Notas del Trabajo Especial

Rocío Perez Sbarato

Abril 2025

Índice

Índice de figuras

1. Conjuntos no bien fundados y diagramas de estructuras reflexivas

Uno de los objetivos centrales del trabajo es caracterizar los modelos de sistemas formales reflexivos y señalar algunas propiedades recurrentes. Para ello, es posible proponer una representación formal aprovechando las posibilidades

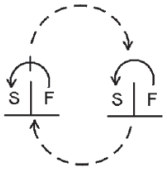

Figura 1: Diagrama de la estructura de la paradoja del Mentiroso.

que nos brinda la teoría de conjuntos ***ZFA*** para representar circularidad. Es aún más interesante establecer una relación con los análisis formales de las estructuras detrás de muchas paradojas, realizados por Grim y Rescher, plasmados en un tipo de diagramas especial para la ocasión.

1.1. La paradoja del Mentiroso

1.1.1. Diagrama estructural

En *Reflexivity: From Paradox to Consciousness*, Grim y Rescher se proponen mostrar que las distintas paradojas tienen una forma o estructura muy parecida. Este eje central que subyace al absurdo de las paradojas puede ser expresado con un tipo de diagrama que podemos ver en la Figura 1. En esta figura, tenemos la separación de una oración entre sujeto y predicado, lo cual nos permite construir la oración que se refiere a sí misma. Del lado izquierdo se encuentra el sujeto y del lado derecho el predicado. La flecha de estos diagramas representa que el predicado se aplica al sujeto. El predicado debe ser algo que pueda decirse de un sujeto. En nuestro caso, el sujeto se llama **S** y debe referirse a sí misma, es decir, tenerse a sí misma como sujeto.

Dicho esto, podemos notar que la paradoja semántica que se ve expresada en esa circularidad no es cualquier oración auto-referencial sino la expresión de la paradoja del Mentiroso: "esta oración es falsa". Lo paradójico de esta oración se puede ver en la oscilación de la Figura 1, donde si S - que representa "esta oración es falsa es falsa, entonces no puede ser que sea falsa. Por otro lado, si no es falsa, entonces tiene que serlo. Resumiendo, la paradoja contiene auto-referencia y negación.

Similarmente, la paradoja de Russell contiene un ciclo. Sea R es el conjunto russelliano compuesto por cualquier conjunto que no pertenezca a sí mismo. Entonces, si R no pertenece a sí mismo, entonces debe pertenecer a sí mismo. En la Figura 2, la flecha representa la relación de pertenencia. A diferencia de la paradoja del Mentiroso, la de Russell solo trabaja con componentes con una estructura única y no de dos partes.

1.1.2. Proposición, ecuación y su *labeled graph* asociado

A su vez, los grafos tienen sus nodos y sus aristas. Se trabaja con *labeled graphs* y se define el decorado de estos grafos como el conjunto de los hijos de

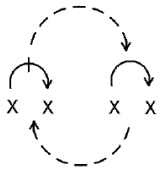

Figura 2: Diagrama de la estructura de la paradoja de Russell

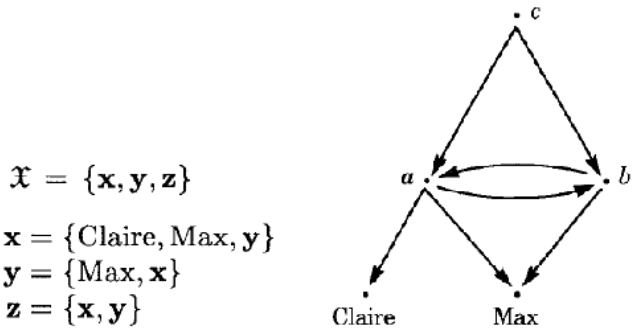

Figura 3: Relación entre decorado y soluciones.

cada nodo. Si no tiene hijos, entonces ese nodo se decora con su *label*. Luego, los nodos de los grafos son conjuntos y las aristas son establecidas por la relación de pertenencia invertida entre conjuntos. En la Teoría de Conjuntos con AFA incorporado, estos grafos pueden ser tanto bien fundados como no serlo. Por ejemplo, en la Figura 3 se ve un grafo no bien fundado.

Ahora bien, uno de los conceptos clave desarrollados por Barwise y Moss es el *Solution Lemma*. Tiene distintas aristas y colores, en general este lemma significa que cada sistema de ecuaciones tiene una única solución. Esto solo vale en la Teoría de Conjuntos ***ZFA*** (con el Axioma de Anti-Fundación), no en la teoría ***ZFC*** que conocemos. En respecto a los grafos, plantea que cada *labeled graph* tiene un único decorado. Podemos pensar que un conjunto es cualquier colección de objetos cuya relación de pertenencia puede ser ilustrada a través de un grafo. En esta Teoría, puede pasar que un conjunto tenga varios grafos que lo representan, pero en el fondo todos deben tener el mismo decorado.

Otra cosa importante a destacar es que las soluciones de sistemas de ecuaciones y las decoraciones de los grafos se corresponden. En *Vicious Circles* está demostrada la equivalencia entre el *Solution Lemma* en general y para grafos. Teniendo en cuenta estas dos ideas, es claro que las soluciones a las ecuaciones

son un mapeo que asigna valores a las variables indeterminadas. En la Figura 3, esas variables indeterminadas son $X = \{x, y, z\}$. Para entender mejor este concepto ir a la sección Definiciones útiles.

La solución o las soluciones de cada ecuación del sistema se encuentran en el decorado del grafo asociado. Por ejemplo, la Figura X. muestra un grafo G_p asociado a la proposición $p = \langle E, q, 0\rangle$, la cual nos dice que la proposición **q** no tiene la propiedad E. La solución a la ecuación $p = \langle E, q, 0\rangle$ es el grafo asociado a G_p. Recordar que, en Teoría de Conjuntos, un par ordenado $\langle a, \langle b, c\rangle\rangle$ es pensado como {a, {b, c}}.

Esta estructura matemática enmarcada en ***ZFA*** permite la circularidad en estos grafos, como se ve en las Figuras 5 y 4. En esta última, la solución de la ecuación es ella misma, lo que se puede ver en el *loop* a su propia raíz. Notar que en este caso tomamos la proposición p y la cambiamos para obtener la auto-referencialidad $q = \langle E, q, 0\rangle$. Esta no es necesariamente la paradoja del Mentiroso, pero sin lugar a dudas tiene su estructura. Si $E =$ "esta oración es verdadera", entonces q representa "esta oración no tiene la propiedad "es verdadera. O sea, la oración q es "esta oración es falsa". Podemos representar la paradoja de Russell mediante una ecuación auto-referencial en nuestro marco de ecuaciones del tipo $q = \langle E, q, 0\rangle$. En este caso, definimos la expresión $E(x)$ como

$$E(x) = x \notin x$$

y construimos el par ordenado $q = \langle E, q, 0\rangle$. Este objeto representa "el conjunto de todos los conjuntos que no se contienen a sí mismos". Evaluar si $q \in q$ equivale a evaluar la condición $E(q)$, es decir $q \in q \iff q \notin q$ lo cual genera una contradicción. De este modo, reproducimos en nuestro sistema la estructura lógica de la paradoja de Russell.

1.1.3. Relación entre diagrama y ecuación

Sea S la sentencia con sujeto y predicado que vimos en la **sección 1.1.1** y q la proposición que vimos en 1.1.2. Su equivalencia surge de la reflexividad, negación y la propiedad/predicado que se aplica a la misma estructura auto-referencial. presente en ambas. Además, las propiedades $E =$ "es verdaderoz el predicado $F =$ "es falso"son opuestas. Es decir, $E = \neg F$ o $F = \neg E$. Entonces, "esta oración es falsa"no cumple E si y solo si cumple F. Esto es equivalente a decir que S cumple F si y solo si S cumple $\neg E$, q cumple F si y solo si q cumple $\neg E$ y por lo tanto S es equivalente a q en tanto ambas representan el mismo estado de verdad respecto de la evaluación de E. O sea, S es falsa si y solo si q no es verdadera.

Algo similar podríamos decir para la generalización del diagrama de la paradoja del mentiroso (Figura 6) y otros valores para la propiedad E. Es crucial estudiar esto aplicado a la paradoja de Russell, paradojas semánticas como heterological o teoremas como Halting problem y el de Incompletitud. En el caso de estos dos últimos deberá tener otro enfoque, puesto que son teoremas y el absur-

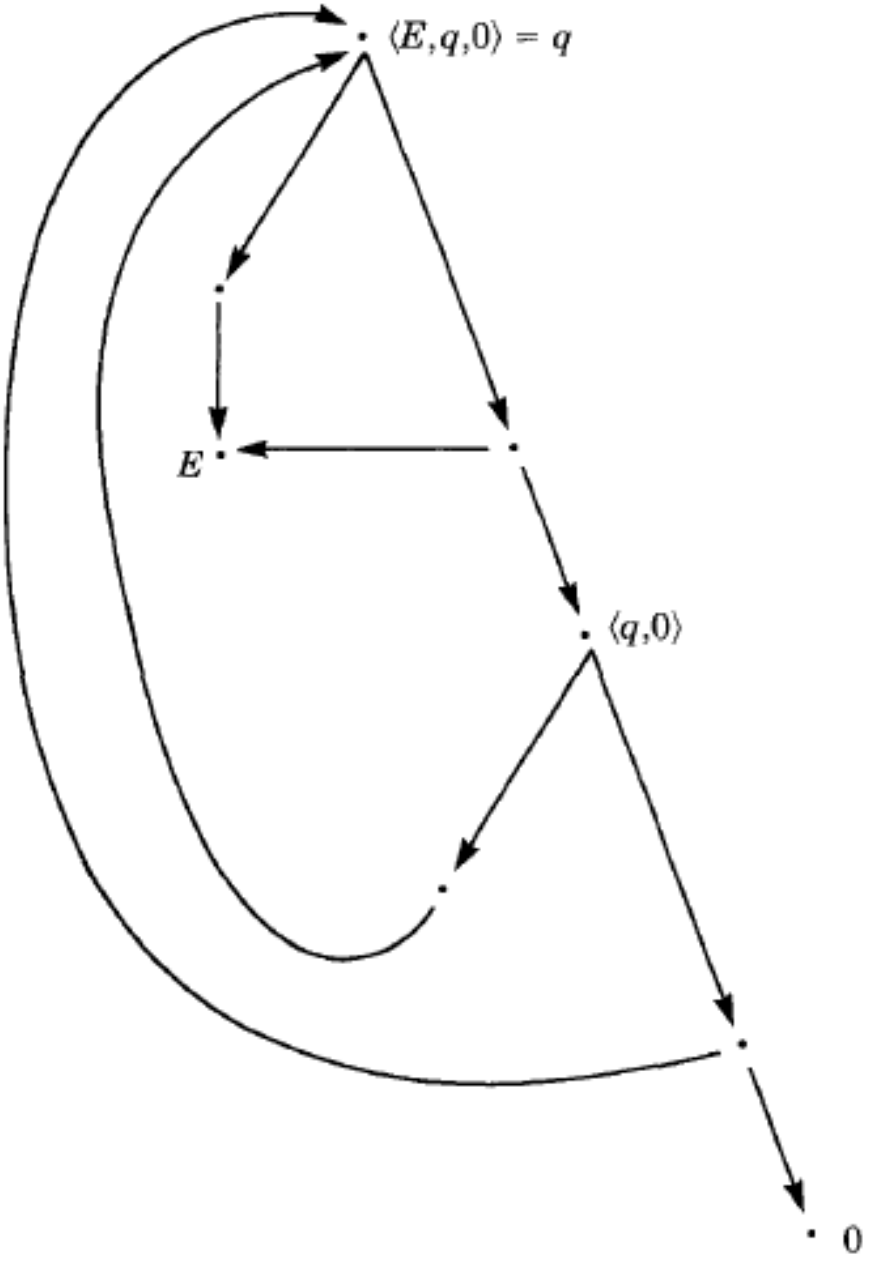

Figura 4: Grafo de $q = \langle E, q, 0\rangle$.

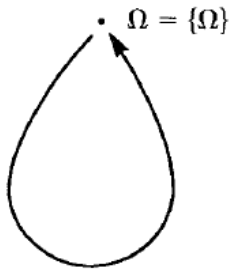

Figura 5: Grafo de la ecuación $\Omega = \{\Omega\}$.

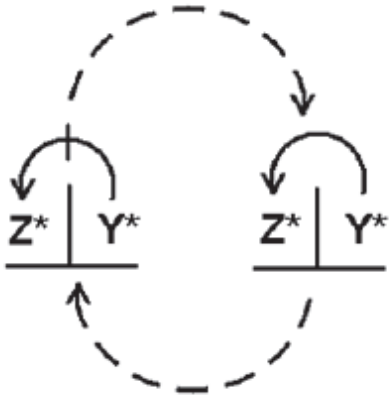

Figura 6: Diagrama de la estructura de la paradoja del Mentiroso y demás paradojas con una estructura de dos partes.

do surge de una construcción de prueba. Sin embargo, se mantiene la estructura de "sujeto y predicado"del Mentiroso.

Una aproximación al análisis para la paradoja de Russell consiste en algo similar. La propiedad E indica "no pertenecer a sí mismo", que es negación de la propiedad $P(x) = x \in x$. Es decir, $E = \neg P$ y, por lo tanto, $P = \neg E$. Si $q \in q$, entonces no cumple E, y por definición de E, eso implica que $q \notin q$. Por el contrario, si $q \notin q$, entonces cumple E, lo cual implica que $q \in q$. Por lo tanto, S y q son equivalentes en tanto ambas representan una estructura auto-referencial basada en la aplicación de una propiedad a sí misma, y ambas conducen a una contradicción lógica al evaluarse.

Una diferencia entre los diagramas y los grafos es que la Figura 4 representa la oración que causa una paradoja, mientras que diagramas como el de la Figura 1 presentan el "bucle extraño"de la paradoja, es decir, la dinámica de auto-referencia que genera la contradicción. Es por eso que podemos establecer la relación anterior, donde la propiedad E y el predicado P son opuestos: $E(x) = \neg P(x)$. En el caso del Mentiroso, E puede ser "es verdaderaz P = "es falsa", mientras que en la paradoja de Russell, $E(x) = x \notin x$ y $P(x) = x \in x$. En ambos casos, la paradoja emerge cuando una entidad se evalúa respecto de una propiedad que la niega al cumplirse, generando así una estructura lógicamente inestable.

Surgen las interrogantes de cómo lograr que el diagrama solo represente la oración y cómo lograr que el grafo represente el bucle extraño. La respuesta a la primera pregunta es simple, pues S es esa misma representación. La respuesta a la segunda pregunta parece ser más complicada a la hora de formalizarla, el hecho de que $q = \langle E, q, 0\rangle$ y $q = \langle E, q, 1\rangle$ tienen el mismo grafo podría generar un problema. Sin embargo, podemos justificar, mediante una explicación lógica similar a Grim y Rescher, que $q = \langle E, q, 0\rangle$ y $q = \langle E, q, 1\rangle$ generan un "bucle extraño".

Cabe aclarar que esta relación entre P y E existe porque ambas son lo que se dice de la unidad auto-referencial, independientemente de si esta propiedad se aplica o no se aplica a tal unidad.

1.2. Prototipos de sistemas reflexivos

1.2.1. Paradoja de Russell

En un principio, buscamos modelar el conjunto X del diagrama de la Figura 2. También es posible el camino de modelar el par ordenado $q = \langle E, q, 0\rangle$ mediante hypergraphs.

1.2.2. Intento con bug

Es importante destacar que existe un bug en Haskell[1], donde GHC tiene el error `ghc: panic! (the 'impossible' happened)` al ejecutar el siguiente programa:

[1]Código fuente: `https://okmij.org/ftp/Haskell/impredicativity-bites.html`

Figura 7: Diagrama de $R \notin R$

Listing 1: Modelo en Haskell de la Paradoja de Russell

```haskell
{-# LANGUAGE GADTs, KindSignatures, EmptyDataDecls #-}

data False -- Fantasma
data J c = J (c ())

{- Si el conjunto no pertence a si mismo, entonces pertenece al conjunto
    russelliano R -}
data R :: * -> * where
   MkR :: (c (J c) -> False) -> R (J c)

{- La funcion f toma como argumento el mismo R (J R) que construye -}
cond_false :: R (J R) -> False
cond_false x@(MkR f) = f x

absurd :: False
absurd = cond_false (MkR cond_false) -- Ciclo de self-reference
```

En este caso, `f` es el predicado $P(x) := x \notin x$. Mientras que P construye el conjunto R, el predicado se aplica a este mismo conjunto.

```
cond_false (MkR cond_false)

    cond_false (MkR f)

       == f (MkR f)

  == cond_false (MkR cond_false)
```

La condición `cond_false` es similar a preguntar "¿El conjunto R pertenece a sí mismo?". Más formalmente, se trata de evaluar la propiedad P aplicada a R, es decir, determinar si $P(R)$ es verdadera o falsa, o bien, si $R \in \{x : x \notin x\}$. En el ejemplo, se asigna el valor de falsedad a esta pregunta, lo cual como ya vimos lleva a un ciclo infinito.

Es posible establecer la relación con este predicado `cond_false` y $q = \langle E, q, 0 \rangle$. Ambos proponen que el conjunto ruselliano no pertenece a sí mismo, si tomamos la definición de E que dimos en **Proposición, ecuación y su labeled graph asociado** (sección 1.1.2). Notar que P es exactamente este E, q es exactamente `cond_false`. Por esto mismo, podemos decir que es similar a la Figura 7.

1.2.3. Modelos de autoreferencialidad en Haskell sin bugs

Este programa[2] es muy similar al anterior, lo que cambia es el uso de `inline`. Aunque no termina de correr, puesto que genera un ciclo infinito, permite modelar la circularidad de la paradoja sin bugs.

Listing 2: Otra versión del modelo en Haskell de la Paradoja de Russell

```
data False

-- Conjunto ruselliano
data R = MkR {proj :: R -> False}

-- R pertenece a si mismo?
f :: R -> False
f = \x -> proj x x
-- La clave: evitar optimizaciones de GHC
{-# noinline f #-}

omega :: False
omega = f (MkR f) -- Ciclo de self-reference

main = do
  print (omega `seq` ())
```

En el código `data False` define un tipo vacío, sin constructores. Representa una proposición lógicamente falsa. El tipo `R` representa un conjunto russelliano, es decir, un conjunto que contiene elementos que no se contienen a sí mismos. Cada valor de tipo `R` tiene una función llamada `proj` de tipo `R ->False`.

El selector de campo `proj` se genera automáticamente por Haskell, y tiene tipo `proj :: R ->(R ->False)`. Así, `proj x` es una función que toma otro `R` y produce un `False`.

Por otro lado, la función `f` toma un valor `x :: R` y aplica su propio `proj` a sí mismo: `proj x x`. Es decir, pregunta si `x` "se contiene a sí mismo", y se aplica a sí mismo para decidirlo. Esta forma es directamente análoga a la construcción del conjunto de todos los conjuntos que no se contienen a sí mismos pero tiene el componente de la auto-referencia.

El pragma {`-# noinline f #-`} es crucial: le indica al compilador que no optimice ni expanda la definición de `f`, para evitar que el compilador descubra anticipadamente el bucle infinito. Esto fuerza a que la auto-referencia se mantenga en tiempo de ejecución. Notar que el código anterior se arregla usando la misma estrategia.

Listing 3: Arreglo del primer intento de modelo en Haskell de la Paradoja de Russell

```
cond_false :: R (J R) -> False
```

[2] Código fuente en uno de los comentarios del post en `https://www.reddit.com/r/haskell/comments/5nzxf6/is_the_following_encoding_of_russels_paradox/`

```
cond_false x@(MkR f) = f x
{-# noinline cond_false #-}
```

La expresión `omega = f (MkR f)` construye una instancia de `R` con la función `f`, y luego se aplica a sí misma. El resultado es un bucle de auto-aplicación que nunca termina...

$$\begin{aligned}\omega &= \texttt{f (MkR f)} \\ &= \texttt{proj (MkR f) (MkR f)} \\ &= \texttt{f (MkR f)} \\ &= \ldots\end{aligned}$$

Finalmente, en `main`, se intenta forzar la evaluación de `omega` usando `seq` para imprimir `()`. Sin embargo, como `omega` nunca termina, el programa entra en un bucle infinito ($\perp$).

2. Definiciones útiles

2.1. Solución

- The Liar, página 50.

> By an *assignment* for $\mathbb{X}$ in V_A we mean a function $f : \mathbb{X} \to V_A$ which assigns an element $f(\mathbf{x})$ of V_A to each indeterminate $\mathbf{x} \in \mathbb{X}$. Any such assignment f extends in a natural way to a function $\hat{f} : V_A[\mathbb{X}] \to V_A$. Intuitively, given some $a \in V_A[\mathbb{X}]$ one simply replaces each $\mathbf{x} \in \mathbb{X}$ by its value $f(\mathbf{x})$. Rather than write $\hat{f}(a)$, we write $a[f]$, or even more informally, $a(\mathbf{x}, \mathbf{y}, \ldots)$ and $a(f(\mathbf{x}), f(\mathbf{y}), \ldots)$.

Definición

Una asignación $f : V_A[\mathbb{X}] \to V_A$ es una *solución de una ecuación* $\mathbf{x} = a(\mathbf{x}, \mathbf{y}, \ldots) \in V_A[\mathbb{X}]$ si

$$f(\mathbf{x}) = a(f(\mathbf{x}), f(\mathbf{y}), \ldots) \in V_A$$

Notar que $V_A[\mathbb{X}] = V_{A'}$
- f es una solución de un sistema de ecuaciones en $\mathbb{X}$ si es una solución de cada ecuación del sistema.