Interplay between Domain Mu-Calculus and Formal Languages

Guo-Qiang Zhang

Department of EECS
Case Western Reserve University

Summer Topology Conference, DC, July 9-12, 2003
Domain logic

- Idea:
  - types as topological spaces/domains
  - properties as open sets
  - programs as points/continuous functions

- Goal: to provide a systematic way to generate program logics from denotational semantics.
Extending expressive power: the domain mu-calculus is a fixed-point extension of propositional domain logic (Abramsky 89, 91, Z. 89, 91), in a similar sense that the modal mu-calculus is a fixed-point extension of propositional modal mu-calculus (Kozen and others).
Interplay of mu-formulas and languages

- If a certain class of languages can be encoded as mu-formulas, and the language class is undecidable, then that mu-calculus is undecidable.

- If, on the other hand, each mu-formula of a given type can be encoded as a language, and the language class is decidable, then the mu-calculus is decidable.
Set up: types and formulas

- **Types**
  \[ \sigma ::= 1 \mid \sigma \otimes \tau \mid \sigma \oplus \tau \mid \sigma_\bot \mid t \mid \text{rec } t.\sigma \]
  (additional types can be added)

- **Formulas**
  \[ t, f, x_0, x_1, \ldots \] are formulas of any type \( \sigma \). Use superscripts to make the type information explicit.

  If \( \varphi \) is a formula of type \( \sigma \) and \( \psi \) a formula of type \( \tau \), then \( \varphi_\uparrow \) is a formula of type \( \sigma_\bot \), \( \varphi \cdot \psi \) is a formula of type \( \sigma \otimes \tau \), and \( \text{inl } \varphi, \text{inr } \psi \) are formulas of type \( \sigma \oplus \tau \).

  If \( \varphi \) is a formula of type \( \sigma[\text{rec } t.\sigma/t] \), then \( \varphi \) is a formula of \( \text{rec } t.\sigma \).

  If \( \varphi, \psi \) are formulas of \( \sigma \), then \( \varphi \land \psi, \varphi \lor \psi \), and \( \mu x^\sigma.\varphi \) are formulas of \( \sigma \).
Interpretation of types

Each closed type $\sigma$ is interpreted as a Scott domain, with 1 one-point cpo, $\otimes$ smash product, $\oplus$ coalesced sum, $(\ )\bot$ lifting, and $\text{rect} \cdot \sigma$ recursively defined domain.

- The coalesced sum $D_1 \oplus D_2$, has the bottom $\bot_{D_1 \oplus D_2}$ and tagged elements of the form $< x_i, i >$ such that $x_i$ belongs to $(D_i \setminus \{ \bot_{D_i} \})$ for $i = 1, 2$. Elements with the same tag inherit the ordering of their components, while elements with distinct tags are incomparable.

- The smash product $D_1 \otimes D_2$, consists of elements in $(D_1 \setminus \{ \bot_{D_1} \}) \times (D_2 \setminus \{ \bot_{D_2} \})$, ordered coordinatewise, together with the bottom element $(\bot_{D_1}, \bot_{D_2})$. The smash product is the same as the standard cartesian product, except that $(a, \bot_{D_2}) = (\bot_{D_1}, b) = \bot_{D_1 \otimes D_2}$. 
Interpretation of formulas: intra-type

$[\varphi]^{\sigma} : \mathcal{L}_{\sigma} \rightarrow [\mathcal{E} \rightarrow \Omega(D_{\sigma})]$ is defined by structural induction, where $\mathcal{E}$ is the set of environments fixing the interpretation of free variables.

- $[[\varphi_1]^{\sigma} \downarrow \rho$ is $[[\varphi]^{\sigma} \rho$ residing in $D_{\sigma \downarrow}$.
- $[[\varphi \cdot \psi]^{\sigma} \otimes \tau \rho$ is $([[[\varphi]^{\sigma} \rho] \times ([[\psi]^{\tau} \rho])$. If either $(a, \perp)$ or $(\perp, b)$ is a member of $([[[\varphi]^{\sigma} \rho] \times ([[\psi]^{\tau} \rho)$, then it is $D_{\sigma \otimes \tau}$.

When $\perp_{D_{\sigma}}$ is not a member of $[[\varphi]^{\sigma} \rho$, $[[\text{inl } \varphi]^{\sigma} \oplus \tau \rho$ is defined to be the set $[[\varphi]^{\sigma} \rho$ residing in the “left part” of $D_{\sigma \oplus \tau}$. Otherwise, $[[\text{inl } \varphi]^{\sigma} \oplus \tau \rho$ is defined to be the whole space $D_{\sigma \oplus \tau}$. $[[\text{inr } \varphi]^{\sigma} \oplus \tau \rho$ is defined similarly.

- $[[\varphi]^{\text{rec } t. \sigma} \rho$ is defined to be the set
  \[\{\epsilon_{\sigma}(u) \mid u \in [[\varphi]^{\sigma}[(\text{rec } t. \sigma) \setminus t] \rho\}\]
Example

\[ N_\bot := \text{rec } t.(1_\bot \oplus t) \). Formulas:

- \text{inl } t\uparrow \text{ denotes } \{0\} \text{ (abbreviated as 0)}
- \text{inr inr 0} \text{ denotes } \{2\}, \text{ and}
- \mu x. (0 \lor \text{inr inr } x) \text{ denotes the set of even numbers.}

To see the first item above, note that since \( t\uparrow \) is a formula of type \( 1_\bot \), \text{inl } t\uparrow \text{ is a formula of type } 1_\bot \oplus N_\bot. \text{ The isomorphism } \epsilon_{N_\bot} : (1_\bot \oplus N_\bot) \rightarrow N_\bot \text{ sends } \bot \text{ to } \bot, \text{ the top of } 1_\bot \text{ to 0, and } n \text{ to the successor of } n \text{ in general.} \]
Interpretation of formulas: inner-type

- \([t]^{\sigma} \rho = D_{\sigma}\), \([f]^{\sigma} \rho = \emptyset\), and \([x]^{\sigma} \rho = \rho(x)\).
- \(\land\) is interpreted as the intersection and \(\lor\) as the union.
- \([\mu x. \varphi(x)]^{\sigma} \rho\) is the least fixed point of the operator \(\Phi\) induced by \(\varphi\) on the complete lattice of Scott open sets over \(D_{\sigma}\), where \(\Phi(X) =_{\text{def}} [\varphi]^{\sigma} \rho[x \mapsto X]\).

Use Knaster-Tarski Fixed-Point Theorem for the last item.
Intuition: syntactic unwinding

- For $\mu x.(0 \lor \text{inr}^2 x)$, unwinding gives:

  \[
  \begin{align*}
  \mu x.(0 \lor \text{inr}^2 x) \\
  &\equiv 0 \lor \text{inr}^2 (\mu x.(0 \lor \text{inr}^2 x)) \\
  &\equiv 0 \lor \text{inr}^2 (0 \lor \text{inr}^2 (\mu x.(0 \lor \text{inr}^2 x))) \\
  &\equiv 0 \lor (\text{inr}^2 0) \lor \text{inr}^4 (\mu x.(0 \lor \text{inr}^2 x)) \\
  \ldots \\
  &\equiv 0 \lor (\text{inr}^2 0) \lor (\text{inr}^4 0) \lor \ldots \\
  &\lor (\text{inr}^{2k} 0) \lor \text{inr}^{2k+2} (\mu x.(0 \lor \text{inr}^2 x))
  \end{align*}
  \]

- In general, $\mu x.\varphi(x)$ corresponds to the infinite union $\bigcup_{i \geq 0}[^{i} \varphi(f)]$. 
The inner-type proof rules include the standard boolean axioms for

- distributivity,
- commutativity, and
- associativity.

with respect to $\land$, $\lor$.

Park’s rules for reasoning about the least fixed-point:

$$\varphi(\mu x.\varphi(x)) \leq \mu x.\varphi(x) \quad \frac{\psi(\varphi) \leq \varphi}{\mu x.\psi(x) \leq \varphi}$$
Proof system: meta-predicates

- $T(\varphi)$ if every sub-formula $t$ of $\varphi$ occurs inside a lifting-context $(\_\_\_)^{\uparrow}$. $T$ for “termination”: if $T(\varphi)$ then $\bot \notin [\varphi]$. We have $\frac{T(\varphi(f))}{T(\mu x. \varphi(x))}$.

- $P(\varphi)$ if every sub-formula of $\varphi$ is free of $f$, disjunction, conjunction, and the least fixed-point operator $\mu$. $P$ for “prime open”.

- **Contracting context** defined inductively:
  - $(\_\_)^{\uparrow}$, $\text{inl}(\_)$, $\text{inr}(\_)$, are contracting contexts.
  - $p \cdot (\_)$ is a contracting context if $T(p)$, and similarly, $(\_ \cdot q$ is a contracting context if $T(q)$.
  - if $\varphi(\_)$ and $\psi(\_)$ are contracting contexts, then so is the composition $\varphi(\psi(\_))$. 

Proof system: intra-type

- **Lifting.** \((f^{\sigma})^\uparrow = f^{\sigma \bot}\), and \((\cdot)^\uparrow\) distributes over \(\land\) and \(\lor\).

- **Smash product.**
  - \(f \cdot \psi = \varphi \cdot f = f\)
  - \(\frac{P(\psi)}{t \cdot \psi = t}\)
  - \(\frac{P(\varphi)}{\varphi \cdot t = t}\)
  - \(\cdot\) distributes over \(\land\), \(\lor\) on both left and right.

- **Coalesced sum.**
  - \(\text{inl } t = \text{inr } t = t\), \(\text{inl } f = \text{inr } f = f\)
  - \(\frac{T(\varphi)}{\text{inl } \varphi \land \text{inr } \psi = f}\)
  - \(\text{inl}\) and \(\text{inr}\) distribute over \(\land\) and \(\lor\).

- **Contracting context**
  - \(\frac{T(p)}{p \leq \varphi(p)}\)
  - \(\frac{p = f}{\text{where } \varphi(\bullet) \text{ is a contracting context}}\)
**Example**

If \( \phi \) is distributive, then we can prove that

\[
\mu x. (p_1 \lor p_2 \lor \phi(x)) = [\mu x. p_1 \lor \phi(x)] \lor [\mu y. p_2 \lor \phi(y)]
\]

**Proof.** Let \( p \equiv (\mu x. p_1 \lor \phi(x)) \lor (\mu y. p_2 \lor \phi(y)) \) and \( q(x) \equiv p_1 \lor p_2 \lor \phi(x) \). Then by the distributivity of \( \phi \)

\[
q(p) \equiv p_1 \lor p_2 \lor \phi(p)
\]

\[
= p_1 \lor p_2 \lor \phi((\mu x. p_1 \lor \phi(x)) \lor (\mu y. p_2 \lor \phi(y)))
\]

\[
= p_1 \lor p_2 \lor \phi(\mu x. p_1 \lor \phi(x)) \lor \phi(\mu y. p_2 \lor \phi(y))
\]

\[
= (p_1 \lor \phi(\mu x. p_1 \lor \phi(x))) \lor (p_2 \lor \phi(\mu y. p_2 \lor \phi(y)))
\]

\[
= (\mu x. p_1 \lor \phi(x)) \lor (\mu y. p_2 \lor \phi(y))
\]

\[
\equiv p
\]

which gives \( \mu x. q(x) \leq p \), the non-trivial direction.
Another example

Abbreviate \( \text{inl } t \uparrow \) as 0 and \( \text{inr } \) as s.

Want to prove \((\mu x.0 \lor s^2 x) \land (\mu x.s(0) \lor s^2 x) = f\). We have

\[
(\mu x.0 \lor s^2 x) \land (\mu x.s(0) \lor s^2 x) \\
= (\mu x.0 \lor s^2 x) \land s(\mu x.0 \lor s^2 x) \\
= (0 \lor s^2(\mu x.0 \lor s^2 x)) \land s(\mu x.0 \lor s^2 x) \\
= (0 \land s(\mu x.0 \lor s^2 x)) \lor (s^2(\mu x.0 \lor s^2 x) \land s(\mu x.0 \lor s^2 x)) \\
= s^2(\mu x.0 \lor s^2 x) \land s(\mu x.0 \lor s^2 x) \\
= s[(\mu x.0 \lor s^2 x) \land (\mu x.s(0) \lor s^2 x)]
\]

Step 4 uses \( \frac{T(\varphi)}{\text{inl } \varphi} \land \frac{T(\psi)}{\text{inr } \psi} = f \) to obtain \( (0 \land s(\mu x.0 \lor s^2 x)) = f \)

Now use \( \frac{T(p)}{p \leq \text{inr } p} \) with \( p \equiv (\mu x.0 \lor s^2 x) \land (\mu x.s(0) \lor s^2 x) \)
Rest of the talk

Interplay between regular languages over $\Sigma$ and terminating formulas of type $P = \Sigma \perp (\Sigma \otimes P)$

- A set $\Sigma$ of size $n$ can be represented as the coalesced sum $1 \perp (1 \perp (1 \perp \cdots))$ with $(n - 1)$ times of $\perp$ operations.

- Each distinct symbol of $\Sigma$ can be uniquely identified with a formula $\text{inr}^k \text{inl} t$ for some $k \geq 0$. For convenience, we use standard symbols such as $a, b, c$ to range over both elements of $\Sigma$ and their corresponding formulas over $\text{rec} t. \Sigma \perp (\Sigma \otimes t)$.

- Strings can then be encoded accordingly in an unambiguous way. For example, the string $abab$ corresponds to the formula $\text{inr} (a \cdot (\text{inr} (b \cdot (\text{inr} (a \cdot (\text{inl} b)))))$. 

A terminating \( \mu \)-formula \( \varphi \) of type \( P \) determines a language by the following inductive definition:

- \( \mathcal{R}(f) = \emptyset \).
- \( \mathcal{R}(\text{inl } a) = \{a\} \) for each \( a \in \Sigma \).
- \( \wedge \) corresponds to intersection and \( \vee \) corresponds to union.

- If \( \mathcal{R}(\varphi) = A \), and \( a \in \Sigma \), then
  \( \mathcal{R}(\text{inr } (a \cdot \varphi)) = \{aw \mid w \in A\} \).
- \( \mathcal{R}(\mu x. \varphi(x)) = \bigcup_{i \geq 0} \mathcal{R}(\varphi^i(f)) \).

Question: what class of languages is determined by \( \mathcal{R}(\varphi) \)?

Problem: infinite union may give rise to any language.
Results

Theorem

- For each terminating formula $\varphi$, $R(\varphi)$ is an $\varepsilon$-free regular language, and for every such language $L$, there exists a terminating formula $\varphi$ such that $R(\varphi) = L$.

- There is an effective procedure to find the regular language determined by a terminating formula.

- For terminating formulas $\varphi, \psi$, $[\varphi] \subseteq [\psi]$ if and only if $R(\varphi) \subseteq R(\psi)$, and the problems of semantic containment $[\varphi] \subseteq [\psi]$ and emptiness $[\varphi] = \emptyset$ are decidable.
The regularity of $\mathcal{R}(\varphi)$: I

$\varphi$ a terminating formula. First rename all of its bound variables so that they are all distinct from each other. Then derive a system of language equations associated with $\varphi$.

- The only sub-formulas of $t$, $f$ and variables are the formulas themselves.
- The sub-formulas of $\varphi \land \psi$ and $\varphi \lor \psi$ consist of the formulas themselves together with sub-formulas of $\varphi$ and $\psi$.
- The sub-formulas of $\text{inr}(a \cdot \varphi)$ consist of the formula itself, $a$, and the sub-formulas of $\varphi$.
- The sub-formulas of $\mu x. \varphi(x)$ consist of the formula itself and sub-formulas of $\varphi(x)$. 
The regularity of $\mathcal{R}(\varphi)$: II

For each sub-formula $\psi$ of $\varphi$, introduce a language variable $X_\psi$ associated with it. Then build a system of equations as follows:

- If $p$ is a closed, propositional sub-formula of $\varphi$, then add equation $X_p = p$.
- If $\psi_1 \land \psi_2$ is a sub-formula of $\varphi$, then add equation $X_{\psi_1 \land \psi_2} = X_{\psi_1} \land X_{\psi_2}$. Similarly for $\land$.
- If $\psi \equiv \text{inr}(a \cdot \varphi)$ is a sub-formula of $\varphi$, then add equation $X_\psi = a \cdot X_\varphi$.
- Finally, if $\mu z. \psi$ is a sub-formulas of $\mu x. \varphi(x)$, then add equation $Z = X_\psi$. 
Example

The formula $\mu x. (\text{inl } 0 \lor \text{inr} (\text{inl } 0 \cdot x))$ determines the following system of equations:

\[
\begin{align*}
X_0 &= 0 \\
Y &= 0 \cdot X \\
Z &= X_0 \lor Y \\
X &= Z
\end{align*}
\]
Solving language equations

**Key idea:** if solution of systems of language equations is regular and unique, and \( \bigcup_{i \geq 0} R(\varphi^i(f)) \) is part of the solution, then \( R(\mu x.\varphi(x)) \) is regular.

A system of language equations is a collection of equations

\[
\begin{align*}
X_1 &= \varphi_1(X_1, \cdots, X_n) \\
\cdots \\
X_n &= \varphi_n(X_1, \cdots, X_n)
\end{align*}
\]

where each \( \varphi_i \) is an expression built up inductively from

- variables \( X_i, \ i = 0, \ldots n \), and regular languages.
- union and intersection.
- *left-concatenation*, i.e., when forming a concatenation, the left operand must be a constant (language).
Solution

A language vector \((L_1, \ldots, L_n)\) is said to be a solution of the system of language equations if by substituting the variables \(X_i\) by their corresponding languages \(L_i\) for \(i = 0, \ldots, n\), we obtain a system of language equalities

\[
\begin{align*}
L_1 &= \varphi_1(L_1, \cdots, L_n) \\
\vdots \\
L_n &= \varphi_n(L_1, \cdots, L_n)
\end{align*}
\]
**ε**-property

W.r.t \( \{X_i = \varphi_i(X_1, \ldots, X_n) \mid i = 1, \ldots, n\} \), the set of expressions having the \( \varepsilon \)-property is defined as

- any constant has the \( \varepsilon \)-property.
- a variable \( X_i \) has the \( \varepsilon \)-property if \( \varphi_i(X_1, \ldots, X_n) \) has the \( \varepsilon \)-property.
- a left-concatenation \( L \cdot \psi \) has the \( \varepsilon \)-property if either \( \varepsilon \not\in L \), or \( \psi \) has the \( \varepsilon \)-property.
- a conjunction (disjunction) has the \( \varepsilon \)-property if each of its conjuncts (disjuncts) has the \( \varepsilon \)-property.

The equation system is said to have the \( \varepsilon \)-property if every variable \( X_i \) has the \( \varepsilon \)-property for \( i = 0, \ldots, n \) w.r.t. to the given equation system.
Lemma. An equation system with the $\epsilon$-property has a unique solution.

Proof steps

- Leiss’s result: an equation system with the $\lambda$-property has a unique solution
- Reduce an equation system with the $\epsilon$-property to an equivalent equation system with the $\lambda$-property

Example.

\[
\begin{align*}
X_1 &= L \\
X_2 &= X_1
\end{align*}
\]

This equation system has the $\epsilon$-property. It fails to have the $\lambda$-property of Leiss.
\( U_{i \geq 0} R(\varphi^i(f)) \) is part of the solution

Fixed-point formulation of equation system:

- A system of language equations
  \[ X_i = \varphi_i(X_1, \ldots, X_n), \quad i = 0, \ldots, n \]
  induces a function
  \( \Phi : (2^{\Sigma^*})^n \to (2^{\Sigma^*})^n \)
  with
  \[ \Phi(L_1, \ldots, L_n) = \text{def} \ (\varphi_1(L_1, \ldots, L_n), \ldots, \varphi_n(L_1, \ldots, L_n)) \]
  for each language vector \((L_1, \ldots, L_n) \in (2^{\Sigma^*})^n\).

- Under coordinatewise set-inclusion, \((2^{\Sigma^*})^n\) is a complete lattice, and \(\Phi\) is a Scott continuous function (note that negation is not an operator considered here).

- By the continuity of \(\Phi\), the least fixed-point \(L\) of \(\Phi\) is
  \[ L := \bigcup_{i \geq 0} \Phi^i(\emptyset, \ldots, \emptyset). \]
  This implies that a solution to
  \[ X = X_{\varphi(x)} \]
  can be given as \( \bigcup_{i \geq 0} R(\varphi^i(f)) \).
Regular languages as mu-formulas

- Any regular language is accepted by some deterministic finite automaton.
- A deterministic finite automaton can be translated to an equation system

\[
\begin{align*}
X_1 &= \varphi_1(X_1, \cdots, X_n) \\
\vdots \\
X_n &= \varphi_n(X_1, \cdots, X_n)
\end{align*}
\]

where each \( \varphi_i(X_1, \cdots, X_n) \) takes a simple linear form

\[
a_1 \cdot X_{i_1} \lor a_2 \cdot X_{i_2} \lor \cdots \lor a_m \cdot X_{i_m} \lor L_i,
\]

with \( L_i = \{\epsilon\} \) if \( X_i \) is a final state, and \( L_i = \emptyset \) otherwise.

- The solution to the equation system can be expressed by a vector of mu-formulas using the idea of Gaussian elimination.
Gaussian elimination

\[
\begin{align*}
X_1 &= \varphi_1(X_1, \cdots, X_n) \\
\vdots \\
X_n &= \varphi_n(X_1, \cdots, X_n)
\end{align*}
\]

Replacing all occurrences of \(X_1\) by the formula \(\xi_1(X_2, \ldots, X_n) \equiv \mu X_1. \varphi_1(X_1, \cdots, X_n)\) in equations 2 to \(n:\)

\[
\begin{align*}
X_2 &= \varphi_2(\xi_1(X_2, X_3, \ldots, X_n), \cdots, X_n) \\
\vdots \\
X_n &= \varphi_n(\xi_1(X_2, X_3, \ldots, X_n), \cdots, X_n)
\end{align*}
\]
Gaussian elimination continued

Next, replace all occurrences of $X_2$ by the formula

$$\xi_2(X_3, \ldots, X_n) \equiv \mu X_2. \varphi_2(\xi_1(X_2, X_3, \ldots, X_n), \ldots, X_n)$$

in equations 3 to $n$ to obtain

$$\begin{cases} 
X_3 = \varphi_3(\xi_1(\xi_2, X_3, \ldots, X_n), \xi_2, \ldots, X_n) \\
\vdots \\
X_n = \varphi_n(\xi_1(\xi_2, X_3, \ldots, X_n), \xi_2, \ldots, X_n)
\end{cases}$$

In $n$ steps, we obtain $\xi_n \equiv \mu X_n. \varphi'(X_n)$ where $\varphi'(X_n)$ is a formula without variables other than $X_n$; so $\xi_n$ is closed.
Backwards substitution

Now start substituting backwards, to obtain a closed formula for $\xi_{n-1}$, and then $\xi_{n-2}$, etc., until we obtain a closed formula for $\xi_1$.

$\xi_1$ would be the $\mu$-formula to encode the language $L$ except for one problem: the presences of $\varepsilon$ in some places for which we have no corresponding formula to represent.

However, this problem can be overcome by noting that the linear form for $X_1$ does not involve $\varepsilon$ (we only consider $\varepsilon$-free languages here!), and $a \cdot (\mu X. \varphi(X))$ is a formula equivalent in meaning to $\mu X. (a \cdot \varphi(X))$. The $\varepsilon$-term in $\varphi(X)$ can then be absorbed by using the distributivity of $\cdot$ over $\lor$, since $\varphi$ is a linear term.
Gaussian elimination example

Here is a set of language equations for some DFA, where $X_1$ represents the initial state:

\[
\begin{align*}
X_1 &= aX_2 \lor bX_1 \\
X_2 &= aX_2 \lor bX_3 \lor \epsilon \\
X_3 &= bX_3 \lor aX_1 \lor \epsilon
\end{align*}
\]

- Replace $X_1$ by $\mu X_1. (aX_2 \lor bX_1)$:

\[
\begin{align*}
X_2 &= aX_2 \lor bX_3 \lor \epsilon \\
X_3 &= bX_3 \lor a(\mu X_1. (aX_2 \lor bX_1)) \lor \epsilon
\end{align*}
\]

- Replace $X_2$ by $\mu X_2. (aX_2 \lor bX_3 \lor \epsilon)$:

\[
X_3 = bX_3 \lor a(\mu X_1. (a(\mu X_2. (aX_2 \lor bX_3 \lor \epsilon)) \lor bX_1)) \lor \epsilon
\]

- The solution to this equation is expressed by

\[
\mu X_3. (bX_3 \lor a(\mu X_1. (a(\mu X_2. (aX_2 \lor bX_3 \lor \epsilon)) \lor bX_1)) \lor \epsilon)
\]
Substituting backwards . . .

- Replace $X_3$ by the last formula for $X_2$:
  \[
  \mu X_2. (aX_2 \lor bX_3 \lor \epsilon) \\
  = \mu X_2. \epsilon \lor aX_2 \lor \\
  b(\mu X_3. \epsilon \lor bX_3 \lor a(\mu X_1. bX_1 \lor a(\mu X_2. \epsilon \lor aX_2 \lor bX_3)))
  \]

- Substitute $X_2$ by this formula for $X_1$:
  \[
  \mu X_1. bX_1 \lor \\
  a(\mu X_2. \epsilon \lor aX_2 \lor \\
  b(\mu X_3. \epsilon \lor bX_3 \lor a(\mu X_1. bX_1 \lor a(\mu X_2. \epsilon \lor aX_2 \lor bX_3))))
  \]
Eliminating $\varepsilon$

Finally, the occurrences of $\varepsilon$'s are removed by the distributive law ($\cdot$ over $\lor$), and bound variables have been renamed for clarity:

$$
\mu X_1. bX_1 \lor
\mu X_2. a \lor aaX_2 \lor
a(\mu X_3. b \lor bbX_3 \lor
ba(\mu Y_1. bY_1 \lor
(\mu Y_2. a \lor aaY_2 \lor abX_3))))
$$
Concluding remarks

- Use automata-theoretic ideas to tackle decidability issues of domain $\mu$-calculus.

- Question: is the mu-calculus for $T = \Sigma_{\bot} + (T \otimes T)$ decidable?

- How about completeness?

- Tree automata may be relevant for these questions.

- No magic bullet here: these questions could well be inherently hard. Case in point: the star-height problem, the generalized star-height problem (still open!)
<table>
<thead>
<tr>
<th></th>
<th>Modal $\mu$-calculus</th>
<th>Domain $\mu$-calculus</th>
</tr>
</thead>
<tbody>
<tr>
<td>“fragment”</td>
<td>restricting formulas</td>
<td>restricting types</td>
</tr>
<tr>
<td>finite model prop.</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>decidability</td>
<td>yes</td>
<td>open</td>
</tr>
<tr>
<td>completeness</td>
<td>yes</td>
<td>open</td>
</tr>
<tr>
<td>duality results</td>
<td>yes</td>
<td>open</td>
</tr>
</tbody>
</table>

Some progresses:
Kozen83, Bonsanque-Kok99, Walukiewicz00
Axioms and rules

( $\mu$-axiom )

( $\mu$-rule )

$\varphi(\mu x. \varphi(x)) \leq \mu x. \varphi(x)$

$\frac{\varphi(\varphi) \leq \varphi}{\mu x. \varphi(x) \leq \varphi}$
Theorem. \( \varphi(\mu x. \varphi(x)) = \mu x. \varphi(x) \)

\[
\begin{array}{ll}
\varphi(\mu x. \varphi(x)) & \leq \mu x. \varphi(x), \\
\varphi(\varphi(\mu x. \varphi(x))) & \leq \varphi(\mu x. \varphi(x)), \\
\mu x. \varphi(x) & \leq \varphi(\mu x. \varphi(x)), \\
\mu x. \varphi(x) & = \varphi(\mu x. \varphi(x)).
\end{array}
\]

- \( \mu \)-axiom
- Monotonicity
- \( \mu \)-rule
- \( \mu \)-axiom