Academic Writing

William Cook
Univ. of Texas Austin

Encode
a complex web
of ideas

…as a linear stream of text.

how?

JOSEPH M. WILLIAMS

STYLE

TOWARD CLARITY
AND GRACE

about me

William Cook

- High school drop-out
- PhD Brown 1989
- HP Labs: Foundations of OOP
  - Learn writing the hard way
- Industry
  - AppleScript
  - BAM!, Net-It, Allegis
- Assistant Prof UT 2003

paper
organization

≠

research
process

Criteria

Significance	Motivate why the research is important or useful. Explain what problem it addresses.
Clarity	Organize the paper well and write clearly. Make sure you support your claims.
Novelty	Extend the frontier of knowledge. Explicitly relate your research to previous work.
Correctness	Critically evaluate and support your claims with proofs, an implementation, examples, or experiments.

Clarity

- Subject of sentence names a <u>character</u>

- Verbs name <u>action</u> involving characters

Missing Subjects

"Termination occurred after 23 iterations"

Missing Subjects

character

"The program terminated after 23 iterations"

action

Weak Verbs

"The algorithm <u>supports</u> effective garbage collection in distributed systems"

Stronger

"The algorithm collects garbage effectively in distributed systems"

Nominalization

Noun instead of verb/adjective

Verb NOM

Verb	*Nominalization*
discover	discovery
move	movement
collaborate	collaboration

Adjective NOM

Adjective	*Nominalization*
difficult	difficult<u>y</u>
applicable	applicabil<u>ity</u>
different	difference

empty verb + NOM

"The police <u>conducted</u> an <u>investigation</u> of the matter"

Verb = Action

"The police
<u>investigated</u>
the matter"

Many more cases

See "the book"

Cohesion

Managing information flow

Sentences

subject	• ideas already mentioned • familiar ideas
verb	• action
object	• new ideas

Topics form a logical sequence of ideas

Emphasis

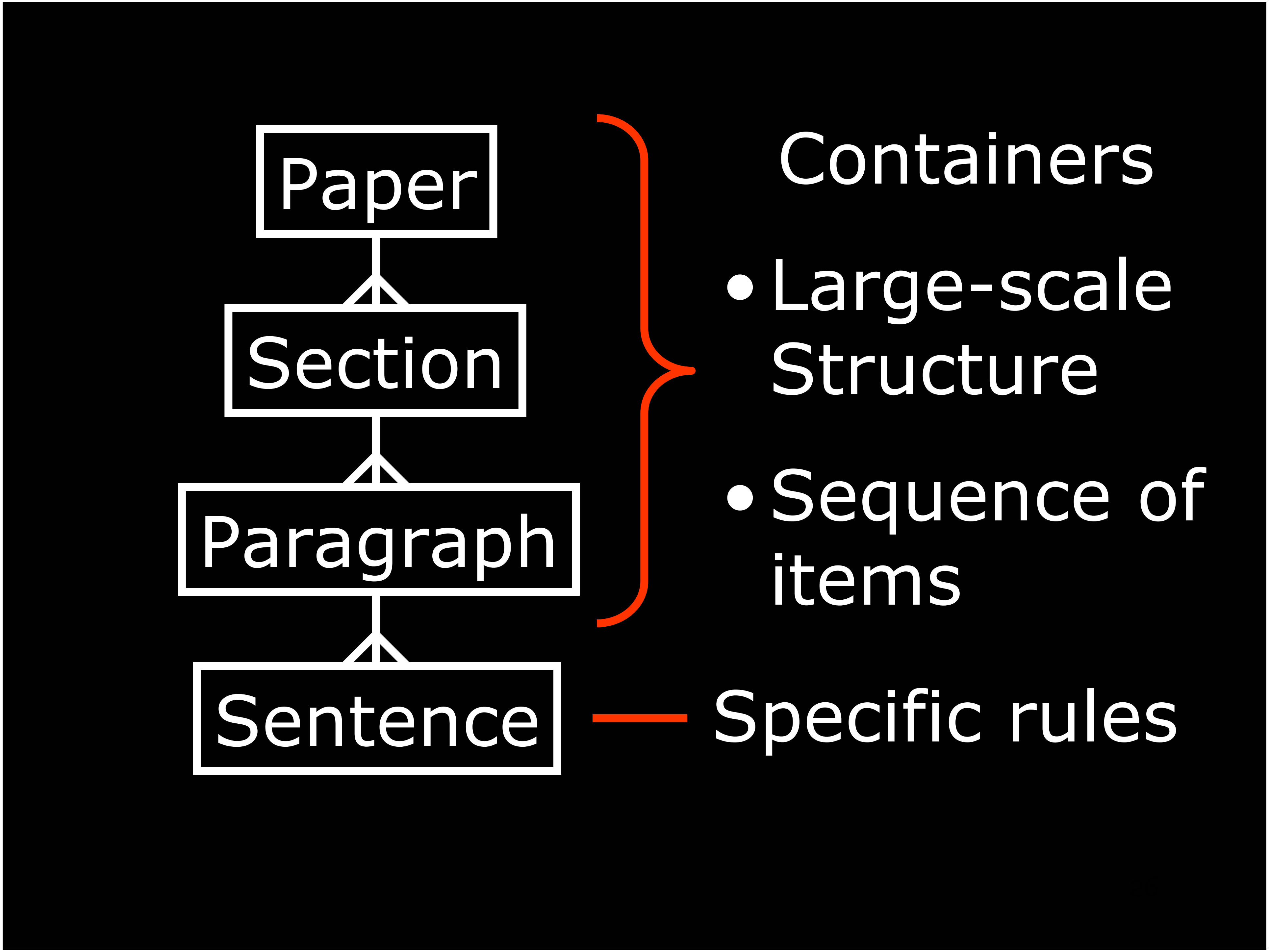
Put important things at the end

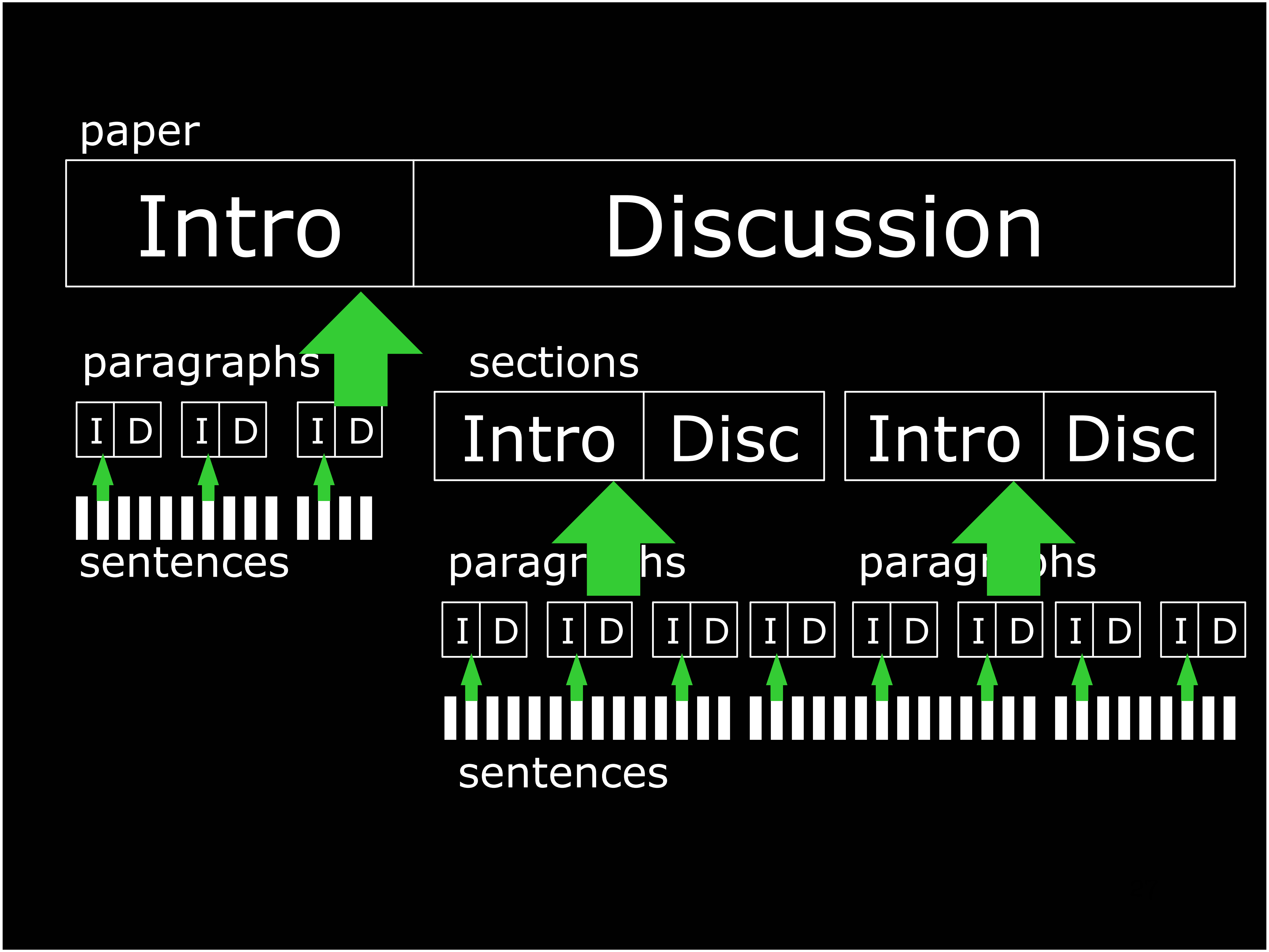
sentence	final words
paragraph	last sent.
section	last para.

Coherence

Where's the point?

The Point


Paper
Section
Paragraph
Sentence
Containers
• Large-scale Structure
• Sequence of items
Specific rules


paper
Intro
Discussion
paragraphs
I D
I D
I D
sentences
sections
Intro
Disc
Intro
Disc
paragraphs
paragraphs
I D
I D
I D
I D
I D
I D
I D
I D
sentences

Active

Passive

Passive
is fine,
if it is more
coherent

Active

"Our partners were old friends… but they let us down. The partners broke the agreement."

Passive

"We thought we had a good agreement. Then we found out who killed it. The agreement was broken by the partners."

Miscellaneous Rules

Section Title Rule

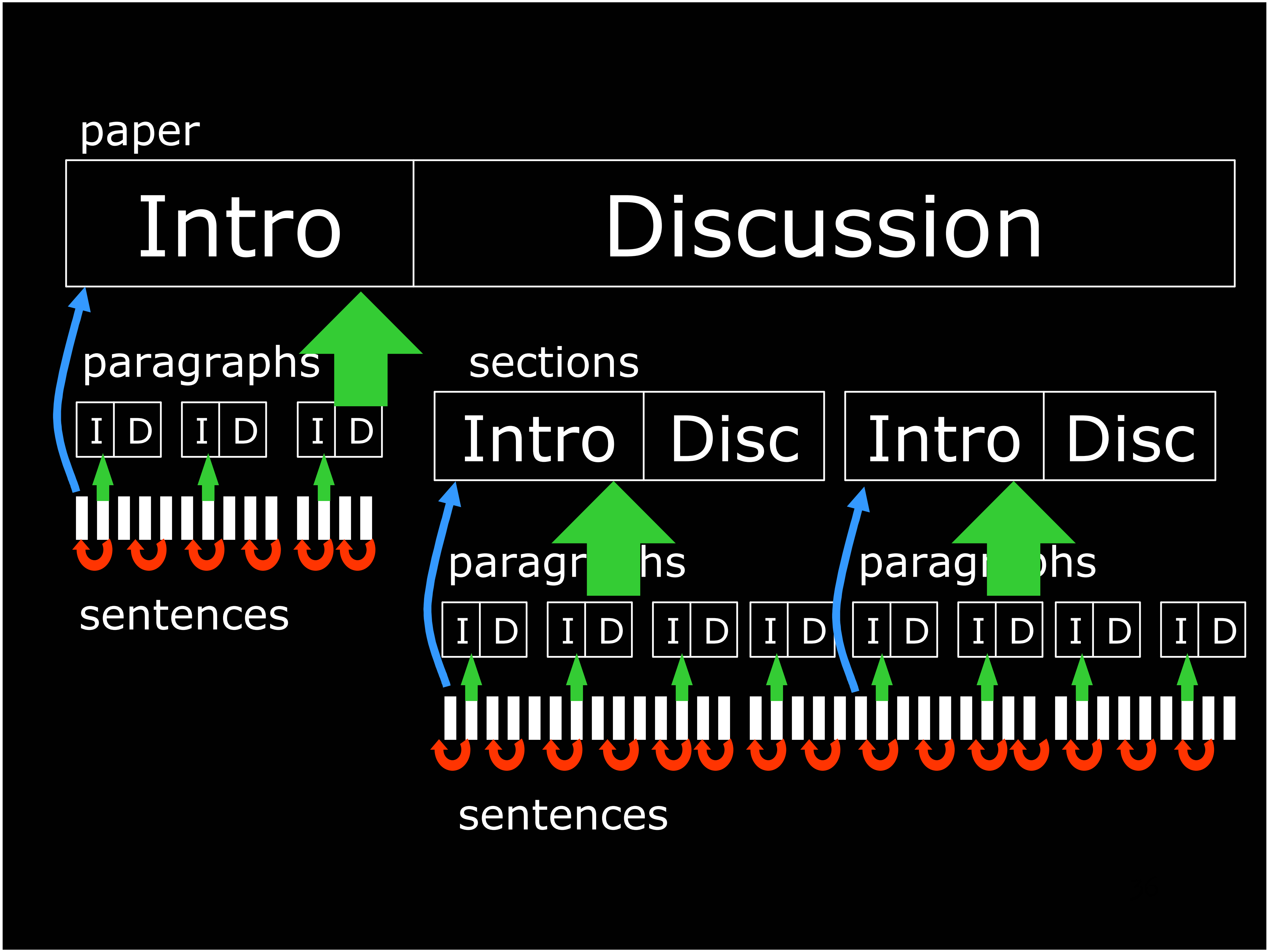
First sentence
of every section:
Must include the
section title

(except intro/conclusion)

Little Piggy Rule

Avoid “we” as subject,
unless it is
something you,
the author,
actually did

Summary


paper
Intro
Discussion
paragraphs
I D
I D
I D
sentences
sections
Intro
Disc
Intro
Disc
I D
I D
I D
I D
I D
I D
I D
I D
sentences

Examples

This paper formalizes the notion of virtual classes, in the form of the language *vc*, an extension of Featherweight Java. We present its dynamic semantics and static type rules, and show that the type system is sound.

Let us introduce virtual classes by analogy. Mainstream object-oriented languages invariably enable (virtual) methods to mean different things in context of objects of different type, at the syntactic level by means of overriding definitions of methods in subclasses, and in the dynamic semantics by means of late binding in method invocations. A1

Virtual classes are class valued attributes of objects, and they can also be refined (to subclasses) in context of a subclass; at the syntactic level there are introductory and further-binding declarations, and at the dynamic level there is late binding. ….

Virtual classes are class-valued attributes of objects. They are analogous to virtual *methods* in traditional object-oriented languages: they follow similar rules of definition, overriding and reference. In particular, virtual classes are defined within an object's class. They can be overridden and extended in subclasses, and they are accessed relative to an object instance, using late binding. This last characteristic is the key to virtual classes: it introduces a dependence between static types and dynamic instances, because dynamic instances contain classes that act as types.

…

A2

2 ISA Description

The PowerPC ISA has some features that make it different from the Alpha and PISA ISAs. For example, the Alpha ISA has 25 instructions with 4 formats and the PISA ISA has 135 instructions with 4 formats. Not all of these instructions are implemented in the simulator. In this section, we describe features of the ISA that are implemented in the simulator.

3 TRIPS Architecture

The TRIPS architecture is designed to address key challenges posed by next-generation technologies—power efficiency, high concurrency on a latency-dominated physical substrate, and adaptability to the demands of diverse applications [10, 12]. It uses an EDGE ISA [2], which has two defining characteristics: *block atomic execution* and *direct instruction communication*. The ISA aggregates large groups of instructions into blocks which are logically fetched, executed, and committed as an atomic unit by the hardware. This model amortizes the cost of per-instruction overheads such branch predictions over a large number of instructions. With direct instruction communication, instructions within a block send their results directly to the consumers without writing the value to the register file, enabling lightweight intra-block dataflow execution.