Formalizing Both Refraction-Based and Sequential Executions of Production Rule Programs

Bruno Berstel-Da Silva

Albert-Ludwigs-Universität Freiburg

IBM France Lab

RuleML

Montpellier – August 29, 2012

Motivation

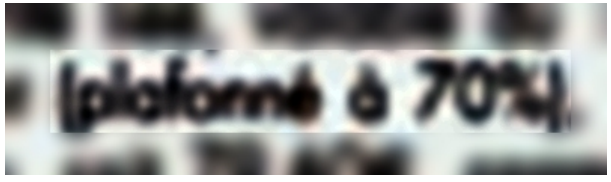
$\mathcal{R} = \{\mathtt{r}_1, \ldots\}$

$\mathtt{r}_1(\mathtt{p}) : \ldots \rightarrow \mathtt{p.discount} := \mathtt{p.age}$

Motivation

$$\mathcal{R} = \{\mathtt{r}_1, \mathtt{r}_2\}$$

$$\mathtt{r}_1(\mathtt{p}) : \ldots \rightarrow \mathtt{p.discount} := \mathtt{p.age}$$



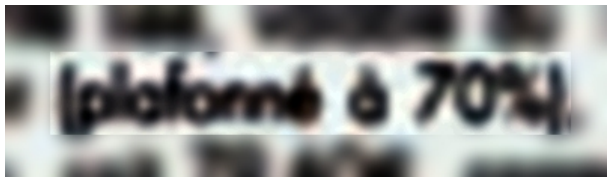
"(limited to 70 %)"

$$\mathtt{r}_2(\mathtt{p}) : \mathtt{p.discount} > 70 \rightarrow \mathtt{p.discount} := 70$$

Motivation

$$\mathcal{R} = \{\mathtt{r}_1, \mathtt{r}_2\}$$

$$\mathtt{r}_1(\mathtt{p}) : \ldots \rightarrow \mathtt{p.discount} := \mathtt{p.age}$$



"(limited to 70 %)"

$$\mathtt{r}_2(\mathtt{p}) : \mathtt{p.discount} > 70 \rightarrow \mathtt{p.discount} := 70$$

$$\{\ldots\}\,\mathcal{R}\,\{\forall p\,(p.discount \leq 70)\}$$

Challenges

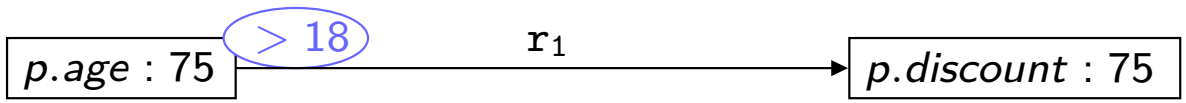
$$\mathtt{r}_1(\mathtt{p}) : \mathtt{p.age} \geq 18 \rightarrow \mathtt{p.discount} := \mathtt{p.age}$$
$$\mathtt{r}_2(\mathtt{p}) : \mathtt{p.discount} > 70 \rightarrow \mathtt{p.discount} := 70$$

$$\{\mathsf{true}\}\,\mathcal{R}\,\{\forall p\,(p.\mathit{discount} \leq 70)\}$$

Challenges

$\mathtt{r}_1(\mathtt{p}) : \mathtt{p.age} \geq 18 \rightarrow \mathtt{p.discount} := \mathtt{p.age}$
$\mathtt{r}_2(\mathtt{p}) : \mathtt{p.discount} > 70 \rightarrow \mathtt{p.discount} := 70$

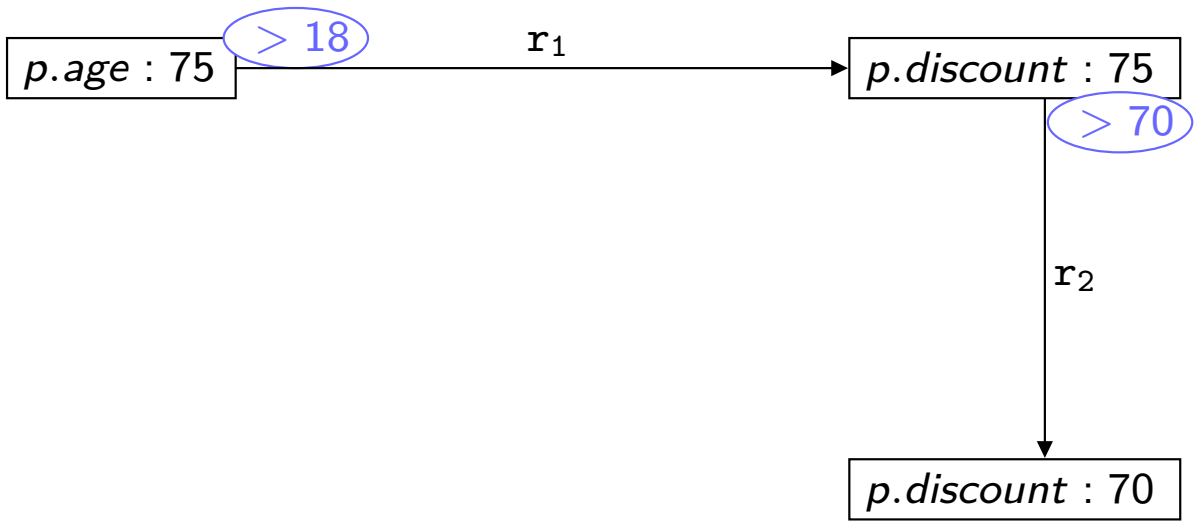
$$\{\text{true}\}\, \mathcal{R}\, \{\forall p\, (p.discount \leq 70)\}$$



Challenges

$$\mathtt{r}_1(\mathtt{p}) : \mathtt{p.age} \geq 18 \rightarrow \mathtt{p.discount} := \mathtt{p.age}$$
$$\mathtt{r}_2(\mathtt{p}) : \mathtt{p.discount} > 70 \rightarrow \mathtt{p.discount} := 70$$

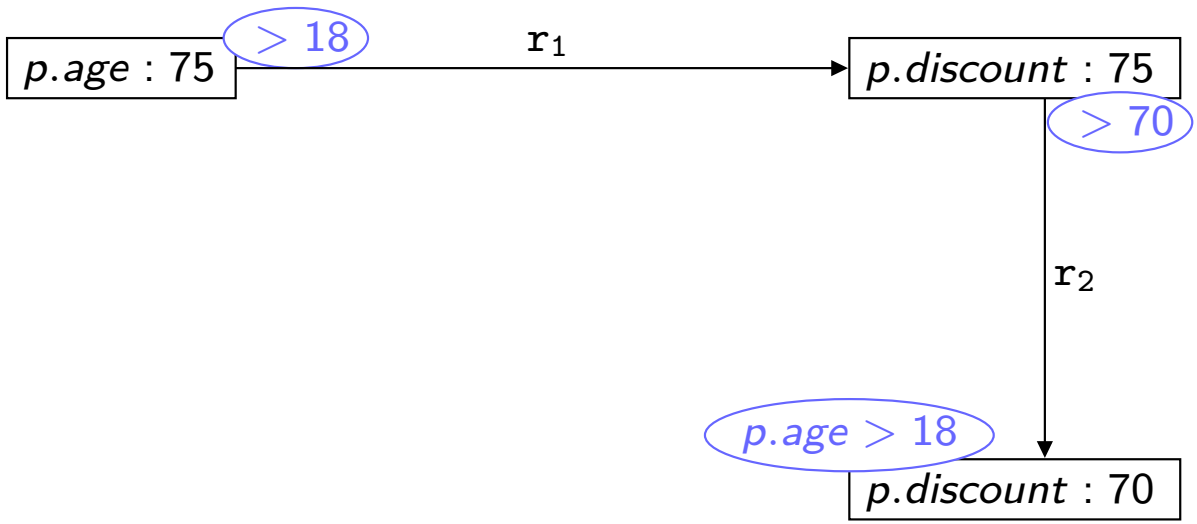
$$\{\text{true}\}\, \mathcal{R}\, \{\forall p\, (p.discount \leq 70)\}$$



Challenges

$\mathtt{r}_1(\mathtt{p}) : \mathtt{p.age} \geq 18 \rightarrow \mathtt{p.discount} := \mathtt{p.age}$
$\mathtt{r}_2(\mathtt{p}) : \mathtt{p.discount} > 70 \rightarrow \mathtt{p.discount} := 70$

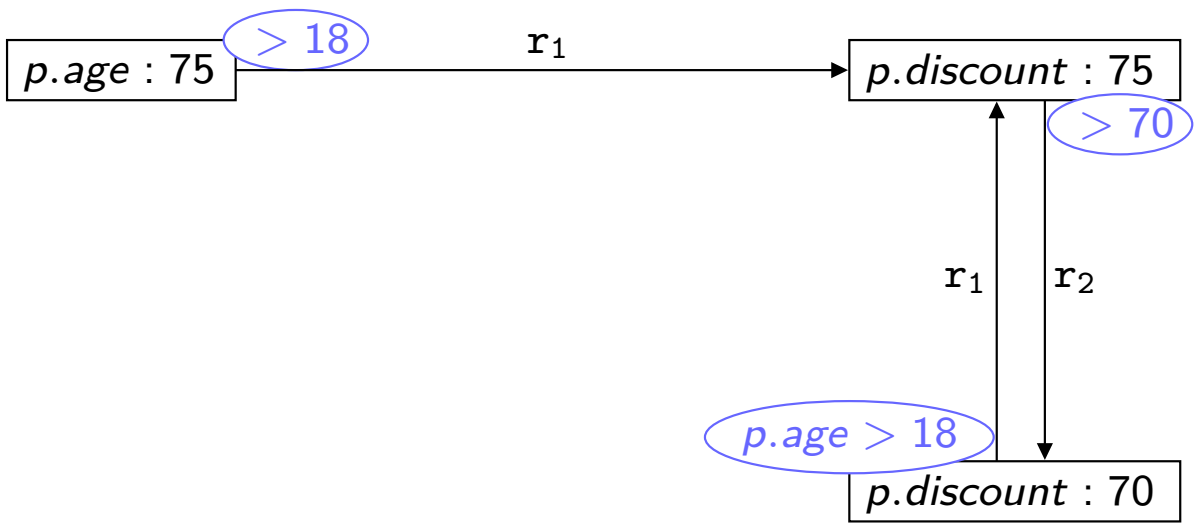
$$\{\text{true}\}\, \mathcal{R}\, \{\forall p\, (p.discount \leq 70)\}$$



Challenges

$\mathtt{r}_1(\mathtt{p}) : \mathtt{p.age} \geq 18 \rightarrow \mathtt{p.discount} := \mathtt{p.age}$
$\mathtt{r}_2(\mathtt{p}) : \mathtt{p.discount} > 70 \rightarrow \mathtt{p.discount} := 70$

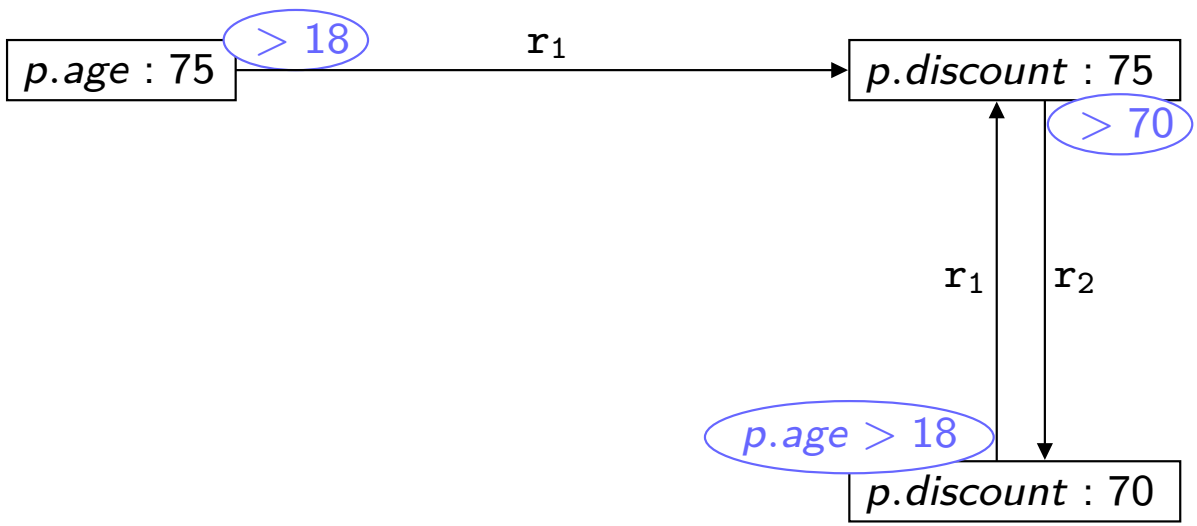
$$\{\text{true}\}\, \mathcal{R}\, \{\forall p\, (p.discount \leq 70)\}$$



Challenges

$\mathtt{r}_1(\mathtt{p}) : \mathtt{p.age} \geq 18 \rightarrow \mathtt{p.discount} := \mathtt{p.age}$
$\mathtt{r}_2(\mathtt{p}) : \mathtt{p.discount} > 70 \rightarrow \mathtt{p.discount} := 70$

$$\{\text{true}\}\,\mathcal{R}\,\{\forall p\,(p.discount \leq 70)\}$$



→ Control: applicability vs. eligibility

Rete... What Else?

- Rete handles eligibility through refraction
  - C. Forgy, 1982: *Rete: A Fast Algorithm for the Many Patterns/Many Objects Match Problem*
  - Fages et al., 1992: *Sémantique opérationnelle et compilation des systèmes de production*
  - Cirstea et al., 2004: *Production Systems and Rete Algorithm Formalisation*
  - W3C recommendation, 2010: *Rule Interchange Format, Production Rule Dialect*

Rete... What Else?

- Rete handles eligibility through refraction
  - C. Forgy, 1982: *Rete: A Fast Algorithm for the Many Patterns/Many Objects Match Problem*
  - Fages et al., 1992: *Sémantique opérationnelle et compilation des systèmes de production*
  - Cirstea et al., 2004: *Production Systems and Rete Algorithm Formalisation*
  - W3C recommendation, 2010: *Rule Interchange Format, Production Rule Dialect*

- Business Rules Management Systems introduced sequential execution
  - IBM Websphere Operation Decision Management (formerly ILOG JRules)
  - FICO Blaze Advisor (formerly Nexpert)
  - Red Hat JBoss Rules (formerly Drools)
  - etc.

Outline

- Motivation
- Challenges
- Rete... What Else?
- Formalizing the Execution Semantics
- Refraction-Based Execution
- Sequential Execution
- Comparison
- Conclusion

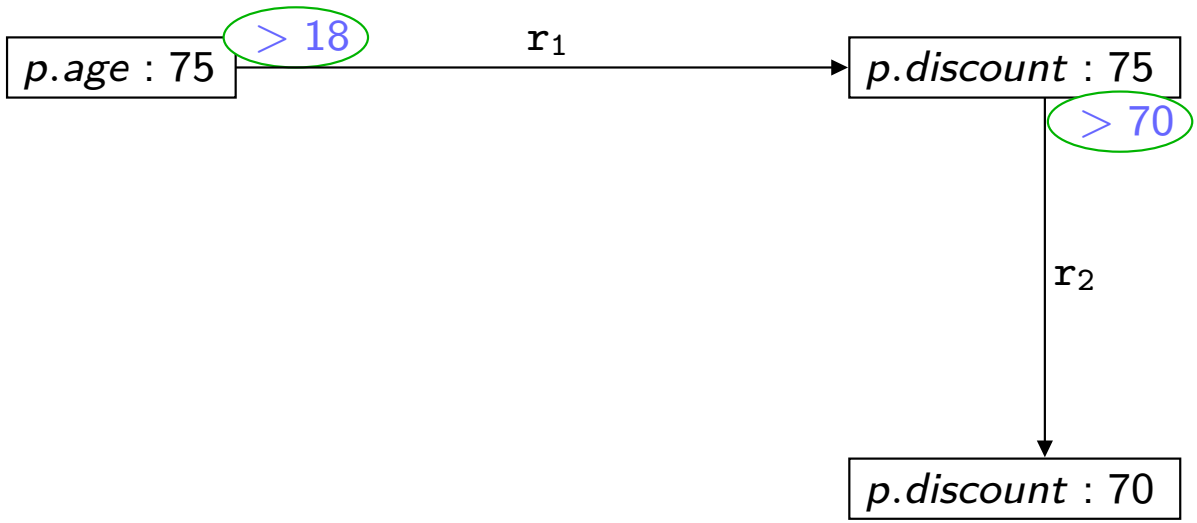
Formalizing the Execution Semantics

- A rule program **execution** is a sequence of transitions between configurations by **applicable** **and** **eligible** rule instances.
- A **configuration** is $\langle E, s \rangle$, where E are the eligible rule instances.
- In a **transition** $\langle E_{k-1}, s_{k-1} \rangle \xrightarrow{R_k} \langle E_k, s_k \rangle$, the eligibility strategy updates the set of eligible rule instances: $E_k = \mathcal{E}(E_{k-1}, R_k, \dots)$.

$$\frac{\{R_k\} = \mathcal{S}(A_{s_{k-1}} \cap E_{k-1}) \quad s_{k-1} \xrightarrow{R_k} s_k \quad E_k = \mathcal{E}(E_{k-1}, \dots)}{\langle E_{k-1}, s_{k-1} \rangle \xrightarrow{R_k} \langle E_k, s_k \rangle}$$

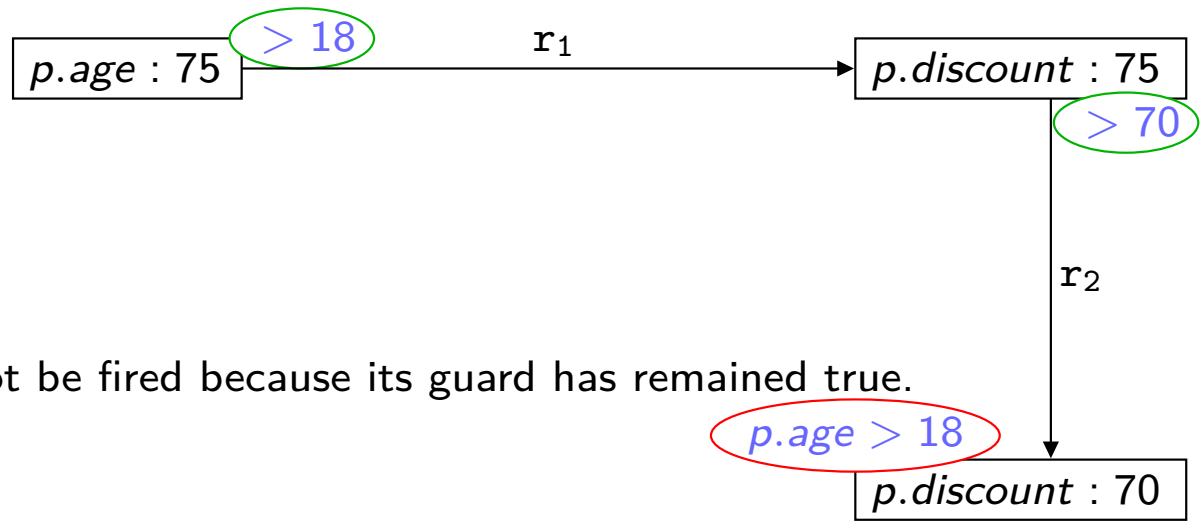
Refraction-Based Execution

"A rule must not be fired more than once as long as the reasons for firing it hold." [W3C RIF-PRD recommendation]



Refraction-Based Execution

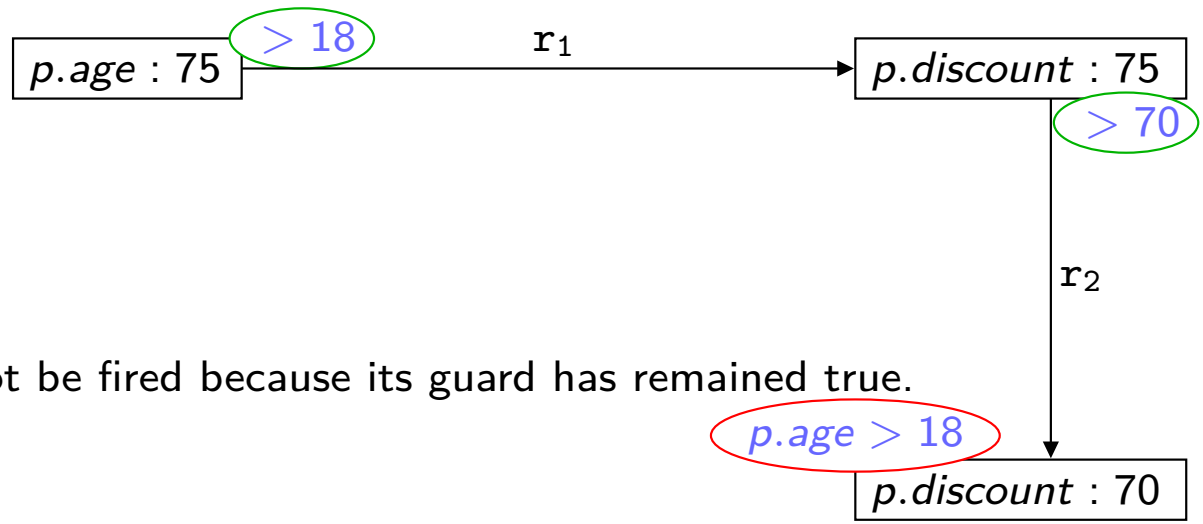
"A rule must not be fired more than once as long as the reasons for firing it hold." [W3C RIF-PRD recommendation]



Rule $\mathtt{r}_1$ will not be fired because its guard has remained true.

Refraction-Based Execution

"A rule must not be fired more than once as long as the reasons for firing it hold." [W3C RIF-PRD recommendation]

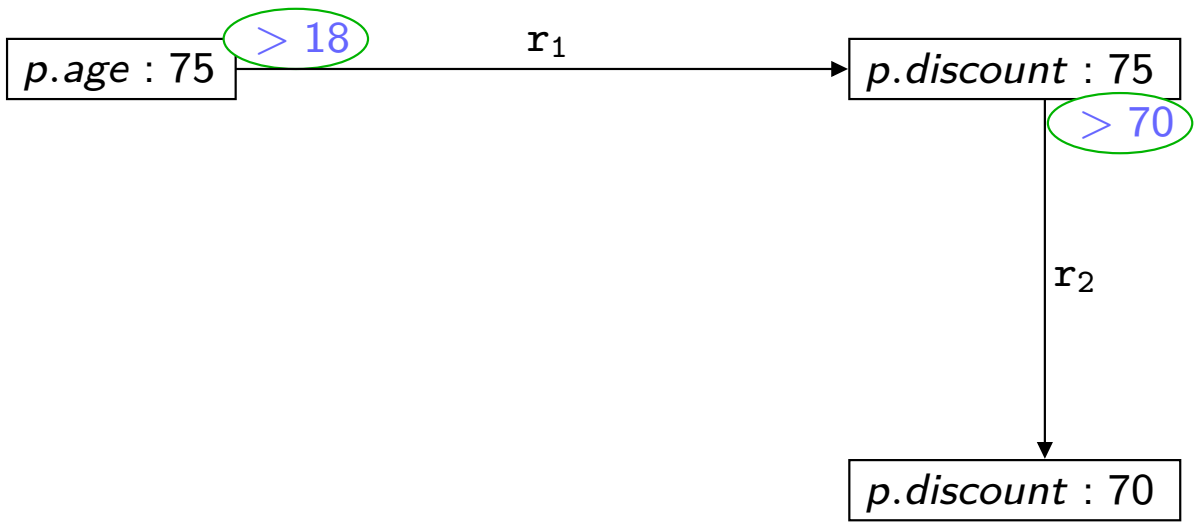


Rule $\mathtt{r}_1$ will not be fired because its guard has remained true.

$$\mathcal{E}_{\text{ref}}(E_{k-1}, R_k, s_k) = E_{k-1} \setminus \{R_k\} \cup \{R \mid R \text{ is not applicable in } s_k\}$$

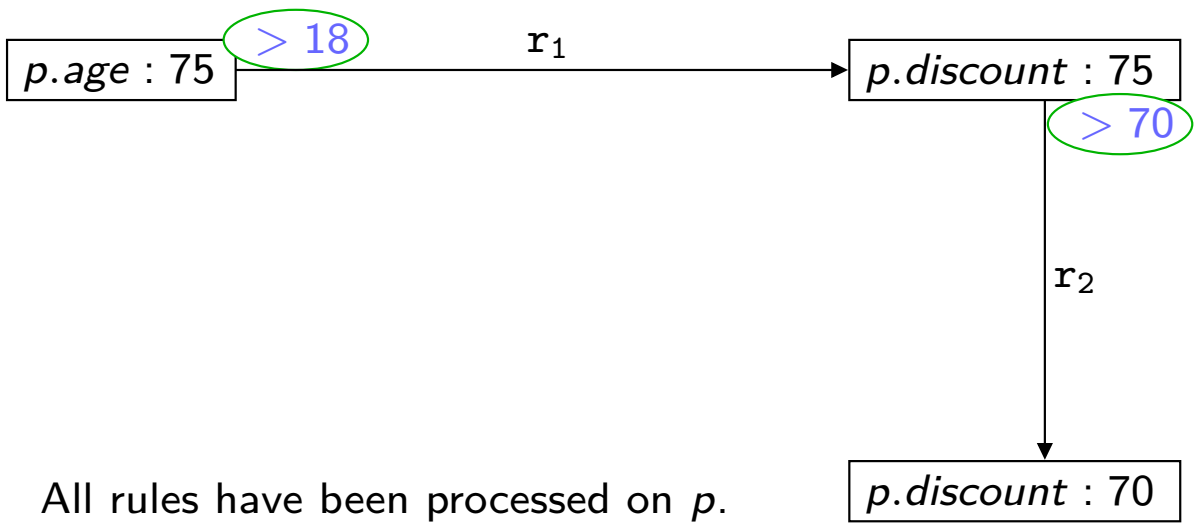
Sequential Execution

Objects are taken **in sequence** from the working memory, and each rule **in sequence** is tentatively executed on them.



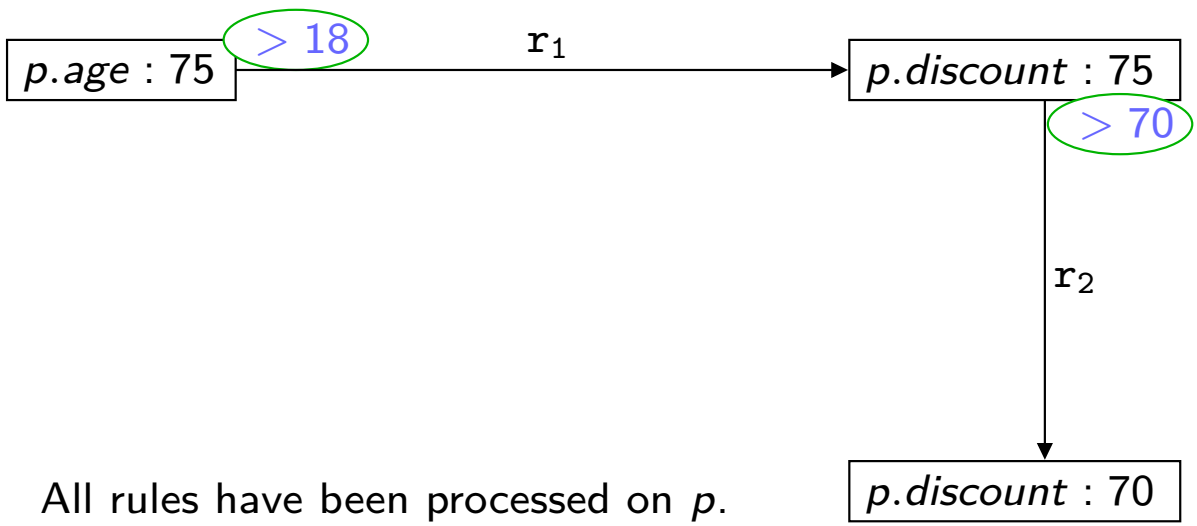
Sequential Execution

Objects are taken **in sequence** from the working memory, and each rule **in sequence** is tentatively executed on them.



Sequential Execution

Objects are taken **in sequence** from the working memory, and each rule **in sequence** is tentatively executed on them.

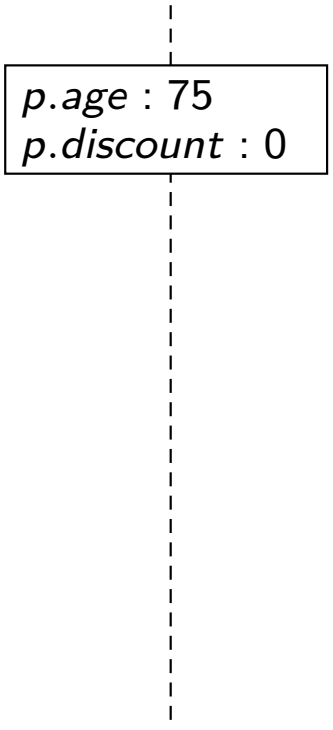


$$\mathcal{E}_{\text{seq}}(E_{k-1}, R_k) = \{R \in E_{k-1} \mid R_k <_{\text{seq}} R\}$$

No, it's not the same

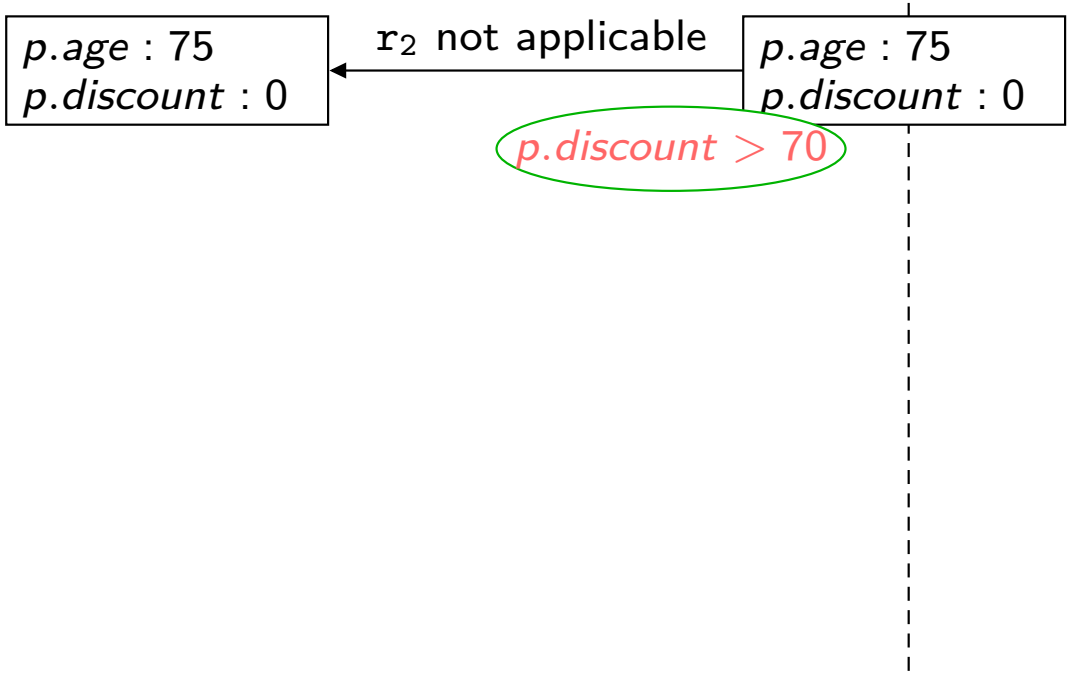
Sequential – with $\mathtt{r}_2 <_{\text{seq}} \mathtt{r}_1$

Refraction – try $\mathtt{r}_2$ first



No, it's not the same

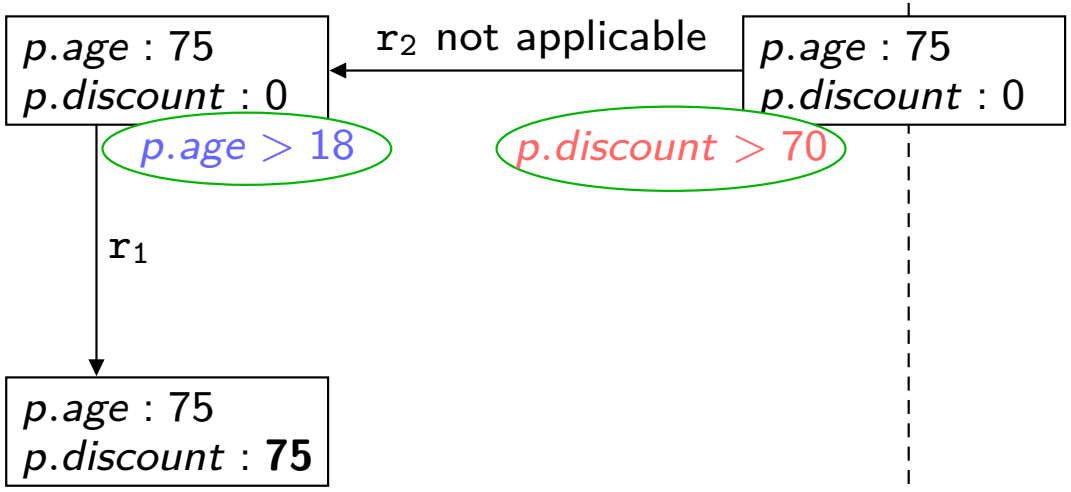
Sequential – with $\mathtt{r}_2 <_{\text{seq}} \mathtt{r}_1$



Refraction – try $\mathtt{r}_2$ first

No, it's not the same

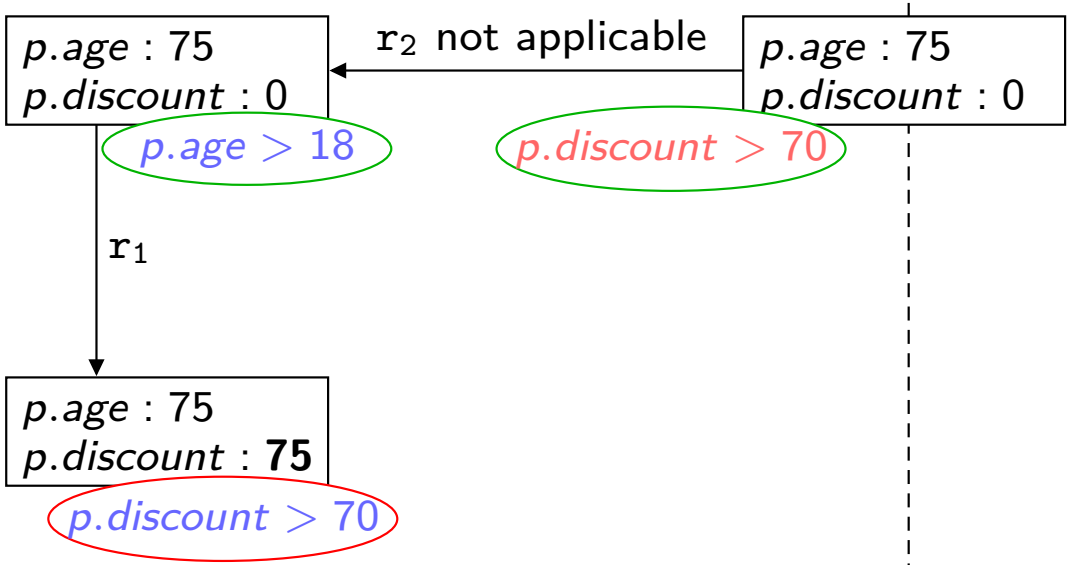
Sequential – with $\mathtt{r}_2 <_{\text{seq}} \mathtt{r}_1$



Refraction – try $\mathtt{r}_2$ first

No, it's not the same

Sequential – with $\mathtt{r}_2 <_{\text{seq}} \mathtt{r}_1$

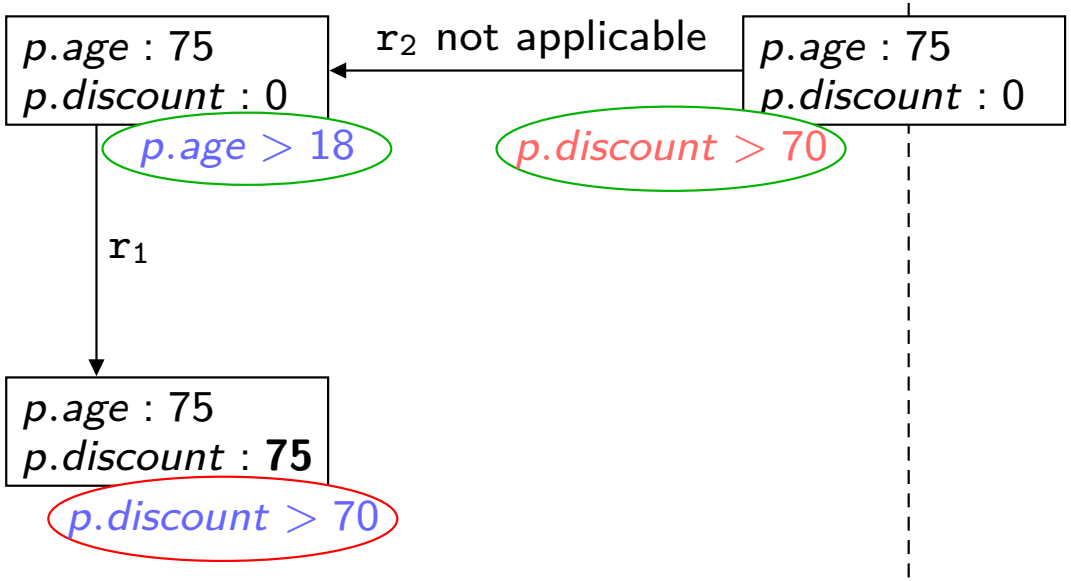


No rule eligible ($\mathtt{r}_2$ applicable).

Refraction – try $\mathtt{r}_2$ first

No, it's not the same

Sequential – with $\mathtt{r}_2 <_{\text{seq}} \mathtt{r}_1$



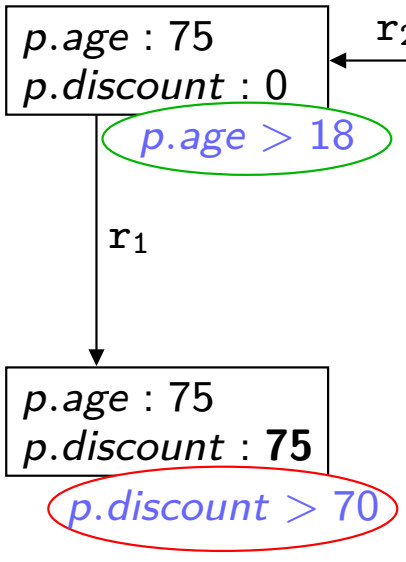
No rule eligible ($\mathtt{r}_2$ applicable).

Refraction – try $\mathtt{r}_2$ first

- In the sequential execution, rules are given no second chance.

No, it's not the same

Sequential – with $\mathtt{r}_2 <_{\text{seq}} \mathtt{r}_1$

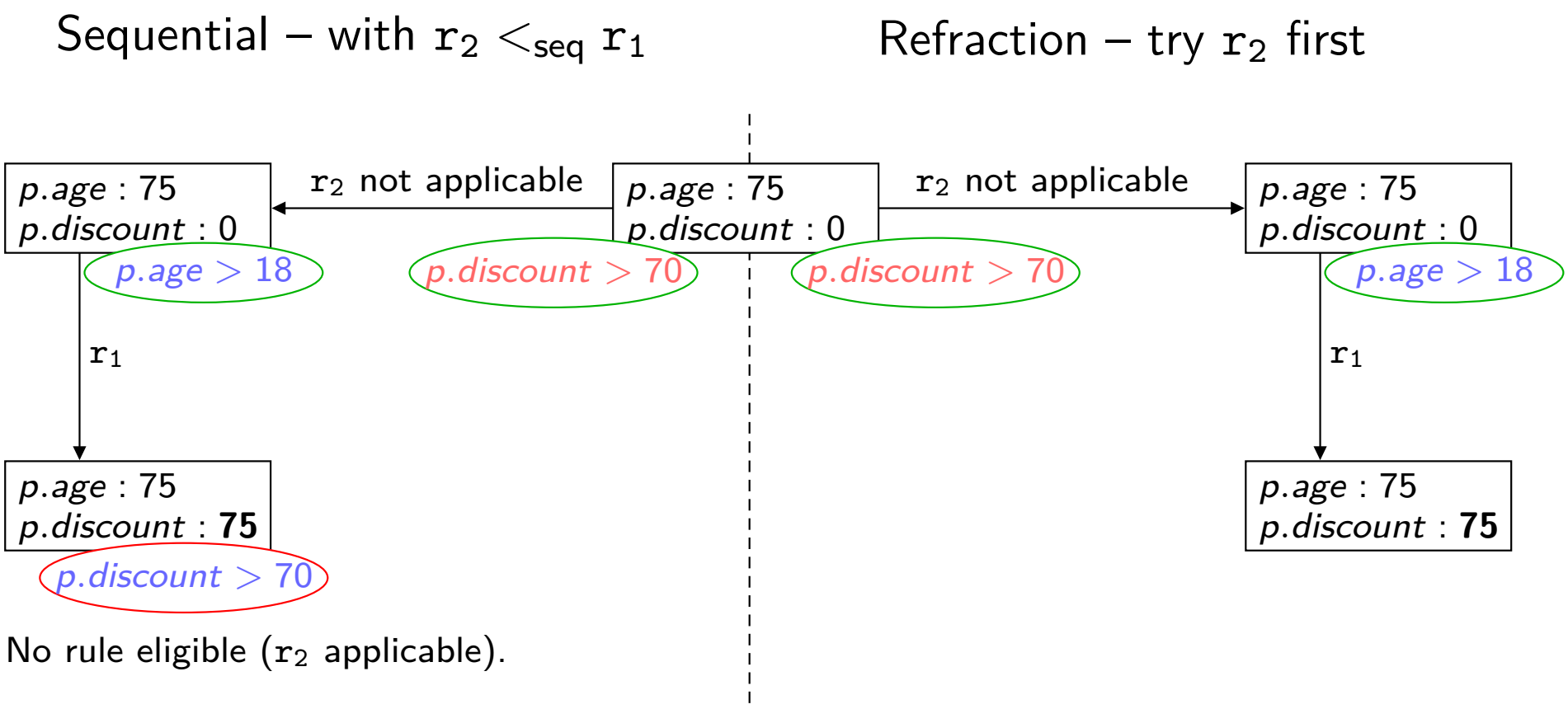


No rule eligible ($\mathtt{r}_2$ applicable).

Refraction – try $\mathtt{r}_2$ first

- In the sequential execution, rules are given no second chance.

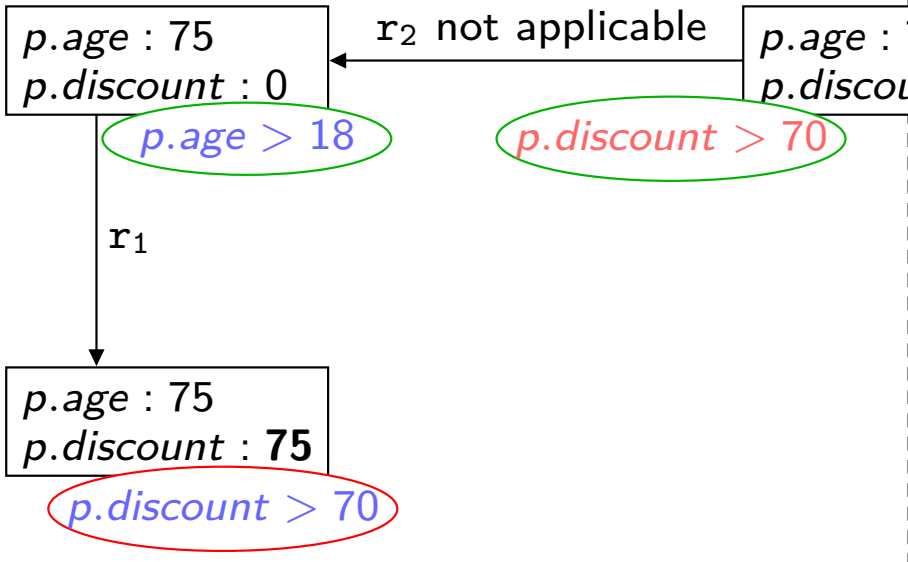
No, it's not the same



- In the sequential execution, rules are given no second chance.

No, it's not the same

Sequential – with $\mathtt{r}_2 <_{\text{seq}} \mathtt{r}_1$



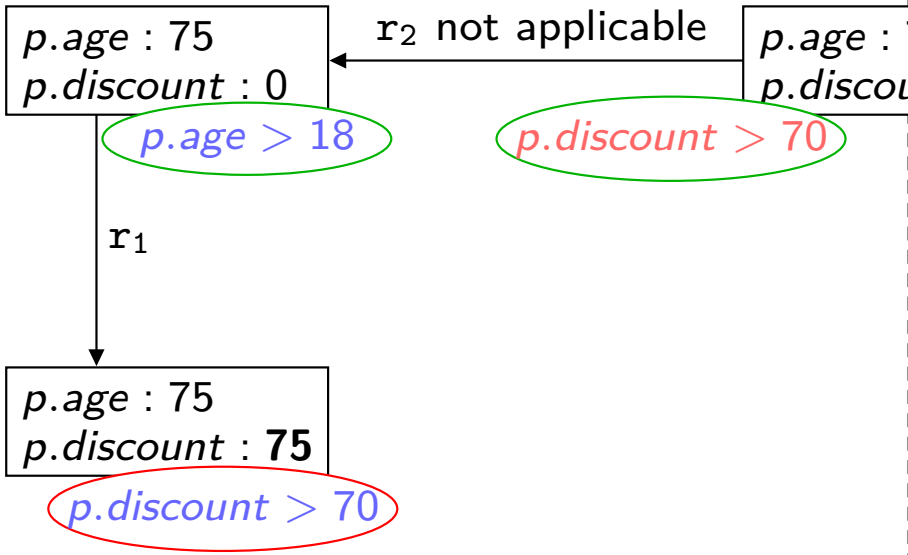
No rule eligible ($\mathtt{r}_2$ applicable).

Refraction – try $\mathtt{r}_2$ first

- In the sequential execution, rules are given no second chance.

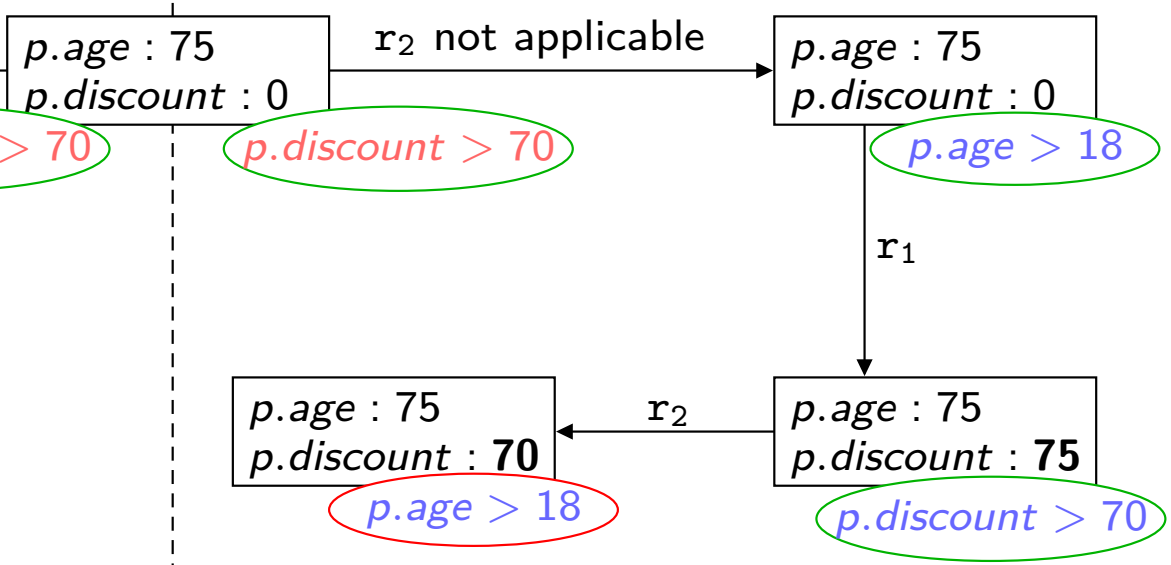
No, it's not the same

Sequential – with $\mathtt{r}_2 <_{\text{seq}} \mathtt{r}_1$



No rule eligible ($\mathtt{r}_2$ applicable).

Refraction – try $\mathtt{r}_2$ first

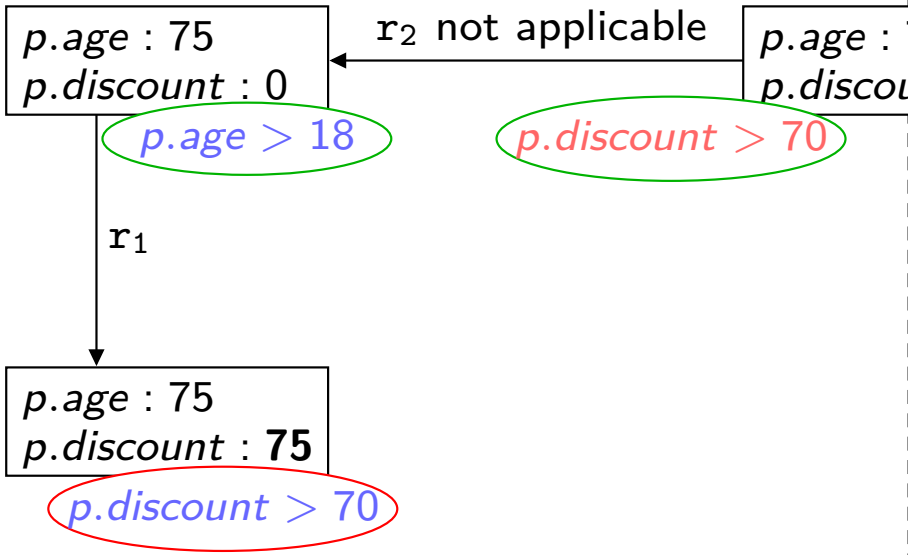


No rule eligible ($\mathtt{r}_1$ applicable).

- In the sequential execution, rules are given no second chance.

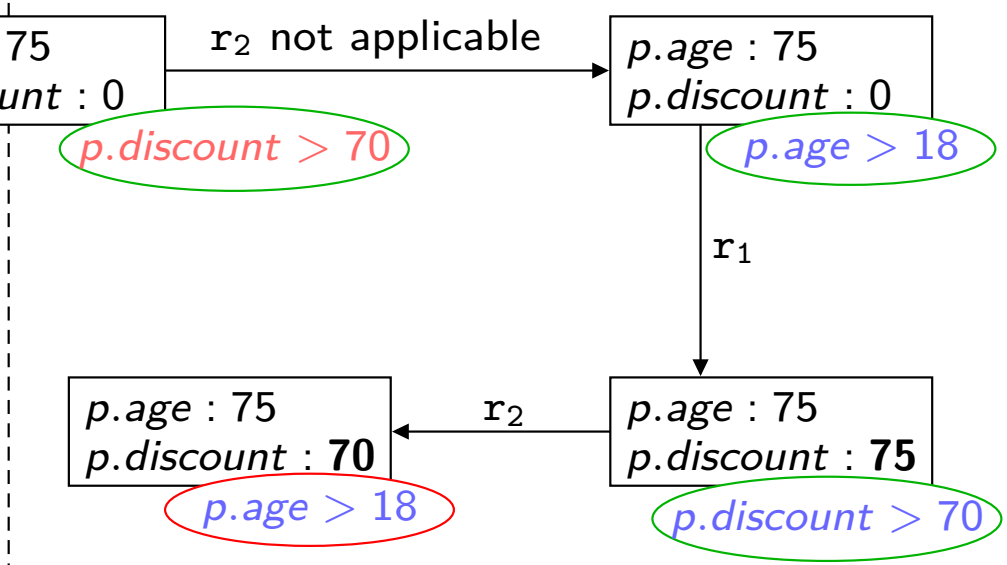
No, it's not the same

Sequential – with $\mathtt{r}_2 <_{\text{seq}} \mathtt{r}_1$



No rule eligible ($\mathtt{r}_2$ applicable).

Refraction – try $\mathtt{r}_2$ first



No rule eligible ($\mathtt{r}_1$ applicable).

- In the sequential execution, rules are given no second chance.
- Is refraction more intuitive or less predictible?

Conclusion

- Contribution
  - A formal description of the execution semantics of business rules programs.
- Short-term benefits
  - Provide such a formalization.
  - Cover Rete's refraction as well as sequential execution.
- Long-term motivations
  - Verification of Business Rules programs.
  - Better compilation for faster execution.
  - More generally, enhance understanding of rule programs.