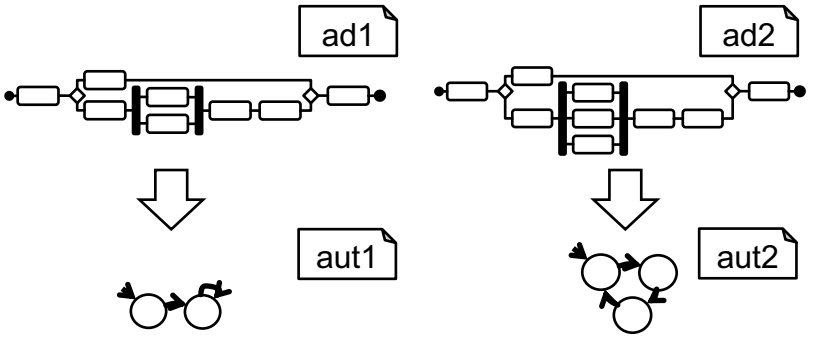
SE SOFTWARE ENGINEERING

RWTH AACHEN UNIVERSITY

Semantic Differencing of Activity Diagrams by a Translation into Finite Automata

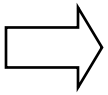


$sem(\text{ad1}) \subseteq sem(\text{ad2}) \Leftrightarrow \mathcal{L}(aut1) \subseteq \mathcal{L}(aut2)$

Kautz, Rumpe
Software Engineering
RWTH Aachen

http://www.se-rwth.de/

Agenda



1. Context
2. Example
3. Activity Diagram Variant
4. Translation to Finite Automata

Activity Diagram Evolution Analysis

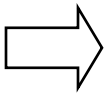
- Syntactic AD evolution analysis
  - $ad_1 \oplus \Delta(ad_1, ad_2) = ad_2$
  - But what about semantics?

- $ad_1 \neq ad_2$ but $sem(ad_1) = sem(ad_2)$ is possible

- Semantic AD evolution analysis
  - $sem(ad_1) \subseteq sem(ad_2)$?
  - If not, output witness $w \in sem(ad_1) \setminus sem(ad_2)$

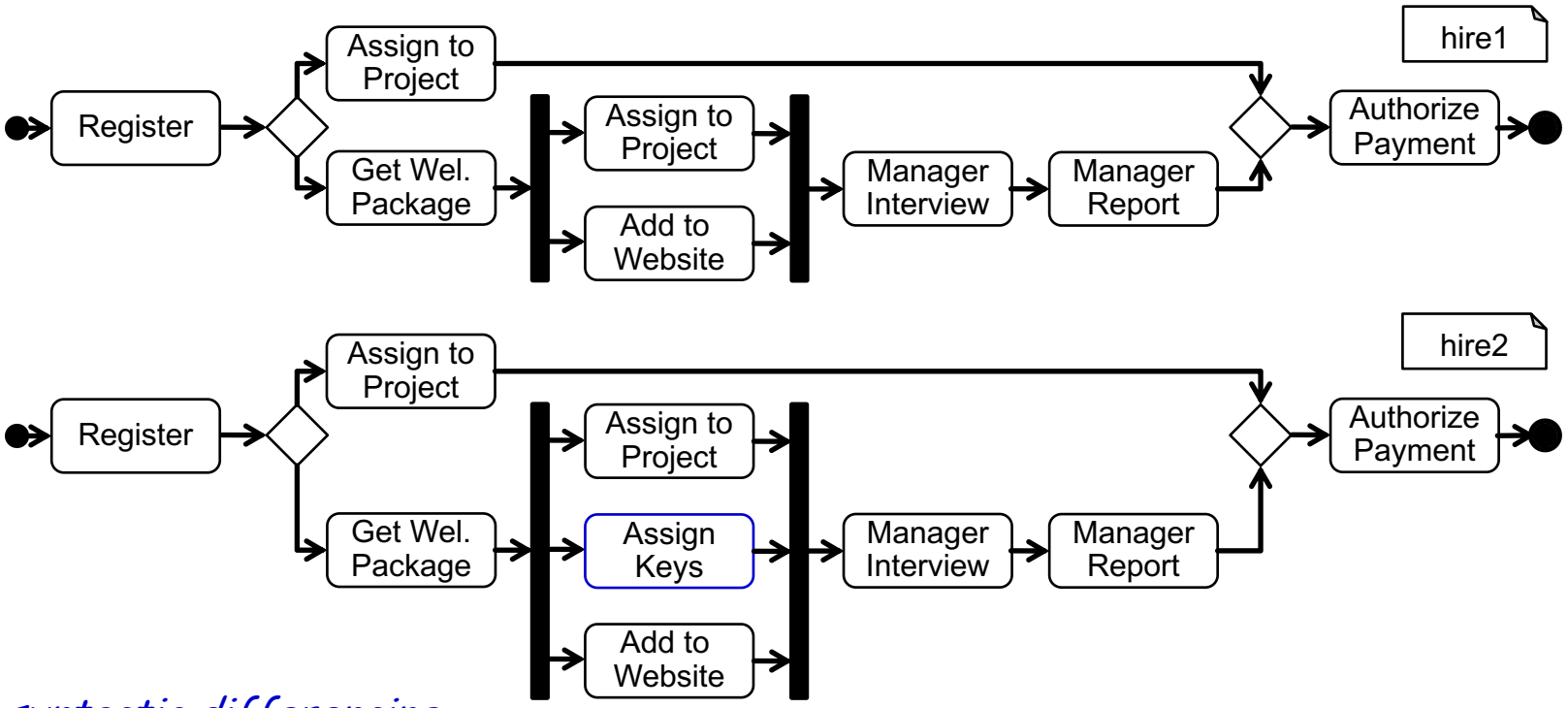
- Contribution:
  - Simple translation from AD subclass to automata
  - Reduction of $sem(ad_1) \subseteq sem(ad_2)$ to automata language inclusion

Agenda



1. Context
2. Example
3. Activity Diagram Variant
4. Translation to Finite Automata

Hiring Workflow

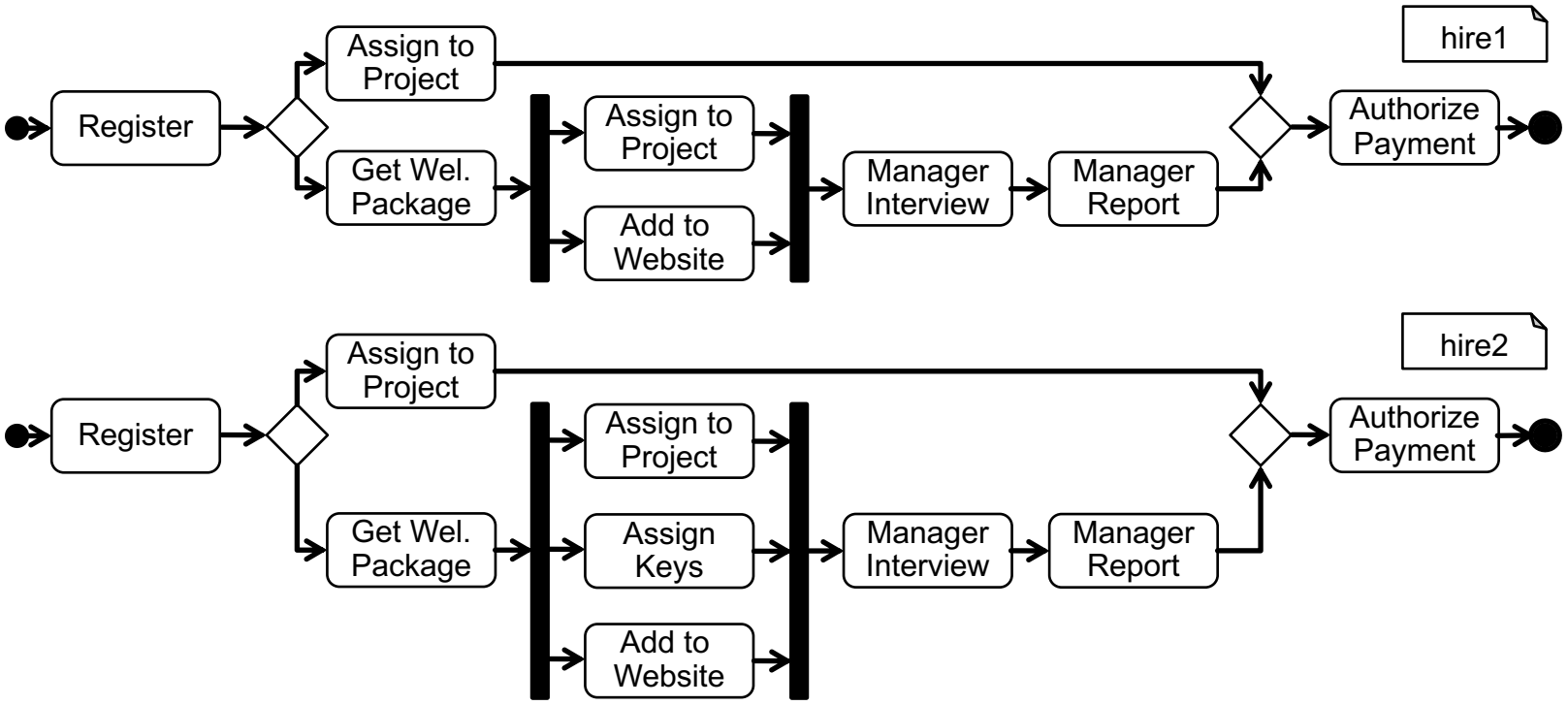


syntactic differencing

$\Delta(\text{hire1}, \text{hire2})$: Action labeled “Assign Keys” added
$\Delta(\text{hire2}, \text{hire1})$: Action labeled “Assign Keys” removed

No information about the semantic impact of the syntactic changes

Hiring Workflow

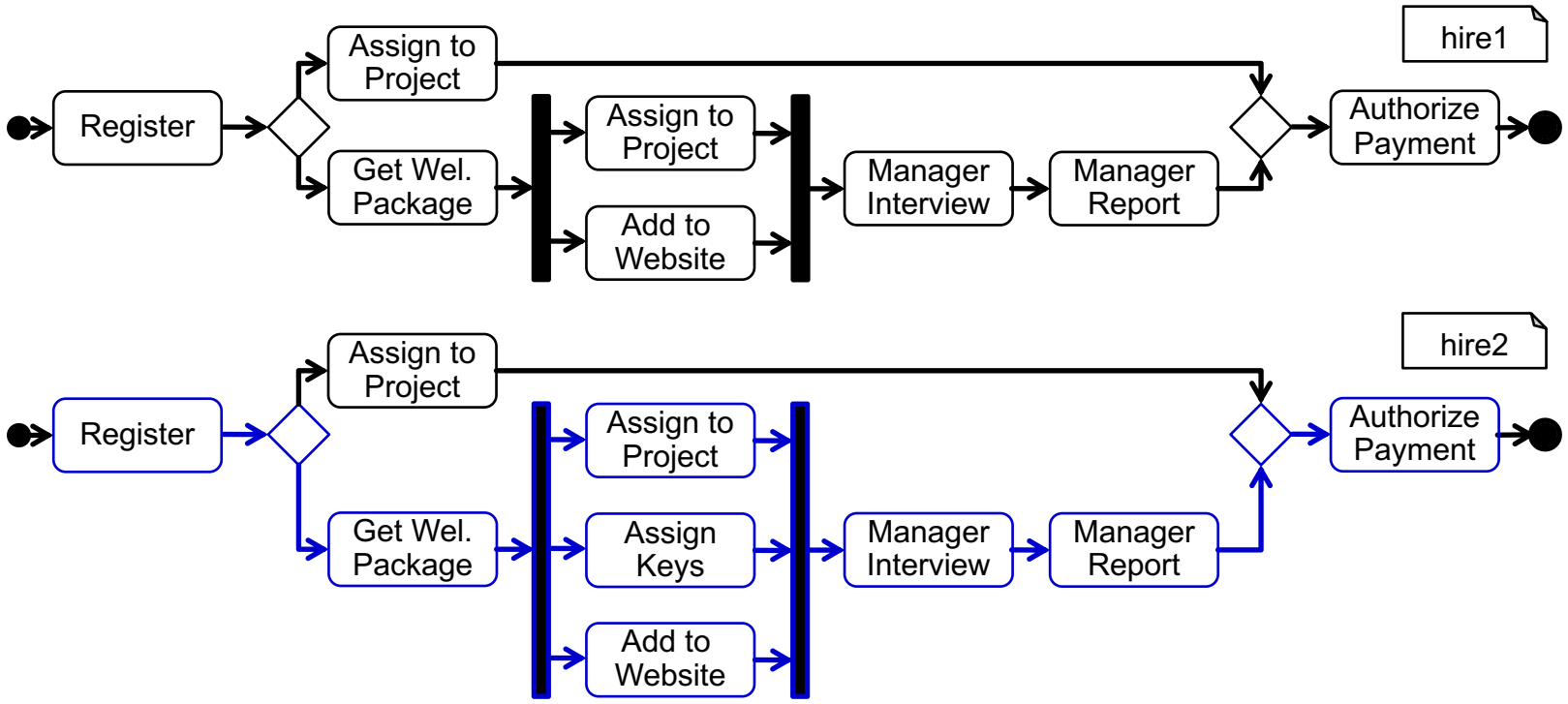


semantic differencing

$sem(\text{hire1}) \subseteq sem(\text{hire2})$

Every execution trace of hire1 is also an execution trace of hire2

Hiring Workflow

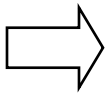


semantic differencing

exists $w \in sem(\text{hire2}) \setminus sem(\text{hire1})$

There are execution traces of hire2 that are no execution traces of hire1

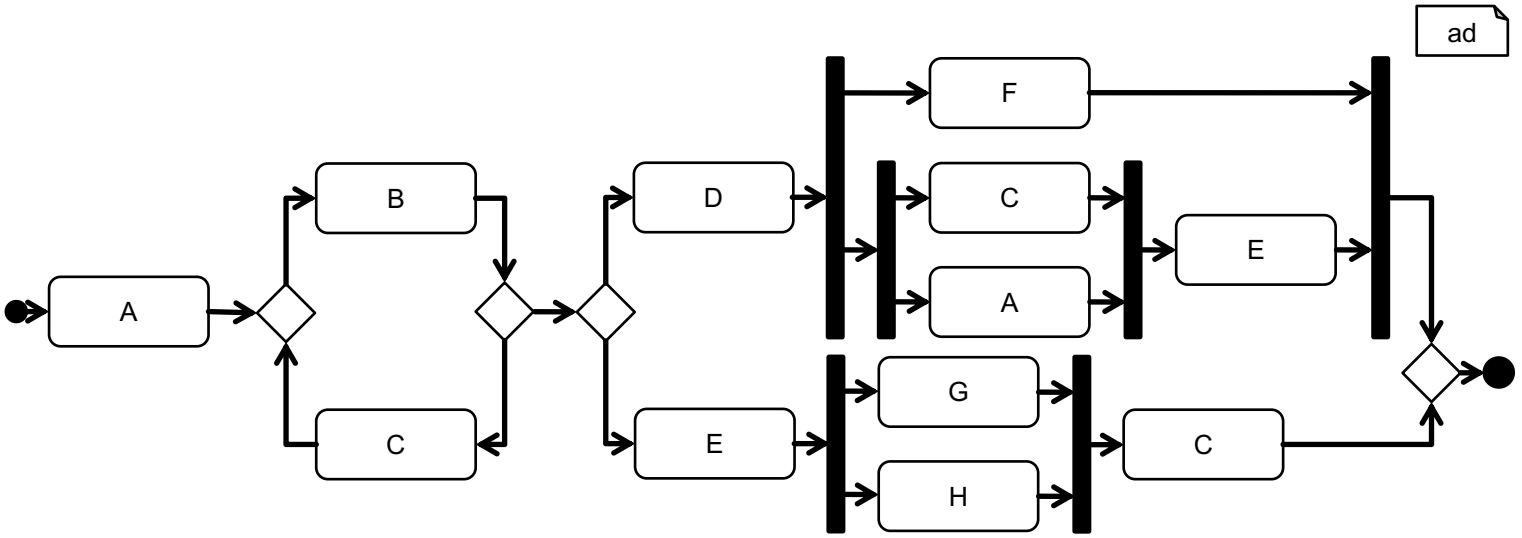
Agenda



1. Context
2. Example
3. Activity Diagram Variant
4. Translation to Finite Automata

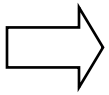
Activity Diagram Variant

- Action, Initial, Final, Fork, Join, Decison, Merge nodes
- Multiple actions with same label
- Loops permitted
- Nesting of control flow nodes and loops
- Semantics: Set of all execution traces – no true parallelism



$\langle A,B,C,B,E,H,G,C \rangle \in sem(ad)$

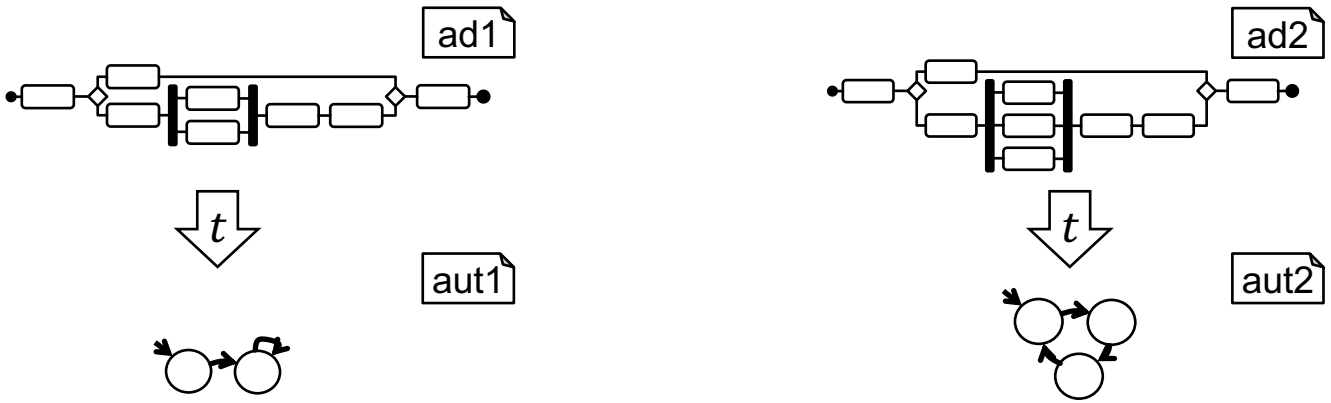
Agenda



1. Context
2. Example
3. Activity Diagram Variant
4. Translation to Finite Automata

Semantic Differencing

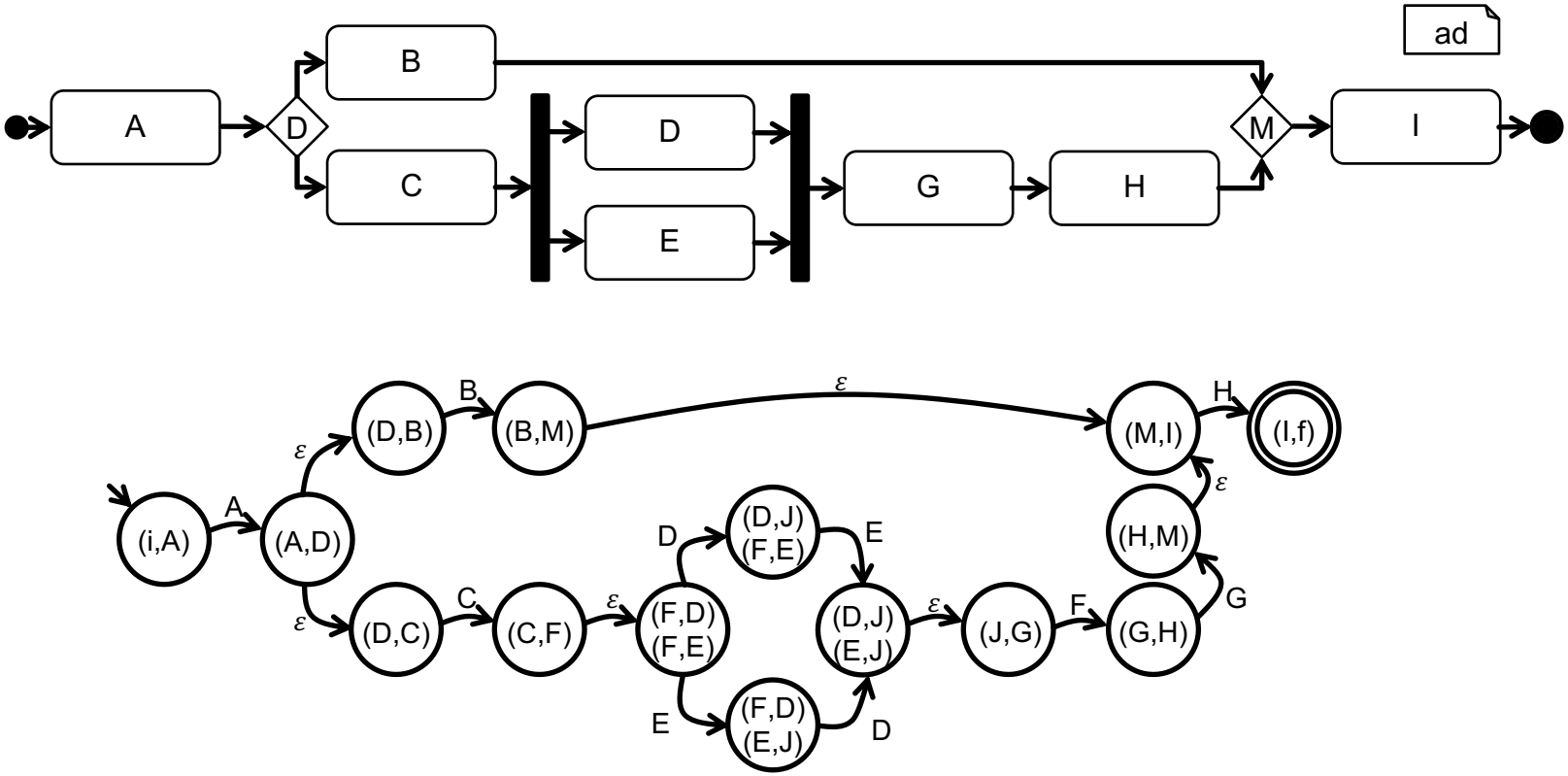
- Define translation $t: AD \rightarrow Aut$ from ADs to automata
- Define semantics $sem(ad) = \mathcal{L}(t(ad))$
- Reduce semantic AD differencing to language inclusion checking



$$sem(\text{ad1}) \subseteq sem(\text{ad2}) \Leftrightarrow \mathcal{L}(aut1) \subseteq \mathcal{L}(aut2)$$

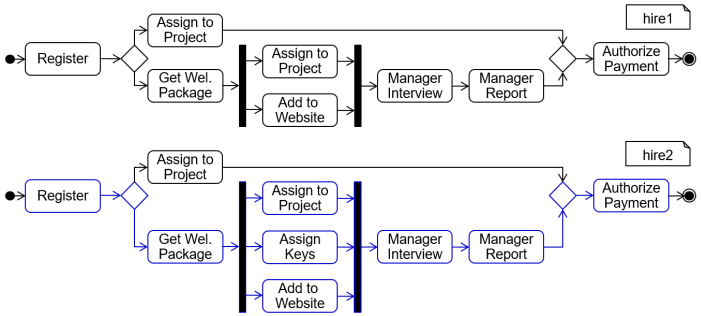
Translation to Automata

- Set of AD edges ↔ Automaton state
- AD action label ↔ Automaton transition label



Thanks

Hiring Workflow



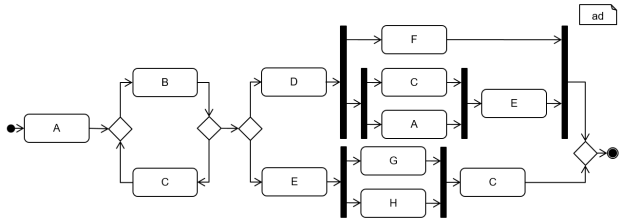
semantic differencing

↳ exists $w \in sem(\text{hire2}) \setminus sem(\text{hire1})$

There are execution traces of hire2 that are no execution traces of hire1

Activity Diagram Variant

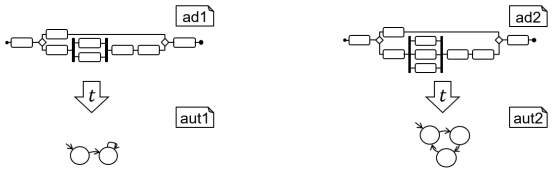
- Action, Initial, Final, Fork, Join, Decison, Merge nodes
- Multiple actions with same label
- Loops permitted
- Nesting of control flow nodes and loops
- Semantics: Set of all execution traces – no true parallelism



⟨A,B,C,B,E,H,G,C⟩ ∈ $sem(ad)$

Semantic Differencing

- Define translation $t: AD \rightarrow Aut$ from ADs to automata
- Define semantics $sem(ad) = \mathcal{L}(t(ad))$
- Reduce semantic AD differencing to language inclusion checking



$sem(\text{ad1}) \subseteq sem(\text{ad2}) \Leftrightarrow \mathcal{L}(aut1) \subseteq \mathcal{L}(aut2)$

Translation to Automata

- Set of AD transitions ↔ Automaton state
- AD action label ↔ Automaton transition label