1 Things to do

Add two new passes:

- `expose-allocation-pointer`
- `expose-memory-operands`

Make minor changes to:

- `uncover-frame-conflict`
- `finalize-frame-locations`
- `uncover-register-conflict`
- `finalize-locations`
- `expose-frame-var`

Make changes to:

- `remove-complex-opera*`
- `flatten-set!`
- `impose-calling-conventions`
- `select-instruction`

`expose-allocation-pointer` runs after or before `impose-calling-conventions`. `expose-memory-operands` runs after or before `expose-frame-var`.

Q: Why is `expose-memory-operands` so late? I have to do so many minor changes.

A: Even if you do it earlier you have to do the same for `index-opnd`s. Worse, more code needs adjustment if you want to target another architecture.

1.1 `expose-allocation-pointer`

This pass converts

```
(set! x (alloc expr))
```

to

```
(begin
  (set! x ap)
  (set! ap (+ ap expr))
```

where *ap* refers to `allocation-pointer-register` in helpers.scm.

1.2 `expose-memory-operands`

This pass converts

```
(mset! base offset Triv)
```

to

`(set! (make-disp-opnd` *base offset*`)` *Triv*`)`

or

`(set! (make-disp-opnd` *offset base*`)` *Triv*`)`

or

`(set! (make-index-opnd` *base offset*`)` *Triv*`)`

depending on types of *base* and *offset*. Also do the same for `mref`.

1.3 `remove-complex-opera*`

This pass needs to handle `mset!`, `mref` and `alloc`.

1.4 `flatten-set!`

you may need to flatten `mset!` in `flatten-set!` if you choose not to reduce *expr* to *Triv*.

1.5 `impose-calling-conventions`

This pass requires new `match`-clauses to handle new operators. Also remember to add `allocation-pointer-register` to *Loc*`*` (live locations) in calls.

1.6 `select-instructions`

This pass requires new `match`-clauses to handle new operators. It's OK to have

`(mset!` *reg reg Triv*`)`

and

`(mset!` *reg reg*`)`

because you can use `make-index-opnd`. In assembly it corresponds to `(`*reg*`, `*reg*`)`.

Remark: In fact the followings are OK in assembly: `(`*reg*`, `*reg*`)`, *n*`(`*reg*`)`, *n*`(`*reg*`, `*reg*`)`.

1.7 passes that need minor changes

These passes require a new `match`-clause to handle `mset!` and maybe `mref`, depending on your code.