

---

**Compilerbau**

---

Blatt 13

Abgabe: 6.2.2002

1. [5 Punkte] Erweitern Sie den Compiler um Codegenerierung für `If`. Sie können dabei voraussetzen, daß das Programm typkorrekt ist.
2. [3 Punkte] Erweitern Sie den Compiler um Codegenerierung für `Let`.
3. [6 Punkte] Erweitern Sie den Compiler um eine Funktion

```
val emit_cont :  
    Cps.cont -> int -> (Cps.ident -> identinfo) -> Iset.statement list
```

welche eine Continuation, den Level der Umgebung und die Übersetzungszeitumgebung als Argumente nimmt und den Code der Continuation generiert, sowie ein Continuation-Objekt für sie anlegt.

4. [3 Punkte] Erweitern Sie den Compiler um Codegenerierung für `LetCont`. Verwenden Sie die Prozedur `emit_cont` aus Aufgabe 3.
5. [2 Punkte] Erweitern Sie den Compiler um Codegenerierung für `Call`. Verwenden Sie die Prozedur `emit_cont` aus Aufgabe 3.
6. [11 Punkte] Erweitern Sie den Compiler um Exception-Handling. Implementieren Sie dazu die Primitiva `try` und `raise`. Legen Sie für auch einen Standard-Exception-Handler an; sein Code darf einfach aus der `trap`-Instruktion bestehen.
7. [je 1 Punkt](Bonus) Bestellen Sie bei Martin Gasbichler weitere Primitiva und erweitern Sie den Compiler um Codegenerierung für diese. Jedes Primitiva wird nur an eine Gruppe ausgegeben.