

Allocation de registres : implémentation

Alexis Nasr
Carlos Ramisch
Manon Scholivet
Franck Dary

Compilation – L3 Informatique
Département Informatique et Interactions
Aix Marseille Université

Allocation de registres

- L'allocation de registres repose sur trois packages
 - Le package `graph`, représentation générique de graphes.
 - Le package `intset`, représentation d'ensembles d'entiers
 - Le package `ig`, graphe d'interférence.

La classe ColorGraph

- Classe du package graph
- Suit de très près l'implémentation de coloration de graphe vue en cours
- Quelques petites différences :
 - Les sommets ne sont pas enlevés du graphe, mais ajoutés à l'ensemble `enLeves`
 - Les K couleurs sont représentées par les entiers $\{0, \dots, K - 1\}$.
 - L'absence de couleur est représentée par la constante `NOCOLOR` qui vaut -1 .
- Le tableau `int2Node` permet d'accéder rapidement à un sommet du graphe à partir de son identifiant.

La classe ColorGraph

```
public class ColorGraph {
    public Graph      G;           // le graphe a colorer
    public int        R;           // nombre de sommets
    public int        K;           // nombre de couleurs
    private Stack<Integer> pile;
    public IntSet     enlevés;     // sommets enlevés
    public IntSet     deborde;    // sommets qui débordent
    public int[]      couleur;     // tableau des couleurs
    public Node[]     int2Node;
    static int        NOCOLOR = -1;

    public          ColorGraph(Graph G, int K, int[] phi)
    public IntSet   couleursVoisins(int t)
    public int      choisissCouleur(IntSet colorSet)
    public int      nbVoisins(int t)
    public int      simplification()
    public void     selection()
    public void     debordement()
    public void     colore()
```

Le package ig

```
public class Ig {
    public Graph      graph; //le graphe d'interférence
    public FgSolution fgs;  //les ensembles in et out
    public int        R;    //le nombre de registres fictifs
    public Nasm       nasm; //le pré-nasm
    public Node       int2Node[];

    public          Ig(FgSolution fgs){...}
    public void     construction(){...}
    public int[]    getPrecoloredTemporaries(){...}
    public void     affiche(String baseFileName){...}
```

- Le tableau `int2Node` permet d'accéder à un sommet de graphe à partir de son identifiant.
- La méthode `getPrecoloredTemporaries` renvoie un tableau de taille `R`, indiquant pour chaque registre, sa couleur.
- La méthode `construction`, construit le graphe d'interaction à partir de la solution `fgs` du graphe d'analyse.