

Compiler Construction

~ Exam 2021 (middle+back-end) review ~

Question 1.

Parmi les réponses suivantes indiquez ce qui relève du middle-end:

- Les optimisations génériques
- SSA
- Le découpage en blocs de base
- La linéarisation
- La gestion du static-link
- l'allocation des registres
- la réorganisation des instructions

Question 1.

Parmi les réponses suivantes indiquez ce qui relève du middle-end:

- Les optimisations génériques
- SSA
- Le découpage en blocs de base
- La linéarisation
- La gestion du static-link
- l'allocation des registres
- la réorganisation des instructions

Question 2.

Quels sont les avantages d'une représentation intermédiaire basée sur la pile?

- Compacité
- Proche des micro-processeurs actuels
- Facile a comprendre
- Augmentation de la pression sur les registres

Question 2.

Quels sont les avantages d'une représentation intermédiaire basée sur la pile?

- Compacité
- Proche des micro-processeurs actuels
- Facile a comprendre
- Augmentation de la pression sur les registres

Question 3.

Les blocs d'activation ne doivent pas stocker

- Le program-counter
- Le static link
- L'adresse de retour
- Les arguments de la fonction

Question 3.

Les blocs d'activation ne doivent pas stocker

- Le program-counter
- Le static link
- L'adresse de retour
- Les arguments de la fonction

Question 4.

```
let function f(n: int): int
=
  let function g(): int =
    let function h(): int = n + 2
    in h() + 1
    end
  in g() + 1
  end
in
  f()
end
```

Question 4.

Dans le code précédent quel doit être le static link de la fonction g?

- Il n'y a pas besoin de static link pour g
- g
- h
- f

Question 4.

Dans le code précédent quel doit être le static link de la fonction g?

- Il n'y a pas besoin de static link pour g
- g
- h
- f

Question 5.

Qu'est ce qu'une proto-traduction?

- Quelque chose qui transforme une expression en instruction
- Quelque chose qui transforme une instruction en condition
- Quelque chose qui traduit une condition en expression
- Quelque chose qui fait une traduction partielle
- Je n'ai jamais entendu parlé de proto-traduction

Question 5.

Qu'est ce qu'une proto-traduction?

- Quelque chose qui transforme une expression en instruction
- Quelque chose qui transforme une instruction en condition
- Quelque chose qui traduit une condition en expression
- Quelque chose qui fait une traduction partielle
- Je n'ai jamais entendu parlé de proto-traduction

Question 6.

Qu'est ce que la runtime

- L'ensemble des valeurs prises par les variables durant l'exécution
- La bibliothèque standard d'un langage de programmation
- Un ensemble de fonction prédéfinies pour un langage de programmation
- Tout ce qui n'est pas connu à la compilation.

Question 6.

Qu'est ce que la runtime

- L'ensemble des valeurs prises par les variables durant l'exécution
- La bibliothèque standard d'un langage de programmation
- Un ensemble de fonction prédéfinies pour un langage de programmation
- Tout ce qui n'est pas connu à la compilation.

Question 7.

```
    j := 101
    i := 0
label L1
    cjump i <> j, L3, L2
label L3
    j := j + 1
label L5
    i := i+1
    jump L1
label L2
    cjump j == 42, L4, L5
label L4
    j := 0
    jump L5
```

Question 7.

Combien y a t'il de block de base dans le code précédent? Ne cherchez pas à analyser ce code, il ne fait rien de particulier.

- 0
- 1
- 2
- 3
- 4
- 5

Question 7.

Combien y a t'il de block de base dans le code précédent? Ne cherchez pas à analyser ce code, il ne fait rien de particulier.

- 0
- 1
- 2
- 3
- 4
- 5

Question 8.

```
a := 42
```

```
b := 51
```

```
c := 21
```

```
label L1
```

```
c := c+1
```

```
cjump a <> 51, L3, L2
```

```
a := a + 5
```

```
label L5
```

```
b := b + 128
```

```
label L2
```

```
c := c - 1
```

```
cjump L4
```

```
label L4
```

```
b := b - 1
```

```
jump b == 42, L4, L1
```

Question 8.

Considérons le code précédent. Combien de jumps et de labels faut il rajouter pour n'avoir que des basic-block.

- Il n'y a déjà que des basic-block
- 2 labels et 3 jumps
- 3 labels et 3 jumps
- 3 labels et 2 jumps
- 1 label et 2 jumps
- 1 label et 1 jumps
- 2 label et 1 jumps
- 0 label et 1 jump
- 1 jump et 0 label

Question 8.

Considérons le code précédent. Combien de jumps et de labels faut il rajouter pour n'avoir que des basic-block.

- Il n'y a déjà que des basic-block
- 2 labels et 3 jumps
- 3 labels et 3 jumps
- 3 labels et 2 jumps
- 1 label et 2 jumps
- 1 label et 1 jumps
- 2 label et 1 jumps
- 0 label et 1 jump
- 1 jump et 0 label

Question 9.

```
a := 42
cjump a = 42, L0, L1
L0:
  b := 51
  c := 69
  jump L1
L1:
  a := a + 12
  d := 1337 + c
  if a < d goto L1
  b := b * c
  return b
```

Question 9.

Ce code satisfait-il les contraintes des micro-processeurs? Rappel cjump <cond>, <iftrue>, <iffalse>.

- Non
- Oui

Question 9.

Ce code satisfait-il les contraintes des micro-processeurs? Rappel cjump <cond>, <iftrue>, <iffalse>.

- Non
- Oui

Question 10.

L'objectif de l'étude de la vivacité des variables est de:

- d'insérer les appels aux destructeurs
- calculer le graphe de flot de contrôle
- préparer l'allocation des registres
- préparer l'inliner

Question 10.

L'objectif de l'étude de la vivacité des variables est de:

- d'insérer les appels aux destructeurs
- calculer le graphe de flot de contrôle
- préparer l'allocation des registres
- préparer l'inliner

Question 11.

```
    i := 0
    a := 21
L1:
    b := b*2
    cjump b < 1337, L5, L3
L2:
    b := b + a
    cjump b < 1337, L1, L3
L3:
    a := a + i
    cjump b = 307, L4, L3
L4:
    b := i + b
    return b
```

Question 11.

Combien de noeuds / transitions seront représentés dans le graphe de flot de contrôle associé à ce programme?

- 1 noeuds
- 2 noeuds
- 3 noeuds
- 4 noeuds
- 5 noeuds
- 6 noeuds
- 7 noeuds
- 8 noeuds
- 9 noeuds
- 10 noeuds
- 11 noeuds
- 12 noeuds

Question 11.

Combien de noeuds / transitions seront représentés dans le graphe de flot de contrôle associé à ce programme?

- 1 noeuds
- 2 noeuds
- 3 noeuds
- 4 noeuds
- 5 noeuds
- 6 noeuds
- 7 noeuds
- 8 noeuds
- 9 noeuds
- 10 noeuds
- 11 noeuds
- 12 noeuds

Question 12.

```
    i := 0
    a := 21
L1:
    b := b*2
    cjump b < 1337, L4, L2
L2:
    b := b + a
    cjump b < 1337, L1, L3
L3:
    a := a + i
    cjump b = 307, L4, L3
L4:
    b := i + b
    return b # b is liveout
```

Question 12.

Calculez la vivacité pour le programme précédent et indiquez quelles sont les affirmations vraies. (Note, la numérotation des lignes commence a 1).

- $\text{in}[1] = \{ b \}$
- $\text{in}[2] = \{ b \}$
- $\text{out}[11] = \{ b, i \}$
- $\text{out}[11] = \{ a, b, i \}$
- $\text{out}[4] = \text{in} [4] = \{ a, i, b \}$
- $\text{out}[4] = \text{in} [4] = \{ i, b \}$

Question 13.

```
# live in: k j
g := [j + 12]
h := k + 1
f := g * h
e := [j + 16]
m := [j + 8]
b := [f]
c := e + m
d := c
k := m + 4
j := b
# live out: d k j
```

Question 13.

Combien de noeuds aura le graphe d'interférence ci-dessous

- 1
- 2
- 4
- 5
- 7
- 8
- 10
- 11
- 13
- 14

Question 13.

Combien de noeuds aura le graphe d'interférence ci-dessous

- 1
- 2
- 4
- 5
- 7
- 8
- 10
- 11
- 13
- 14

Question 14.

Considérant le programme de la question 13, peut on trouver une allocation des registres n'utilisant que 3 registres?

- Oui
- Non

Question 15.

Considérant le programme de la question 13, y a-t-il des possibilités de coalescing?

- Oui
- Non

Question 16.

Considérant le programme de la question 13, le programme est-il sous forme SSA?

- Oui
- Non