Clases de complejidad sintácticas y semánticas

IIC3810

Sintaxis versus Semántica

Sintaxis: Manipulación de símbolos

Semántica: Interpretación de símbolos

Sintaxis efectiva para una clase de complejidad

Dado un alfabeto Σ:

Sintaxis efectiva para una clase de complejidad

Dado un alfabeto Σ:

- Un lenguaje L es un conjunto de palabras en Σ^*: $L \subseteq \Sigma^*$

Sintaxis efectiva para una clase de complejidad

Dado un alfabeto Σ:

- Un lenguaje L es un conjunto de palabras en Σ^*: $L \subseteq \Sigma^*$
- Una clase de complejidad $\mathcal{C}$ es un conjunto de lenguaje: $\mathcal{C} \subseteq 2^{\Sigma^*}$

Sintaxis efectiva para una clase de complejidad

Dado un alfabeto Σ:

- Un lenguaje L es un conjunto de palabras en Σ^*: $L \subseteq \Sigma^*$
- Una clase de complejidad $\mathcal{C}$ es un conjunto de lenguaje: $\mathcal{C} \subseteq 2^{\Sigma^*}$

¿Cuándo decimos que existe una sintaxis efectiva para $\mathcal{C}$?

Sintaxis efectiva para una clase de complejidad

Considere un lenguaje S y una función $\lambda : S \to 2^{\Sigma^*}$

Sintaxis efectiva para una clase de complejidad

Considere un lenguaje S y una función $\lambda : S \to 2^{\Sigma^*}$

- $\lambda(s)$ indica cuál es el lenguaje asociado a $s \in S$

Sintaxis efectiva para una clase de complejidad

Considere un lenguaje S y una función $\lambda : S \to 2^{\Sigma^*}$

- $\lambda(s)$ indica cuál es el lenguaje asociado a $s \in S$
  - Por ejemplo, si s es una MT, entonces $\lambda(s)$ es el lenguaje aceptado por s

Sintaxis efectiva para una clase de complejidad

Considere un lenguaje S y una función $\lambda : S \to 2^{\Sigma^*}$

- $\lambda(s)$ indica cuál es el lenguaje asociado a $s \in S$
  - Por ejemplo, si s es una MT, entonces $\lambda(s)$ es el lenguaje aceptado por s

S es una sintaxis efectiva para $\mathcal{C}$ si:

Sintaxis efectiva para una clase de complejidad

Considere un lenguaje S y una función $\lambda : S \to 2^{\Sigma^*}$

- $\lambda(s)$ indica cuál es el lenguaje asociado a $s \in S$
  - Por ejemplo, si s es una MT, entonces $\lambda(s)$ es el lenguaje aceptado por s

S es una sintaxis efectiva para $\mathcal{C}$ si:

- Para cada $L \in \mathcal{C}$, existe $s \in S$ tal que $L = \lambda(s)$

Sintaxis efectiva para una clase de complejidad

Considere un lenguaje S y una función $\lambda : S \to 2^{\Sigma^*}$

- $\lambda(s)$ indica cuál es el lenguaje asociado a $s \in S$
  - Por ejemplo, si s es una MT, entonces $\lambda(s)$ es el lenguaje aceptado por s

S es una sintaxis efectiva para $\mathcal{C}$ si:

- Para cada $L \in \mathcal{C}$, existe $s \in S$ tal que $L = \lambda(s)$
- Para cada $s \in S$, existe $L \in \mathcal{C}$ tal que $\lambda(s) = L$

Sintaxis efectiva para una clase de complejidad

Considere un lenguaje S y una función $\lambda : S \to 2^{\Sigma^*}$

- $\lambda(s)$ indica cuál es el lenguaje asociado a $s \in S$
  - Por ejemplo, si s es una MT, entonces $\lambda(s)$ es el lenguaje aceptado por s

S es una sintaxis efectiva para $\mathcal{C}$ si:

- Para cada $L \in \mathcal{C}$, existe $s \in S$ tal que $L = \lambda(s)$
- Para cada $s \in S$, existe $L \in \mathcal{C}$ tal que $\lambda(s) = L$
- El lenguaje S es decidible

Una sintaxis efectiva para una NP

Vamos a ver un ejemplo de sintaxis efectiva para NP utilizando notación lógica

Una sintaxis efectiva para una NP

Vamos a ver un ejemplo de sintaxis efectiva para NP utilizando notación lógica

Sea $\mathcal{L}$ el vocabulario $\{E(\cdot, \cdot)\}$ y STRUCT[$\mathcal{L}$] el conjunto de todas las $\mathcal{L}$-estructuras con dominio $\{1, \ldots, n\}$ para algún $n \in \mathbb{N}$

Una sintaxis efectiva para una NP

Vamos a ver un ejemplo de sintaxis efectiva para NP utilizando notación lógica

Sea $\mathcal{L}$ el vocabulario $\{E(\cdot, \cdot)\}$ y STRUCT[$\mathcal{L}$] el conjunto de todas las $\mathcal{L}$-estructuras con dominio $\{1, \ldots, n\}$ para algún $n \in \mathbb{N}$

Un lenguaje L es un subconjunto de STRUCT[$\mathcal{L}$]

Una sintaxis efectiva para una NP

Vamos a ver un ejemplo de sintaxis efectiva para NP utilizando notación lógica

Sea $\mathcal{L}$ el vocabulario $\{E(\cdot, \cdot)\}$ y STRUCT[$\mathcal{L}$] el conjunto de todas las $\mathcal{L}$-estructuras con dominio $\{1, \ldots, n\}$ para algún $n \in \mathbb{N}$

Un lenguaje L es un subconjunto de STRUCT[$\mathcal{L}$]

- L es un lenguaje puesto que cada estructura puede ser representada como un string

Una sintaxis efectiva para una NP

Dada una oración φ sobre el vocabulario $\mathcal{L}$, definimos el lenguaje:

$$L_\varphi \quad = \quad \{\mathfrak{A} \in \mathsf{STRUCT}[\mathcal{L}] \mid \mathfrak{A} \models \varphi\}$$

Una sintaxis efectiva para una NP

Dada una oración φ sobre el vocabulario $\mathcal{L}$, definimos el lenguaje:

$$L_\varphi \quad = \quad \{\mathfrak{A} \in \mathsf{STRUCT}[\mathcal{L}] \mid \mathfrak{A} \models \varphi\}$$

L_φ define el lenguaje asociado a una oración φ

Una sintaxis efectiva para una NP

Dada una oración φ sobre el vocabulario $\mathcal{L}$, definimos el lenguaje:

$$L_\varphi \quad = \quad \{\mathfrak{A} \in \mathsf{STRUCT}[\mathcal{L}] \mid \mathfrak{A} \models \varphi\}$$

L_φ define el lenguaje asociado a una oración φ

- En términos de la notación anterior tenemos que $\lambda(\varphi) = L_\varphi$

Una sintaxis efectiva para una NP

Ejercicios

1. Sea φ la siguiente oración en lógica de primer orden:

$$\exists x \forall y \, (x \neq y \to E(y, x))$$

Una sintaxis efectiva para una NP

Ejercicios

1. Sea φ la siguiente oración en lógica de primer orden:

$$\exists x \forall y \, (x \neq y \rightarrow E(y, x))$$

¿Cuál es el lenguaje L_φ?

Una sintaxis efectiva para una NP

Ejercicios

1. Sea φ la siguiente oración en lógica de primer orden:

$$\exists x \forall y \, (x \neq y \to E(y, x))$$

   ¿Cuál es el lenguaje L_φ?

2. Sea ψ la siguiente oración en lógica de segundo orden:

$$\begin{aligned}\exists R \exists G \, \big[& \forall x \, (R(x) \vee G(x)) \wedge \\ & \forall x \, (\neg R(x) \vee \neg G(x)) \wedge \\ & \forall x \forall y \, (E(x, y) \to \neg R(x) \vee \neg R(y)) \wedge \\ & \forall x \forall y \, (E(x, y) \to \neg G(x) \vee \neg G(y)) \big]\end{aligned}$$

Una sintaxis efectiva para una NP

Ejercicios

1. Sea φ la siguiente oración en lógica de primer orden:

$$\exists x \forall y \, (x \neq y \rightarrow E(y, x))$$

¿Cuál es el lenguaje L_φ?

2. Sea ψ la siguiente oración en lógica de segundo orden:

$$\begin{aligned} \exists R \exists G \, \big[& \forall x \, (R(x) \vee G(x)) \wedge \\ & \forall x \, (\neg R(x) \vee \neg G(x)) \wedge \\ & \forall x \forall y \, (E(x, y) \rightarrow \neg R(x) \vee \neg R(y)) \wedge \\ & \forall x \forall y \, (E(x, y) \rightarrow \neg G(x) \vee \neg G(y)) \big] \end{aligned}$$

¿Cuál es el lenguaje L_ψ?

Una sintaxis efectiva para una NP

Una fórmula φ en lógica de segundo orden es existencial si $\varphi = \exists R_1 \cdots \exists R_k \, \psi$, donde cada $\exists R_i$ es un cuantificador existencial de segundo orden y ψ es una fórmula de primer orden

Una sintaxis efectiva para una NP

Una fórmula φ en lógica de segundo orden es existencial si $\varphi = \exists R_1 \cdots \exists R_k \, \psi$, donde cada $\exists R_i$ es un cuantificador existencial de segundo orden y ψ es una fórmula de primer orden

Teorema (Fagin)

El conjunto de oraciones en lógica de segundo orden es una sintaxis efectiva para NP

Una sintaxis efectiva para NP basado en MTs

Considere el conjunto $\{M \mid M$ es una MT no determinista que funciona en tiempo polinomial$\}$

Una sintaxis efectiva para NP basado en MTs

Considere el conjunto $\{M \mid M$ es una MT no determinista que funciona en tiempo polinomial$\}$

¿Forma este conjunto una sintaxis efectiva para NP?

Una sintaxis efectiva para NP basado en MTs

Considere el conjunto $\{M \mid M$ es una MT no determinista que funciona en tiempo polinomial$\}$

¿Forma este conjunto una sintaxis efectiva para NP?

- No, puesto que no es decidible

Una sintaxis efectiva para NP basado en MTs

¿Cómo podemos modificar el conjunto anterior para tener una sintaxis efectiva para NP?

Una sintaxis efectiva para NP basado en MTs

¿Cómo podemos modificar el conjunto anterior para tener una sintaxis efectiva para NP?

Considere el conjunto $\{(M, p) \mid M$ es una MT no determinista y p es un polinomio$\}$

Una sintaxis efectiva para NP basado en MTs

¿Cómo podemos modificar el conjunto anterior para tener una sintaxis efectiva para NP?

Considere el conjunto $\{(M, p) \mid M$ es una MT no determinista y p es un polinomio$\}$

- (M, p) representa a la MT M restringida a ejecutar a lo más $p(|x|)$ pasos con entrada x

Una sintaxis efectiva para NP basado en MTs

¿Cómo podemos modificar el conjunto anterior para tener una sintaxis efectiva para NP?

Considere el conjunto $\{(M, p) \mid M$ es una MT no determinista y p es un polinomio$\}$

- (M, p) representa a la MT M restringida a ejecutar a lo más $p(|x|)$ pasos con entrada x

Ejercicio

Demuestre que $\{(M, p) \mid M$ es una MT no determinista y p es un polinomio$\}$ es una sintaxis efectiva para NP

Sintaxis efectivas para otras clases de complejidad

Ejercicios

1. De una sintaxis efectiva para P

Sintaxis efectivas para otras clases de complejidad

Ejercicios

1. De una sintaxis efectiva para P

2. De una sintaxis efectiva para Σ_2^P

Sintaxis efectivas para otras clases de complejidad

Ejercicios

1. De una sintaxis efectiva para P
2. De una sintaxis efectiva para Σ_2^P
   - Considere un lenguaje con tuplas de la forma (M_1, p_1, M_2, p_2)

Sintaxis efectivas para otras clases de complejidad

Ejercicios

1. De una sintaxis efectiva para P
2. De una sintaxis efectiva para Σ_2^P
   - Considere un lenguaje con tuplas de la forma (M_1, p_1, M_2, p_2)
3. De una sintaxis efectiva para $\text{EXPTIME}^{\text{NP}}$

Problemas completos basados en la sintaxis efectiva

Considere la sintaxis efectiva $\{(M, p) \mid M$ es una MT no determinista y p es un polinomio$\}$ para NP

Problemas completos basados en la sintaxis efectiva

Considere la sintaxis efectiva $\{(M, p) \mid M$ es una MT no determinista y p es un polinomio$\}$ para NP

A partir de esta sintaxis efectiva definimos el siguiente lenguaje:

$$\begin{aligned} U \ = \ \{(M, x, 1^{p(|x|)}) \mid & \ M \text{ es una MT no determinista,} \\ & \ p \text{ es un polinomio y } M \text{ con tiempo restringido a } p \text{ acepta } x\} \end{aligned}$$

Problemas completos basados en la sintaxis efectiva

Considere la sintaxis efectiva $\{(M, p) \mid M$ es una MT no determinista y p es un polinomio$\}$ para NP

A partir de esta sintaxis efectiva definimos el siguiente lenguaje:

$$\begin{aligned} U \ = \ \{(M, x, 1^{p(|x|)}) \mid\ & M \text{ es una MT no determinista,} \\ & p \text{ es un polinomio y } M \text{ con tiempo restringido a } p \text{ acepta } x\} \end{aligned}$$

Ejercicio

Demuestre que U es NP-completo

Un resultado fundamental

Teorema

Sea $\mathcal{C}$ una clase de complejidad de lenguajes decidibles que es cerrada bajo reducciones many-to-one de tiempo polinomial. Si $\mathcal{C}$ tiene un problema completo bajo este tipo de reducciones, entonces $\mathcal{C}$ tiene una sintaxis efectiva.

Un resultado fundamental

Teorema

Sea $\mathcal{C}$ una clase de complejidad de lenguajes decidibles que es cerrada bajo reducciones many-to-one de tiempo polinomial. Si $\mathcal{C}$ tiene un problema completo bajo este tipo de reducciones, entonces $\mathcal{C}$ tiene una sintaxis efectiva.

Ejercicio

Demuestre el teorema

Una extensión del resultado

Teorema

Sea $\mathcal{C}$ *una clase de complejidad de lenguajes decidibles que es cerrada bajo reducciones de Turing de tiempo polinomial. Si* $\mathcal{C}$ *tiene un problema completo bajo este tipo de reducciones, entonces* $\mathcal{C}$ *tiene una sintaxis efectiva.*

Una extensión del resultado

Teorema

Sea $\mathcal{C}$ una clase de complejidad de lenguajes decidibles que es cerrada bajo reducciones de Turing de tiempo polinomial. Si $\mathcal{C}$ tiene un problema completo bajo este tipo de reducciones, entonces $\mathcal{C}$ tiene una sintaxis efectiva.

Ejercicio

Demuestre el teorema

Clases de complejidad semánticas

Una clase de complejidad se considera semántica si no se conoce una sintaxis efectiva para ella

Clases de complejidad semánticas

Una clase de complejidad se considera semántica si no se conoce una sintaxis efectiva para ella

Ejemplo

1. ¿Es $\text{NP} \cap \text{co-NP}$ una clase de complejidad semántica?

Clases de complejidad semánticas

Una clase de complejidad se considera semántica si no se conoce una sintaxis efectiva para ella

Ejemplo

1. ¿Es NP $\cap$ co-NP una clase de complejidad semántica? Sí, se considera semántica

Clases de complejidad semánticas

Una clase de complejidad se considera semántica si no se conoce una sintaxis efectiva para ella

Ejemplo

1. ¿Es NP $\cap$ co-NP una clase de complejidad semántica? Sí, se considera semántica
2. ¿Es BPP una clase de complejidad semántica?

Clases de complejidad semánticas

Una clase de complejidad se considera semántica si no se conoce una sintaxis efectiva para ella

Ejemplo

1. ¿Es NP $\cap$ co-NP una clase de complejidad semántica? Sí, se considera semántica
2. ¿Es BPP una clase de complejidad semántica? Sí, también se considera semántica

Dos consecuencias fundamentales

No esperamos que NP $\cap$ co-NP tenga problemas completos

Dos consecuencias fundamentales

No esperamos que NP $\cap$ co-NP tenga problemas completos

Tampoco esperamos que BPP tenga problemas completos

Dos consecuencias fundamentales

No esperamos que $\mathsf{NP} \cap \text{co-}\mathsf{NP}$ tenga problemas completos

- Ni bajo reducciones many-to-one ni bajo reducciones de Turing

Tampoco esperamos que BPP tenga problemas completos

- Ni bajo reducciones many-to-one ni bajo reducciones de Turing