Notes : XDD

Romain PRETET[1] and Camille VEDANI[2]

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 relève des prises de notes durant la lecture des articles donnés par Monsieur Cassé. Le but de cela est d'accéder rapidement aux propriétés des XDD pour les optimiser.

Introduction

Les **eXecution Decision Diagrams (XDD)** sont une structure de données utilisée pour représenter de manière compacte les temps d'exécution d'une séquence d'instructions en présence de latences variables. Ils sont particulièrement utiles dans l'analyse du **Worst-Case Execution Time (WCET)** pour les systèmes temps-réel. Ce cours présente les concepts fondamentaux des XDD, leurs propriétés mathématiques, et leur utilisation dans l'analyse des temps d'exécution.

1 Bloc de Base (BB)

Un **bloc de base (BB)** est une séquence d'instructions dans un programme où :

1. Le flux de contrôle ne peut entrer dans le bloc qu'à partir de la première instruction.
2. Le flux de contrôle ne peut sortir du bloc qu'à partir de la dernière instruction.

En d'autres termes, un BB est une séquence linéaire d'instructions sans sauts (branches) internes, ce qui simplifie l'analyse du temps d'exécution. Les BB sont utilisés dans l'analyse WCET pour diviser le programme en unités plus petites et plus gérables.

2 Définition des XDD

Un XDD est une structure de données récursive qui représente un ensemble de temps induits par différentes configurations d'événements. Il est défini comme suit :

▶ **Définition 1.** Un XDD est défini récursivement par :

$$\text{XDD} = \text{LEAF}(k) \mid \text{NODE}(e, \bar{f}, f)$$

où :

— $k \in \mathbb{Z}$ est une constante représentant un temps d'exécution.

— $e \in \mathcal{E}$ est un événement (par exemple, un accès mémoire qui peut être un hit ou un miss).

— $\bar{f}$ et f sont des sous-XDD représentant les temps d'exécution si l'événement e est inactif ou actif, respectivement.

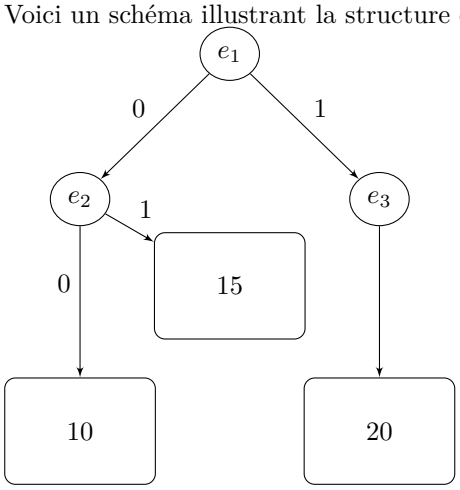
▶ **Exemple 2.** Un XDD représentant le temps d'exécution d'une instruction en fonction de deux événements e_1 et e_2 peut être écrit comme suit :

$$\text{NODE}\left(e_1, \text{NODE}\left(e_2, \text{LEAF}(10), \text{LEAF}(15)\right), \text{LEAF}(20)\right)$$

Cela signifie :
— Si e_1 est inactif, le temps d'exécution est 10 si e_2 est inactif, et 15 si e_2 est actif.
— Si e_1 est actif, le temps d'exécution est 20.

3 Schéma des XDD

Voici un schéma illustrant la structure d'un XDD :



4 Propriétés des XDD

Les XDD possèdent plusieurs propriétés importantes qui garantissent leur efficacité et leur compacité.

4.1 Canonicité

Un XDD est dit **canonique** s'il satisfait les propriétés suivantes :

1. **Ordre** : Les événements dans un XDD suivent un ordre total prédéfini.
2. **Compacité** : Aucun nœud dans un XDD ne peut avoir deux sous-XDD identiques.

▶ **Définition 3.** Un XDD est canonique si pour tout nœud $\text{NODE}(e, \bar{f}, f)$:
— Les sous-XDD $\bar{f}$ et f sont distincts.
— Les événements dans les sous-XDD suivent l'ordre total prédéfini.

4.2 Opérations sur les XDD

Les XDD supportent deux opérations principales : l'addition ($\otimes$) et le maximum ($\oplus$). Ces opérations sont définies récursivement comme suit :

▶ **Définition 4.** Pour deux XDD f_1 et f_2, et une opération binaire $\square$ sur $\mathbb{Z}$:

$$f_1 \odot f_2 = \begin{cases} \text{LEAF}(k_1 \square k_2) & \text{si } f_1 = \text{LEAF}(k_1) \text{ et } f_2 = \text{LEAF}(k_2) \\ g_1 \odot g_2 & \text{si } f_1 = \text{NODE}(e, \bar{g_1}, g_1) \text{ et } f_2 = \text{NODE}(e, \bar{g_2}, g_2) \\ f_1 \odot \bar{g_2} & \text{si } f_2 = \text{NODE}(e, \bar{g_2}, g_2) \text{ et } \text{evt}(f_1) \leq e \\ \bar{g_1} \odot f_2 & \text{si } f_1 = \text{NODE}(e, \bar{g_1}, g_1) \text{ et } \text{evt}(f_2) \leq e \\ \text{NODE}(e, \bar{g_1} \odot \bar{g_2}, g_1 \odot g_2) & \text{si } f_1 = \text{NODE}(e, \bar{g_1}, g_1) \text{ et } f_2 = \text{NODE}(e, \bar{g_2}, g_2) \end{cases}$$

▶ **Exemple 5.** Pour deux XDD $f_1 = \text{NODE}(e_1, \text{LEAF}(10), \text{LEAF}(20))$ et $f_2 = \text{NODE}(e_1, \text{LEAF}(15), \text{LEAF}(25))$, l'addition $f_1 \otimes f_2$ donne :

$$f_1 \otimes f_2 = \text{NODE}(e_1, \text{LEAF}(25), \text{LEAF}(45))$$

5 Utilisation des XDD dans l'analyse WCET

Les XDD sont utilisés pour représenter les temps d'exécution des blocs de base (BB) dans l'analyse WCET. Ils permettent de prendre en compte les latences variables dues aux événements tels que les accès mémoire (cache hit/miss) ou les prédictions de branchement.

5.1 Calcul du WCET avec XDD

Le WCET d'un BB est calculé en utilisant les XDD pour représenter les temps d'exécution pour chaque configuration d'événements. Les opérations $\otimes$ et $\oplus$ sont utilisées pour combiner les temps d'exécution des instructions en tenant compte des dépendances dans le pipeline.

▶ **Exemple 6.** Pour un BB avec deux instructions I_1 et I_2, les temps d'exécution peuvent être représentés par des XDD f_1 et f_2. Le WCET du BB est calculé comme suit :

$$\text{WCET} = f_1 \oplus f_2$$

6 Conclusion

Les XDD sont une structure de données puissante pour représenter de manière compacte les temps d'exécution en présence de latences variables. Ils permettent de réduire la complexité combinatoire de l'analyse WCET et d'améliorer la précision des estimations. Les propriétés mathématiques des XDD garantissent leur efficacité et leur compacité, ce qui les rend particulièrement adaptés à l'analyse des systèmes temps-réel.