Verificación del μ-calculus
IIC3800

Pablo Barceló

Complejidad de verificación del μ-calculus

Recordemos que el model checking es el siguiente problema: Dado un sistema de transición $\mathcal{M}$, un estado e, y una fórmula ϕ en el μ-calculus, ¿es cierto que $(\mathcal{M}, e) \models \phi$?

La complejidad del model checking del μ-calculus está en $\mathrm{NP} \cap \mathrm{coNP}$.

De hecho, está en NP porque podemos adivinar el valor de los mayores puntos fijos (y verificar que son puntos fijos), y luego computar los menores puntos fijos mediante iteraciones.

Está además en coNP porque la lógica es cerrada bajo negación.

Complejidad de verificación del μ-calculus

Esta cota no es buena, pero no implica inmediatamente que sea imposible efectuar la verificación de especificaciones en esta lógica:

- Ejemplo: Aunque el model checking de LTL es NP-hard, muchas de las aplicaciones de verificación utilizan LTL como lenguaje de especificación.

Complejidad de verificación del μ-calculus

Esta cota no es buena, pero no implica inmediatamente que sea imposible efectuar la verificación de especificaciones en esta lógica:

- Ejemplo: Aunque el model checking de LTL es NP-hard, muchas de las aplicaciones de verificación utilizan LTL como lenguaje de especificación.

Pregunta: ¿Cuáles son las condiciones ideales bajo las cuales es posible realizar la verificación de un lenguaje de especificación?

Complejidad de verificación del μ-calculus

Esta cota no es buena, pero no implica inmediatamente que sea imposible efectuar la verificación de especificaciones en esta lógica:

- Ejemplo: Aunque el model checking de LTL es NP-hard, muchas de las aplicaciones de verificación utilizan LTL como lenguaje de especificación.

Pregunta: ¿Cuáles son las condiciones ideales bajo las cuales es posible realizar la verificación de un lenguaje de especificación?

- Tener un algoritmo que tome tiempo a lo más lineal en el tamaño de $\mathcal{M}$ y exponencial en el tamaño de ϕ.

Complejidad de verificación del μ-calculus

Esta cota no es buena, pero no implica inmediatamente que sea imposible efectuar la verificación de especificaciones en esta lógica:

- Ejemplo: Aunque el model checking de LTL es NP-hard, muchas de las aplicaciones de verificación utilizan LTL como lenguaje de especificación.

Pregunta: ¿Cuáles son las condiciones ideales bajo las cuales es posible realizar la verificación de un lenguaje de especificación?

- Tener un algoritmo que tome tiempo a lo más lineal en el tamaño de $\mathcal{M}$ y exponencial en el tamaño de ϕ.

Analizaremos si eso es posible para el caso del μ-calculus.

Algoritmo (ingenuo) de verificación del μ-calculus

Ocupemos el algoritmo más trivial que existe, es decir el de aplicación directa de la semántica (sólo mostraremos el caso en que la fórmula es de la forma $\mu Y.\phi(Y)$):

function $\mathrm{Evaluate}(\mu Y.\phi(Y), \sigma)$
    $Q_{val} := \emptyset$;
    repeat
        $Q_{old} := Q_{val}$;
        $Q_{val} := \mathrm{Evaluate}(\phi, \sigma[Y \rightarrow Q_{val}])$;
    until $Q_{val} = Q_{old}$;
    return Q_{val};
end function

Algoritmo (ingenuo) de verificación del μ-calculus

Pregunta: ¿Cuál es la complejidad de este algoritmo en términos de $|\mathcal{M}|$ y $|\psi|$?

Algoritmo (ingenuo) de verificación del μ-calculus

Pregunta: ¿Cuál es la complejidad de este algoritmo en términos de $|\mathcal{M}|$ y $|\psi|$?

- Cada punto fijo requiere computar $O(|E|)$ iteraciones.

Algoritmo (ingenuo) de verificación del μ-calculus

Pregunta: ¿Cuál es la complejidad de este algoritmo en términos de $|\mathcal{M}|$ y $|\psi|$?

- Cada punto fijo requiere computar $O(|E|)$ iteraciones.
- Cada iteración toma tiempo $O(|M| \cdot |\psi|)$.

Algoritmo (ingenuo) de verificación del μ-calculus

Pregunta: ¿Cuál es la complejidad de este algoritmo en términos de $|\mathcal{M}|$ y $|\psi|$?

- Cada punto fijo requiere computar $O(|E|)$ iteraciones.
- Cada iteración toma tiempo $O(|M| \cdot |\psi|)$.
- La fórmula del cuerpo del operador de punto fijo más interno es llamada $O(|E|^k)$ veces, donde k es la profundidad de anidación de operadores μ y ν en ψ.

Algoritmo (ingenuo) de verificación del μ-calculus

Pregunta: ¿Cuál es la complejidad de este algoritmo en términos de $|\mathcal{M}|$ y $|\psi|$?

- Cada punto fijo requiere computar $O(|E|)$ iteraciones.
- Cada iteración toma tiempo $O(|M| \cdot |\psi|)$.
- La fórmula del cuerpo del operador de punto fijo más interno es llamada $O(|E|^k)$ veces, donde k es la profundidad de anidación de operadores μ y ν en ψ.
- Todo el algoritmo toma tiempo $O(|M| \cdot |\psi| \cdot |E|^k)$.

Algoritmo (ingenuo) de verificación del μ-calculus

Pregunta: ¿Cuál es la complejidad de este algoritmo en términos de $|\mathcal{M}|$ y $|\psi|$?

- Cada punto fijo requiere computar $O(|E|)$ iteraciones.
- Cada iteración toma tiempo $O(|M| \cdot |\psi|)$.
- La fórmula del cuerpo del operador de punto fijo más interno es llamada $O(|E|^k)$ veces, donde k es la profundidad de anidación de operadores μ y ν en ψ.
- Todo el algoritmo toma tiempo $O(|M| \cdot |\psi| \cdot |E|^k)$.

Esto no es ideal: En el mejor caso (profundidad de anidación $= 1$) el algoritmo es cuadrático en $|\mathcal{M}|$.

Algoritmo (más inteligente) de verificación del μ-calculus

Un mejor algoritmo fue presentado por Emerson y Lei en 1986. Su complejidad es $O((|\psi| \cdot |\mathcal{M}|)^a)$, donde a es la alternación de ψ (asumimos que ψ sólo tiene negaciones aplicadas a proposiciones y variables).

Aunque ésto no es ideal, si nos da una muy buena cota para fórmulas con poca alternación (de hecho, es lineal para fórmulas sin alternación).

Algoritmo (más inteligente) de verificación del μ-calculus

La idea aquí es tratar de procesar en una sola iteración cada fórmula ψ de la forma

$$\mu Y_1.\phi_1(Y_1, \mu Y_2.\phi_2(Y_1, Y_2, \mu Y_3.\phi_3(\ldots, \mu Y_n.\phi_n(Y_1, \ldots, Y_n)))) \ldots),$$

o, equivalentemente, con μ reemplazado por ν.

La idea es que para computar la i-ésima aproximación Y_j^i al valor final de Y_j, para $j \geq 2$, no es necesario empezar en $\emptyset$.

Problema abierto

Dado lo que sabemos hasta ahora, no es posible contestar la pregunta de si el model checking del μ-calculus se puede realizar en tiempo $O(|2^{\phi}| \cdot |\mathcal{M}|)$.

Una pregunta abierta más interesante es si el model checking del μ-calculus está en PTIME (ésto se cree que es el caso para todos los problemas en $\mathrm{NP} \cap \mathrm{coNP}$).