

Documentation du code de Hashmap

Romain PRETET¹ and Camille VEDANI²

1 Université III Paul Sabatier, France, romain.pretet@univ-tlse3.fr

2 Université III Paul Sabatier, France, camille.vedani@univ-tlse3.fr



Résumé

Ce document correspond à la documentation du code, des fichiers `LocalBBXddTimeProcessor.cpp` et `LocalBBXddTimeProcessor.hpp`

Lien pour modifier le document : <https://fr.overleaf.com/2337163131nkmgwkbwwtcs#24145d>

Lien pour consulter les fichiers du code : https://github.com/camved/TER2025_XDD/tree/main/Code

1 Introduction

Le processeur `LocalBBXddTimeProcessor` calcule le temps d'exécution pire cas (WCET) des blocs de base (BB) en utilisant des XDD (eXtended Decision Diagrams). Il fait partie du framework OTAWA pour l'analyse de temps d'exécution.

2 Fonctionnalités

2.1 Caractéristiques principales

- Calcul des temps locaux des blocs de base
- Support du parallélisme (optionnel)
- Optimisation via découpage des blocs d'instructions

2.2 Features

LOCAL_BBTIMES_FEATURE Garantit le calcul des temps locaux

BBTIMES_FEATURE Stocke les temps relatifs aux prédécesseurs

2.3 Propriétés

BBTIMES Stocke le temps XDD d'un BB (attaché aux arrêtes)

SPLIT_THRESHOLD Seuil pour le découpage des blocs (défaut : 12)



© Romain PRETET, Camille VEDANI;
sous licence Creative Commons CC-BY

3 Implémentation

3.1 Structure de classe

```
class LocalBBXddTimeProcessor : public BBProcessor {
    // Configuration
    void setup(WorkSpace *ws);
    void destroy(WorkSpace *ws);

    // Traitement principal
    void processBB(WorkSpace *ws, CFG *cfg, Block *b);
    void processEdge(Edge *e);
    void computeTime(Edge *e, const Vector<...>& xsteps,
                    XddTimingState& state);

    // Statistiques
    void dumpGlobalStats(io::Output &out);
};
```

3.2 Méthodes clés

3.2.1 processEdge

Calcule le temps pour une arrête donnée :

1. Collecte les étapes d'exécution (xsteps)
2. Applique les splits si nécessaire
3. Calcule le temps final

3.2.2 computeTime

Implémente le calcul effectif :

- Compile les xsteps en matrice XDD
- Applique la matrice à l'état courant
- Met à jour le temps dans BBTIMES

4 Exemple d'utilisation

```
// Configuration
WorkSpace *ws = ...;
LocalBBXddTimeProcessor processor;
Processor::run(ws, processor);

// Acces aux resultats
Edge *e = ...;
Xdd bb_time = BBTIMES(e);
cout << "WCET:␣" << bb_time << endl;
```

5 Optimisations

- Découpage des blocs via `SPLIT_THRESHOLD`
- Calcul parallèle (si `XDD_PARA` défini)
- Minimisation des XDD pour performance