Teoría de lenguajes: Práctica 1

Agustín Santiago Gutiérrez

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Octubre 2013

Objetivo de la práctica

- Un objeto de estudio central de la primer parte de la materia lo constituyen los autómatas finitos. Serán resultados importantes de la teórica la equivalencia entre los distintos tipos de autómatas finitos (determinista, no determinista, con transiciones λ), las expresiones regulares, y las gramáticas regulares.
- Esta práctica tiene por objetivo fundamental ejercitar el manejo de autómatas finitos: construcción de autómatas para lenguajes, propiedades, manipulación. Dado que son equivalentes, se incorporan algunos ejercicios sobre gramáticas regulares. Las expresiones regulares son un tema que da para muchos más ejercicios y por eso se separa en otra práctica diferente.

División

Si bien los ejercicios están numerados en forma continua y no hay secciones marcadas, la práctica se encuentra dividida implícitamente en cuanto a la temática de los ejercicios de la siguiente manera

- 1) Ejercicios 1,2 y 3 : Construcción de autómatas para lenguajes.
- 2) Ejercicios 4 y 5 : Construcción de un autómata en base a otros.
- 3) Ejercicios 7,8 y 9 : Ejercicios más avanzados de construcción de autómatas, basados en las ideas de los anteriores.
- 4) Ejercicios 10, 11 : Ejercicios de gramáticas regulares.

El ejercicio 6 es un ejercicio teórico sobre AFDs, y no guarda una relación directa clara con otros ejercicios. El ejercicio 10 también es teórico, pero trata sobre gramáticas regulares.

Parte 1 (Construcción de autómatas)

Los primeros 3 ejercicios consisten en la construcción de autómatas para lenguajes particulares, con dificultad creciente.

- El ejercicio 1 muestra ejemplos sencillos en los que se puede usar el estado del autómata para almacenar información para decidir aceptar o no.
- El ejercicio 2 se enfoca en la construcción de lenguajes definidos por la presencia o ausencia de subcadenas, prefijos y sufijos particulares. Estos ejemplos van dando cuenta a una escala muy pequeña de una aplicación posible de los autómatas, que es la búsqueda de un patrón en un texto, que es uno de los usos habituales de las expresiones regulares en la práctica.

Parte 1 (Construcción de autómatas)

- El ejercicio 3 pide la construcción de autómatas que reconozcan lenguajes más complejos, y es una muestra muy interesante de cómo los autómatas se pueden usar en la práctica para reconocer elementos básicos de lenguajes de programación, ya que los lenguajes pedidos se enfocan en constantes, identificadores y valores numericos tipicos de un lenguaje de programación.

Parte 2 (Construcción de autómatas sobre otros)

- El ejercicio 4 pide describir como sería la construcción de varios autómatas nuevos, partiendo de un autómata que acepta un cierto lenguaje L. Aquí se introducen construcciones sencillas pero importantes.
- Como ejemplo concreto, el item 4-a pide dar una construccion para un automata que reconozca el complemento de L, supuesto que tenemos un automata Γ para L. Si Γ es AFD la construcción es muy sencilla: basta complementar el conjunto de estados de aceptación y dejar todo lo demás igual.
- No obstante su aparente sencillez, este ejercicio implicará más adelante que dada una expresión regular, existe otra que acepta el lenguaje complementario, algo completamente no evidente.

Parte 2 (Construcción de autómatas sobre otros)

- El ejercicio 5 resulta similar al anterior, pero ahora se parte de dos lenguajes L_1 y L_2, y se pide dar automatas para los lenguajes correspondientes a la unión, intersección, concatenación y resta.
- Este ejercicio junto con el anterior tienen una gran importancia práctica, ya que su solución provee de métodos efectivos que podemos usar para operar con lenguajes fácilmente.
- Por ejemplo si en un ejercicio se pide obtener un autómata para las cadenas que cumplen las propiedades P_1, P_2 y P_3, pero no la P_4, basta encontrar autómatas que identifiquen las cadenas correspondientes a cada una, y luego podemos operar conjuntistamente utilizando las construcciones que resuelven este ejercicio.

Ejercicio 6

- El ejercicio 6 resulta un poco aislado del resto de la práctica. Pide demostrar propiedades formales sobre la relación de transición de un AFD.
- Dado que los AFD son los autómatas más simples para razonar, pero no por eso menos poderosos o fundamentales que otros autómatas finitos, es buena idea que los alumnos ejerciten la demostración de propiedades formales sobre estos autómatas.
- Demostraciones de propiedades análogas para otros tipos de autómatas son similares.

Parte 3 (Construcción de autómatas avanzados)

Como ya dijimos, los ejercicios 7, 8 y 9 son ejercicios integradores sobre construcción de autómatas, que toman ideas de los ejercicios anteriores para construir lenguajes concretos.

- El ejercicio 7 es un ejemplo concreto que puede resolverse aprovechando la construcción del lenguaje concatenación, y pide construir un AFD para el lenguaje $\{a^k | k = 2i + 3j \wedge 0 \leq i \wedge 0 \leq j\}$
- Podemos primero obtener AFDs para los lenguajes $\{a^k | k = 2i \wedge 0 \leq i\}$, y $\{a^k | k = 3j \wedge 0 \leq j\}$, y luego usar una construcción para la concatenación, obteniendo el pedido.
- Pero también podemos observar que el lenguaje coincide con con $\{a^k | k \neq 1\}$, y obtener directamente un autómata para esta descripción más sencilla.
- Lo que observamos es que si se pretendía que los alumnos aprovechen las construcciones de ejercicios anteriores en este ejercicio, conviene cambiar un poco las definiciones de los lenguajes de manera que la estructura final no sea tan simple, y convenga utilizar esas construcciones en lugar de dar el autómata final directamente.

Parte 3 (Construcción de autómatas avanzados)

Como ya dijimos, los ejercicios 7, 8 y 9 son ejercicios integradores sobre construcción de autómatas, que toman ideas de los ejercicios anteriores para construir lenguajes concretos.

- El ejercicio 7 es un ejemplo concreto que puede resolverse aprovechando la construcción del lenguaje concatenación, y pide construir un AFD para el lenguaje $\{a^k | k = 2i + 3j \wedge 0 \leq i \wedge 0 \leq j\}$
- Podemos primero obtener AFDs para los lenguajes $\{a^k | k = 2i \wedge 0 \leq i\}$, y $\{a^k | k = 3j \wedge 0 \leq j\}$, y luego usar una construcción para la concatenación, obteniendo el pedido.
- Pero también podemos observar que el lenguaje coincide con con $\{a^k | k \neq 1\}$, y obtener directamente un autómata para esta descripción más sencilla.
- Lo que observamos es que si se pretendía que los alumnos aprovechen las construcciones de ejercicios anteriores en este ejercicio, conviene cambiar un poco las definiciones de los lenguajes de manera que la estructura final no sea tan simple, y convenga utilizar esas construcciones en lugar de dar el autómata final directamente.

Parte 3 (Construcción de autómatas avanzados)

Como ya dijimos, los ejercicios 7, 8 y 9 son ejercicios integradores sobre construcción de autómatas, que toman ideas de los ejercicios anteriores para construir lenguajes concretos.

- El ejercicio 7 es un ejemplo concreto que puede resolverse aprovechando la construcción del lenguaje concatenación, y pide construir un AFD para el lenguaje $\{a^k | k = 2i + 3j \wedge 0 \leq i \wedge 0 \leq j\}$
- Podemos primero obtener AFDs para los lenguajes $\{a^k | k = 2i \wedge 0 \leq i\}$, y $\{a^k | k = 3j \wedge 0 \leq j\}$, y luego usar una construcción para la concatenación, obteniendo el pedido.
- Pero también podemos observar que el lenguaje coincide con con $\{a^k | k \neq 1\}$, y obtener directamente un autómata para esta descripción más sencilla.
- Lo que observamos es que si se pretendía que los alumnos aprovechen las construcciones de ejercicios anteriores en este ejercicio, conviene cambiar un poco las definiciones de los lenguajes de manera que la estructura final no sea tan simple, y convenga utilizar esas construcciones en lugar de dar el autómata final directamente.

Parte 3 (Construcción de autómatas avanzados)

- El ejercicio 8 es un ejemplo más complejo, donde queremos reconocer con un AFD el lenguaje de las cadenas que cumplen simultáneamente 3 condiciones.
- Se resuelve utilizando la construcción del AFD que reconoce la intersección de los lenguajes de los AFD correspondientes.

Parte 3 (Construcción de autómatas avanzados)

- El ejercicio 9 pide dar un autómata finito que acepte el lenguaje de las cadenas sobre $\{a, b\}$ que alternan la paridad de la longitud de sus mesetas (secuencias de letras idénticas maximales).
- Este ejercicio es ilustrativo de como un pequeño autómata sencillo que solo recuerde la mínima información necesaria (en este caso, paridad y letra de la meseta actual, y paridad de la ultima) puede reconocer lenguajes aparentemente más complejos.
- En particular como solo tiene finitos estados, un AFD no puede contar las longitudes para saber la paridad: Al resolver este ejercicio el alumno debería ir incorporando ese concepto de que un autómata solo puede recordar una cantidad finita de información, lo que más adelante en la materia se plasmará en el Lema de pumping.

Parte 4 (Gramáticas regulares)

Finalmente, los últimos dos ejercicios exploran el concepto de gramáticas regulares.

- El ejercicio 10 es como ya mencionamos un ejercicio teórico: Se pide demostrar la equivalencia entre dos formas de expresar las gramáticas regulares.
- El ejercicio 11 pide dar gramáticas regulares para los lenguajes de los ejercicios 1 a 3. Este ejercicio claramente ejercita la equivalencia entre gramáticas regulares y autómatas. En particular, estos ejercicios piden mostrar explícitamente (con un autómata y una gramática regular) la equivalencia de ambos formalismos sobre el conjunto de lenguajes considerados en los ejercicios 1 a 3.

Resolución ejercicio 8

Ahora resolveremos el ejercicio 8:
Dar un autómata finito determinístico que acepte todas las cadenas sobre el alfabeto $\{a, b, c\}$ que cumplan simultáneamente:

- Cada a debe estar seguida inmediatamente de una b.
- La cantidad de b debe ser par.
- La cadena no debe terminar en c.