Come scrivere i simboli

∈ \mem
⊆ \subseteq
∅ \emptyset
∩ \cap
∀ \forall
→ \to
↔ \iff
$_1$ \1
$_2$ \2
...

Teoremi e assiomi

def A⊆B := ∀Z, Z ∈ A → Z ∈ B
axiom ax_extensionality1: ∀A B, (∀Z, Z∈A ↔ Z∈B) → A=B
axiom ax_extensionality2: ∀A B, A = B → (∀Z, Z∈A ↔ Z∈B)
axiom ax_empty: ∀X, (X ∈ ∅)→ False
axiom ax_intersect1: ∀A B, ∀Z, (Z ∈ A∩B → Z∈A ∧ Z∈B)
axiom ax_intersect2: ∀A B, ∀Z, (Z∈A ∧ Z∈B → Z ∈ A∩B)

lab_1
theorem reflexivity_inclusion: ∀A, A ⊆ A
theorem transitivity_inclusion: ∀A B C, A⊆B → B⊆C → A⊆C
theorem subset_to_eq: ∀A B, A⊆B → B⊆A → A=B
theorem eq_to_subset1: ∀A B, A=B → A⊆B
theorem eq_to_subset2: ∀A B, A=B → B⊆A
theorem transitivity_equality: ∀A B C, A=B → B=C → A=C
lab_1
theorem emptyset_is_subset: ∀A, ∅⊆A
theorem intersection_idempotent: ∀A, A∩A = A
theorem intersect_empty: ∀A, A∩∅ = ∅
theorem subseteq_emptyset: ∀X, X⊆∅ → X=∅
theorem intersect_commutative_aux: ∀A B, A∩B ⊆ B∩A
theorem intersect_commutative: ∀A B, A∩B = B∩A
theorem intersect_monotone: ∀A B A' B', A⊆A' → B⊆B' → A∩B ⊆ A'∩B'
theorem intersect_is_subset: ∀A B, A∩B ⊆ A

Sintassi

. assume A: set
∀-introduzione
usato per dimostrare ∀A, P
la conclusione diventa P

. suppose P as H
→-introduzione
usato per dimostrare P → Q
la conclusione diventa Q
si ha una nuova ipotesi P di nome H
dopo P è possibile specificare "that is equivalent to R" per espandere le definizioni contenute in P
in tal caso la nuova ipotesi ha la forma R e non più P
"as H" può essere omesso; in tal caso si può usare l'ipotesi solo al passo successivo con thus

. we split the proof
↔-introduzione
usato per dimostare P ↔ Q
bisogna aprire due sottoprove, la prima di P → Q e la seconda di Q → P
le due sottoprove iniziano con . e sono indentate

∧-introduzione
usato per dimostrare P ∧ Q
bisogna aprire due sottoprove, la prima di P e la seconda di Q
le due sottoprove iniziano con . e sono indentate

. we need to prove P
esplicita cosa si sta dimostrando
non corrisponde a un passo logico
può essere seguito da "that is equivalent to Q" per espandere le definizioni contenute in P

. by H it suffices to prove P
∀-eliminazione + →-eliminazione
forma alternativa di ∀-eliminazione + →-eliminazione
si use quando la conclusione corrente è Q e quando H, dopo l'applicazione di zero o più ∀-eliminazioni, ha la forma P → Q
la nuova conclusione da dimostrare diventa P

. by H1, ..., Hn done
∀-eliminazione + →-eliminazione + ↔-eliminazione + ∧-introduzione + ⊥-eliminazione
si dimostra la conclusione del teorema combinando assieme le n ipotesi tramite un numero arbitrario di applicazione delle regole elencate subito sopra
e ri-spiegate qua sotto
si può usare "thus" prima di "by" per aggiugere l'ultima ipotesi introdotta, anonima o meno
la dimostrazione (o la sotto-dimostrazione) è conclusa

∀-eliminazione: da un'ipotesi ∀x, P si ottiene P in un caso specifico, ottenuto
sostituendo a x qualcosa
Esempio: da ∀A, ∅ ⊆ A si può ricavare ∅ ⊆ ∅ sostituendo ad A l'insieme vuoto ∅
→-eliminazione: da un'ipotesi P → Q e da un'ipotesi P si ricava Q
↔-eliminazione: da un'ipotesi P ↔ Q si ricava sia P → Q che Q → P
∧-introduzione: da un'ipotesi P e da un'ipotesi Q si ricava P ∧ Q
⊥-eliminazione: da un'ipotesi False si ricava qualunque cosa

. by H1, ..., Hn we proved P as H

come il caso precedente, ma invece di dimostrare la conclusione si ricava una nuova ipotesi P alla quale viene data il nome H
dopo P è possibile specificare "that is equivalent to R" per espandere le definizioni contenute in P
in tal caso la nuova ipotesi ha la forma R e non più P
"as H" può essere omesso; in tal caso si può usare l'ipotesi solo al passo successivo con thus
la conclusione da dimostrare non cambia

. by H1, ..., Hn we proved P as H_1 and Q as H_2

come il caso precedente, ma invece di concludere P ∧ Q
si applica un passo di ∧-eliminazione concludendo separatamente
sia P che Q. Alle due conclusioni vengono date i nomi indicati