THE UNIVERSITY of EDINBURGH
informatics

Paul Anderson
(dcspaul@ed.ac.uk)

Herry
(h.herry@sms.ed.ac.uk)

Formalising Configuration Languages

Why is this important in practice?

☞ **A precise definition enables us to create multiple implementations of compiler which are truly compatible and “correct”.**

☞ **A precise definition can be used to implement supporting tools such as IDEs, graphical tools, analysers, etc.**

☞ **Formalisation process gives us a deeper understanding of the language and highlights problems with the language design.**

☞ **Based on the formal semantics, we can prove properties of the configuration making it highly reliable.**

☞ **Formal semantics allows others people to experiment with compatible language extensions.**

```
A extends {
  allow "x";
  deny  "y";
}
B extends A {
  deny  "paul";
}
C extends B {
  deny  "herry";
  allow "all";
}
```

☞ ***Does “herry” have access to machine C?***

$$put : \mathcal{S} \times \mathbb{I} \times \mathcal{V} \rightarrow \mathcal{S}$$

$$put(\varnothing_{\mathcal{S}}, id, v) := \langle id, v \rangle :: \varnothing_{\mathcal{S}}$$

$$put(\langle id, v_s \rangle :: s', id, v) := \langle id, v \rangle :: s'$$

$$put(\langle id_s, v_s \rangle :: s', id, v) := \langle id_s, v_s \rangle :: put(s', id, v)$$