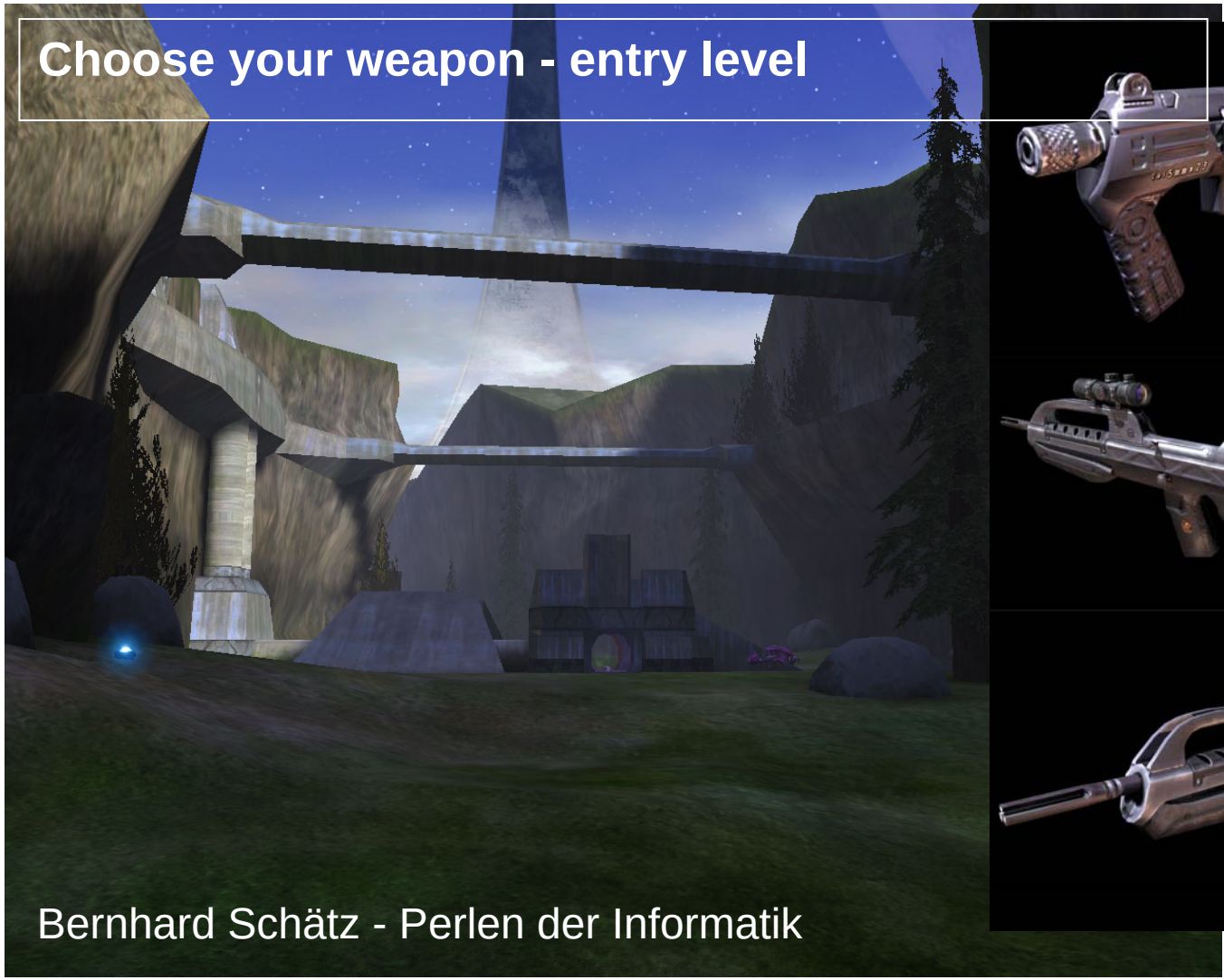
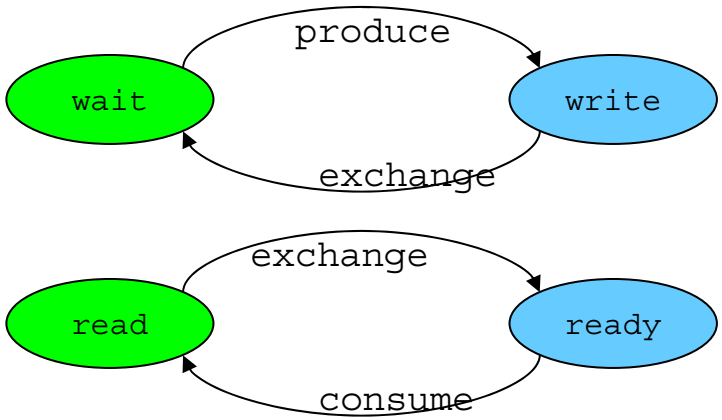
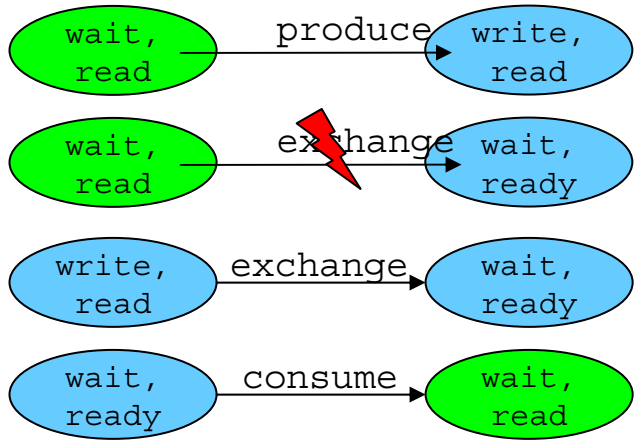
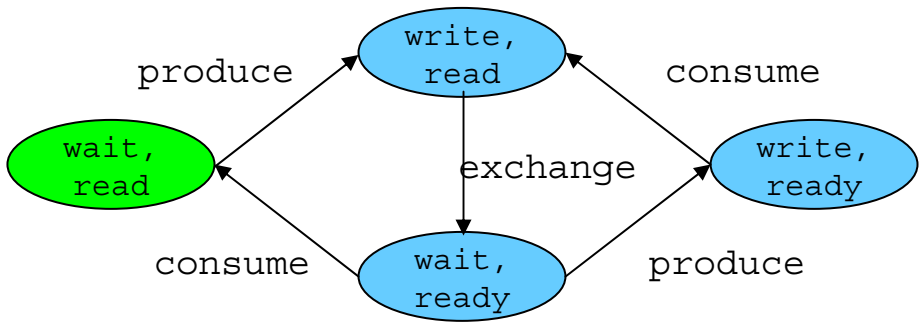
Choose your weapon - entry level

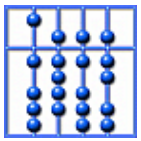


Example: Synchronized Producer Consumer









Models

Classes of models:

- **Algebraic model**: Specification in form of syntactic formulae
  - Concepts: Process
  - Verification: Term equivalence
  - Example: ACP
- **Operational model**: Specification in form of abstract machine
  - Concepts: State, (labeled) transition
  - Verification: (Bi-)Simulation
  - Example: CCS
- **Denotational model**: Specification in form of observable behavior
  - Concepts: Event, (observation) trace
  - Verification: Behavior Inclusion
  - Example: TCSP

Algebraic Semantics: Syntax

$$S = p.x.S \qquad E = p.\tau.B$$
$$R = \bar{x}.c.R \qquad B = c.E + p.F$$
$$F = c.\tau.B$$
$$P = S \mid R$$

Syntax: Process terms over alphabet A

- Blocking process: 0
- Action prefix for some a of A: $a.P$
- Alternative choice: $P + Q$
- Parallel composition: $P \mid Q$
- (Mutual) Recursion: $P = Q$

Algebraic Semantics: Rules

$$0 + P = P$$
$$P + P = P$$
$$P + Q = Q + P$$
$$P + (Q + R) = (P + Q) + R$$

$$0 \mid P = 0$$
$$P \mid Q = Q \mid P$$
$$P \mid (Q \mid R) = (P \mid Q) \mid R$$
$$a.P \mid \overline{a}.Q = \tau.P \mid Q$$
$$a.P \mid b.Q = a.(P \mid b.Q) + b.(a.P \mid Q)$$

$$\frac{P = S[P] \wedge Q = T[Q] \wedge P = Q \Rightarrow S[P] = T[Q]}{P = Q}$$

Algebraic semantics:

- Relates equivalent proceses
- Defined as equations of process terms

Algebraic Semantics: Example

$$S = p.x.S$$
$$R = \bar{x}.c.R$$

$$P = S \mid R$$

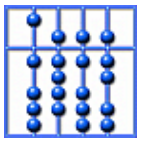
$$E = p.\tau.B$$
$$B = c.E + p.F$$
$$F = c.\tau.B$$

$$\begin{aligned} B &= c.E + p.F \\ &= c.p.\tau.B + p.c.\tau.B \\ &= c.p.\tau.(S \mid c.R) + p.c.\tau.(S \mid c.R) \\ &= c.p.(x.S \mid \bar{x}.c.R) + p.c.(x.S \mid \bar{x}.c.R) \\ &= c.(p.x.S \mid R) + p.(x.S \mid c.\bar{x}.c.R) \\ &= c.(S \mid R) + p.(x.S \mid R) \\ &= (S \mid c.R) + (S \mid c.R) \\ &= S \mid c.R \end{aligned}$$

$$\begin{aligned} P &= S \mid R \\ &= p.x.S \mid R \\ &= p.(x.S \mid R) \\ &= p.(x.S \mid \bar{x}.c.R) \\ &= p.\tau.(S \mid c.R) \\ &= p.\tau.B \\ &= E \end{aligned}$$

Example: Proving equivalence of process terms S_1 and S_2

- Approach:
  - Construct chain of equations
  - Use inductive principles for recursion



Models

Classes of models:

- **Algebraic model**: Specification in form of syntactic formulae
  - Concepts: Process
  - Verification: Term equivalence
  - Example: ACP
- **Operational model**: Specification in form of abstract machine
  - Concepts: State, (labeled) transition
  - Verification: (Bi-)Simulation
  - Example: CCS
- **Denotational model**: Specification in form of observable behavior
  - Concepts: Event, (observation) trace
  - Verification: Behavior Inclusion
  - Example: TCSP

Operational Semantics: Syntax

$$S = p.x.S \qquad E = p.\tau.B$$
$$R = \bar{x}.c.R \qquad B = c.E + p.F$$
$$F = c.\tau.B$$
$$P = S \mid R$$

Syntax: Process terms over alphabet A

- Blocking process: 0
- Action prefix for some a or τ of A: $a.P$
- Alternative choice: $P + Q$
- Parallel composition: $P \mid Q$
- (Mutual) Recursion: $P = Q$

Operational Semantics: Rules

$$\frac{}{a.P \xrightarrow{a} P} \qquad \frac{S = P, P \xrightarrow{a} Q}{S \xrightarrow{a} Q} \qquad \frac{P \xrightarrow{a} P'}{P + Q \xrightarrow{a} P'} \qquad \frac{Q \xrightarrow{a} Q'}{P + Q \xrightarrow{a} Q'}$$

$$\frac{P \xrightarrow{a} P'}{P \mid Q \xrightarrow{a} P' \mid Q} \qquad \frac{Q \xrightarrow{a} Q'}{P \mid Q \xrightarrow{a} P \mid Q'} \qquad \frac{P \xrightarrow{a} P', Q \xrightarrow{\bar{a}} Q'}{P \mid Q \xrightarrow{\tau} P \mid Q'}$$

Assuming disjoint alphabets on a

Operational semantics:

- Defined by (labeled) transition relation
- Constructed in terms of computation steps
  - Start state
  - (Inter)action
  - End state
- Structured operational semantics: Process terms as state space

Operational Semantics: Example

$$\frac{}{p.x.S \xrightarrow{p} x.S}$$

$$\frac{S = p.x.S,\ p.x.S \xrightarrow{p} x.S}{S \xrightarrow{p} x.S}$$

$$\frac{S \xrightarrow{p} x.S}{S \mid R \xrightarrow{p} x.S \mid R}$$

Example: Constructing execution steps

- Construct tableau of rules
- Use structure of process terms
  - Break down terms into atomic terms
  - Build up tableau using rules

Operational Semantics: Verification

$$\approx \text{ is a (strong) bisimulation} \Leftrightarrow \forall P,Q,a.$$

$$P \approx Q \Rightarrow (\forall P'.\, P \xrightarrow{a} P' \Rightarrow \exists Q'.\, Q \xrightarrow{a} Q' \wedge P' \approx Q') \wedge$$

$$(\forall Q'.\, Q \xrightarrow{a} Q' \Rightarrow \exists P'.\, P \xrightarrow{a} P' \wedge P' \approx Q')$$

Verification:

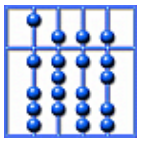
- Relation between process terms S_1 and S_2
- Based on transition relation:
  - Strong Bisimularity: $\exists \approx .\, S_1 \approx S_2$
  - Weak bisimularity: Bisimularity modulo τ
  - Simulation relation: Asymmetric variant

Operational Semantics: Example

$$S = p.x.S \qquad S \mid R \approx E \qquad E = p.\tau.B$$
$$R = \bar{x}.c.R \qquad x.S \mid \bar{x}.c.R \approx \tau.B \qquad B = c.E + p.F$$
$$p.x.S \mid c.R \approx B \qquad F = c.\tau.B$$
$$P = S \mid R \qquad x.S \mid c.R \approx F$$

Example: Proving process bisimularity of process terms S_1 and S_2

- Approach:
  - Construct largest bisimulation relation over transition relation
  - Check containment of process terms S_1 and S_2
- Automatic construction:
  - Using fixed point induction
  - Using 1-step-equivalence as induction principle



Models

Classes of models:

- **Algebraic model**: Specification in form of syntactic formulae
  - Concepts: Process
  - Verification: Term equivalence
  - Example: ACP
- **Operational model**: Specification in form of abstract machine
  - Concepts: State, (labeled) transition
  - Verification: (Bi-)Simulation
  - Example: CCS
- **Denotational model**: Specification in form of observable behavior
  - Concepts: Event, (observation) trace
  - Verification: Behavior Inclusion
  - Example: TCSP

Denotational Semantics: Syntax

$$S = \mu X : p.x.X$$
$$R = \mu X : x.c.X$$
$$P = S \mid R$$

$$E = p.B$$
$$B = \mu X : c.p.X + p.c.X$$

Syntax: Process terms over alphabet A

- Blocking process: 0
- Action prefix for some a of A: $a.P$
- Alternative choice: $P + Q$
- Parallel composition: $P \mid Q$
- (Mutual) Recursion: $\mu P : T(P)$

Denotational Semantics: Mapping

$$F[0] = \{(<>, R) \mid R \subseteq A\}$$
$$F[a.P] = \{(<>, R) \mid R \subseteq A \setminus \{a\}) \cup \{(a \bullet t, R) \mid (t, R) \in F[P]\}$$
$$F[P + Q] = \{(t, R) \mid ((t, R) \in F[P] \cap F[Q]) \vee (t \neq <> \wedge (t, R) \in F[P] \cup F[Q])\}$$
$$F[P \mid Q] = \{(t, R_1 \cup R_2) \mid (t \uparrow \alpha P, R_1) \in F[P] \wedge (t \uparrow \alpha Q, R_2) \in F[Q]\}$$
$$F[\mu P : T(P)] = lfp_i F[T^i](A^* \times 2^A)$$

Denotational semantics:

- Defined by mapping into (Scott) domain
- Constructed via inductive definition
  - Compositonal
  - Aligned according to process term structure
- Recursive definitions: Using fixed point induction

Denotational Semantics: Example

$$F[S] = \{(<>,\emptyset),(<>,\{x\}),(p,\emptyset),(p,\{p\}),(p \bullet x,\emptyset),(p \bullet x,\{x\}),\ldots\}$$
$$F[R] = \{(<>,\emptyset),(<>,\{c\}),(x,\emptyset),(x,\{x\}),(x \bullet c,\emptyset),(x \bullet c,\{c\}),\ldots\}$$
$$F[S \mid R] = \{(<>,\emptyset),(<>,\{c\}),(<>,\{x\}),(<>,\{c,x\}),$$
$$(p,\emptyset),(p,\{p\}),(p,\{c\}),(p,\{p,c\}),$$
$$(p \bullet x,\emptyset),(p \bullet x,\{x\}),\ldots\}$$

Example: Constructing behaviors

- Construct behaviors as elements of Scott domain
- Use structure of process terms
  - Break down terms into atomic terms
  - Build up behaviors using rules
  - Use fixed point induction

Operational Semantics: Verification

$$P \leq Q \Leftrightarrow F[P] \subseteq F[Q]$$

Verification:

- Relation between process terms S_1 and S_2
- Based on denotation of process terms:
  - Equivalence: $P \leq Q \wedge Q \leq P$
  - Refinement: Ordering in Scott domain

Operational Semantics: Example

$$S = \mu X : p.x.X$$
$$R = \mu X : x.c.X$$
$$E = p.B$$
$$B = \mu X : c.p.X + p.c.X$$
$$P = S \mid R$$

$$F[S \mid R] = \{(<>,\emptyset),(<>,\{c\}),(<>,\{x\}),(<>,\{c,x\}),$$
$$(p,\emptyset),(p,\{p\}),(p,\{c\}),(p,\{p,c\}),$$
$$(p \bullet x,\emptyset),(p \bullet x,\{x\}),...\}$$
$$= F[E]$$

Example: Proving equivalence of process terms S_1 and S_2

- Approach:
  - Construct denotation of process terms S_1 and S_2
  - Check set equivalence

Conclusion

- Algebraic Semantics
  - Advantage: Semantics is term language
  - Disadvantages: Consistency, implementability
- Operational Semantics:
  - Advantages: Implementability
  - Disadvantages: Verification reduced to simulation
- Denotational Semantics:
  - Advantages: Observational approach
  - Disadvantages: Complex