Parameterisierte Algorithmen

WS 2007/08 in Trier

Henning Fernau
fernau@uni-trier.de

Parameterisierte Algorithmen | Gesamtübersicht

- Einführung
- Grundbegriffe
- Problemkerne
- Suchbäume
- Graphparameter
- Weitere Methoden
- Komplexitätstheorie—parameterisiert

Themen

- parameterisierte Reduktionen
- Komplexitätsklassen
- Vollständigkeitsbeweise

Reduktionen

Es seien (P, k) und (P', k') parameterisierte Probleme (formal gegeben als Sprachen über den Alphabetn Σ bzw. Σ').
Eine *FPT*-(many-one)-Reduktion von (P, k) auf (P', k') ist eine Abbildung $R : (\Sigma^* \times \mathbb{N}) \to ((\Sigma')^* \times \mathbb{N})$ mit:
(1) $\forall (x, k) \in \Sigma^* \times \mathbb{N} : ((x, k) \in L(P) \iff R(x, k) \in L(P'))$
(2) R ist in Zeit $f(k) \cdot p(|x|)$ berechenbar für bel. f und Polynom p.
(3) Es gibt Funktion $g : \mathbb{N} \to \mathbb{N}$, sodass $\forall k \in \mathbb{N} : k' \leq g(k)$ für alle x, sodass $R(x, k) = (x', k')$.
Schreibweise: $(P, k) \leq^{\mathrm{FPT}} (P', k')$.

FPT und parameterisierte Reduktionen

Satz 1 *FPT ist abgeschlossen unter FPT-Reduktionen, d.h., gilt* $(P, k) \leq^{FPT} (P', k')$ *und* $(P', k') \in FPT$*, so auch* $(P, k) \in FPT$*.*

Beweis: Betrachte Reduktion R, die in Zeit $f(k) \cdot p(|x|)$ eine Instanz (x, k) von (P, k) auf eine Instanz (x', k') von (P', k') transformiert, wobei $k' \leq g(k)$. Die Instanz (x', k') kann in Zeit $h(k') \cdot q(|x'|)$ gelöst werden. Die Gesamtlaufzeit dieses Algorithmus ist:

$$f(k) \cdot p(|x|) + h(k') \cdot q(|x'|) \leq f(k) \cdot p(|x|) + h(g(k)) \cdot q(f(k) \cdot p(|x|)) \leq f'(k) \cdot p'(|x|)$$

FPT und parameterisierte Reduktionen

Lemma 2 *Die Relation* $\leq^{\mathrm{FPT}}$ *ist eine Quasiordnung, also reflexiv und transitiv.*

Es ist also sinnvoll, den Begriff der *FPT*-Äquivalenz $\equiv^{\mathrm{FPT}}$ einzuführen.

Lemma 3 *Ist* (P, k) *irgendein nicht-triviales Problem aus FPT und ist* (P', k') *ein nicht-triviales parameterisiertes Problem, so gilt:*
$(P', k') \in \mathrm{FPT}$ *gdw.* $(P, k) \equiv^{\mathrm{FPT}} (P', k')$.

Der eingeführte Begriff der *FPT*-Reduktion scheint also sinnvoll zu sein.

Beispiele für parameterisierte Reduktionen I

Lemma 4 *Das Problem, Cliquen der Mindestgröße k in Graphen zu finden, ist FPT-äquivalent zu dem Problem, unabhängige Mengen der Mindestgröße k in Graphen zu finden.*

Das Graph-Komplement $\bar{G}$ von $G = (V, E)$ ist definiert durch:

$$\bar{G} = (V, \bar{E}) \quad \text{mit} \quad vw \in \bar{E} \iff (v \neq w \wedge vw \notin E)$$

Die Reduktion $(G, k) \mapsto (\bar{G}, k)$ leistet das Gewünschte (beide Richtungen).

Hinweis: Die Abbildung, die einer Instanz ihr parameterisiertes Dual zuordnet, ist i.d.R. keine parameterisierte Reduktion. Das gilt insbesondere für den bekannten Zusammenhang zwischen VERTEX COVER und INDEPENDENT SET.

Beispiele für parameterisierte Reduktionen II

Lemma 5 *Das Problem, dominierende Mengen der Höchstgröße k in Graphen zu finden, ist FPT-äquivalent zu dem Problem, eine Knotenüberdeckung der Höchstgröße k in einem Hypergraphen zu finden.*

Beweis: Statt eine dominierende Menge $D \subseteq V$ in einem Graphen $G = (V, E)$ zu suchen, kann man auch nach einer Knotenüberdeckung in dem Hypergraphen $H = (V, \{N_G[v] \mid v \in V\})$ fahnden.
Statt nach einer Knotenüberdeckung $C \subseteq V$ im Hypergraphen $H = (V, E)$ zu suchen, kann man auch nach einer dominierenden Menge im Graphen $G = (V \cup E, E_1 \cup E_2)$ fahnden mit $E_1 = \{ve \mid v \in V, e \in E, v \in e\}$ und $E_2 = \{uv \mid u, v \in V, u \neq v\}$.
Beobachte:
(1) Eine dominierende Menge D mit $D \cap E \neq \emptyset$ kann man ersetzen durch $D' = (D \cap V) \cup \{v_e \mid \exists e \in D\}$ mit $|D'| \leq |D|$, wobei v_e ein willkürlich aber festes $v \in e$ meint.
(2) Eine dominierende Menge $D \subseteq V$ in G ist auch eine Knotenüberdeckung in H.

Das Halteproblem auf Turing-Maschinen

ist einer der natürlichen Kandidaten, um “harte Probleme” zu definieren.
In der “klassischen Berechenbarkeit” ist das allgemeine Halteproblem das typische Beispiel für ein unentscheidbares Problem.
Schränkt man die Schrittzahl, die eine Turing-Maschine machen darf, geeignet ein, so gelangt man in der “klassischen Komplexitätstheorie” zu typischen *NP*-harten Problemen.
Man kann hierbei sowohl Ein- als auch Mehrband-Turing-Maschinen betrachten.
Die einfachste Variante ist vielleicht:
KURZE NTM-AKZEPTANZ

Eingabe: Einband-NTM M, Eingabe x
Parameter: $k \in \mathbb{N}$
Frage: Gibt es Berechnung von M auf x, die in $\leq k$ Schritten akzeptiert?

Das Halteproblem auf Turing-Maschinen: Glaubensfragen

Aufgabe: Wie kann man eine nichtdeterministische TM durch eine deterministische simulieren ?
Am einfachsten geht das durch systematisches Durchprobieren aller möglichen Verzweigungen.
$\rightsquigarrow$ Für das Problem NTM-Akzeptanz führt das auf eine (deterministische) Laufzeit von n^k für ein von M abhängiges n.
Jedenfalls "auf diese Weise" liegt das Problem nicht in *FPT*.
Eine ähnliche Intuition, die einen dazu verleitet anzunehmen, dass KURZE NTM-AKZEPTANZ nicht in Polynomzeit (auf deterministischen Maschinen) gelöst werden kann, führt einen auch auf den Gedanken, das "geht nicht" in *FPT*-Zeit.

$\rightsquigarrow$ W[1]: Klasse von parameterisierten Problemen (P, k) mit
$(P, k) \leq^{FPT}$ KURZE NTM-AKZEPTANZ.

Das Halteproblem auf Turing-Maschinen: Glaubensfragen

Wenn wir nicht glauben, dass KURZE NTM-AKZEPTANZ in *FPT* liegt, heißt das:

(1) Wir vermuten, dass es Probleme gibt, die in W[1] liegen, aber nicht in *FPT*.
Ein Kandidat dafür ist KURZE NTM-AKZEPTANZ.
(2) Da *FPT* unter *FPT*-Reduktionen abgeschlossen ist, liegt es nahe, Folgendes zu definieren:
(P, k) W[1]-hart: gdw. KURZE NTM-AKZEPTANZ $\leq^{FPT} (P, k)$.
(P, k) W[1]-vollständig: gdw. KURZE NTM-AKZEPTANZ $\equiv^{FPT} (P, k)$.

Vergleichen Sie mit den Ihnen bekannten “klassischen Komplexitätsbegriffen” !
Entsprechende Sprechweisen verwenden wir für die anderen noch einzuführenden parameterisierten Komplexitätsklassen.

Ein Beispiel für ein W[1]-Problem

Klar: $(P, k) \in \text{FPT} \implies (P, k) \in W[1]$.
Wir erwähnten schon: UNABHÄNGIGE MENGEN ist "wohl nicht" in *FPT*.

Lemma 6 UNABHÄNGIGE MENGEN *liegt in W[1].*

Unsere Reduktion speichert die Adjazenzrelation von G in der Übergangstabelle von M_G.
Dabei genehmigt sich die Reduktion für jeden Knoten ein Extra-Eingabezeichen.
M_G rät in den ersten k Schritten k Knoten und schreibt sie auf sein Arbeitsband.
In den folgenden $\mathcal{O}(k^2)$ Schritten [bitte ausführen!] überprüft M_G (deterministisch!), ob die geratene Knotenmenge eine unabhängige Menge bildet.
Es gilt: G enthält unabhängige Menge der Größe k gdw. M_G akzeptiert das leere Wort in $f(k) \in \mathcal{O}(k^2)$ Schritten.
Mitteilung: UNABHÄNGIGE MENGEN ist W[1]-vollständig.
Folgerung: CLIQUE ist W[1]-vollständig.

Dominierende Mengen—Philosophisches zu W[1]

Um Mitgliedschaft in W[1] zu beweisen, kann man allgemein folgende Art von Algorithmen / Turingmaschinen bauen:

(1) Es werden $\mathcal{O}(f(k))$ "Objekte" geraten und aufs Arbeitsband geschrieben.

Hierbei ist zu beachten, dass jedes dieser Objekte nur $\mathcal{O}(1)$ Speicherplatz benötigt (oder vielleicht $\mathcal{O}(g(k))$), damit die Zeit für das Aufschreiben der Rateergebnisse (und auch die für das spätere Lesen) nicht zu lange dauert. Dazu sind geeignete Kodierungen zu wählen.

(2) Die "Richtigkeit" der geratenen Objekte wird in Zeit $\mathcal{O}(h(k))$ überprüft.

Hierbei ist wieder wichtig, dass die Turingmaschine geeignete Informationen in ihrem endlichen Speicher hat, die das Überprüfen eines einzelnen Objektes (oder einer endlichen Menge der geratenen Objekte) in Zeit $\mathcal{O}(1)$ (oder vielleicht $\mathcal{O}(j(k))$) gestattet.

Frage: Was geht "schief", will man diesen Ansatz dazu benutzen, um einen W[1]-Algorithmus für DOMINATING SET zu entwickeln ?

Ein schwierigeres Halteproblem auf Turing-Maschinen

KURZE NTM-MEHRBAND-AKZEPTANZ

Eingabe: Mehrband-NTM M, Eingabe x

Parameter: $k \in \mathbb{N}$

Frage: Gibt es Berechnung von M auf x, die in $\leq k$ Schritten akzeptiert?

$\leadsto$ W[2]: Klasse von parameterisierten Problemen (P, k) mit

$$(P, k) \leq^{\text{FPT}} \text{KURZE NTM-MEHRBAND-AKZEPTANZ}.$$

Intuition: Mehrband-Akzeptanz ist "schwieriger" als Einband-Akzeptanz, da wir $\mathcal{O}(p(n))$ viele Bänder parallel benutzen können; wie diese verwendet werden, beschreibt die Reduktion.

Ein schwierigeres Halteproblem auf Turing-Maschinen

Lemma 7 DOMINIERENDE MENGEN *liegt in W[2].*

Für jeden Knoten v des Graphen G gibt es ein Hilfsband $h[v]$.

Auf das (weitere) Arbeitsband raten wir k Knoten.

In einem Schritt schreiben wir $*$ auf jedes Hilfsband.

Für alle geratenen Knoten u schreiben wir # auf jedes Hilfsband $h[v]$ mit $v \in N_G[u]$.

Die Maschine akzeptiert schließlich, falls das rechteste Zeichen auf jedem Hilfsband ein # ist (das kann man mit einem "Linksschritt" auf allen Hilfsbändern prüfen).

Mitteilung: DOMINIERENDE MENGEN ist W[2]-vollständig.
Folgerung: HITTING SET ist W[2]-vollständig.

Ein noch schwierigeres Halteproblem auf Turing-Maschinen

NICHTDETERMINISMUSBESCHRÄNKTE NTM-AKZEPTANZ

Eingabe: Einband-NTM M, Eingabe x, unärkodierte Zahl n

Parameter: $k \in \mathbb{N}$

Frage: Gibt es Berechnung von M auf x, die in $\leq n$ Schritten akzeptiert und dabei $\leq k$ nichtdeterministische Schritte macht?

$\rightsquigarrow$ W[P]: Klasse von parameterisierten Problemen (P, k) mit

$$(P, k) \leq^{FPT} \text{NICHTDETERMINISMUSBESCHRÄNKTE NTM-AKZEPTANZ}.$$

Satz 8 $W[2] \subseteq W[P]$.

"Klar", wie man DOMINIERENDE MENGEN mit diesem Formalismus löst:

Das "Einband-Problem" war ja nur im Verifikationsteil.

Mitteilung: Es gibt eine ganze W-Hierarchie:

$$FPT = W[0] \subseteq W[1] \subseteq W[2] \subseteq W[3] \subseteq \cdots \subseteq W[P].$$

Zwei Verallgemeinerungen von W[P]

(1) XP: die mit folgendem Modell lösbaren parameterierten Probleme:
EXP-DTM-HALTEPROBLEM

Eingabe: Deterministische TM M, n unärkodiert, Eingabe x
Parameter: $k \in \mathbb{N}$
Frage: Akzeptiert M die Eingabe x in höchstens n^k Schritten ?

(2) para-*NP*: die mit nichtdeterministischen Algorithmen "in *FPT*-Zeit" lösbaren parameterisierten Probleme.

Mitteilung: $W[P] \subseteq (XP \cap \text{para} - NP)$.

Hart oder weich ?

Es ist oft schwierig, Problemen an der “Nasenspitze” anzusehen, ob sie in *FPT* liegen oder aber W[1]-hart sind.

MERKMALSMENGE (FEATURE SET)
Eingabe: Merkmalsmenge $F = \{1, \ldots, n\}$, Messreihe $X = \{x_1, \ldots, x_m\}$ mit $x_i : F \to \{0, 1\}$, Zielfunktion $t : X \to \{0, 1\}$
Parameter: $k \in \mathbb{N}$
Frage: Gibt es Merkmalsteilmenge $M \subseteq F$, $|M| \leq k$ mit

$$\forall x, y \in X : (x \neq y \wedge t(x) \neq t(y)) \implies \exists f \in M : x(f) \neq y(f) \quad ?$$

Hart oder weich ?

MERKMALSMENGE (FEATURE SET)
Eingabe: Merkmalsmenge $F = \{1, \ldots, n\}$, Messreihe $X = \{x_1, \ldots, x_m\}$ mit $x_i : F \to \{0, 1\}$, Zielfunktion $t : X \to \{0, 1\}$
Parameter: $k \in \mathbb{N}$
Frage: Gibt es Merkmalsteilmenge $M \subseteq F$, $|M| \leq k$ mit

$$\forall x, y \in X : (x \neq y \wedge t(x) \neq t(y)) \implies \exists f \in M : x(f) \neq y(f) \quad ?$$

Satz 9 MERKMALSMENGE *ist W[2]-hart.*

Beweis: Wir zeigen, wie man DOMINATING SET mit FEATURE SET lösen könnte. Sei $G = (V, E)$ vorgelegt, mit $V = \{v_1, \ldots, v_n\}$. Bilde $X = \{x_0, \ldots, x_n\}$ und $F = \{1, \ldots, n\}$ mit $x_0(f) = 0$ für alle $f \in F$, $t(x_0) = 0$; und $x_i(j) = 1 \iff v_i \in N[v_j]$ sowie $t(x_i) = 1$ für $1 \leq i \leq n$. $D = \{x_{i_j} \mid 1 \leq j \leq k\}$ ist Dominierungsmenge von G gdw. $M = \{i_j \mid 1 \leq j \leq k\}$ ist zulässige Merkmalsmenge von der FEATURE SET Instanz.

Hart oder weich ?

MERKMALSMENGE (FEATURE SET)
Eingabe: Merkmalsmenge $F = \{1, \ldots, n\}$, Messreihe $X = \{x_1, \ldots, x_m\}$ mit $x_i : F \to \{0, 1\}$, Zielfunktion $t : X \to \{0, 1\}$
Parameter: $k \in \mathbb{N}$
Frage: Gibt es Merkmalsteilmenge $M \subseteq F$, $|M| \leq k$ mit

$$\forall x, y \in X : (x \neq y \wedge t(x) \neq t(y)) \implies \exists f \in M : x(f) \neq y(f) \quad ?$$

TESTMENGE (TEST SET)
Eingabe: Menge möglicher Tests $F = \{1, \ldots, n\}$, Objekte $X = \{x_1, \ldots, x_m\}$ mit $x_i : F \to \{0, 1\}$
Parameter: $k \in \mathbb{N}$
Frage: Gibt es Testmenge $M \subseteq F$, $|M| \leq k$ mit

$$\forall x, y \in X : x \neq y \implies \exists f \in M : x(f) \neq y(f) \quad ?$$

Hart oder weich ?

TESTMENGE (TEST SET)
Eingabe: Menge möglicher Tests $F = \{1, \ldots, n\}$, Objekte $X = \{x_1, \ldots, x_m\}$ mit $x_i : F \to \{0, 1\}$
Parameter: $k \in \mathbb{N}$
Frage: Gibt es Testmenge $M \subseteq F$, $|M| \leq k$ mit

$$\forall x, y \in X : x \neq y \implies \exists f \in M : x(f) \neq y(f) \quad ?$$

Satz 10 TESTMENGE *ist in FPT.*

Beweis: k Tests können maximal 2^k Objekte trennen. Daher gibt es höchstens 2^{2^k} Tests. Bei mehr als 2^k Objekten haben wir also eine NEIN-Instanz, andernfalls einen “kleinen Kern”.

Rückblick

Vorlesung fokussiert auf Techniken für *FPT*-Algorithmen.
Es gib aber auch “die böse Welt”: eine Härte-Theorie analog zur klassischen Komplexitätstheorie.
Längst nicht alle parameterisierten Probleme liegen in *FPT*.
Selbst wenn sie in *FPT* liegen, so gibt es manchmal nur sehr unschöne parameterisierte Algorithmen.
Solche “bösen Beispiele” umfassen “defensive Allianzen” und “Testmengen”.
Auch die $\mathcal{O}$-Notation kann “Böses” verbergen, wie die aus dem Beweis der Wagnerschen Vermutung sich ergebenen “Algorithmen” zeigen.
Insbesondere Suchbaum- und Problemkern-Algorithmen verbergen aber nichts Schlimmes, selbst in der $\mathcal{O}^*$-Notation.

Rückblick: Wahl der Technik

Manchmal hat man ein “Gefühl”, ob ein Problem W[1]-hart ist oder nicht.
Mitgliedschaft in *FPT* beweist man oft schnell durch Aufzeigen eines Problemkerns oder durch die Wohlquasiordnungsmethode.
Sofern man so keinen linearen Kern gefunden hat, sind derartige *FPT*-Algorithmen aber meist von theoretischem Charakter.
Allerdings ist eine derart gefundene Sammlung von Reduktionsregeln zumindest heuristisch wertvoll.
Suchbaum-Methoden liefern hingegen “meist” $\mathcal{O}^*(c^k)$-Verfahren, ebenso dynamisches Mengenprogrammieren, iteratives Verbessern oder Farbkodieren.
Bei Suchbäumen sind die Basen meist am kleinsten.
“Wirklich gut” sind wohl kombinierte Verfahren.
“Moderne Suchbaumanalysen” wählen oft den “Measure&Conquer”-Ansatz;
das wäre eine eigene Vorlesung wert gewesen. . .

Nachweise

Die parameterisierte Komplexitätstheorie in all ihren Verästelungen finden Sie ausführlich in den Büchern von Downey/Fellows und Flum/Grohe dargelegt und kürzer im 13. Kapitel des Buches von Niedermeier.

Der hier beschrittene “Turing-Weg” zur parameterisierten Komplexität wurde von M. Cesati in der nachstehenden Arbeit vorgeschlagen: “The Turing way to parameterized complexity”, *Journal of Computer and System Sciences*, 67:654-685, 2003.
Ergänzungen hierzu finden Sie im Buch von Flum und Grohe.
Im gleichen JCSS-Band finden Sie auch eine Arbeit von C. Cotta und P. Moscato über die W[2]-Vollständigkeit von FEATURE SET.