ChatGPT

TypeScript型注釈自動生成・進化の先行研究

TypeScript型注釈（型定義ファイル）の自動生成・メンテナンスに関し、Kristensen & Møller (2017) はツール**tsinfer**と**tsevolve**を提案している [1]。著者らによれば、tsinfer/tsevolve は「新しい型定義ファイルの構築と、ライブラリ進化に伴う型定義の共進化を支援する」ツールであり、実験では「tsinfer と tsevolve は tscheck（既存ツール）に比べ高速かつ精度良く動作する」と報告されている [1]。以下、それぞれの仕組みと特長を概説する。

tsinfer: 型定義ファイル推論ツールの概要と限界

tsinferは、JavaScriptライブラリの動作から型を推論して.ts定義を生成するツールである。具体的には、ライブラリの初期化をブラウザで実行し「**その結果の状態のスナップショットを記録**」し、次にスナップショット中のすべての関数・オブジェクトに対して「**軽量な静的解析**」を行う [2]。この解析により、各関数の引数型・戻り値型やクラス・フィールド構造を推定し、TypeScriptの型宣言(.d.ts)を出力する仕組みである [2]。Kristensenらは、既存のTSCheckが関数の戻り値型のみチェックしてそれ以外を「正しい」と仮定するのに対し、tsinferは「すべての関数（パラメータや戻り値）、さらにクラスやフィールドの型を推論する」と強調している [3]。このようにtsinferは静的解析に基づくルールベース手法であり、機械学習などの確率的・統計的手法は用いていない。

しかしながら、tsinferにはいくつか限界もある。実行時スナップショットにはライブラリ内部の非公開メソッドやプライベートなプロパティも含まれてしまい、開発者が意図しないAPIを型定義に含めてしまう可能性がある [4]。また、静的解析の制約・抽象化には改善の余地があると指摘されており、より正確な型推論には「ランタイムベースのアプローチ」の方が有効であるとの見解も示されている [5]。実験では主要ライブラリを対象にtsinferの高速性・精度が示された一方で、生成された型定義ファイルの**品質評価やコンパイル可能性検証**は議論されておらず、あくまで人手によるレビューや既存宣言との差分評価に依存している [5] [3]。

tsevolve: 型定義ファイルの進化支援ツール

tsevolveは、ライブラリのバージョンアップに伴う型定義ファイルの更新を支援するツールである。古いバージョンの実装（old.js）・新しいバージョン（new.js）、そして古いバージョン用の型定義ファイル(old.d.ts)を入力とし、差分から更新すべき型宣言を抽出する [6]。具体的には両バージョンの実装に対してtsinferを適用して型定義を生成し、**ツリー構造として両型定義を比較**する。相違があった箇所を変更候補とし、さらに旧型定義に未定義の要素に関する差分を除去するフィルタリングを行うことで、無駄な警告を低減する仕組みである [6] [7]。出力には各変更点に対応する実装コードが付加されており、解析精度には限界があるため「必要に応じて手動で確認・修正する」運用が想定されている [8]。

評価では、tsevolveの出力リストに含まれる変更点を手動で分類した結果、**True Positive (TP)**（実際に型定義に反映すべき変更）が大半を占め、誤警告(**False Positive**)も多くは出力上で簡単に識別可能であったと報告されている [9] [10]。Kristensenらは「TSevolveの出力は、対応すべき変更を主に指摘している」と結論付けており、実際にその変更案を使ってPRを作成し多くが受け入れられたことも報告している [10]。ただしこちらも、あくまで型定義の差分検出支援に特化しており、**型定義全体の品質指標**や型安全性の保証、実際のコンパイル成功検証までは扱っていない。

未対応の課題と本研究との相違点

以上の先行ツールには、主に「静的・差分解析による型推論」に特化するゆえの未解決点が存在する。例えば、生成・更新した型定義ファイルが**TypeScriptコンパイラを通過するか**（型チェックでエラーが出ないか）については検証機能がなく、**品質評価指標**（型の網羅性・精度など）も提示されていない。また、最新の研究動向として、Transformer系モデルなどを用いた大規模言語モデル(LLM)ベースの型推論が注目されている点も挙げられる。実際、CodeTIDAL5 のようなTransformerモデルはTypeScript型注釈の予測で約71.3%の精度を達成しており [11]、コードの使用例から確率的に型を補完する手法として効果を示している。一方でtsinfer/tsevolveはこのような機械学習・LLMベースのアプローチを用いておらず、あくまでルールベース解析に依拠している。また、既存プロジェクトのTypeScript移行を自動支援する観点では、**型定義の自動サニタイズ**（未使用型の削除や矛盾解消）や、移行後のコードが正常に型チェックを通るか(移行成功率)といった問題も扱われていない。

本研究では、これらの先行研究の限界を踏まえた上で新規性を打ち出す。具体的には、生成された型定義の**コンパイル可否検証**や**品質定量評価**を導入し、さらにLLMによる型補完や動的情報を活用して型推論精度を高めることを目指す。これにより、tsinfer/tsevolveが手動確認や部分的な解析に頼らざるを得なかった課題を克服し、より実用性・信頼性の高い型注釈自動生成・移行支援を実現する予定である。

参考文献: Kristensen and Møller (2017) [1] [2] [6]、および関連研究 [11] [9]。

---

[1] Inference and Evolution of TypeScript Declaration Files | Springer Nature Link (formerly SpringerLink)
https://link.springer.com/chapter/10.1007/978-3-662-54494-5_6

[2] [4] [5] Generation of TypeScript Declaration Files from JavaScript Code
https://arxiv.org/pdf/2108.08027

[3] [6] [7] [8] [9] [10] Automated Techniques for Creation and Maintenance of TypeScript Declaration Files
https://pure.au.dk/ws/files/161232639/thesis.pdf

[11] [2310.00673] Learning Type Inference for Enhanced Dataflow Analysis
https://arxiv.org/abs/2310.00673