

Beyond Haskell

Ben Lippmeier

University of New South Wales, Australia

Haskell Implementors Workshop

2010/10/1

My Nirvana



photo: lisa nolan @ flickr

My Nirvana

- Realising a program was written in Haskell after I've already been using it for an hour.

My Nirvana

- Realising a program was written in Haskell after I've already been using it for an hour.
- Being pleasantly surprised about the absolute performance of a parallel Haskell program.

My Nirvana

- Realising a program was written in Haskell after I've already been using it for an hour.
- Being pleasantly surprised about the absolute performance of a parallel Haskell program.
- Getting any of the games on Hackage to compile....

Reassuring Signs



[Parallel Haskell: 2-year project to push real world use](#)

Thu, 29 Apr 2010 14:41:09 GMT, by duncan.
Filed under well-typed.

[Well-Typed delivers air traffic analysis tool for NATS](#)

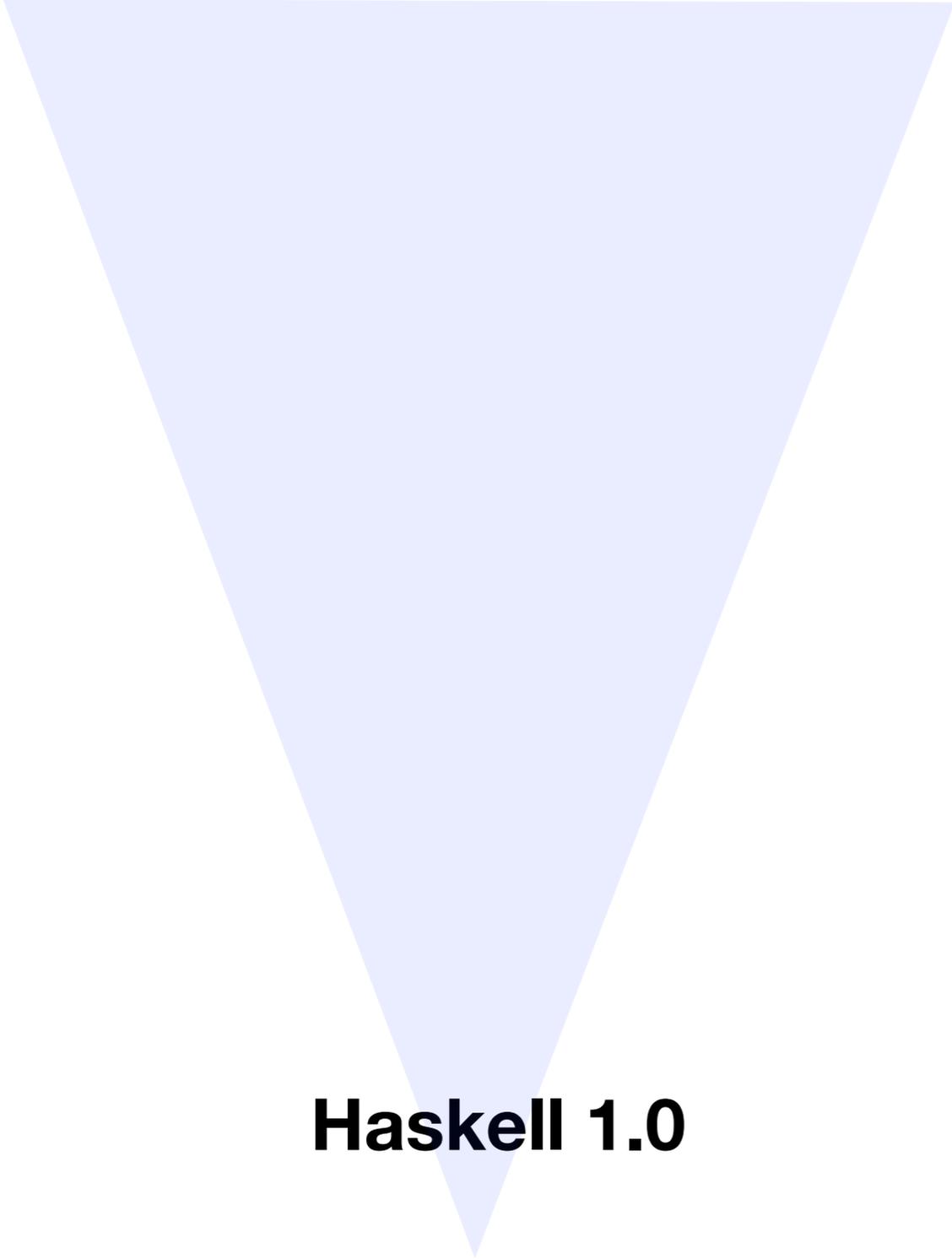
Wed, 05 May 2010 11:04:35 GMT, by duncan.
Filed under well-typed.

Galois, Inc. Wins Three Department of Energy Small Business Research Awards

August 10th, 2010 by Galois

Onwards and Upwards

Onwards and Upwards



Haskell 1.0



photo: sa_ku_ra @ flickr

Onwards and Upwards

Haskell `98

Haskell 1.0



photo: sa_ku_ra @ flickr

Onwards and Upwards

Haskell + MPTCs

Haskell `98

Haskell 1.0



photo: sa_ku_ra @ flickr

Onwards and Upwards

Haskell + GADTs

Haskell + MPTCs

Haskell `98

Haskell 1.0



photo: sa_ku_ra @ flickr

Onwards and Upwards

Haskell + Cheese

Haskell + GADTs

Haskell + MPTCs

Haskell `98

Haskell 1.0



photo: Scorpions and Centaurs @ flickr



photo: sa_ku_ra @ flickr

Relatives

Currrx+

+ Logic Programming



+ Concurrent Objects
+ Realtime constraints
- Default Laziness

ANIA

Disciple

+ Region and Effect Typing
+ Mutability Polymorphism
- Default Laziness



photo: Nadia308 @ flickr

MOAR Features !!!!

Expressiveness

Expressiveness

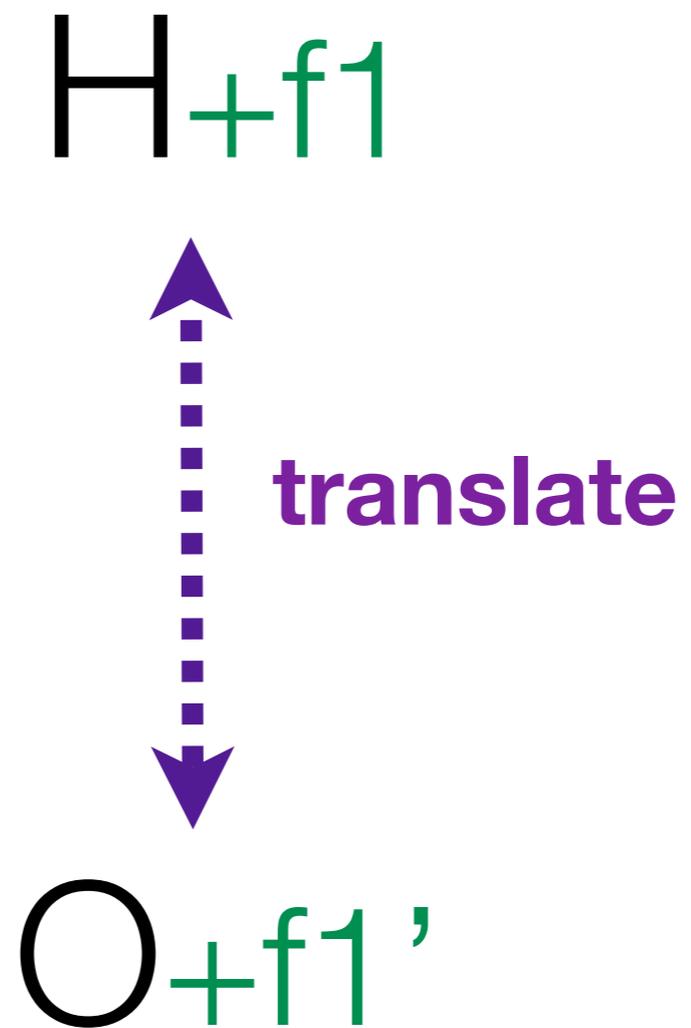
H_{+f1}

Expressiveness

H_{+f1}

O_{+f1}'

Expressiveness



Expressiveness

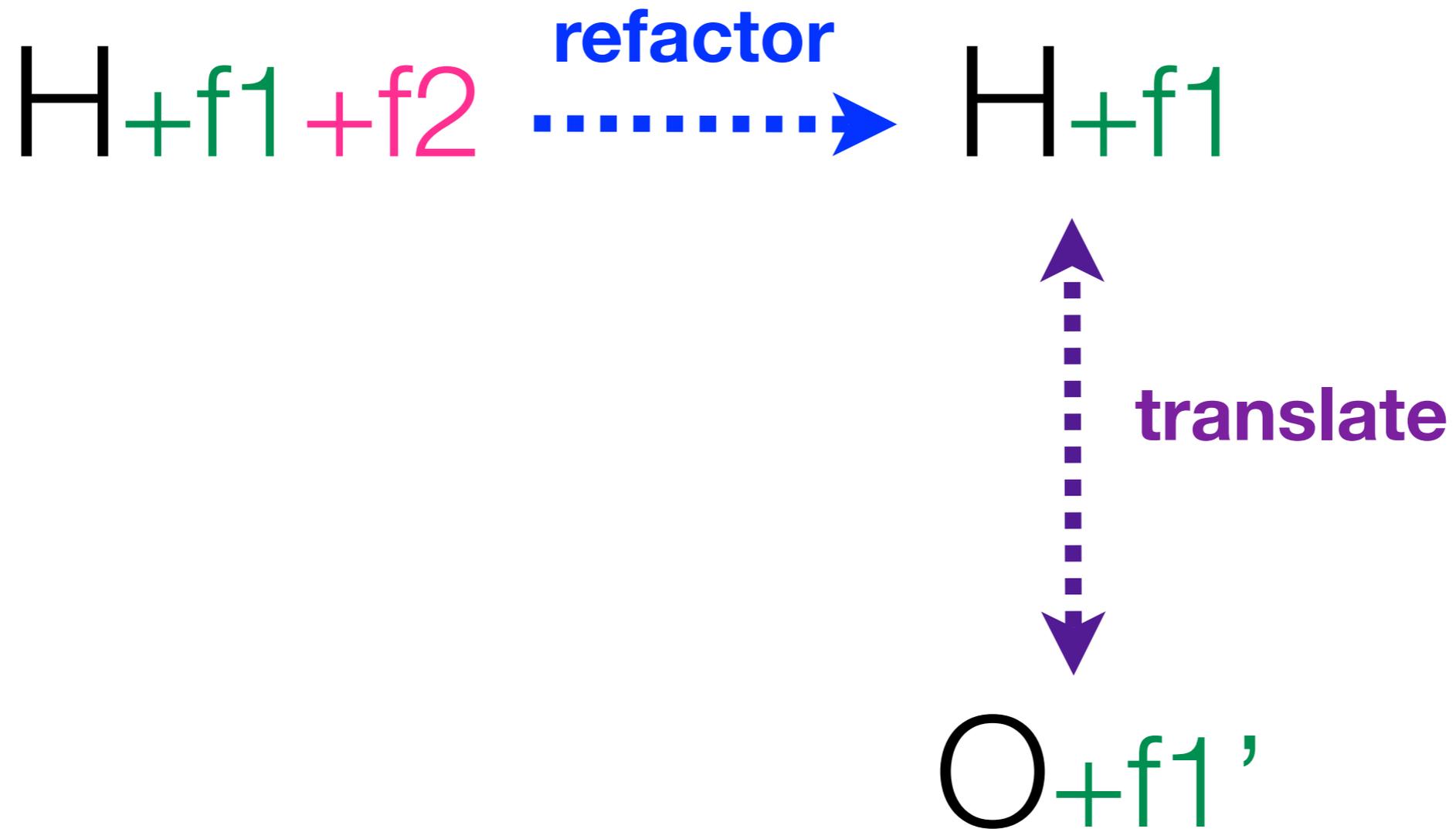
H_{+f1+f2}

H_{+f1}

translate

O_{+f1}'

Expressiveness



Expressiveness

$H+f1+f2$ **refactor** \rightarrow $H+f1$

HIGH EFFORT

GADTs

Pattern Alternatives

Implicit Parameters

if-then-else

LOW EFFORT

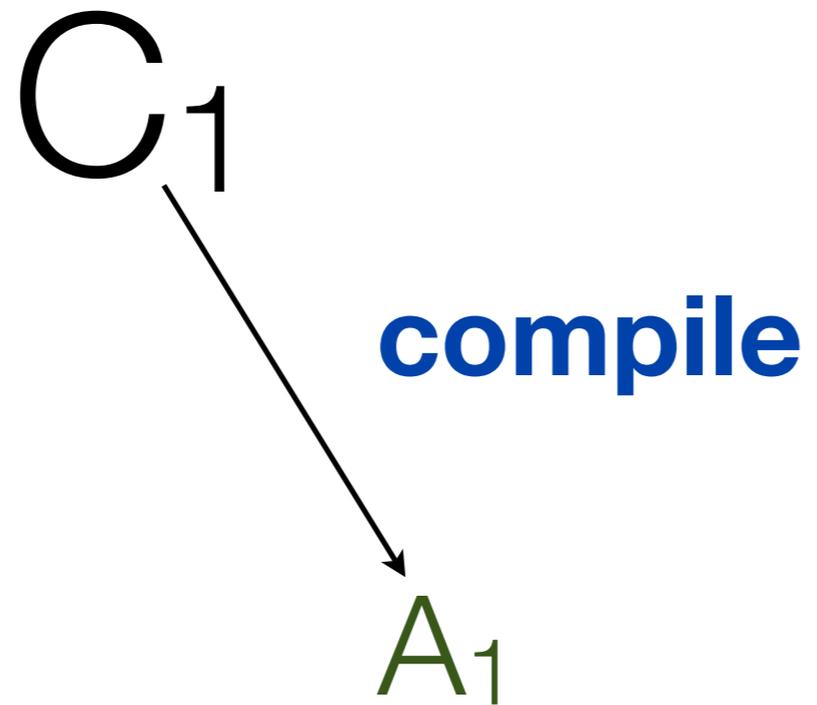
\uparrow **translate** \downarrow

$O+f1'$

Reachability

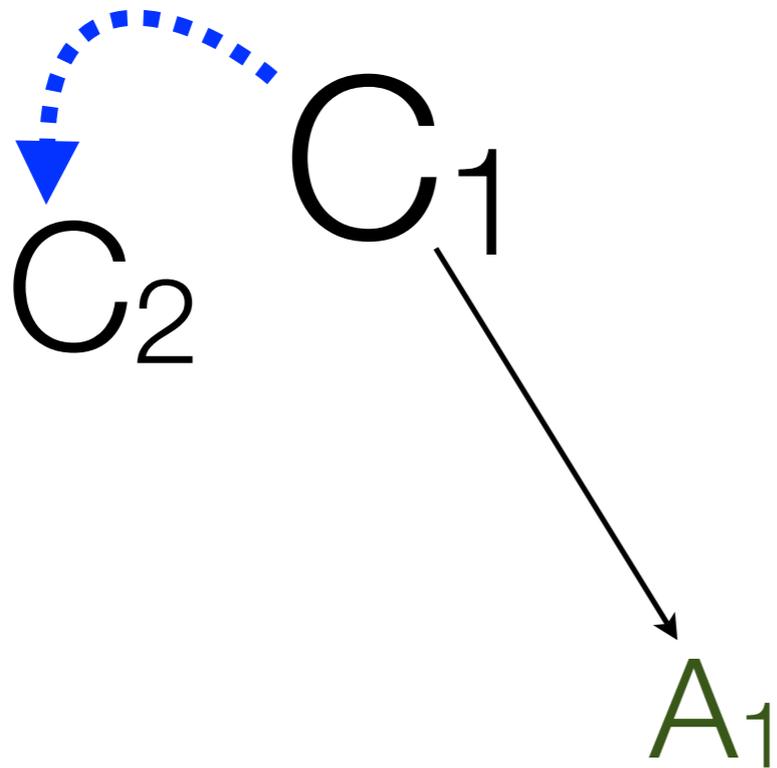
C_1

Reachability

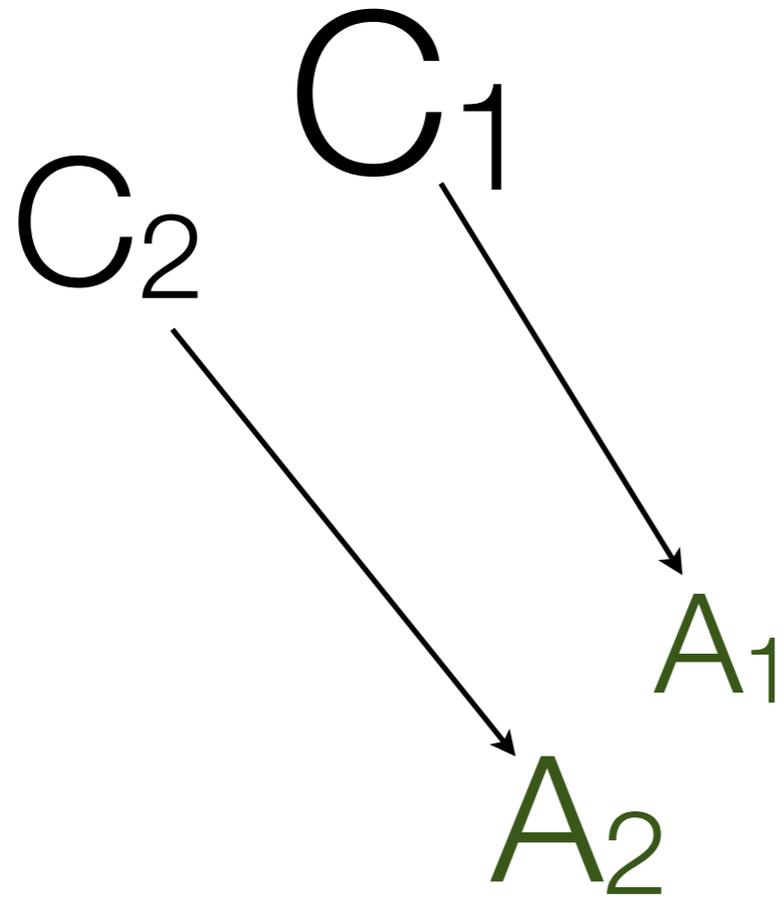


Reachability

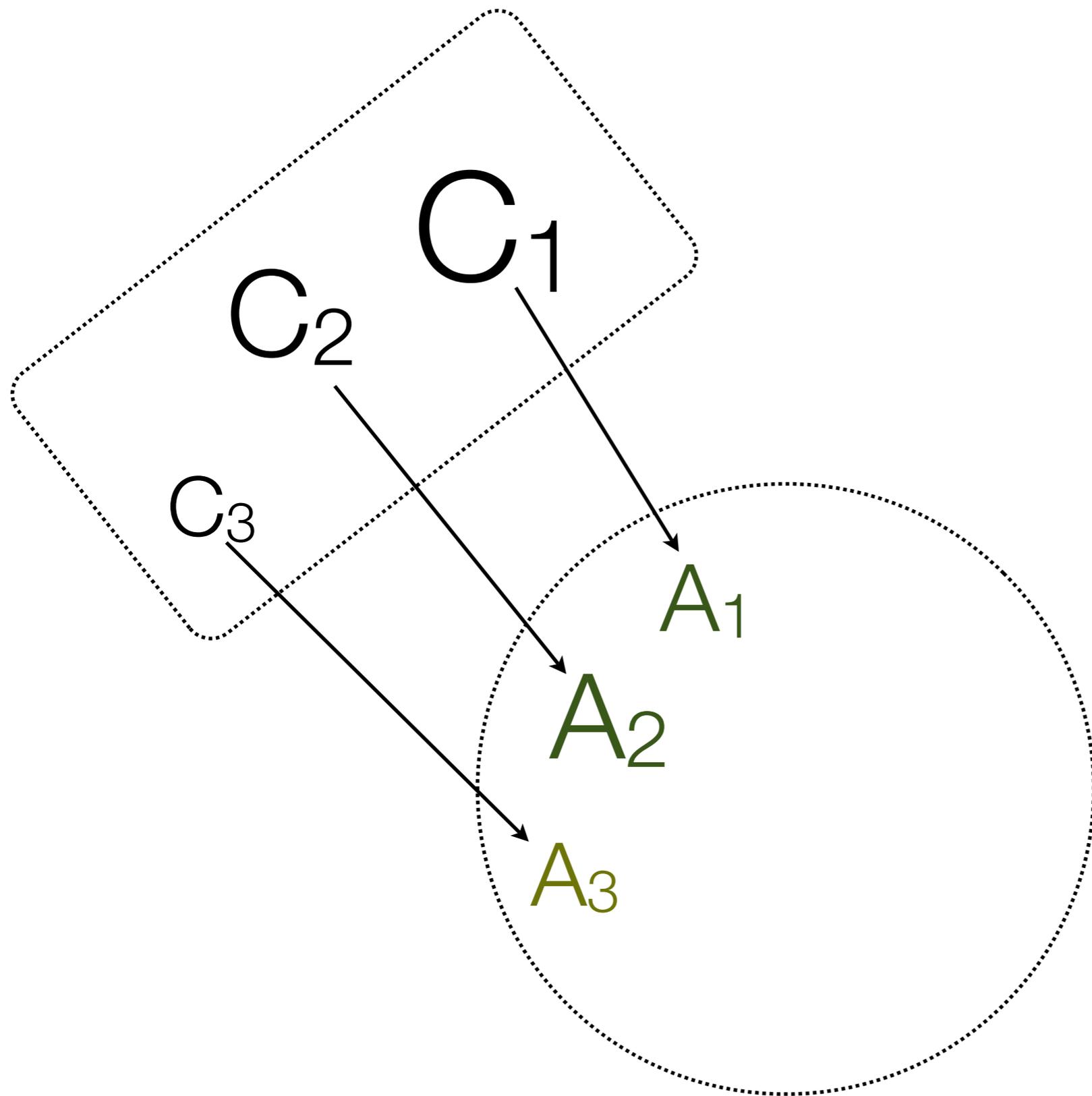
optimise



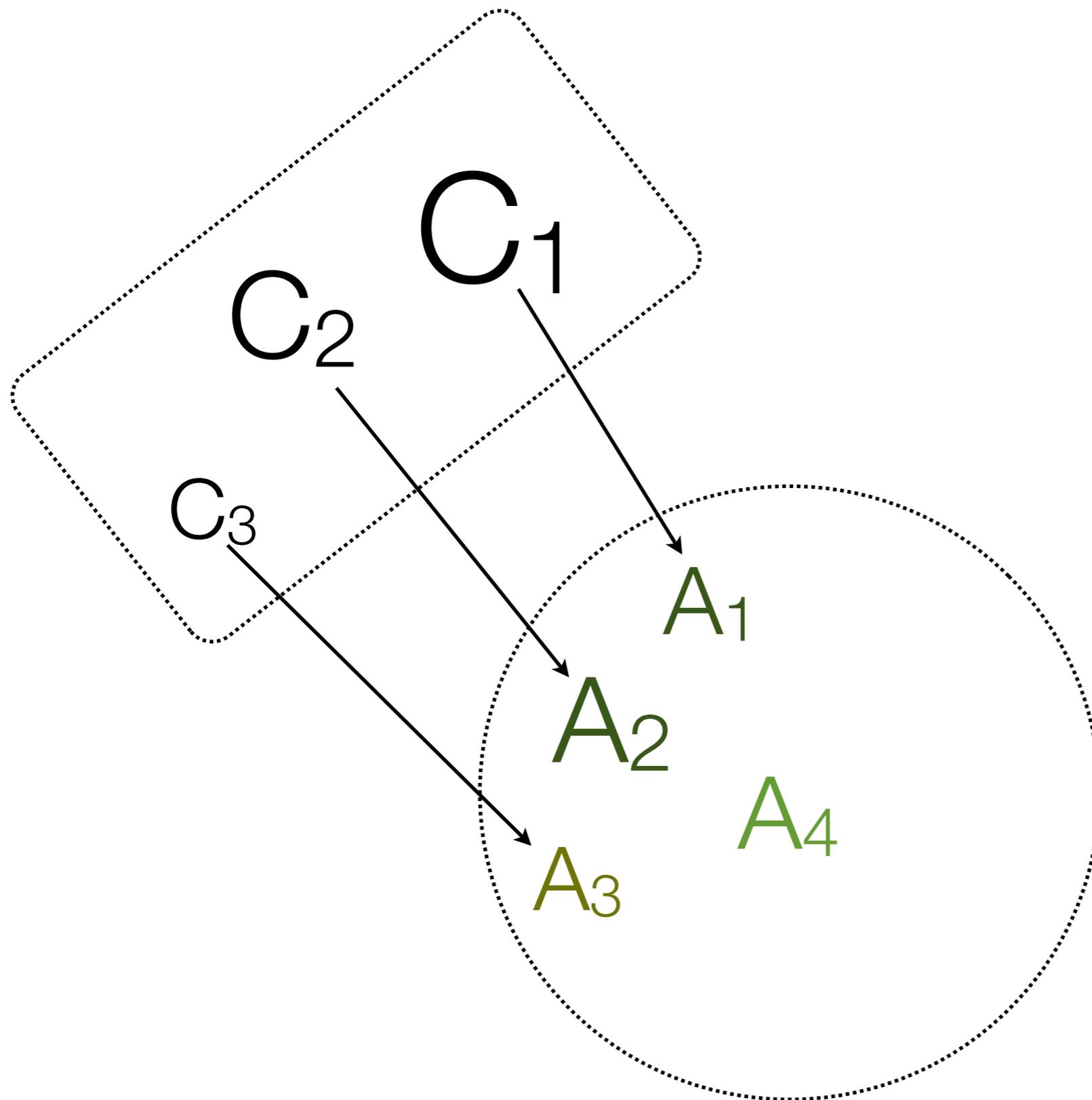
Reachability



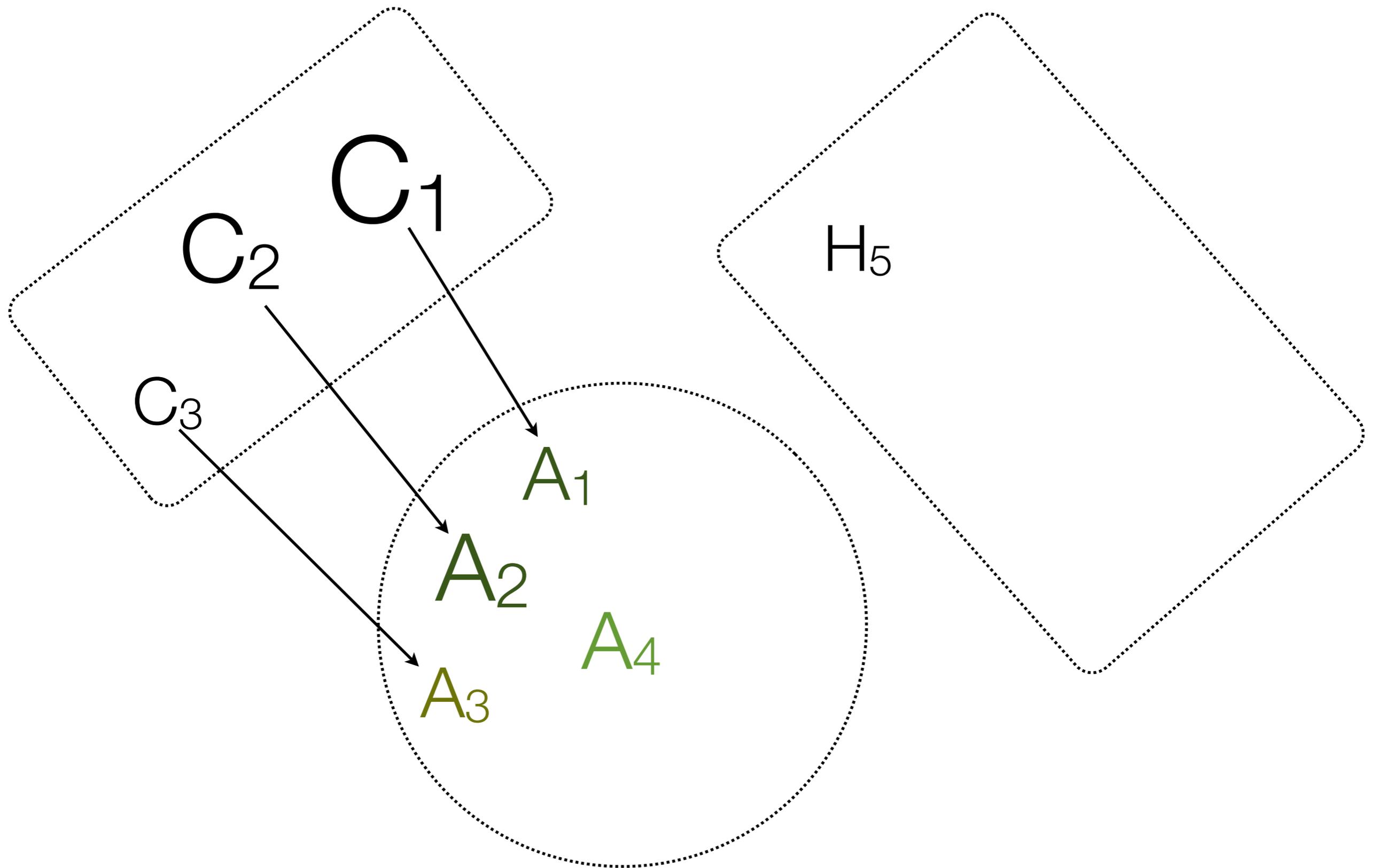
Reachability



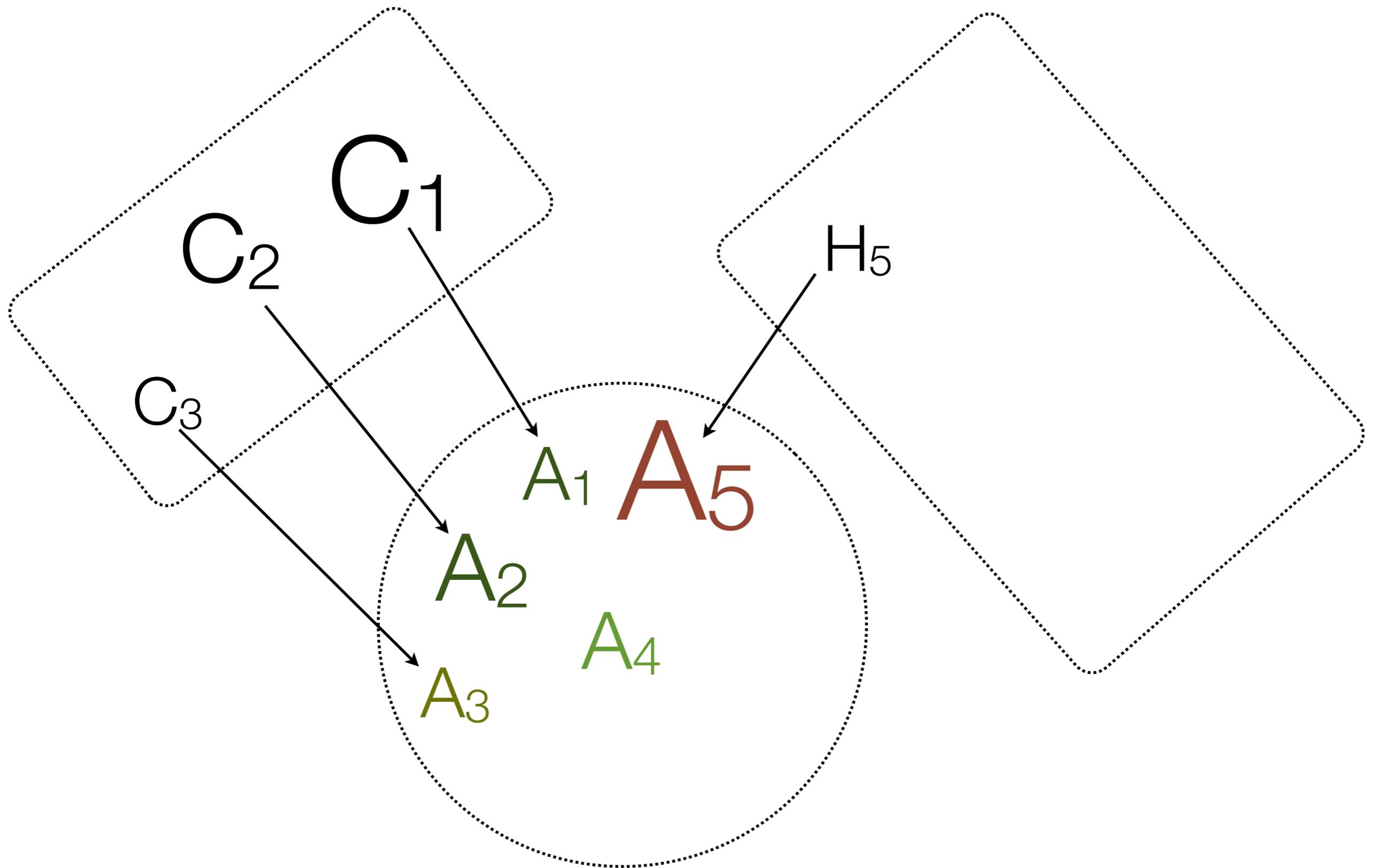
Reachability



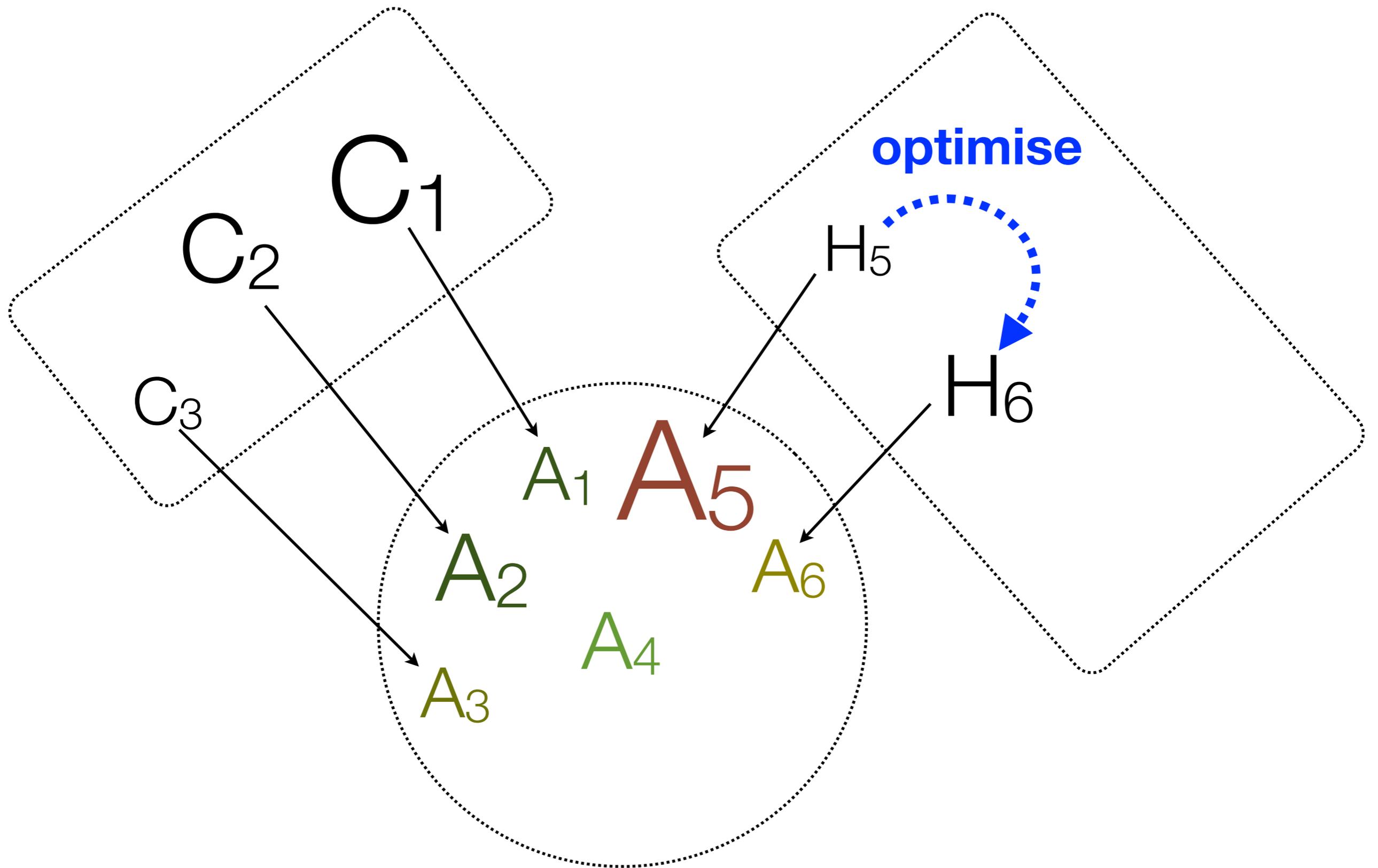
Reachability



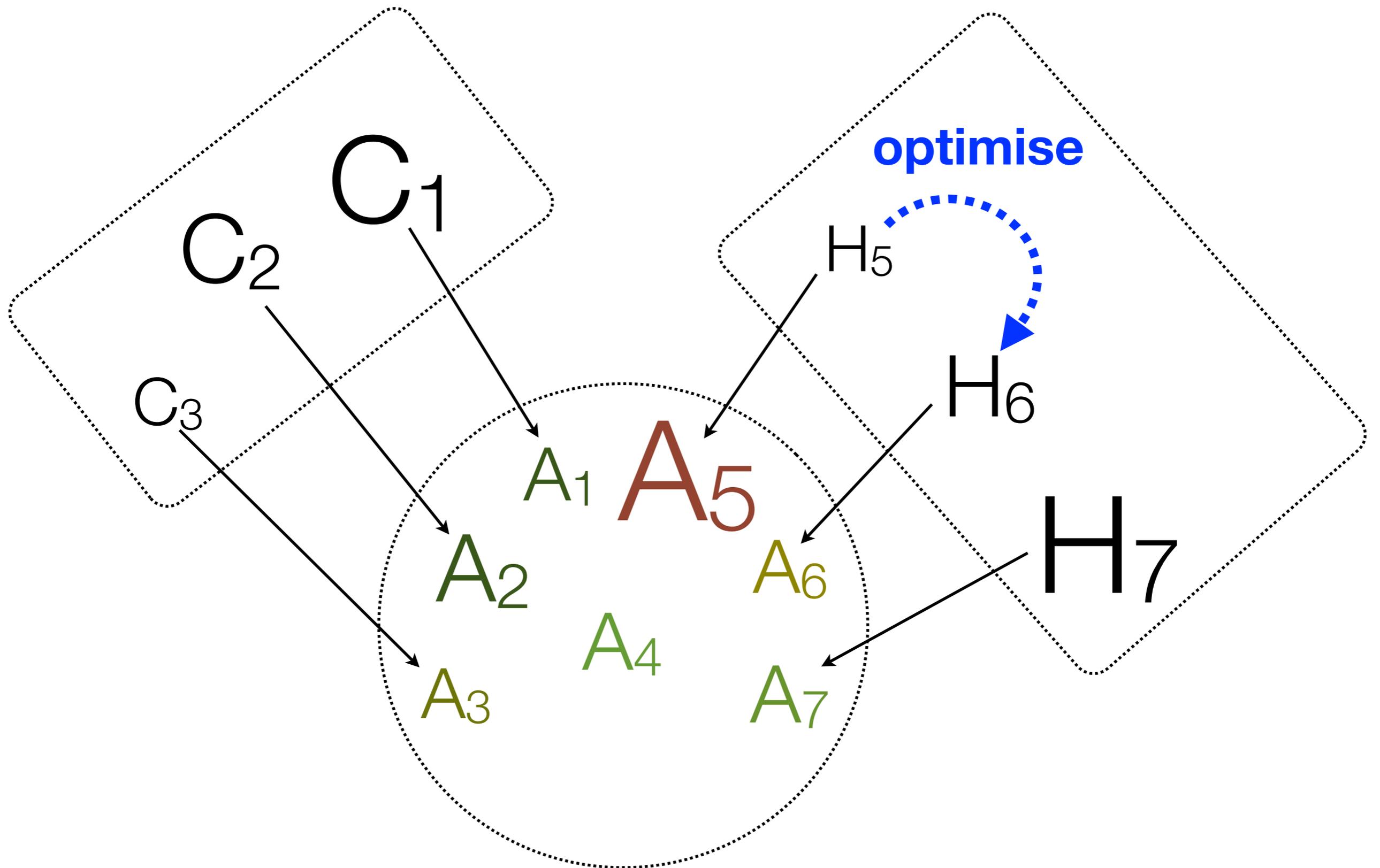
Reachability



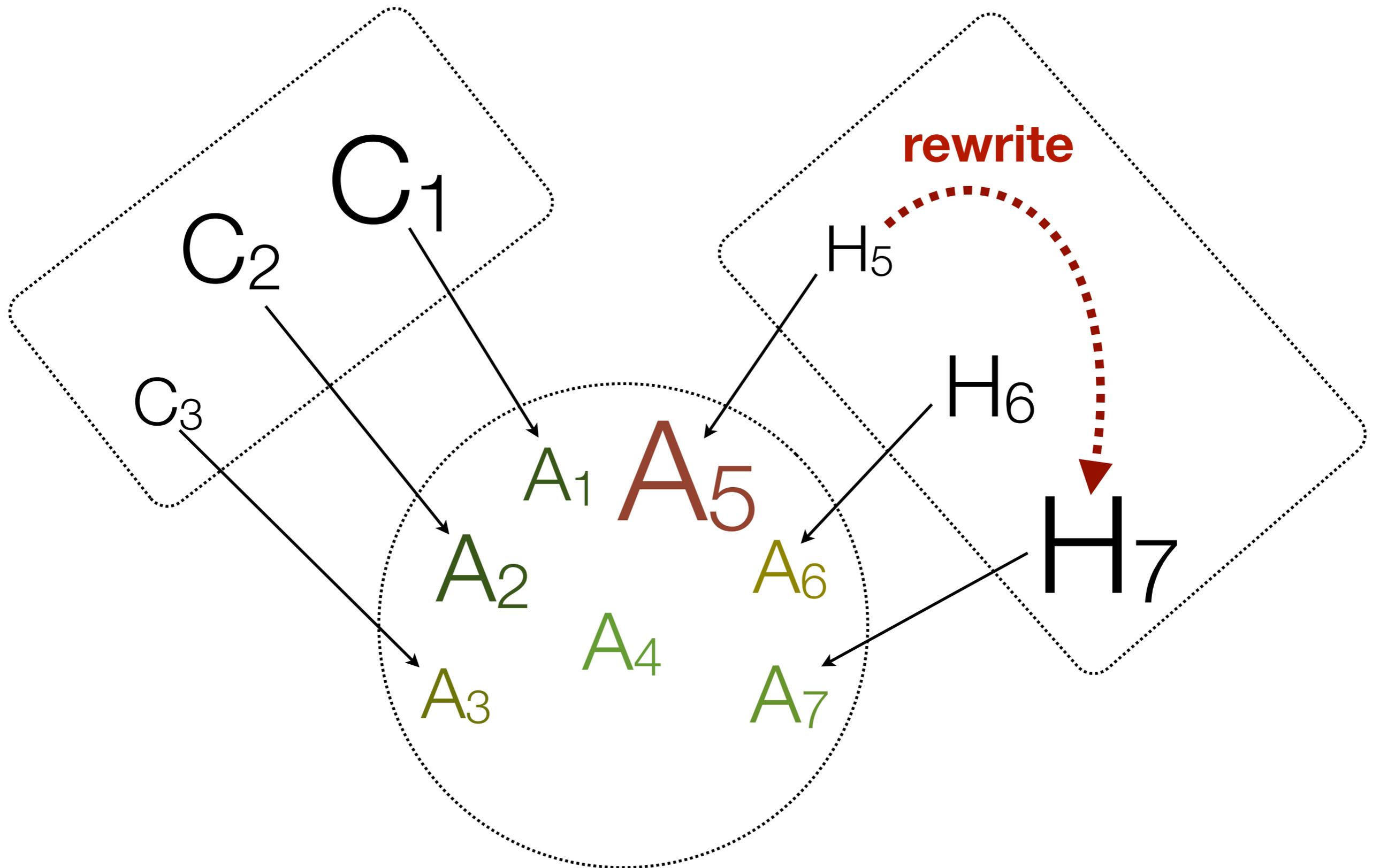
Reachability



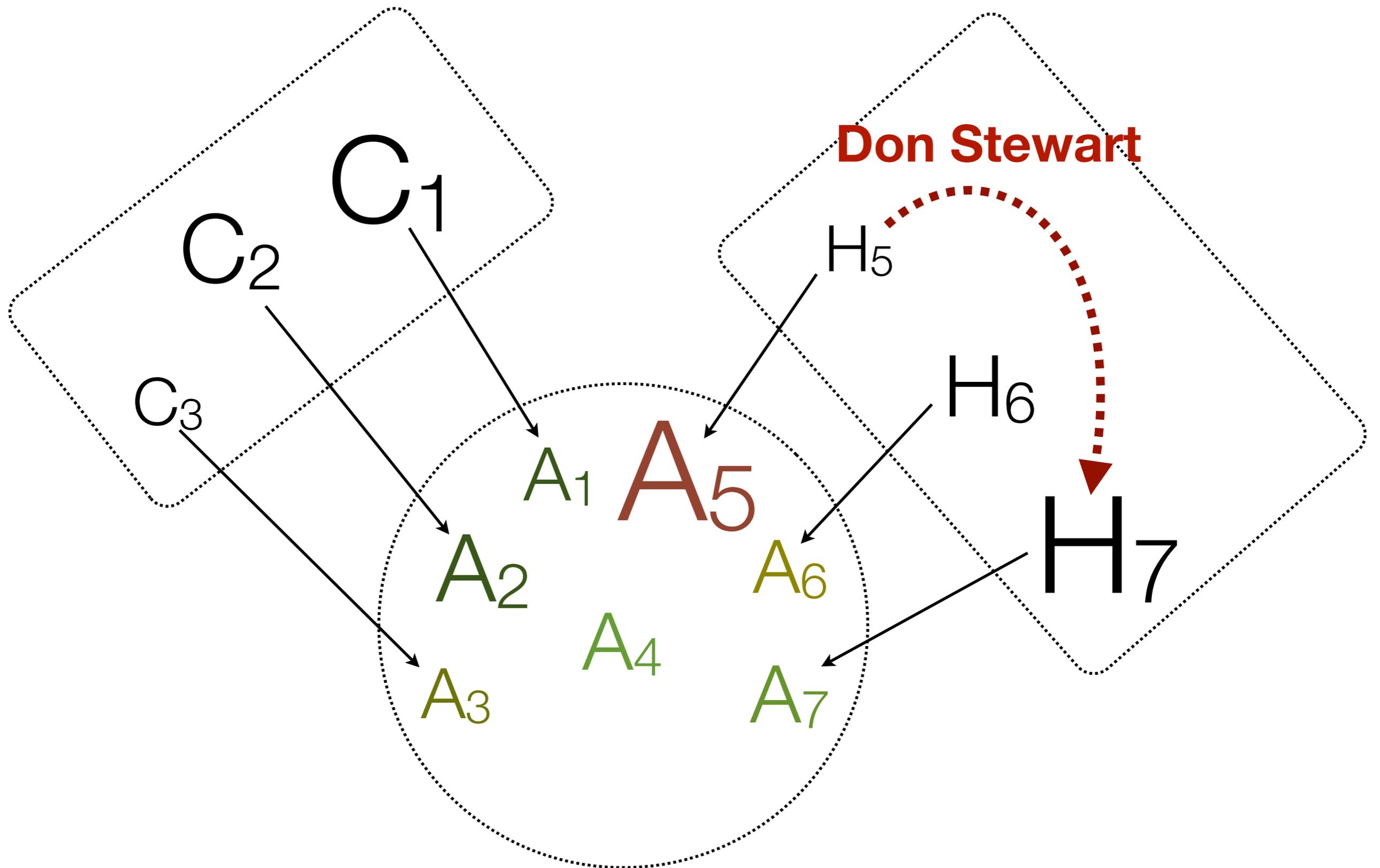
Reachability



Reachability



Reachability



**If I wanted to stare at the core IR
I would have written it in core.**

**If I wanted to stare at the core IR
I would have written it in core.**



Love and Bunnies

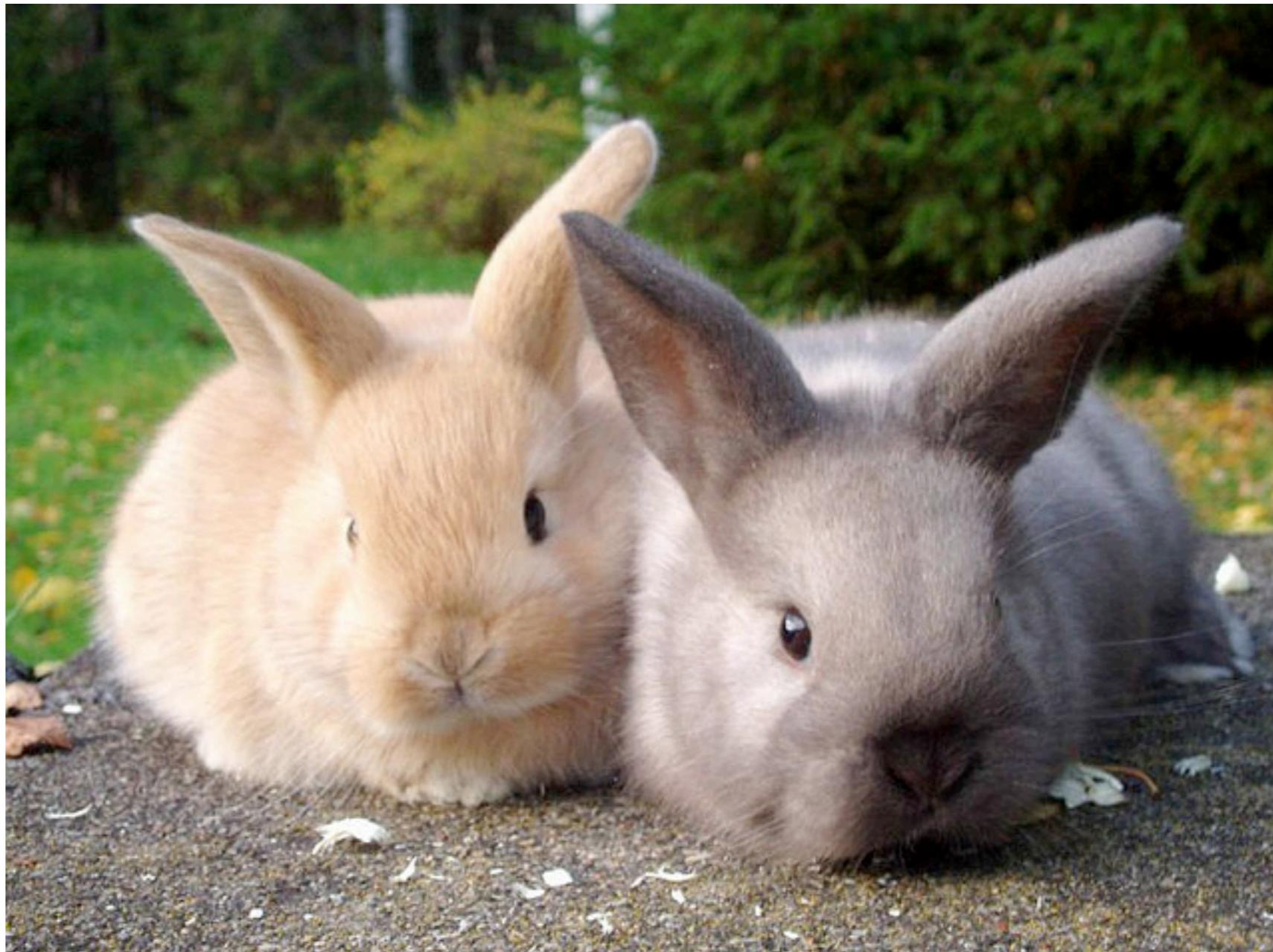


photo: Madeleine_ @ flickr

<vent>

</vent>



photo: aussiegall @ flickr

Talking Points

- What are your experiences optimising code?
Is it a problem? what's the success rate?

Talking Points

- What are your experiences optimising code?
Is it a problem? what's the success rate?
- What about with parallel Haskell?

Talking Points

- What are your experiences optimising code?
Is it a problem? what's the success rate?
- What about with parallel Haskell?
- Have you tried to do something but failed?

Talking Points

- What are your experiences optimising code?
Is it a problem? what's the success rate?
- What about with parallel Haskell?
- Have you tried to do something but failed?
- Are we missing a critical feature or approach?

Talking Points

- What are your experiences optimising code?
Is it a problem? what's the success rate?
- What about with parallel Haskell?
- Have you tried to do something but failed?
- Are we missing a critical feature or approach?
- What would you like to see implemented?