Abusing Ants for Fun and Profit

How Haskell won the ICFP 2004 Programming Contest

Ian Lynagh, Andres Löh, Ganesh Sittampalam and Duncan Coutts

ICFP

International Conference on Functional Programming

Most important Functional Programming conference

Therefore most important Computer Science conference

ICFP

International Conference on Functional Programming

Most important Functional Programming conference

Therefore most important Computer Science conference

Had an associated programming contest for 7 years
Languages

The programming contest is being conducted by ICFP, which has a desire to promote functional languages. However, rather than debate the definition of a “functional programming language,” we will accept submissions implemented using any language whatsoever.

The team

Team Dunkosmiloolump consisted of:
- Duncan Coutts
- Andres Löh (kosmikus)
- Ian Lynagh (Igloo)
- Ganesh Sittampalam (Heffalump)

The team

Team Dunkosmiloolump consisted of:
- Duncan Coutts
- Andres Löh (kosmikus)
- Ian Lynagh (Igloo)
- Ganesh Sittampalam (Heffalump)

Format

Contest start, task posted:
- Friday June 4 at 17:00 BST

Lightning division deadline:
- Saturday June 5 at 17:00 BST

Contest finish:
- Monday June 7 at 17:00 BST
The board

The aim

- $8 \times 3 \times 4 \times 5 = 480$ food initially
- 100,000 timesteps
- Ants are a state machine, max 10,000 states
- Team with most food dropped on their anthill at the end wins

Programming ants

- Move st1 st2
- Turn Left st
- PickUp st1 st2
- Drop st
- Flip p st1 st2
- Mark i st
- Unmark i st

Conditions

- Sense Left Ahead st1 st2 cond
- Foe
- FriendWithFood
- FoeWithFood
- Food
- Rock
- Marker i
- FoeMarker
- Home
- FoeHome
**Fighting**

- After moving, if next to 5 enemy ants, die
- Otherwise, if now the 5th ant surrounding an enemy ant, it dies
- Dead ants turn into 3 food
- Each team has 91 ants. Potentially \((2 \times 91 - 5) \times 3 = 531\) extra food.

---

**The Bird slide**

The PRNG — From half an A4 page down to:

```haskell
mkRng :: Int -> [Int]
mkRng = drop 4
  . map ((`mod` 16384)
    . (`div` 65536))
  . iterate ((`mod` (65536*16384))
    . (1 +)
    . (22695477 *))
```

---

**Tasks**

- Simulator
- Visualiser
- Ant DSL
- Ant

---

**Simulator**

- Given 2 ants and a world, print out info for visualiser
- Slow (used immutable datastructures)
- Worked!
- Clear code allowed easy extensibility
Andres had some code lying around from a hex map based game
Slow (using wxHaskell incorrectly?)
Worked!

Ant
- Always know direction ant is facing
- Walk around looking for food, making home marker if none already
- Randomly turn after a few steps
- If you find a trail to food, follow it
- If you find food, take it home leaving a trail to where you came from
- Action after hitting something depends on what you hit

Ant DSL
- Gotos and labels
- Monadic combinators for fresh label generation
- Inline gotos, common continuation elimination, remove pointless code

Ant Stats
- 3,104 and 3,134 states (max allowed 10,000)
- Against DoNothing.ant, take 30,000-40,000 timesteps to get all the food (runs to 100,000)
Profiler

- Simulator prints stats every 100 timesteps
- gnuplot graphs stats

Results

- Team OCant are an extremely cool bunch of hackers!
- Java and C++ are very suitable for rapid prototyping. (Red Team)
- Haskell and C++ are fine programming tools for many applications. (Frictionless Bananas)
- Haskell is the language of choice for discriminating hackers! (Dunkosmiloolump)
Results

- Team OCant are an extremely cool bunch of hackers!
- Java and C++ are very suitable for rapid prototyping. (Red Team)
- Haskell and C++ are fine programming tools for many applications. (Frictionless Bananas)
- Haskell is the language of choice for discriminating hackers! (Dunkosmiloolump)

Languages

Entries in 2003

Entries in 2004