}$.

$$P_{Dioph} \equiv \begin{cases} 2n_1 + n_3 & = & 2m_1 \\ 2n_2 + n_1 & = & 2m_2 \\ 2n_3 + n_2 & = & 2m_3 \\ m_1 & = & m_3 \\ m_2 & = & m_1 \\ m_3 & = & m_2 \end{cases}$$

| | F(x,y,z) | | | G(x,y,z) | | |
|---|---|---|---|---|---|---|
| $L_1$ | 2 | | | 3 | | |
| $L_2$ | 2 | 2 | 2 | 3 | 3 | 3 |
| | (x,y,z) | (y,z,x) | (z,x,y) | (x,y,z) | (y,z,x) | (z,x,y) |

A solution of the original problem is

$$\{F \mapsto \lambda xyz.2L_1(x,y,z)\theta + 2L_2(x,y,z) + 2L_2(y,z,x) + 2L_2(z,x,y),$$
$$G \mapsto \lambda xyz.3L_1(x,y,z)\theta + 3L_2(x,y,z) + 3L_2(y,z,x) + 3L_2(z,x,y)\}$$

provided that $L_1(x,y,z)\theta = L_1(y,z,x)\theta = L_1(z,x,y)\theta$.

## Pure AC Patterns: General Case

$$E \equiv \lambda\overline{x} \cdot \sum_{i=1}^{n_1} \sum_{\pi\in\Pi} a_{i,\pi} F_i(\overline{x}^{\pi}) = \lambda\overline{x} \cdot \sum_{i=n_1+1}^{n_2} \sum_{\pi\in\Pi} a_{i,\pi} F_i(\overline{x}^{\pi})$$

$\Pi$ is the group of permutations over all the variables of $\overline{x}$.

$$E^{\Pi_0}_{Dioph} \equiv \begin{array}{c} \bigwedge_{i=1}^{n_2} \bigwedge_{(\pi_1,\pi_2)|\Pi_0\circ\pi_1=\Pi_0\circ\pi_2} y_{i,\pi_1} = y_{i,\pi_2} \wedge \\ \bigwedge_{\pi'\in\Pi} \sum_{i=1}^{n_1} \sum_{\pi\in\Pi} a_{i,\pi^{-1}\circ\pi'} y_{i,\pi} = \sum_{i=n_1+1}^{n_2} \sum_{\pi\in\Pi} a_{i,\pi^{-1}\circ\pi'} y_{i,\pi} \end{array}$$

$E_{Dioph}$ is the disjunction $\bigvee_{\Pi_0 \text{ subgroup of } \Pi} E^{\Pi_0}_{Dioph}$.

**Proposition** Let $\mathcal{P}$ be any subset of the minimal solutions of $P_{Dioph}$, which is great enough. Then $\sigma_{\mathcal{P}}$ is a solution of $E$:

$$\sigma_{\mathcal{P}} = \{F_i \mapsto \lambda\overline{x}. \sum_{\Pi_0} \sum_{m\in\mathcal{P}_{\Pi_0}} \sum_{\pi'\in\Pi/\Pi_0} m(i,\pi') L_m(\overline{x}^{\pi'})\}$$

where $L_m, m \in \mathcal{P}_{\Pi_0}$ is a new variable constrained by $\forall\pi \in \Pi_0 \;\; L_m(\overline{x}^{\pi}) = L_m(\overline{x})$.

**Proposition** $\{\sigma_{\mathcal{P}} \mid \mathcal{P} \text{ is great enough}\}$ is a complete set of solutions for $P$.

## Recombination (without the frozen part)

**Solve**
$P_i \quad \rightarrow \; P_i'$
if $P_i$ is not solved and $P_i'$ is a solved form of $P_i$.

**Variable-Replacement**
$F = \lambda\overline{x}.G(\overline{y}) \;\wedge\; P \quad \rightarrow \; F = \lambda\overline{x}.G(\overline{y}) \;\wedge\; P\{F \mapsto \lambda\overline{x}.G(\overline{y})\}$
if both $F$ and $G$ have a free occurrence in $P$.

**Clash**
$F = s \;\wedge\; F = t \quad \rightarrow \; \bot$
if $s$ and $t$ have different constant heads.

**Cycle**
$F_1 = t_1[F_2] \;\wedge\; F_2 = t_2[F_3] \cdots \;\wedge\; F_n = t_n[F_1] \quad \rightarrow \; \bot$
if there is a constant on the path between the head and $F_{i+1(mod\ n)}$ in some $t_i$.

## Recombination (frozen part)

**Compatibility**

$\lambda\overline{x}.F(\overline{x}) = \lambda\overline{x}.F(\overline{x}^{\pi}) \ \wedge\ F = \lambda\overline{x}.u(\overline{x}) \quad \rightarrow\ F = \lambda\overline{x}.u(\overline{x})$

if $\lambda\overline{x}.u(\overline{x}) =_{\eta\beta AC} \lambda\overline{x}.u(\overline{x}^{\pi})$.

**Incompatibility**

$\lambda\overline{x}.F(\overline{x}) = \lambda\overline{x}.F(\overline{x}^{\pi}) \ \wedge\ F = \lambda\overline{x}.u(\overline{x}) \quad \rightarrow\ \perp$

if $\lambda\overline{x}.u(\overline{x})$ is ground and $\lambda\overline{x}.u(\overline{x}) \neq_{\eta\beta AC} \lambda\overline{x}.u(\overline{x}^{\pi})$.

**Propagate**

$\lambda\overline{x}.F(\overline{x}) = \lambda\overline{x}.F(\overline{x}^{\pi}) \ \wedge\ F = \lambda\overline{x}.\gamma(t_1(\overline{x}), \ldots, t_n(\overline{x})) \quad \rightarrow$

$\lambda\overline{x}.t_1(\overline{x}) = \lambda\overline{x}.t_1(\overline{x}^{\pi}) \ \wedge\ \cdots\ \wedge\ \lambda\overline{x}.t_n(\overline{x}) = \lambda\overline{x}.t_n(\overline{x}^{\pi})$

if $\gamma$ is not an $AC$ constant.

**Merge**

$\lambda\overline{x}.F(\overline{x}) = \lambda\overline{x}.F(\overline{x}^{\pi}) \ \wedge\ F = \lambda(\overline{x}).t_1(\overline{x}) + \cdots + t_n(\overline{x}) \quad \rightarrow$

$\lambda(\overline{x}).t_1(\overline{x}) + \cdots + t_n(\overline{x}) = \lambda(\overline{x}).t_1(\overline{x}^{\pi}) + \cdots + t_n(\overline{x}^{\pi})$

If $+$ is an $AC$ constant.

**Result**

**Theorem** The following algorithm terminates and computes a DAG solved form for for unification of higher-order patterns modulo $AC$.

1. Apply as long as possible **VA**,
2. as long as possible do
   (a) apply as long as possible the rules for recombination without frozen equations
   (b) apply the rules for compatibility of frozen equations until a DAG solved form is obtained, or some pure subproblem is made unsolved by **Merge**.

## Conclusion and Further Works

We have a unification algorithm for higher-order patterns modulo $AC$.

### Ongoing Work

Local confluence test for higher-order rewriting modulo $AC$ (à la Jouannaud-Kirchner).

### Future Work

Extensions to $ACU$, $ACI$, $AG$, $BR$, etc.