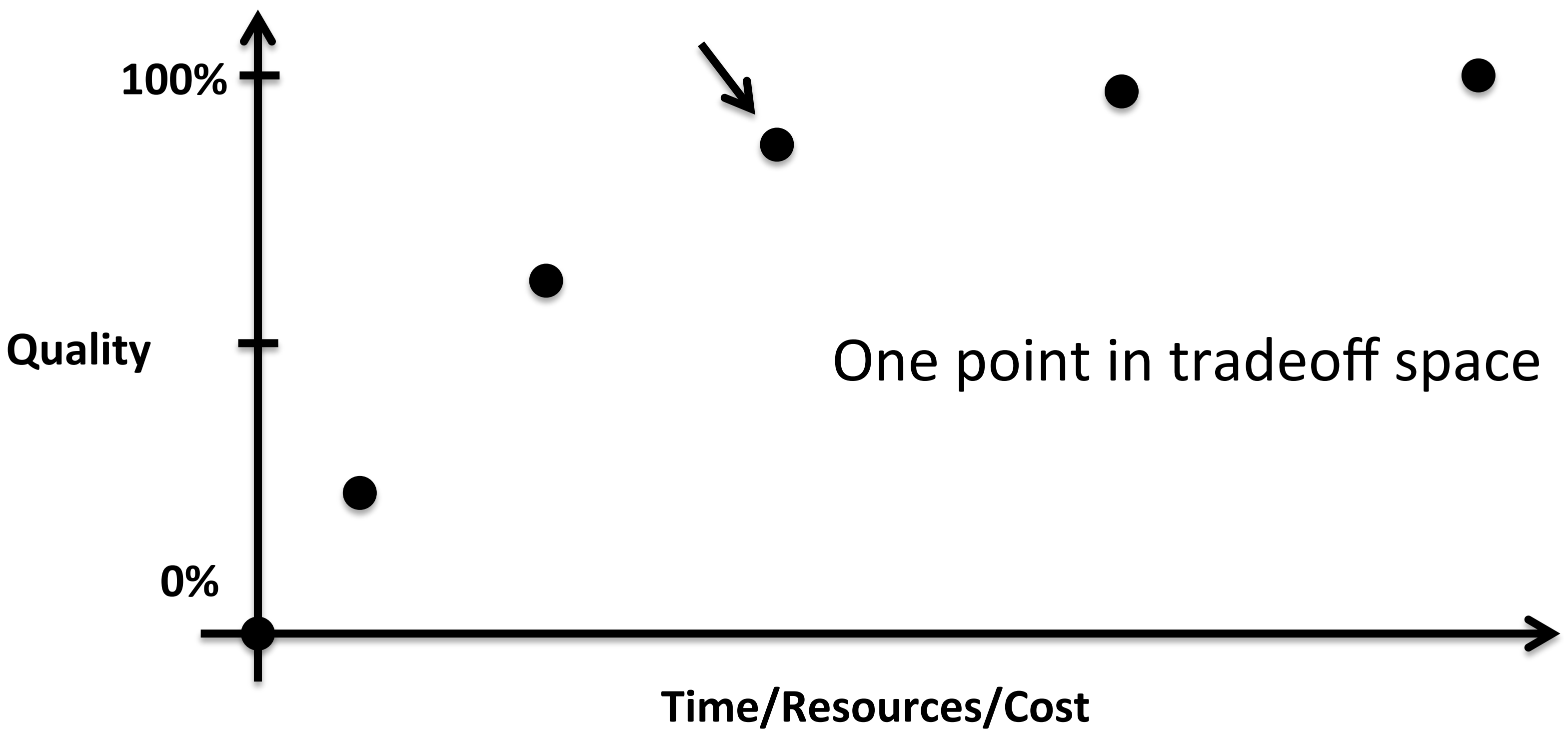
Reasoning about Relaxed Programs

Michael Carbin

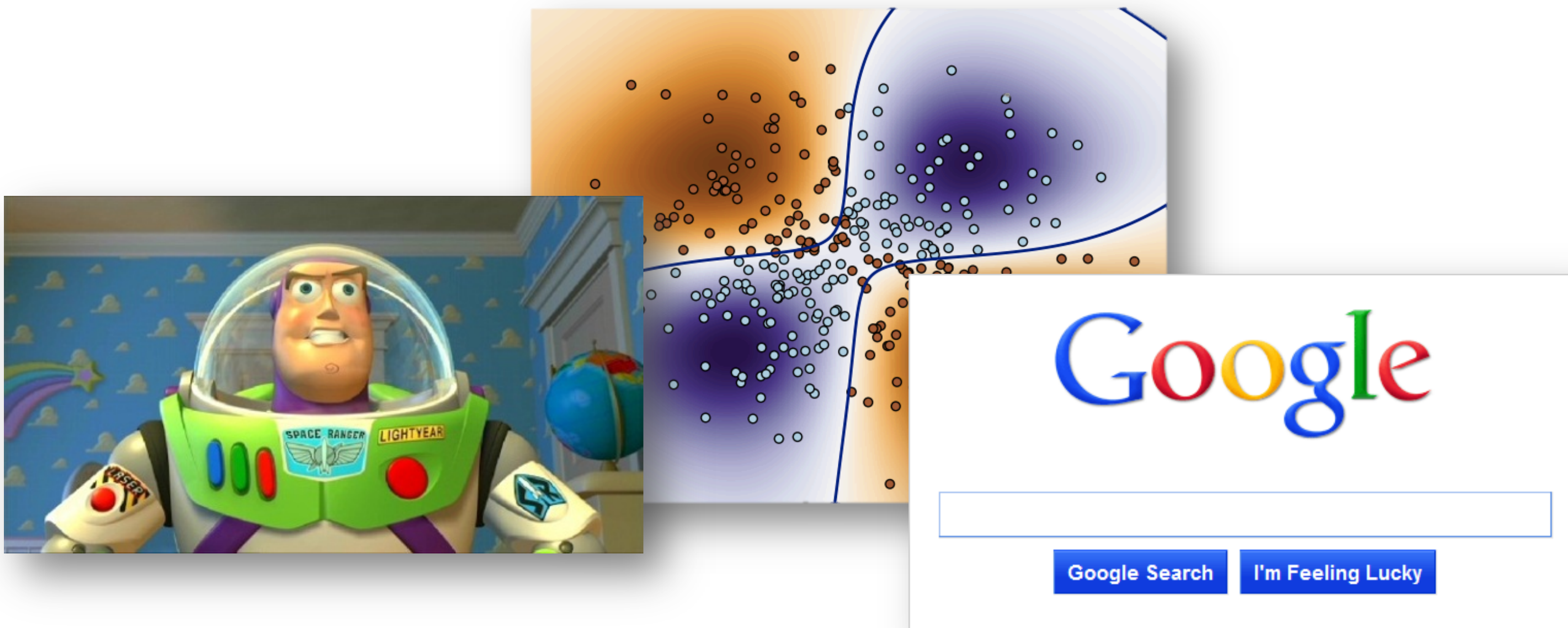
Massachusetts Institute of Technology

How do we verify the safety of relaxed programs?

Standard Program Model

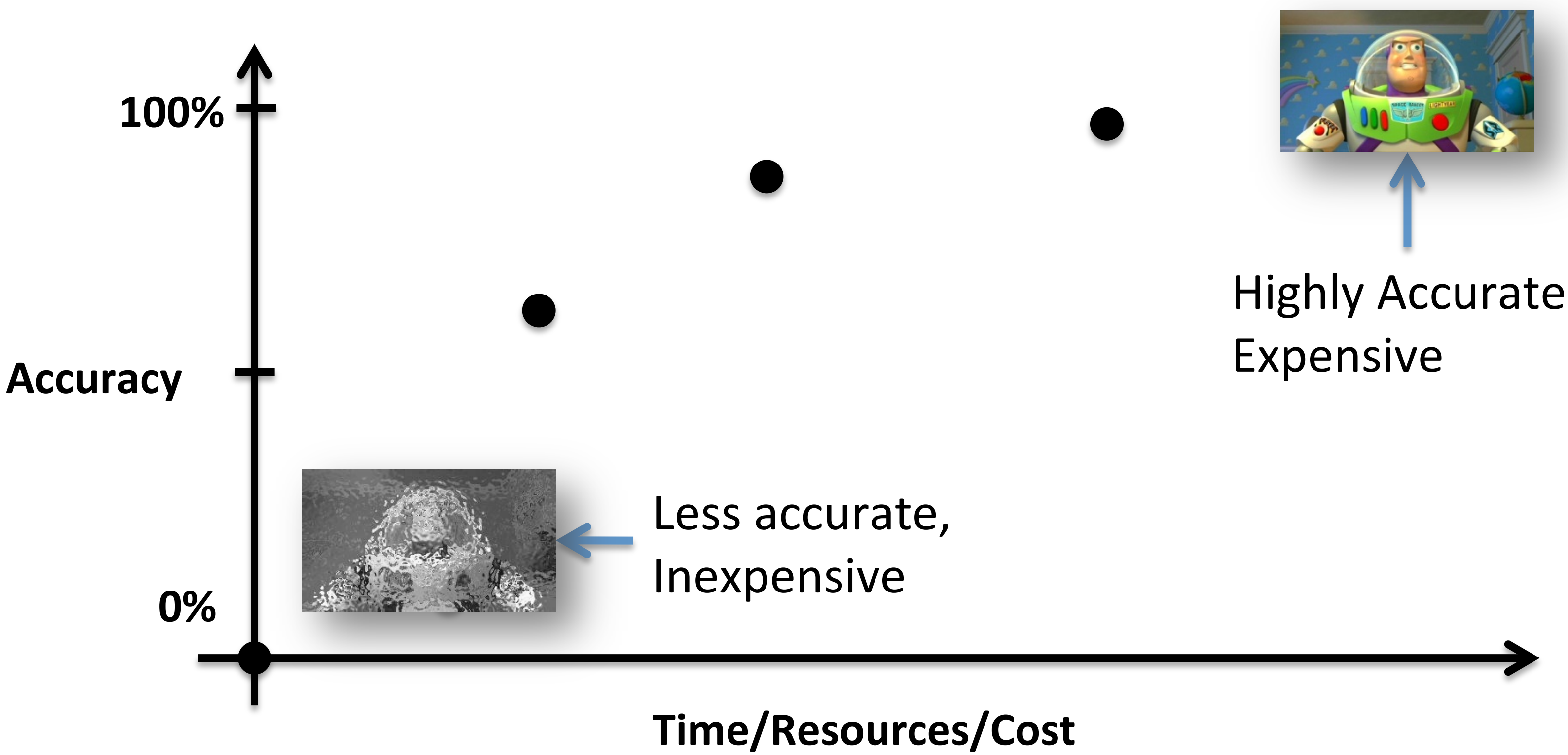


When do we think like this?

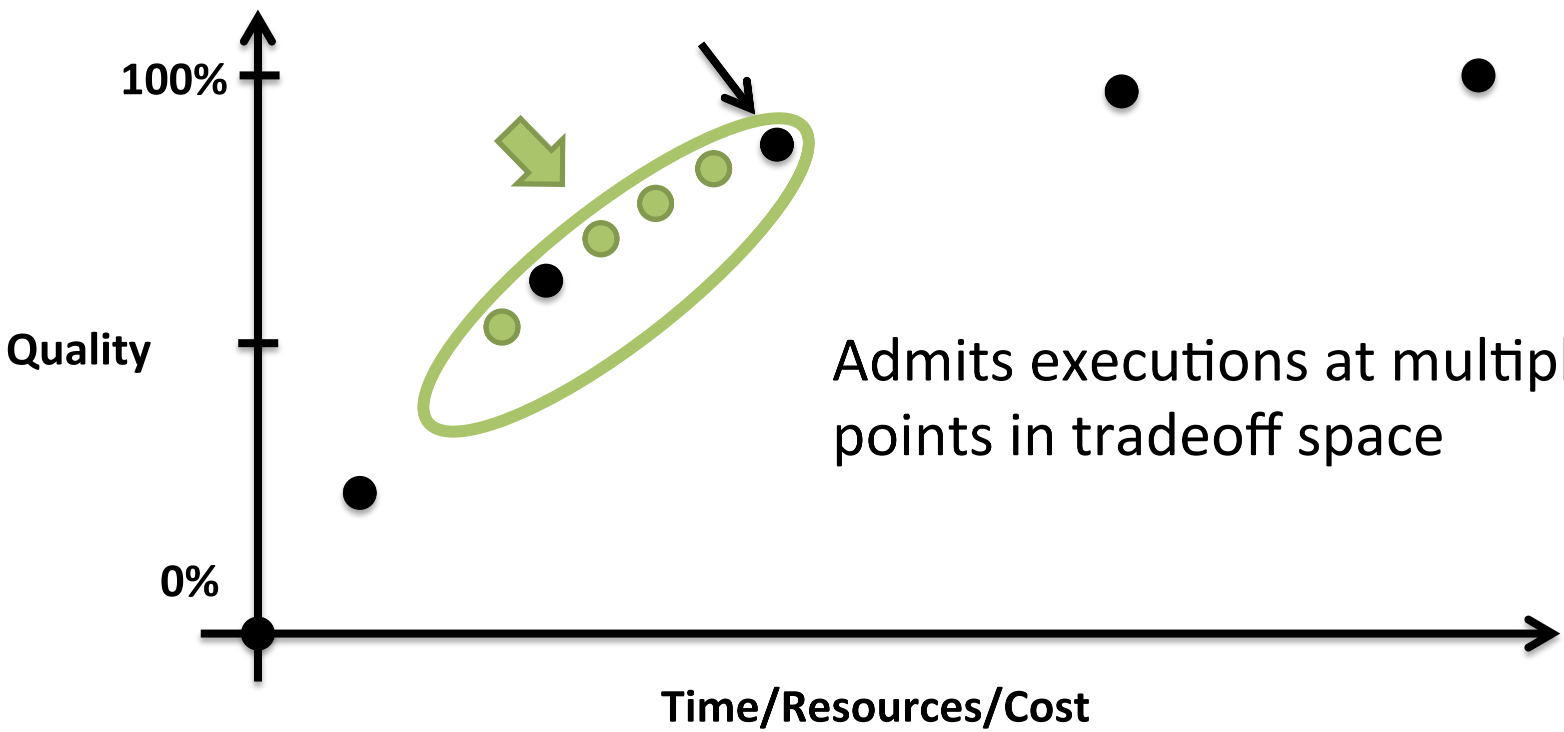


Media Processing, Machine Learning, Search

When do we think like this?



Relaxed Program Model



Admits executions at multiple points in tradeoff space

Relaxed programs can dynamically and automatically adapt

Producing Relaxed Programs

Task Skipping/Loop Perforation - Rinard ICS '06, Misailovic ICSE '10

Dynamic Knobs - Hoffmann ASPLOS '11

Approximate Memories - Lui ASPLOS '11, Sampson PLDI '11

Approximate Memoization - Chaudhuri FSE '11

Racy Parallelization - Misailovic MIT-TR '10, Rinard RACES '12

General Model for Relaxed Programs

A general primitive for relaxed sequential programs [1]:

Modified Variables

Relaxation Predicate

```
relax (n) st (n <= old(n));
for (uint i = 0; i < n; ++i) {...}
```

Loop Perforation!

[1] Proving Acceptability Properties of Nondeterministic Relaxed Approximate Programs. Carbin, Kim, Misailovic, Rinard. PLDI '12

How do we verify the safety of relaxed programs?

Program Logic (Hoare Logic)

$$\{P\}\ s\ \{Q\}$$

Standard Hoare Logic
doesn't capture what we want

```
{x = 1} x = x + 1 {x = 2}
```

Applying Standard Hoare Logic

```
<...>
{ P(x, y) && Q(y) }
relax (x) st (true);
{ Q(y) }
<...>
{ R(x, y) && S(y)}
assert R(x, y) && S(y);
```

Lose P because x is modified

Prove both R and S

- Note: relaxation doesn't modify y
- Why do we need to prove S? If S(y) holds in the original program, then it also holds in the relaxed

Alternative: Relational Program Logic

$$\left\{P_{rel}\right\}\ s\ \left\{Q_{rel}\right\}$$

$$\downarrow$$

```
{x<r> == x<o> && y<r> == y<o>}
relax (x) st (true);
{y<r> == y<o>}
```

Applying Relational Program Logic

```
<...>
{x<r> == x<o> && y<r> == y<o>}
relax (x) st (true);
{ y<r> == y<o> }
<...>
{R(x<r>, y<r>) && y<r> == y<o> }
assert R(x, y) && S(y) ;
```

x different but y the same

Only prove R

If S(y<o>) is true and y<r> == y<o> then S(y<r>) is true

Relational reasoning is the bridge

Relative Safety

If original program satisfies all assertions,

then the relaxed program satisfies all assertions

Established through any means:
verification, testing, code review

In our PLDI paper:

- Full formalization of the relaxed programming model
- Primitives for reasoning about accuracy
- Examples from racy parallelization, approximate memory, and dynamic knobs

Takeaway

```
for (uint i = 0; i < n; ++i) {...}
```

```
for (uint i = 0; i < n; i+=2) {...}
```

Relax Semantics. Preserve Safety. Reuse Proofs