

No Ifs, Ands, or Buts

Uncovering the Simplicity of Conditionals

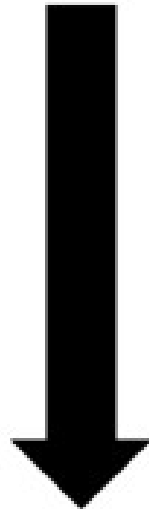
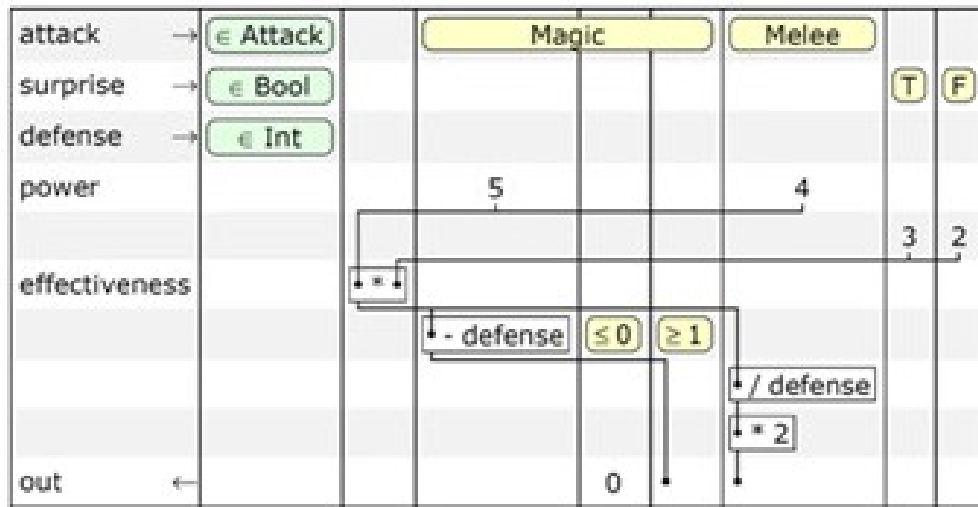
Jonathan Edwards

MIT Computer Science and Artificial Intelligence Lab

Film, który pokazywałam podczas prezentacji, można znaleźć pod adresem:
<http://www.subtextual.org/subtext2.html>

Weronika Majewska

Schematic Tables



Computation



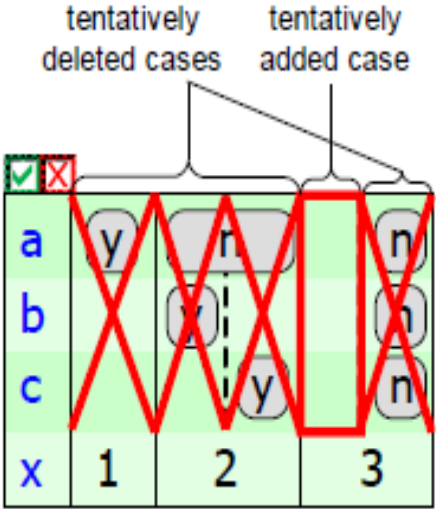
Logic

```

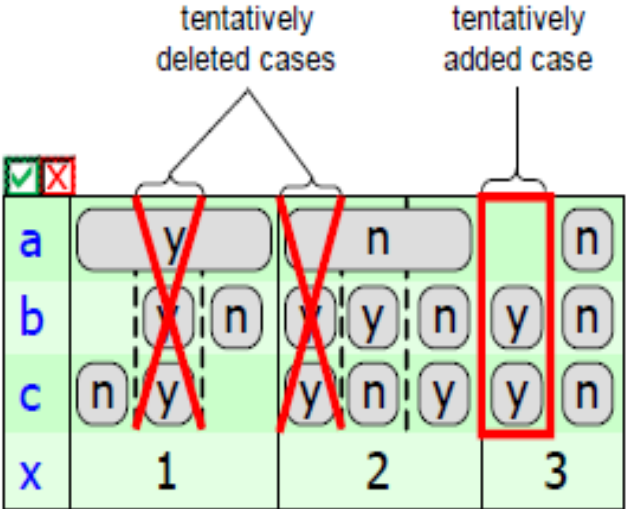
boolean a, b, c;
int x;
if (a) {
    x = 1;
} else if (b | c) {
    x = 2;
} else {
    x = 3;
}

```

a	y	n	
b		y	n
c		y	n
x	1	2	3



(a)



(b)

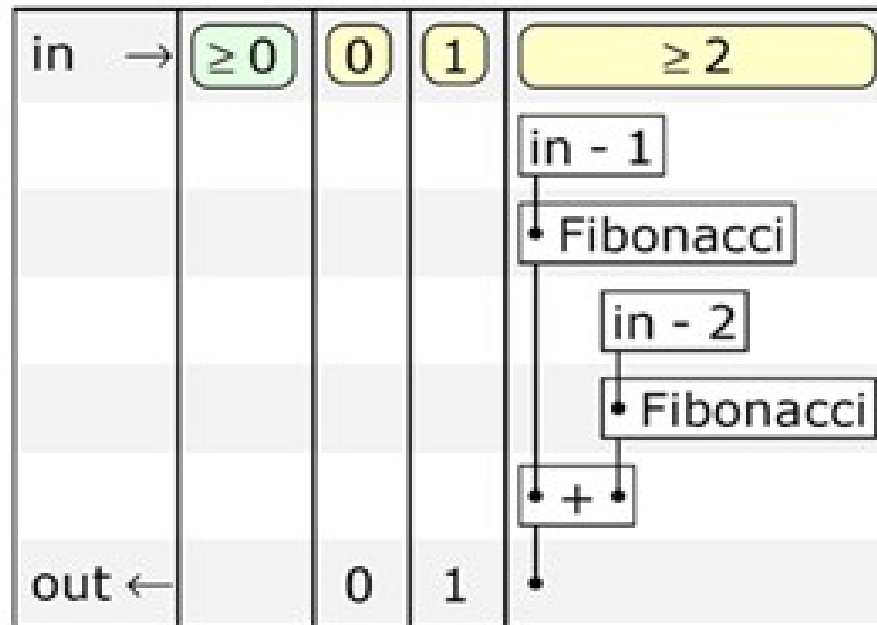
a	y		n	
b		n	y	n
c	n		y	n
x	1	2	3	

(c)

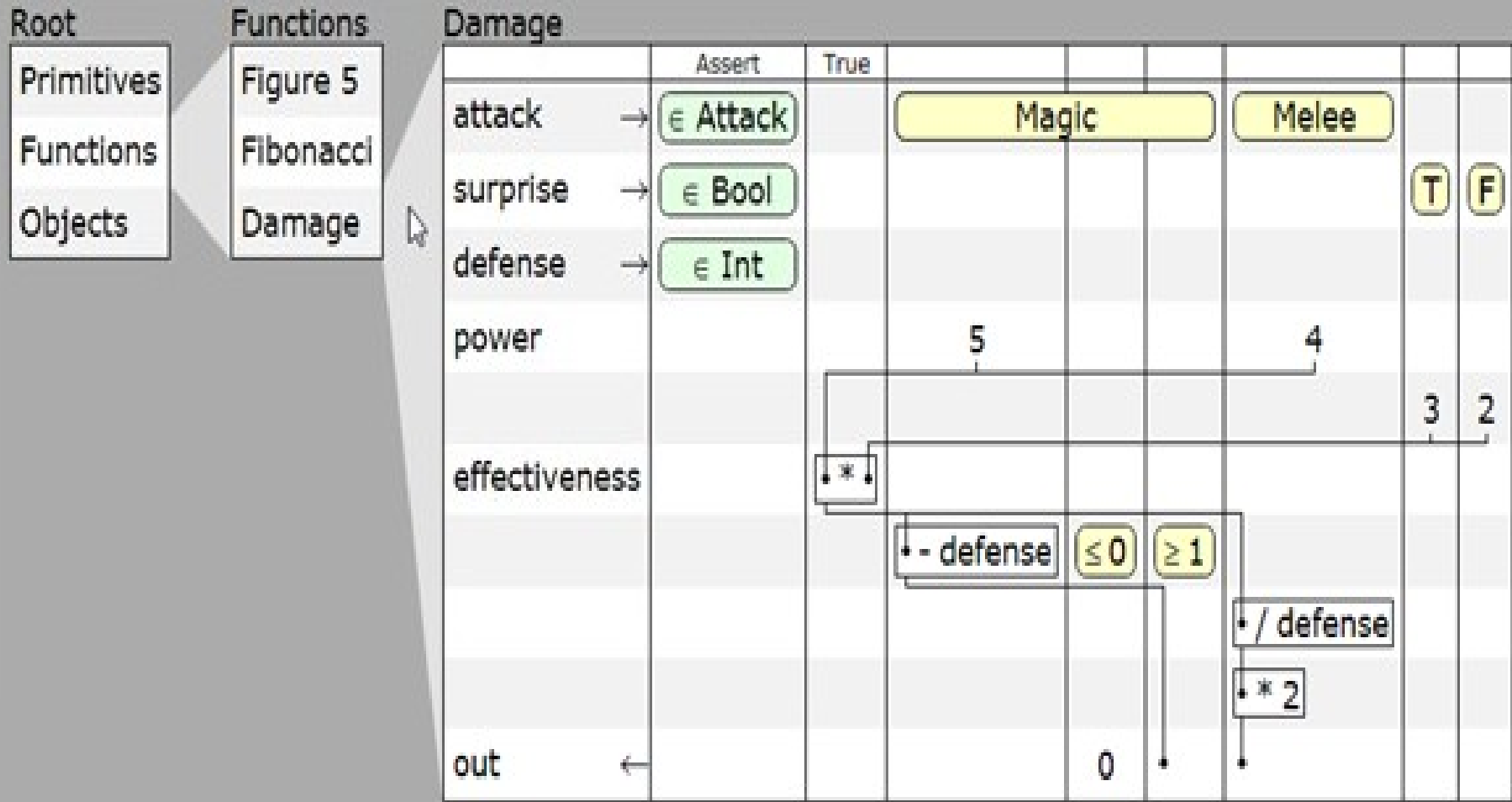
```

static int fibonacci(int n) {
    assert (n >= 0);
    if (n == 0) return 0;
    if (n == 1) return 1;
    return fibonacci(n - 1)
        + fibonacci(n - 2);
}

```



```
enum Attack {Magic, Melee};
int damage(Attack attack, bool surprise, int defense) {
    int power;
    switch (attack){
        case Magic:
            power = 5;
            break;
        case Melee:
            power = 4;
            break;
    }
    int effectiveness = power * (surprise ? 3 : 2);
    switch (attack){
        case Magic:
            if (effectiveness >= defense)
                return effectiveness - defense;
            return 0;
        case Melee:
            return (effectiveness / defense) * 2;
    }
}
```



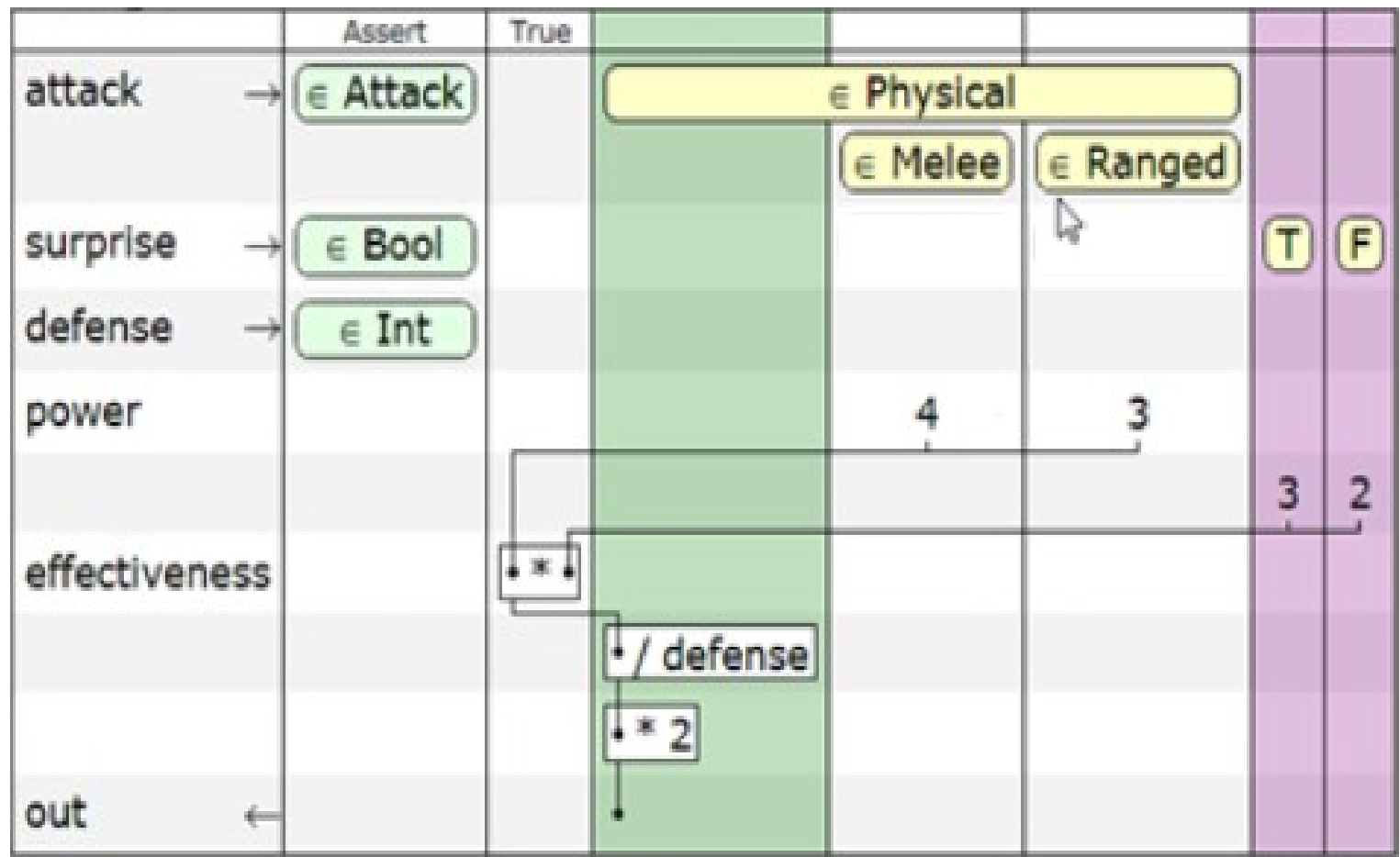
```
abstract class Attack {
    abstract int damage(bool surprise, int defense);
    int effectiveness(bool surprise) {
        return power() * (surprise ? 3 : 2);
    }
    abstract int power();}
```

```
class Magic extends Attack {
    int damage(bool surprise, int defense){
        int theDamage = effectiveness(surprise) - defense;
        if (theDamage > 0)
            return theDamage;
        return 0;
    }
    int power() {
        return 5;
    }
}
```

```
abstract class Physical extends Attack {
    int damage(bool surprise, int defense){
        return (effectiveness(surprise) / defense) * 2;
    }
    int power() {
        return 4;
    }
}
```

```
class Melee extends Physical {
}
```

```
class Ranged extends Physical {
    int power() {
        return 3;
    }
}
```



History is not over

