

# On Narrative *vs.* Computer Programming

**Christos H. Papadimitriou**

UC Berkeley

 "christos"

with many thanks to **Martha Sideri**

# outline

- what is computer programming?
- the elements of c.p. in narrative
- similarities, parallels and connections between c.p. and narrative
- interleaved with above:  
narratives of programming

# what does a program do?

- defines its *data types*
- and the ways these *interact* with one another (through programs)
- a program changes the *state* of its data types
- it may *branch conditionally*
- or it may *repeat* until conditions are met
- or it may *invoke* other programs – or *itself* (!?)

```
class Fighter
```

data type

```
{
```

```
static int count; // CLASS VARIABLE: how many fighters there are
```

```
int strength = 1; // my strength
```

```
int direction; // direction I'm facing
```

```
Battleground place; // the Battleground that I fight on
```

```
int row, column; // where I am
```

```
int newRow, newColumn; // where I want to be
```

```
int lastMoved = -1; // last turn that I did something
```

attributes

birth

```
Fighter (Battleground place, int row, int column) // Construct a Fighter.
```

```
{
```

```
direction = (int) (Math.random () * 4); // face in a direction 0 to 3
```

```
this.place = place; // remember my battleground
```

```
this.row = row; // remember my location
```

```
this.column = column;
```

```
count++; // count me
```

```
void doSomething (int step)
```

interaction



```
{
```

```
// If I've already moved, don't move again
```

```
if (step == lastMoved) return;
```

```
else lastMoved = step;
```

```
// sometimes change direction (about 10% of the time)
```

```
if (Math.random () < 0.10) direction = (int) (Math.random () * 4);
```

```
// figure out where I want to be
```

```
newRow = row; newColumn = column;
```

```
switch (direction)
```

```
{
```

```
case 0: newRow = (row + 1) % place.size;          break;
```

```
case 1: newRow = (place.size + row - 1) % place.size;    break;
```

```
case 2: newColumn = (column + 1) % place.size;          break;
```

```
case 3: newColumn = (place.size + column - 1) % place.size; break;
```

```
}
```

```
// if that space is occupied, fight for it, else just move there
if (place.warzone [newRow][newColumn] != null)
    fight (newRow, newColumn);
else
    move (newRow, newColumn);
}
```

invoke  
interactions



```
void move (int newRow, int newCol) // Do a simple, uncontested move
{
    place.warzone [row][column] = null; // Move from here
    place.warzone [newRow][newColumn] = this; // to here, and
    row = newRow; column = newColumn; // remember where I am now
}
```

```
void fight (int newRow, int newColumn) // Fight someone in that location
{
    Fighter opponent = place.warzone [newRow][newColumn];
}
```

```
if (strength >= opponent.strength) // If I win,
{
    strength += opponent.strength; // take my opponent's strength
    move (newRow, newColumn); // and position;
    Fighter.count--; // he's gone now, reduce count.
}
else
{
    opponent.strength += strength; // But if I lose,
    place.warzone [row][column] = null; // erase myself
    Fighter.count--; // and count me gone.
}
}

public String toString () // Represent a fighter by just his strength
{
    if (strength < 10) return " " + strength; // add a blank if < 10
    else return "" + strength; // else just convert to String
}
```

**conditional  
branching**



```
public class Battleground
{
    int size;           // size of the battleground
    Fighter [][] warzone; // array representing the battleground

    Battleground (int size) // Construct a Battleground.
    {
        warzone = new Fighter [size][size]; // Make the array
        this.size = size; // and remember how big it is.

        for (int i = 0; i < size; i++) // Put a Fighter in 25% of
            for (int j = 0; j < size; j++) // squares (the rest are initially
                if (Math.random () < 0.25) // null).
                    warzone[i][j] = new Fighter (this, i, j);
    }

    void print () // Print the Battleground.
}
```

**another data type**

**birth**

**repetition**



```
void print ()
```

```
// Print the Battleground.
```

```
{  
  for (int i = 0; i < size; i++)  
  {  
    for (int j = 0; j < size; j++)  
    {  
      if (warzone[i][j] == null) System.out.print (" --");  
      else System.out.print (" " + warzone[i][j]);  
    }  
    System.out.println ();  
  }  
}
```

so we can  
see it



```
public static void main (String args[])
```

```
{  
  final int SIZE = 10; // Constant: size of battleground  
  final int STEPS = 10; // Constant: number of steps to run simulation  
  Battleground battleground = new Battleground (SIZE); // Make9 battleground
```

*action!*



```
for (int step = 0; step < STEPS; step++)           // Run for STEPS steps.
{
    System.out.println ("Step " + step + ", " +
        Fighter.count + " fighters:");
    battleground.print ();

    if (Fighter.count == 1) break;                // Quit early if we have a winner,
    for (int i = 0; i < SIZE; i++)                // else loop through battleground
        for (int j = 0; j < SIZE; j++)          // and let each Fighter doSomething.
            if (battleground.warzone[i][j] instanceof Fighter)
                battleground.warzone[i][j].doSomething (step);
}
System.out.println ("At end (" + Fighter.count + " fighters left:");
battleground.print ();
}
}
```

...and that's the whole  
program

# recall: what does a program do?

- defines its *data types*
- and the ways these *interact* with one another (through programs)
- a program changes the *state* of its data types
- it may *branch conditionally*
- or it may *repeat* until conditions are met
- or it may *invoke* other programs – or *itself* (!?)

# data types $\Leftrightarrow$ characters

- their definition creates the *diegesis* (the “world” of the story)
- their complexity is that of the story
- *polymorphism*: same stimuli bring different responses

# data types $\Leftrightarrow$ characters (*cont.*)

- pure diegeses: computer games
- real life as pure diegesis: e.g, the banking world

# programming narratives I

ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS



THE  
MYTHICAL  
MAN-MONTH

FREDERICK P. BROOKS, JR.

*“...The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are [...] so readily capable of realizing grand conceptual structures...”*

Fred Brooks

# repetition? branching? subroutine call? recursion?

- Repetition/iteration is a signature feature of the folk tale:

*“the first day, the older son tried to cross the river, but the dragon...”*

# narrative with branching: the interactive novel

“if you want Guinevere to fall in love with Lancelot and leave Arthur click [here](#)”

*(nb: another form of pure diegesis)*



# program invocation: the nested narrative

- stories within stories (plays too...)

e.g., *The Blind Assassin*



 Iris' s story

 The lovers' story

 *The Blind Assassin*

# program invocation: the nested narrative

- stories within stories (plays too...)

e.g., *The Blind Assassin*



 Iris' s story

 The lovers' story

 *The Blind Assassin*



stack

# program invocation: the nested narrative

- stories within stories (plays too...)

e.g., *The Blind Assassin*



 Iris' s story

 The lovers' story

 *The Blind Assassin*



stack

# program invocation: the nested narrative

- stories within stories (plays too...)

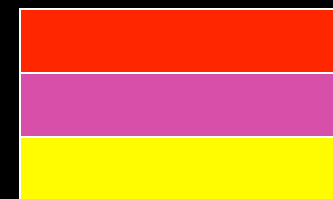
e.g., *The Blind Assassin*



 Iris' s story

 The lovers' story

 *The Blind Assassin*



stack

# program invocation: the nested narrative

- stories within stories (plays too...)

e.g., *The Blind Assassin*



 Iris' s story

 The lovers' story

 *The Blind Assassin*



stack

# program invocation: the nested narrative

- stories within stories (plays too...)

e.g., *The Blind Assassin*



 Iris' s story

 The lovers' story

 *The Blind Assassin*



stack

# program invocation: the nested narrative

- stories within stories (plays too...)

e.g., *The Blind Assassin*



# program invocation: the nested narrative

- stories within stories (plays too...)

e.g., *The Blind Assassin*



 Iris' s story

 The lovers' story

 *The Blind Assassin*



stack



# recursion in narrative?

self-referential  
and self-aware  
narrative

- not a new idea...
- *If on a Winter's Night a Traveler*



# so, what can programs and stories can have in common?

- programs must “compile and run” (i.e., be correct enough so they can be executed on a computer)
- stories must get published, be read, “work”
- programs usually contain *bugs* that prevent them from accomplishing these
- often so do novels
- *bug or feature?*

# programming narratives II



*“A computer can execute millions of instructions in a second. The human brain, in comparison, is painfully slow. The memories of a single year, for instance, took me a full thirty seconds to recall...”*

Ellen Ullman *The Bug*

so, what *else* can programs and stories can have in common?

- programs are *intentions, ploys*; they only have a tentative existence until they are actually executed on a computer
- stories too: they are just the author's intention to induce certain emotional reactions to a reader
- (by the way: genomes also...)

# stories can help in the teaching of programming

*mythematiCS* noun, plural but  
*plural&singular in use, nlgsm/slpn*

from *Gr myth* (= story that serves to  
unfold a world view or explain a practice,  
belief, or natural phenomenon)

**1:** the use of story-telling in the teaching  
of computer science (CS) and  
mathematics

# the three modes of *Myth*ematiCS

1. Historical/biographical context (e.g., Galois, Turing, Al Khwarizmi,...)
2. Storied illustration (e.g., Theseus and Ariadne; “Procopia”; *Copenhagen; Incompleteness* )

“There is no idea worth explaining that cannot be explained by a good story”

3. Extreme mythematics: Embedding the material in a story

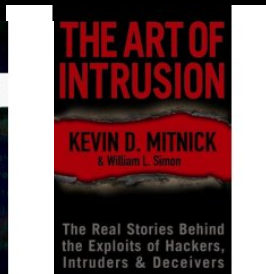
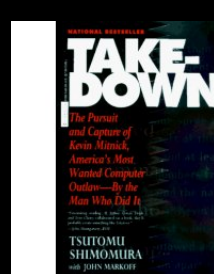
# but aren't they both just grammars?

- in computer programming, linguistic formalism has been rampant in the 1960s-70s
- it has only helped in program analysis, and there mostly with the superficial (syntactic) structure of programs
- in the theory of the narrative: *ditto?*

# programming narratives III: the hacker crime story



the Morris worm



the takedown  
of Kevin Mitnick



the “I love you” virus



both are  
“mind-bogglingly combinatorial”

- The creation of a more than minimally complex narrative appears to involve the same kind of *puzzle-solving* one needs to apply when writing programs (*and proving theorems, by the way*)

*e.g., fill in the blanks...*

underconstrained

*puzzles!*

— — — — —  
*r* — — — — —

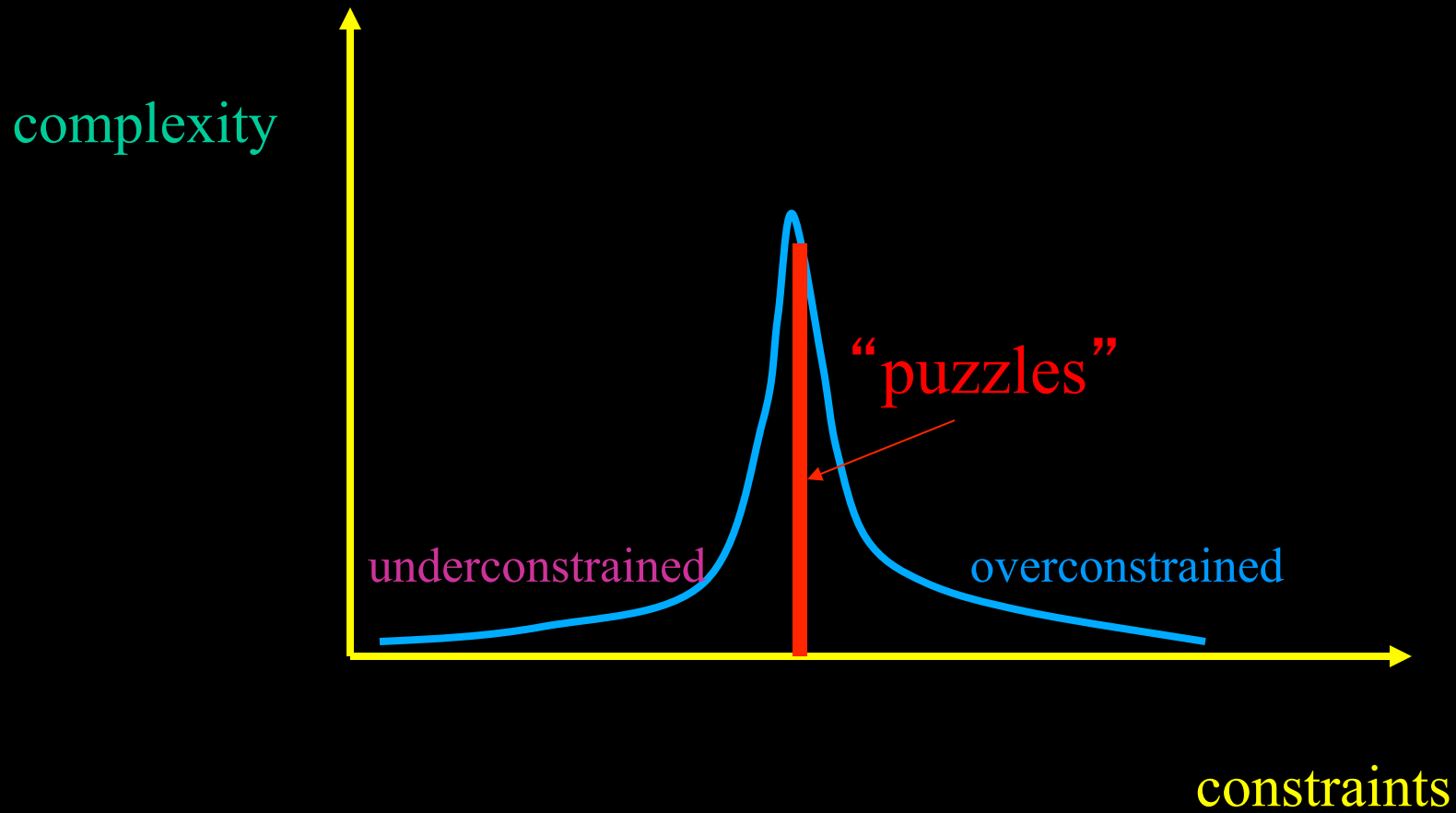
— — *q* — — — — *m*  
— — *r o* — *c*

*r* — *w s b* —

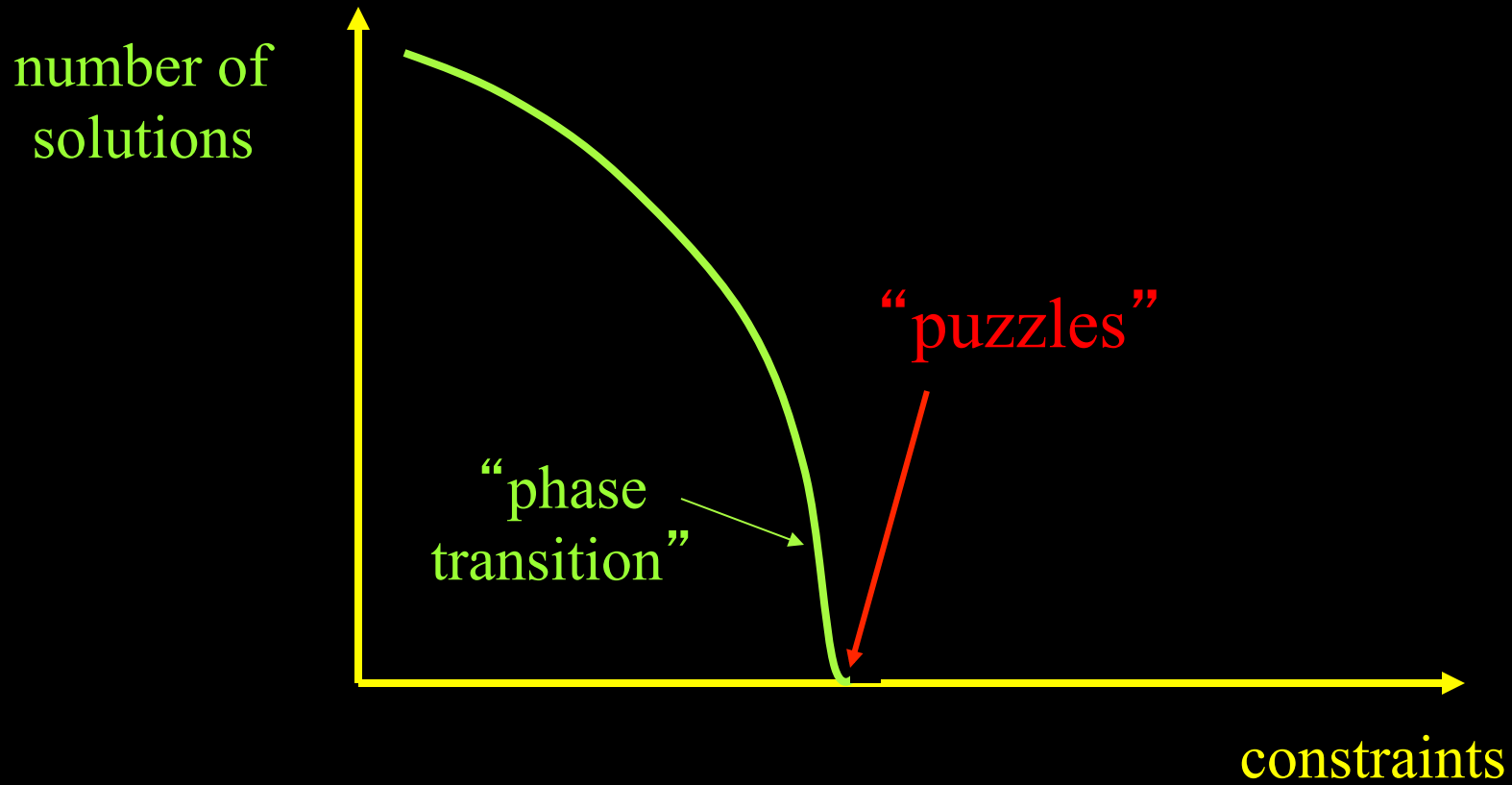
*e f* — *k m a*

overconstrained

# the diagram



# the diagram



# finally: programs of narration (or: the heroes of NI)

- 1960 - 80: understanding language
- post 1990 : the new era
  - narrative interfaces and agents
  - story databases
  - story understanding
  - narrative memory/intelligence
  - story generation

# Human or Computer Poet?

1. is beauty itself  
that they were walking there. All along the new world naked,  
cold, familiar wind -
2. Pink confused with white  
flowers and flowers reversed  
take and spill the shaded flame  
darting it back  
into the lamp's horn
3. The winds of the oozy woods which wear  
the ocean, with azure moss and flowers  
So sweet, the purple even  
I sleep in the arrows  
Of the dome of death.
4. O thou,  
Who moved among some fierce Maenad, even among noise  
and blue  
Between the bones sang, scattered and the silent seas.
5. She eyes me with an ingrown eye,  
in the rhythm of teacup tapping  
thinks of sweeping away crumbs

# Human or Computer Poet?

6. At six I cannot pray:  
Pray for lovers,  
through narrow streets  
And pray to fly  
But the Virgin in their dark wintry bed
7. What seas what shores what granite islands towards my timbers  
and woodthrush calling through the fog  
My daughter.
8. Imagine now a tree in white sails still whirled  
About the leaves  
will be of silences  
Calm and angels
9. -and the sun, dipping into the avenues  
streaking the tops of  
the irregular red houselets, and  
the gay shadows dropping and dropping.
10. The morning and already  
a perfect if slightly paled  
old park turned with young women  
seized in amber

# Answers (1-5)

1. is beauty itself  
that they were walking there. All along the new world naked,  
cold, familiar wind - **Computer**
2. Pink confused with white  
flowers and flowers reversed  
take and spill the shaded flame  
darting it back  
into the lamp's horn **William Carlos Williams**
3. The winds of the oozy woods which wear  
the ocean, with azure moss and flowers  
So sweet, the purple even  
I sleep in the arrows  
Of the dome of death. **Computer**
4. O thou,  
Who moved among some fierce Maenad, even among noise  
and blue  
Between the bones sang, scattered and the silent seas. **Computer**
5. She eyes me with an ingrown eye,  
in the rhythm of teacup tapping  
thinks of sweeping away crumbs **Raymond Kurzweil**



# Answers (6-10)

6. At six I cannot pray:  
Pray for lovers,  
through narrow streets  
And pray to fly  
But the Virgin in their dark wintry bed  
**Computer**
7. What seas what shores what granite islands towards my timbers  
and woodthrush calling through the fog  
My daughter.  
**T.S. Eliot**
8. Imagine now a tree in white sails still whirled  
About the leaves  
will be of silences  
Calm and angels  
**Computer**
9. -and the sun, dipping into the avenues  
streaking the tops of  
the irregular red houselets,  
and  
the gay shadows dropping and dropping.  
**William Carlos Williams**
10. The morning and already  
a perfect if slightly paled  
old park turned with young women  
seized in amber  
**Raymond Kurzweil**