Pillage Games

In[1]:= `Needs["Theorema`"]`

Basic Definitions

```
Definition["1", f[] := "Hello World!" ]

Compute[f[], using → Definition["1"]]
```

Hello World!

TS_In[2]:=
```
Definition["agents", any[n],
  I[n] := {i | i=1,…,n}]

Compute[I[10], using → Definition["agents"],
 builtin → {Builtin ["Sets"], Builtin ["Quantifiers"], Builtin ["Numbers"]}]
```

{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

```
Definition["appL", any[n],
 appL[n] : ⇔ (n ∈ ℕ ⋀ n ≥ 2)]
```

TS_In[3]:=
```
Lemma["1inI", any[n], with[appL[n]],
  {1} ⊆ I[n] ]
```

TS_In[4]:=
```
Lemma["2inI", any[n], with[appL[n]],
  {2} ⊆ I[n] ]

Prove[Lemma["1inI"], using → {Definition["appL"], Definition["agents"]},
 builtin → {Builtin ["Numbers"]}, by → SetTheoryPCSProver,
 ProverOptions → {SimplifyFormula → True}]
```

- ProofObject -

TS_In[5]:= Definition["allocation", any[x, n],

$$\mathrm{allocation}_n[x] :\Leftrightarrow \left(n \in \mathbb{N} \bigwedge \mathrm{istuple}[x] \bigwedge (|x| = n) \bigwedge \underset{i \in I[n]}{\forall} x_i \geq 0 \bigwedge \left(\sum_{i \in I[n]} x_i = 1\right)\right)]$$

TS_In[6]:=

Definition["WC", any[π, n], bound[allocation$_n$[x]],

$$WC[\pi, n] :\Leftrightarrow \underset{\substack{C1, C2 \\ C1 \subset C2 \bigwedge C2 \subseteq I[n]}}{\forall} \underset{x}{\forall} \pi[C2, x] \geq \pi[C1, x]]$$

TS_In[7]:=

Definition["WR", any[π, n], bound[allocation$_n$[x], allocation$_n$[y]],

$$WR[\pi, n] :\Leftrightarrow \left(n \in \mathbb{N} \bigwedge \underset{\substack{C \\ C \subseteq I[n]}}{\forall} \underset{x,y}{\forall} \left(\left(\underset{i \in C}{\forall} y_i \geq x_i\right) \Rightarrow \pi[C, y] \geq \pi[C, x]\right)\right)]$$

TS_In[8]:= $\text{Definition}\Big[\text{"WR contrapositive"}, \text{any}[\pi, n], \text{bound}[\text{allocation}_n[x], \text{allocation}_n[y]],$

$$\text{WR}[\pi, n] :\Leftrightarrow \left(n \in \mathbb{N} \bigwedge \underset{\substack{C \\ C\subseteq I[n]}}{\forall} \underset{x,y}{\forall} \left(\pi[C, x] > \pi[C, y] \Rightarrow \left(\underset{i\in C}{\exists} x_i > y_i \right) \right) \right) \Big]$$

TS_In[9]:=

$\text{Definition}\Big[\text{"SR"}, \text{any}[\pi, n], \text{bound}[\text{allocation}_n[x], \text{allocation}_n[y]],$

$$\text{SR}[\pi, n] :\Leftrightarrow \left(n \in \mathbb{N} \bigwedge \underset{\substack{C \\ C\subseteq I[n] \bigwedge C\neq\emptyset}}{\forall} \underset{x,y}{\forall} \left(\left(\underset{i\in C}{\forall} y_i > x_i \right) \Rightarrow \pi[C, y] > \pi[C, x] \right) \right) \Big]$$

TS_In[10]:=

$\text{Definition}\Big[\text{"powerfunction"}, \text{any}[\pi, n],$

$$\text{powerfunction}[\pi, n] :\Leftrightarrow \bigwedge \begin{cases} \text{WC}[\pi, n] \\ \text{WR}[\pi, n] \\ \text{SR}[\pi, n] \end{cases} \Big]$$

TS_In[11]:=
```
Theory["powerfunction",
  Definition["agents"]
  Definition["allocation"]
  Definition["WC"]
  Definition["WR"]
  Definition["SR"]
  Definition["powerfunction"]
]
```

Lemma 1

TS_In[12]:= $\text{Lemma}\Big[\text{"powerfunction-independent"}, \text{any}[\pi, n, C, x, y],$
$\text{with}[\text{allocation}_n[x] \bigwedge \text{allocation}_n[y] \bigwedge C \subseteq I[n] \bigwedge \text{powerfunction}[\pi, n]],$
$\underset{i\in C}{\forall} (x_i = y_i) \Rightarrow (\pi[C, x] = \pi[C, y])\Big]$

There is a potential problem in the next axiom, since it should be stated for real numbers only. Philosophy, trust in a proof is to read the proof.

TS_In[13]:= $\text{Axiom}[\text{"trichotomy"}, \text{any}[a, b],$
$(a = b) \Leftrightarrow (a \geq b \bigwedge b \geq a)]$

```
PLemma1 = Prove[Lemma["powerfunction-independent"],
  using → {Definition["powerfunction"], Definition["WR"], Axiom["trichotomy"]},
  by → PredicateProver, SearchDepth → 100]
```

- ProofObject -

```
Save["eco1.m", {PLemma1}]
```

```
PLemma2 = Prove[Lemma["powerfunction-independent"],
  using → {Theory["powerfunction"], Axiom["trichotomy"]},
  by → PredicateProver, SearchDepth → 250]
```

- ProofObject -

```
Save["eco1full.m", {PLemma2}]
```

```
? branches
```

```
Option of ProofSimplifier with possible values: Proved, Pending, Failed,
  Disproved and list combinations of these. All (default) means list of all.
```

```
PLemma1Simp = Block[{$RecursionLimit = Infinity}, Transform[PLemma1,
   by → ProofSimplifier, TransformerOptions → {branches → Proved, steps → Useful}]]
```

```
- ProofObject -
```

```
Save["ecosimp1.m", {PLemma1Simp}]
```

```
PLemma2Simp = Block[{$RecursionLimit = Infinity}, Transform[PLemma2,
   by → ProofSimplifier, TransformerOptions → {branches → Proved, steps → Useful}]]
```

```
Not used: {Definition (allocation), Definition (WC), Definition (SR)}

Not used: {Definition (allocation), Definition (WC), Definition (SR)}
```

```
Save["ecosimp1full.m", {PLemma2Simp}]
```

```
Options[SetTheoryPCSProver]
```

```
{DisableProver → {PND}, EarlyRewriting → False, TransformRanges → True,
 AllowIntroduceQuantifiers → False, ApplyBuiltIns → True, BackChaining → False,
 BackChainingDisjunction → False, BackChainingEquivalence → False,
 BackChainingImplication → False, ChooseFromFiniteSet → False,
 DisableInferenceRule → {KBComposeIntersection, KBFiniteChoice, KBInferNonEmpty},
 EarlyCaseDistinction → True, GRWTarget → {kb, goal}, InferMembershipIntersection → False,
 KBRWOnlySimplification → False, MatchExistential → TryAllAtOnce, ModusPonensException → None,
 ModusPonensKB → True, PNDLevel → 1, RWBooleanLiteralCombinations → True,
 RWCombine → False, RWExistentialGoal → False, RWHigherOrder → False, RWInnermost → True,
 RWInsideQuantifiers → False, RWSetOperators → False, RWTuples → True, SemanticMatch → True,
 SimplifyFormula → False, STPFunctionProperties → True, STPLevel → 100,
 STPMembershipByInclusion → False, TryAlternatives → False, TrySubgoal → False,
 TrySubgoalDisjunction → False, TrySubgoalEquivalence → False, TrySubgoalImplication → False,
 UseCyclicRules → False, UseEqualitiesFirst → True, UseNonMembership → True}
```

```
Options[PredicateProver]
```

```
{TryAlternatives → False, BackChaining → False, BackChainingDisjunction → False,
 BackChainingImplication → False, BackChainingEquivalence → False,
 TrySubgoalDisjunction → False, TrySubgoalImplication → False,
 TrySubgoalEquivalence → False, TrySubgoal → False, PNDLevel → 1}
```

```
ProofShow[]
```

```
Depth[PLemma1]
```

```
188
```

```
Depth[PLemma2]
```

```
372
```

```
Depth[PLemma1Simp]
```

```
70
```

```
Depth[PLemma2Simp]
```

```
74
```

Auxiliary Concepts

Permuted Allocation

TS_In[14]:= Definition["perm", any[x, σ, k],
 $perm[x, \sigma]_k := x_{\sigma[k]}$]

TS_In[15]:= Proposition["alloc perm", any[n, σ, x], with[$allocation_n[x] \wedge permutation[\sigma, I[n]]$],
 $allocation_n[perm[x, \sigma]]$]]

TS_In[16]:= Proposition["sum perm", any[σ, A, x], with[permutation[σ, A]],
 $\sum_{i \in A} x_{\sigma[i]} = \sum_{i \in A} x_i$]

```
Prove[Proposition["alloc perm"],
 using → {Proposition["sum perm"], Definition["allocation"], Definition["perm"]},
 by → SetTheoryPCSProver, ProverOptions → {AllowIntroduceQuantifiers → True}]
```

- ProofObject -

Allocation Swap

TS_In[17]:= Definition["swap", any[i, j, k],

$$\sigma_{i,j}[k] := \begin{cases} j \Leftarrow k = i \\ i \Leftarrow k = j \\ k \Leftarrow \text{otherwise} \end{cases}$$

]

TS_In[18]:= Definition["permutation", any[σ, A],
 $permutation[\sigma, A] :\Leftrightarrow \sigma :: A \xrightarrow{bij.} A$]

TS_In[19]:= Proposition["swapperm", any[n], with[appL[n]],
 $permutation[\sigma_{1,2}, I[n]]$]]

```
Prove[Proposition["swapperm"],
 using -> {Definition["swap"], Definition["permutation"]}, by -> SetTheoryPCSProver]
```

```
InverseFunction::ifun :
 Inverse functions are being used. Values may be lost for multivalued inverses. >>
```

$Aborted

```
ProofShow[]
```

TS_In[20]:= Proposition["swap idempotent", any[i], $\sigma_{1,2}[\sigma_{1,2}[i]] = i$]

```
Prove[Proposition["swap idempotent"], using → Definition["swap"], by → SetTheoryPCSProver]
```

- ProofObject -

TS_In[21]:= Lemma["perm swap", any[x],
 $perm[x, \sigma_{1,2}]_2 = x_1$]

```
Prove[Lemma["perm swap"],
 using → {Definition["perm"], Definition["swap"]}, by → SetTheoryPCSProver]
```

- ProofObject -

Additional Axioms

TS_In[22]:=

```
Definition["AN", any[π, n], bound[allocation_n[x], allocation_n[y]],
  AN[π, n] :⇔ ∀_{σ, permutation[σ,I[n]]} ∀_{Cx,Cy,x,y, Cx⊆I[n]∧Cy⊆I[n]} (∀_i (((i ∈ Cx) ⇔ (σ[i] ∈ Cy)) ∧ (x_i = y_{σ[i]})) ⇒
      (π[Cx, x] = π[Cy, y]))]
```

TS_In[23]:=

```
Lemma["AN perm", any[n, π, x], with[appL[n] ∧ AN[π, n] ∧ allocation_n[x]],
  π[{1}, x] = π[{2}, perm[x, σ_{1,2}]]]
```

```
PLemmaANperm =
 Prove[Lemma["AN perm"], using → {Proposition["swapperm"], Proposition["swap idempotent"],
    Proposition["alloc perm"], Lemma["1inI"], Lemma["2inI"], Definition["AN"],
    Definition["perm"], Definition["swap"]}, by → SetTheoryPCSProver, SearchDepth → 80,
  ProverOptions → {AllowIntroduceQuantifiers → True, UseCyclicRules → False,
    EarlyRewriting → False, GRWTarget -> {"goal", "kb"}, DisableProver → {STC}}]
```

```
$Aborted
```

```
Transform[PLemmaANperm, TransformerOptions → {branches → Proved, steps → Useful}]
```

```
ProofShow[]
```

Continuous2 means that the function is continuous in its 2nd component.

TS_In[24]:=

```
Definition["CX", any[π, n],
  CX[π, n] :⇔ continuous2[π]]
```

TS_In[25]:=

```
Definition["RE", any[π, n], bound[allocation_n[x]],
  RE[π, n] :⇔ ∀_{C,i,x, C⊆I[n]} (i ∉ C ∧ π[{i}, x] > 0 ⇒ π[C ⋃ {i}, x] > π[C, x])]
```

TS_In[26]:=

```
Definition["WinLose", any[n, x, y],
  W[n, x, y] := {i | _{i∈I[n]} y_i > x_i}  "W"
  L[n, x, y] := {i | _{i∈I[n]} x_i > y_i}  "L"]
```

TS_In[27]:=

```
Definition["dominates", any[π, n, x, y],
  dominates[y, x, π, n] :⇔ π[W[n, x, y], x] > π[L[n, x, y], x]]
```

```
Lemma["ANdominates-aux", any[n, x, y],
 with[allocation_n[x] ∧ allocation_n[y] ∧ (y_i > x_i ⇔ (i = 1)) ∧ (y_i < x_i ⇔ (i = 2))],
   ∀_{π, powerfunction[π,n]∧AN[π,n]} dominates[y, x, π, n] ⇔ x_1 > x_2]
```

TS_In[28]:=

```
Lemma["ANdominates", any[n, x, y],
  with[n ∈ ℕ ∧ n ≥ 2 ∧ allocation_n[x] ∧ allocation_n[y] ∧ (W[n, x, y] = {1}) ∧ (L[n, x, y] = {2})],
    ∀_{π, AN[π,n]∧powerfunction[π,n]} (dominates[y, x, π, n] ⇒ x_1 > x_2)]
```

```
PLemmaANdominates =
 Prove[Lemma["ANdominates"], using → {Lemma["AN perm"], Definition["dominates"],
    Definition["powerfunction"], Definition["SR"]}, by → SetTheoryPCSProver,
  SearchDepth → 50, ProverOptions → {AllowIntroduceQuantifiers → True,
    UseCyclicRules → True, EarlyRewriting → False, DisableProver → {STKBR, STC}}]
```

```
$Aborted
```

```
ProofShow[]
```

TS_In[29]:= Definition["appAlloc", any[n, x],
 appAlloc[n, x] :⇔ (appL[n] ⋀ $allocation_n$[x])]

TS_In[30]:= Lemma["ANdominates 1", any[n, x, y],
 with[appAlloc[n, x] ⋀ $allocation_n$[y] ⋀ (W[n, x, y] = {1}) ⋀ (L[n, x, y] = {2})],
 $\underset{AN[\pi,n]\wedge powerfunction[\pi,n]}{\underset{\pi}{\forall}}$ (dominates[y, x, π, n] ⇒ $x_1 > x_2$)]

TS_In[31]:= Lemma["ANdominates 2", any[n, x, y],
 with[appAlloc[n, x] ⋀ $allocation_n$[y] ⋀ (W[n, x, y] = {1}) ⋀ (L[n, x, y] = {2})],
 $\underset{AN[\pi,n]\wedge powerfunction[\pi,n]}{\underset{\pi}{\forall}}$ ($x_1 > x_2$ ⇒ dominates[y, x, π, n])]

TS_In[32]:= Definition["appPowAlloc", any[n, π, x],
 appPowAlloc[n, π, x] :⇔ (appAlloc[n, x] ⋀ AN[π, n])]

TS_In[33]:= Lemma["AN all", any[n, π, x], with[appAlloc[n, x] ⋀ AN[π, n]],
 $\bigwedge\begin{cases} \pi[\{1\}, x] = \pi[\{2\}, perm[x, \sigma_{1,2}]] \\ allocation_n[perm[x, \sigma_{1,2}]] \end{cases}$]

```
Prove[Lemma["AN all"], using → {Lemma["AN perm"], Proposition["alloc perm"],
   Proposition["swapperm"], Definition["appAlloc"]}, by → PredicateProver]
```

- ProofObject -

```
PLemmaANdominates = Prove[Lemma["ANdominates 1"],
  using → {Definition["dominates"], Lemma["AN all"], Definition["WR contrapositive"],
    Lemma["2inI"], Definition["powerfunction"], Lemma["perm swap"], Definition["appAlloc"]},
  by → SetTheoryPCSProver, SearchDepth → 80, ProverOptions →
   {AllowIntroduceQuantifiers → False, RWCombine → True, DisableProver → {STC},
    DisableInferenceRule → {¢KBSetEquality, ¢KBInclusion}}]
```

- ProofObject -

```
Block[{$RecursionLimit = ∞}, ProofShow[]]
```

```
Block[{$RecursionLimit = ∞},
 Transform[PLemmaANdominates, TransformerOptions → {branches → Proved, steps → Useful}]]
```

- ProofObject -

```
Block[{$RecursionLimit = ∞}, ProofShow[PLemmaANdominates]]
```

```
Prove[Lemma["ANdominates 2"],
 using → {Definition["dominates"], Lemma["AN all"], Definition["SR"], Lemma["2inI"],
   Definition["powerfunction"], Lemma["perm swap"], Definition["appAlloc"]},
 builtin → {Builtin ["Operators"]["¬", "="]}, by → SetTheoryPCSProver,
 SearchDepth → 80, ProverOptions → {AllowIntroduceQuantifiers → False, RWCombine → True,
   DisableProver → {STP}, DisableInferenceRule → {¢KBSetEquality, ¢KBInclusion}}]
```

- ProofObject -

```
Block[{$RecursionLimit = ∞}, ProofShow[]]

Options[SetTheoryPCSProver]

{DisableProver → {PND}, EarlyRewriting → False, TransformRanges → True,
 AllowIntroduceQuantifiers → False, ApplyBuiltIns → True, BackChaining → False,
 BackChainingDisjunction → False, BackChainingEquivalence → False,
 BackChainingImplication → False, ChooseFromFiniteSet → False,
 DisableInferenceRule → {KBComposeIntersection, KBFiniteChoice, KBInferNonEmpty},
 EarlyCaseDistinction → True, GRWTarget → {kb, goal}, InferMembershipIntersection → False,
 KBRWOnlySimplification → False, MatchExistential → TryAllAtOnce, ModusPonensException → None,
 ModusPonensKB → True, PNDLevel → 1, RWBooleanLiteralCombinations → True,
 RWCombine → False, RWExistentialGoal → False, RWHigherOrder → False, RWInnermost → True,
 RWInsideQuantifiers → False, RWSetOperators → False, RWTuples → True, SemanticMatch → True,
 SimplifyFormula → False, STPFunctionProperties → True, STPLevel → 100,
 STPMembershipByInclusion → False, TryAlternatives → False, TrySubgoal → False,
 TrySubgoalDisjunction → False, TrySubgoalEquivalence → False, TrySubgoalImplication → False,
 UseCyclicRules → False, UseEqualitiesFirst → True, UseNonMembership → True}

ResetComputation[]
```