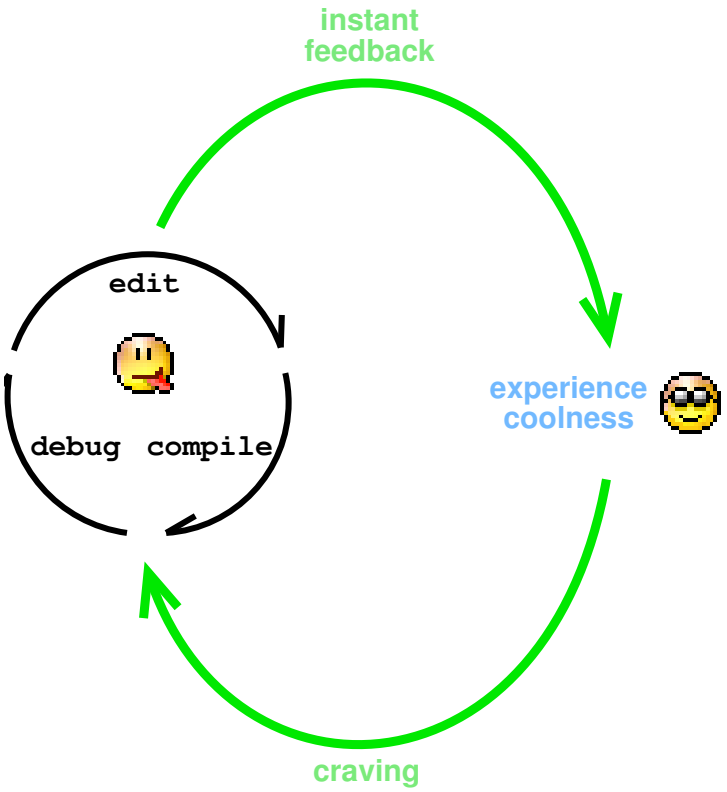
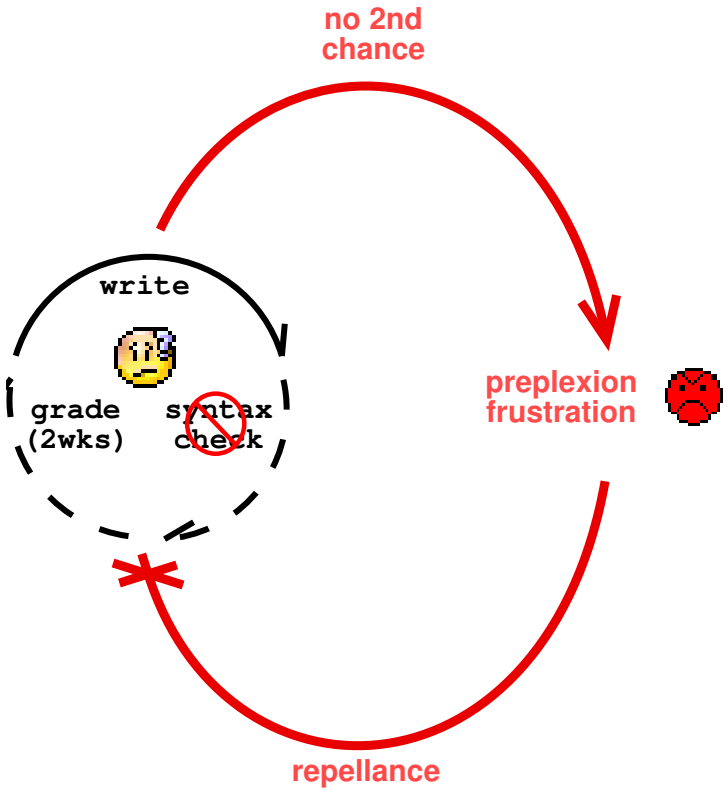
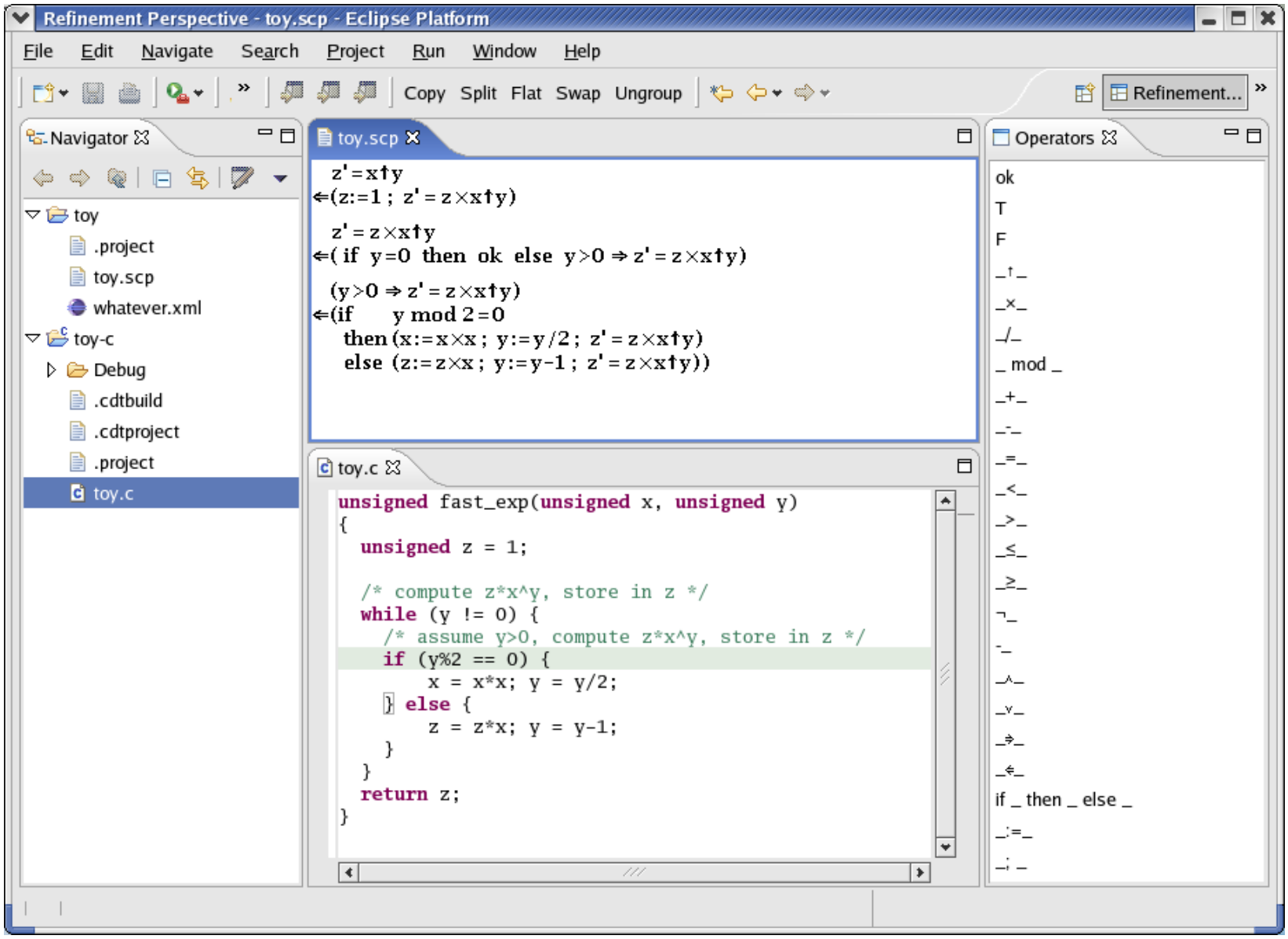
Why we like programming



Why we hate proofs



If only there were an IDE for proofs. . .


Refinement Perspective - toy.scp - Eclipse Platform
File Edit Navigate Search Project Run Window Help
Copy Split Flat Swap Ungroup
Refinement...
Navigator
toy
.project
toy.scp
whatever.xml
toy-c
Debug
.cdtbuild
.cdtproject
.project
toy.c
toy.scp
z' = x↑y
⇐(z:=1 ; z' = z×x↑y)
z' = z×x↑y
⇐(if y=0 then ok else y>0 ⇒ z' = z×x↑y)
(y>0 ⇒ z' = z×x↑y)
⇐(if y mod 2=0
then (x:=x×x ; y:=y/2 ; z' = z×x↑y)
else (z:=z×x ; y:=y-1 ; z' = z×x↑y))
toy.c
unsigned fast_exp(unsigned x, unsigned y)
{
unsigned z = 1;
/* compute z*x^y, store in z */
while (y != 0) {
/* assume y>0, compute z*x^y, store in z */
if (y%2 == 0) {
x = x*x; y = y/2;
} else {
z = z*x; y = y-1;
}
}
return z;
}
Operators
ok
T
F
_↑_
_×_
_/_
_ mod _
_+_
_-_
_=_
_<_
_>_
_≤_
_≥_
¬_
-_
_∧_
_∨_
_⇒_
_⇐_
if _ then _ else _
_:=_
_; _

Useful?

Industrial perspective:

✔ unambiguous language for internal documentation

✔ critical, core fragments of code

✔ delicate hand-optimizations of originally simple algorithms

✘ **not** substitute for testing; **not** panacea

✔ good addition to toolset

? but how many colleagues will use it?

Useful?

Educational perspective:

✔ IDE for discrete math homework!

✔ clear rules of the game—no second-guessing the TA

$\Rightarrow$ graduates mind it less and use it more at work

✘ **not** substitute for other CS/SE homework!

Future Work

- ☞ if a step fails, want explanation/counter-example (see link #4)
- ☞ if a step works by automation, want details
- ☞ syntax+type check, informative messages, heuristics, help
- ☞ bridge to/from code files; proof templates
- ☞ model objects, pointers

Links

1. Homepage: `http://www.cs.utoronto.ca/~trebla/scphEditor/`

2. We model programs with the theory from:

   E. C. R. Hehner, *A Practical Theory of Programming*.
   `http://www.cs.utoronto.ca/~hehner/aPToP/`

3. We currently use this theorem prover at the back:

   HOL: `http://hol.sourceforge.net/`

4. It would be nice to use a counterexample generator such as:

   Alloy: `http://alloy.mit.edu/`