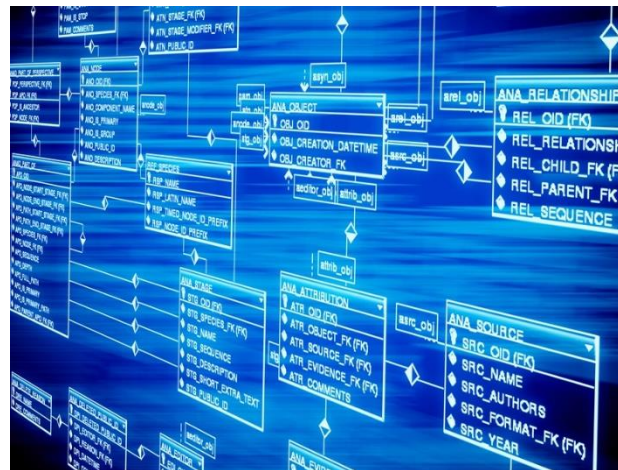


Politecnico di Milano - AA 2018-2019

Prof.ssa Sara Comai

Strategie di progettazione concettuale



Strategie di progetto

Lo sviluppo dello schema si può eseguire seguendo due strategie fondamentali:

- Top-Down
- Bottom-Up

Per quanto riguarda le strategie bottom-up, vedremo i casi:

- "A macchia d'olio"
- Mista

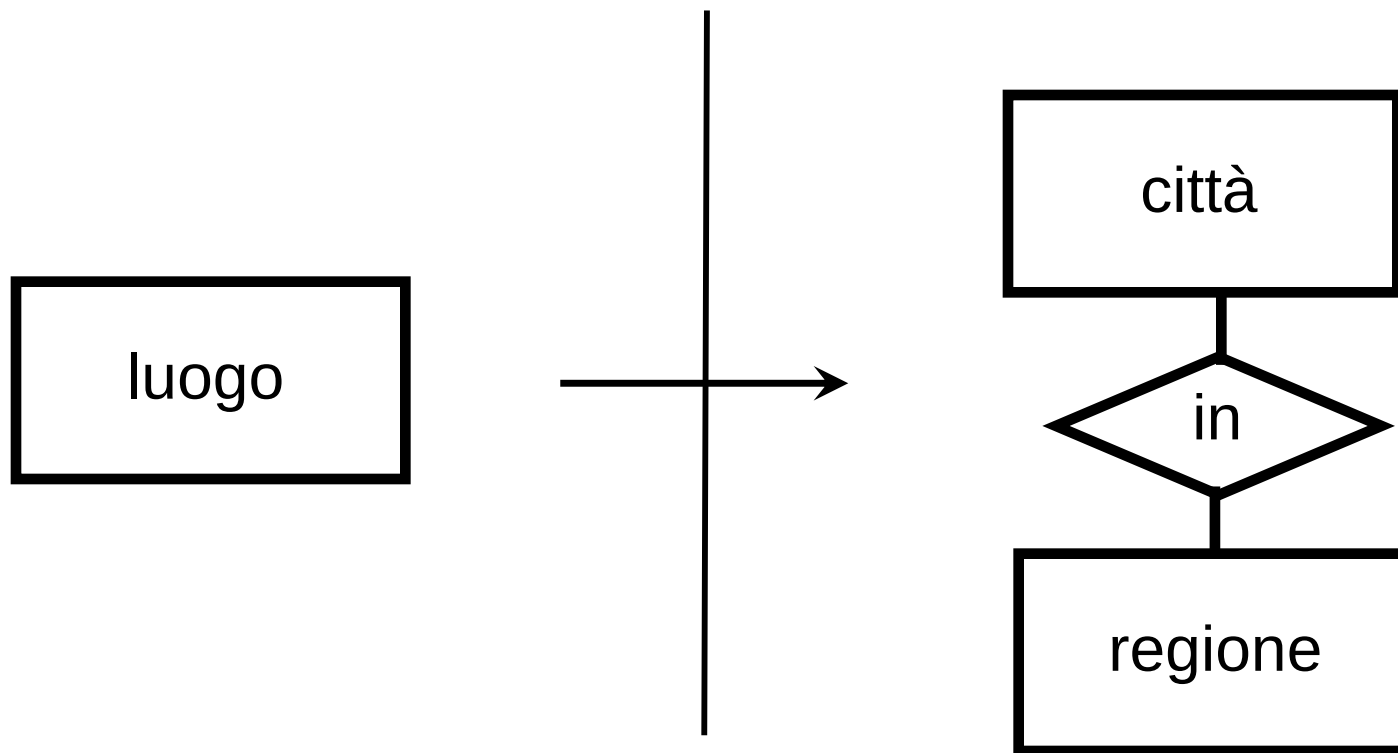
Progetto top-down e bottom-up

- **Top-down**: si procede per raffinamenti a partire da una descrizione che comprende TUTTA la realtà di interesse
 - il progetto è più ordinato e razionale, ma più difficile (il progettista deve possedere una "visione d'insieme")
- **Bottom-up**: si disegnano separatamente aspetti della realtà e poi li si integra costruendo un unico schema
 - si possono prendere decisioni differenti nell'affrontare i sotto-problemi, che si tradurranno in "conflitti" (modelli diversi della stessa realtà).

Strategia top-down “pura”

- A partire dalle specifiche si costruisce uno **schema iniziale**
- Dallo schema iniziale si arriva per raffinamenti successivi a schemi intermedi e poi allo **schema finale**
- I raffinamenti prevedono l'uso di **trasformazioni elementari** (primitive) che operano sul singolo concetto per descriverlo con maggior dettaglio

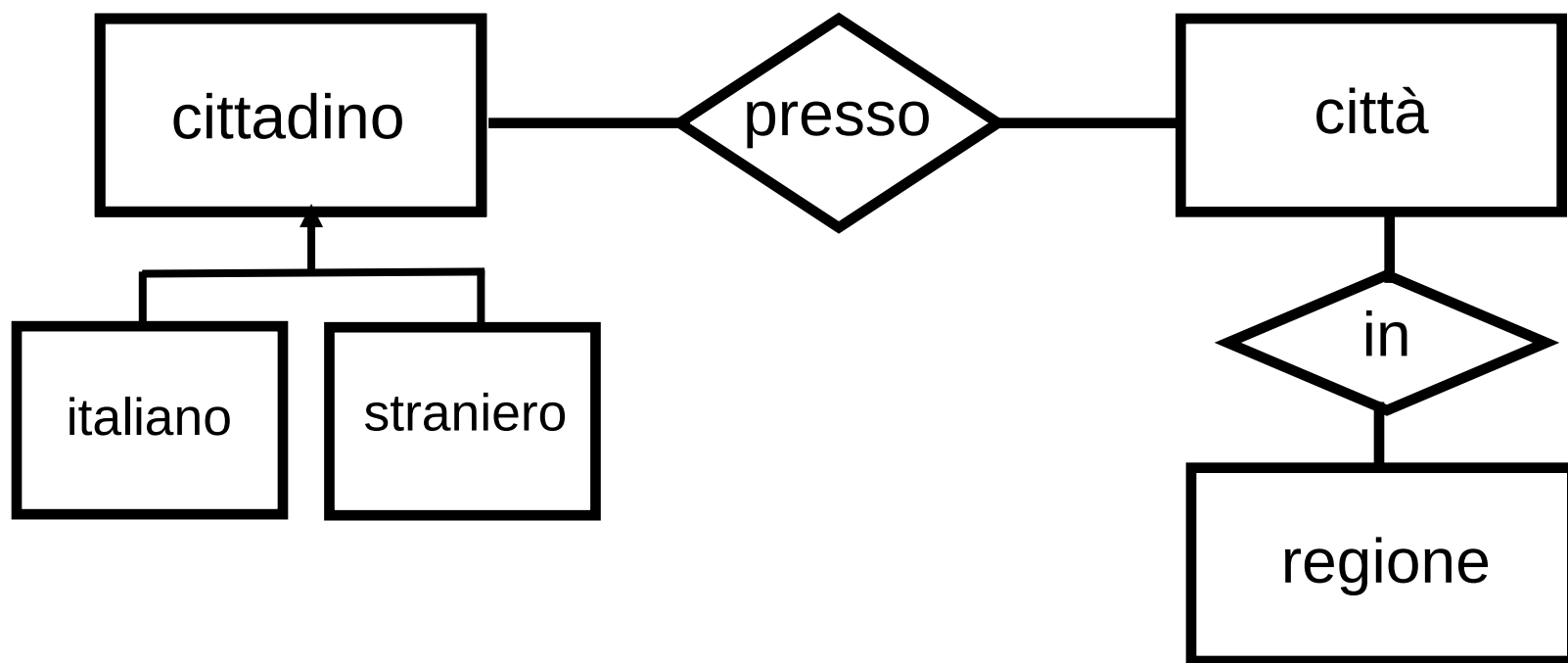
Trasformazione elementare top-down



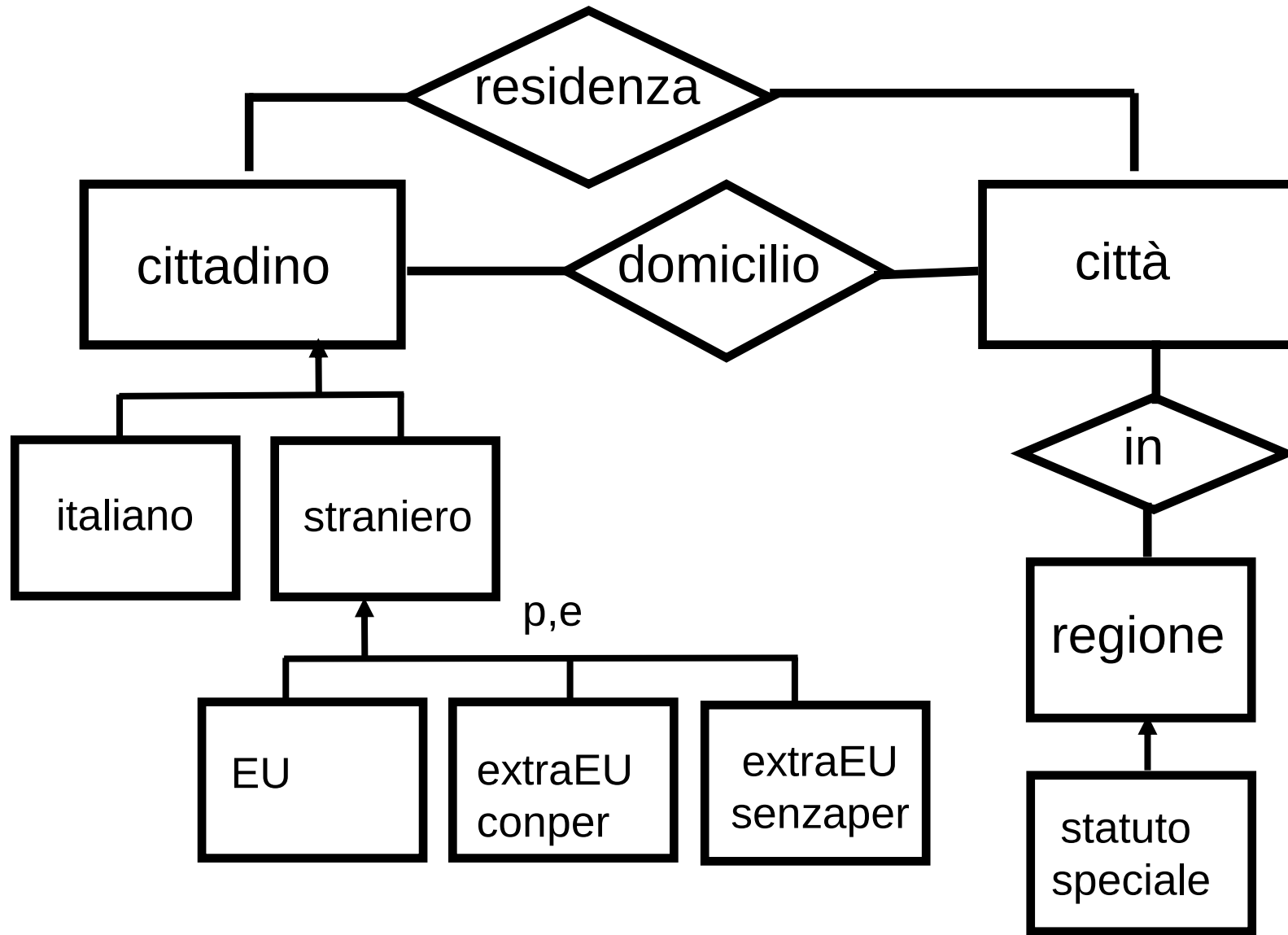
DB ANAGRAFICO, passo 1



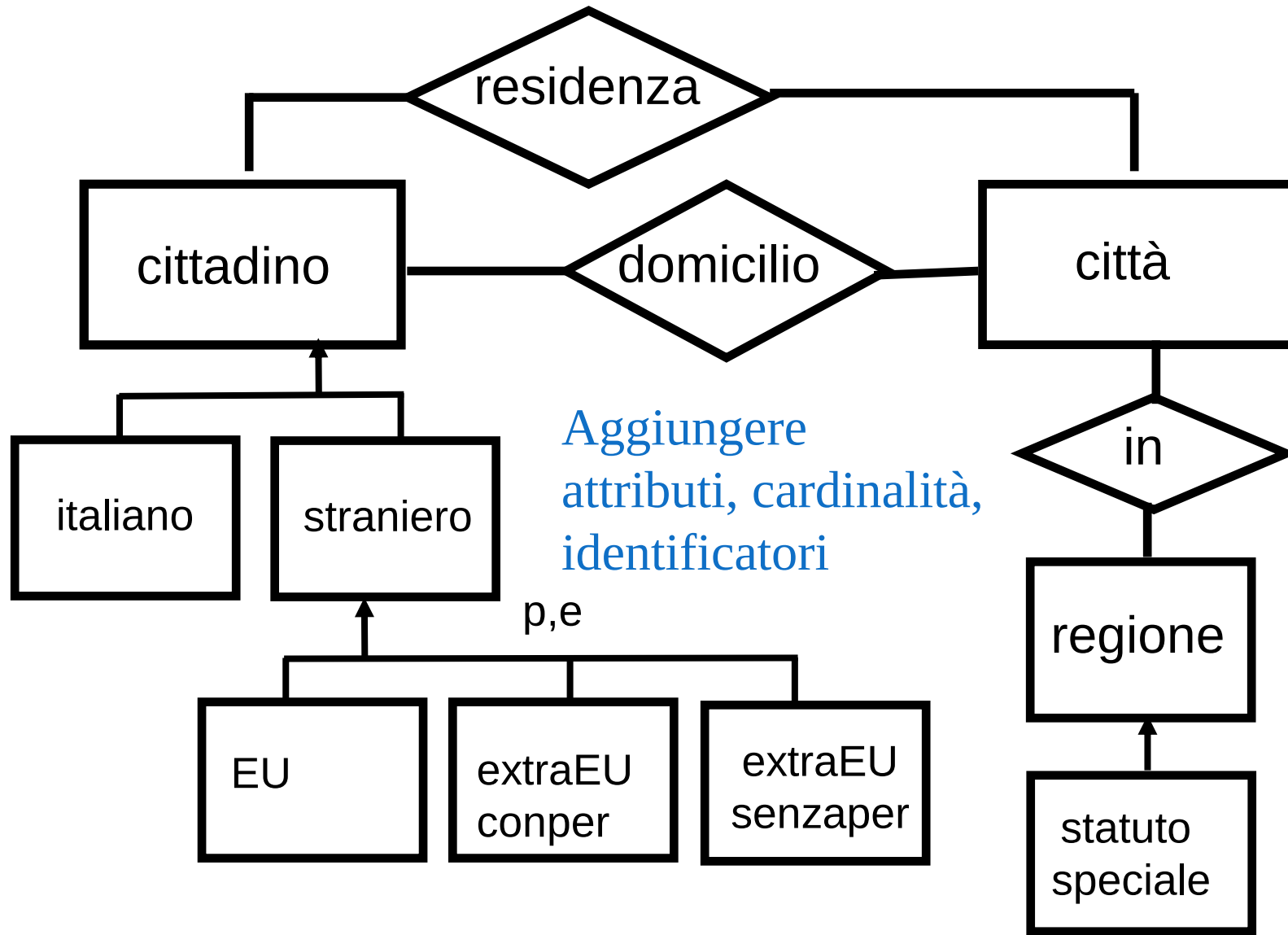
DB ANAGRAFICO, passo 2



DB ANAGRAFICO, passo 3



DB ANAGRAFICO, passo 4



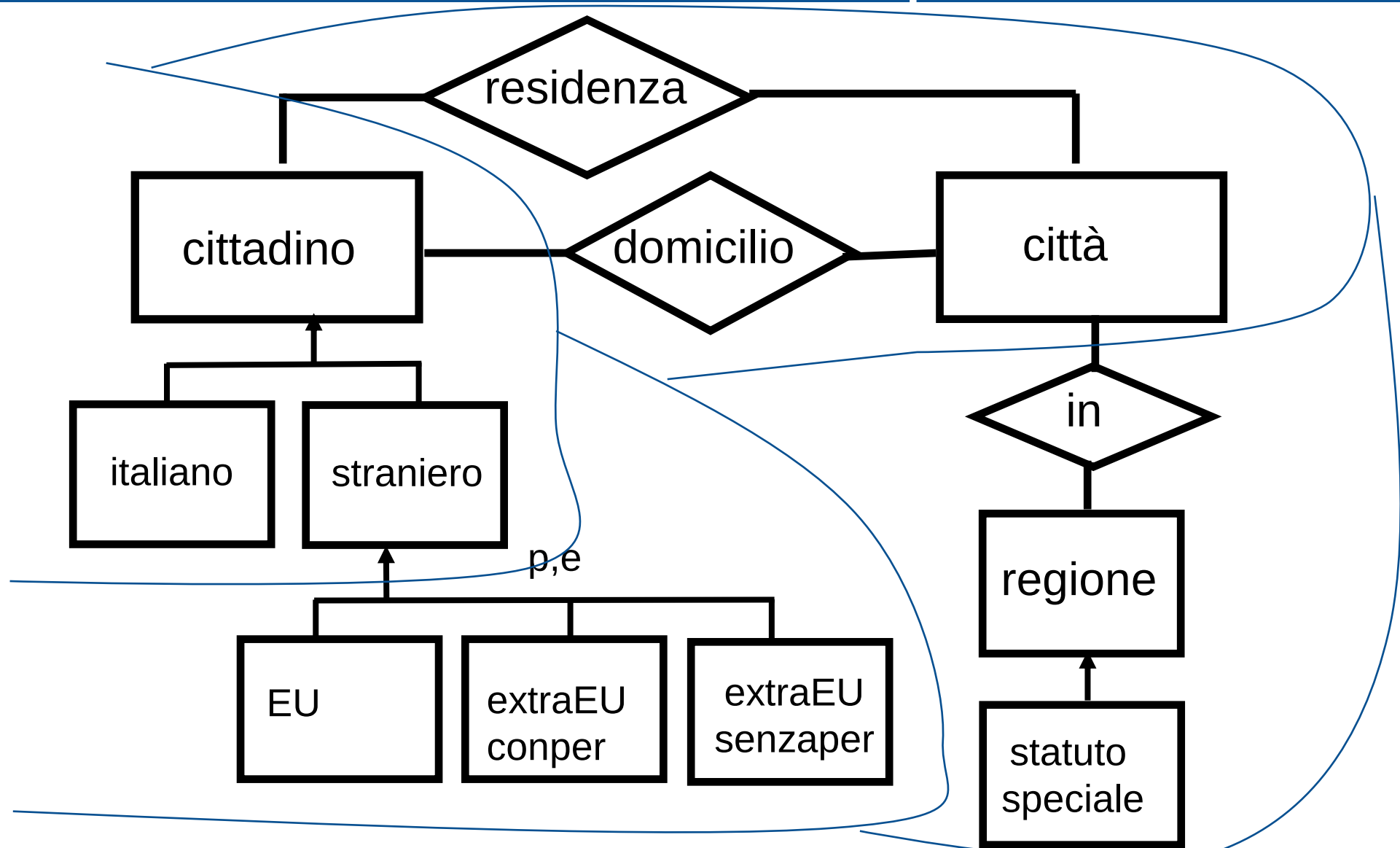
Valutazione di una strategia top down

- Vantaggi:
 - il progettista descrive inizialmente lo schema trascurando i dettagli
 - precisa lo schema **gradualmente**
- Problema:
 - non va bene per applicazioni complesse perché è difficile avere una **visione globale precisa iniziale di tutte** le componenti del sistema

Strategia "a macchia d'olio"

- Le specifiche nascono progressivamente, affrontando i requisiti fino al massimo dettaglio e "avanzando" per sottoproblemi
- La tecnica è adatta a tradurre pian piano una descrizione testuale in un diagramma

DB ANAGRAFICO, passo 4



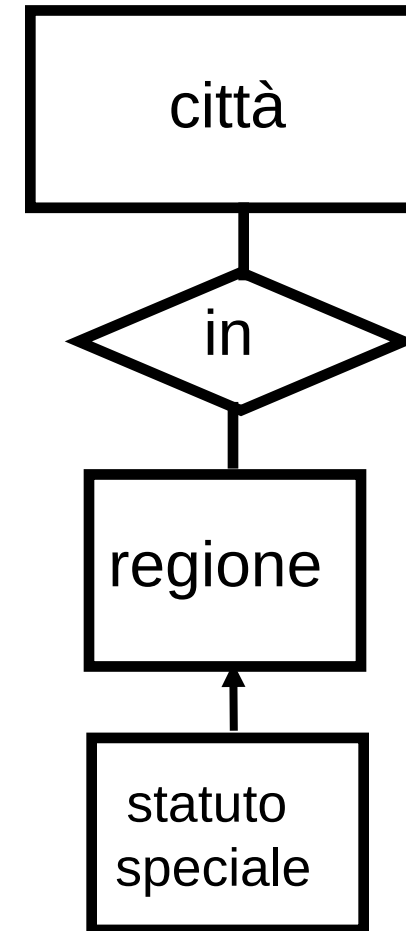
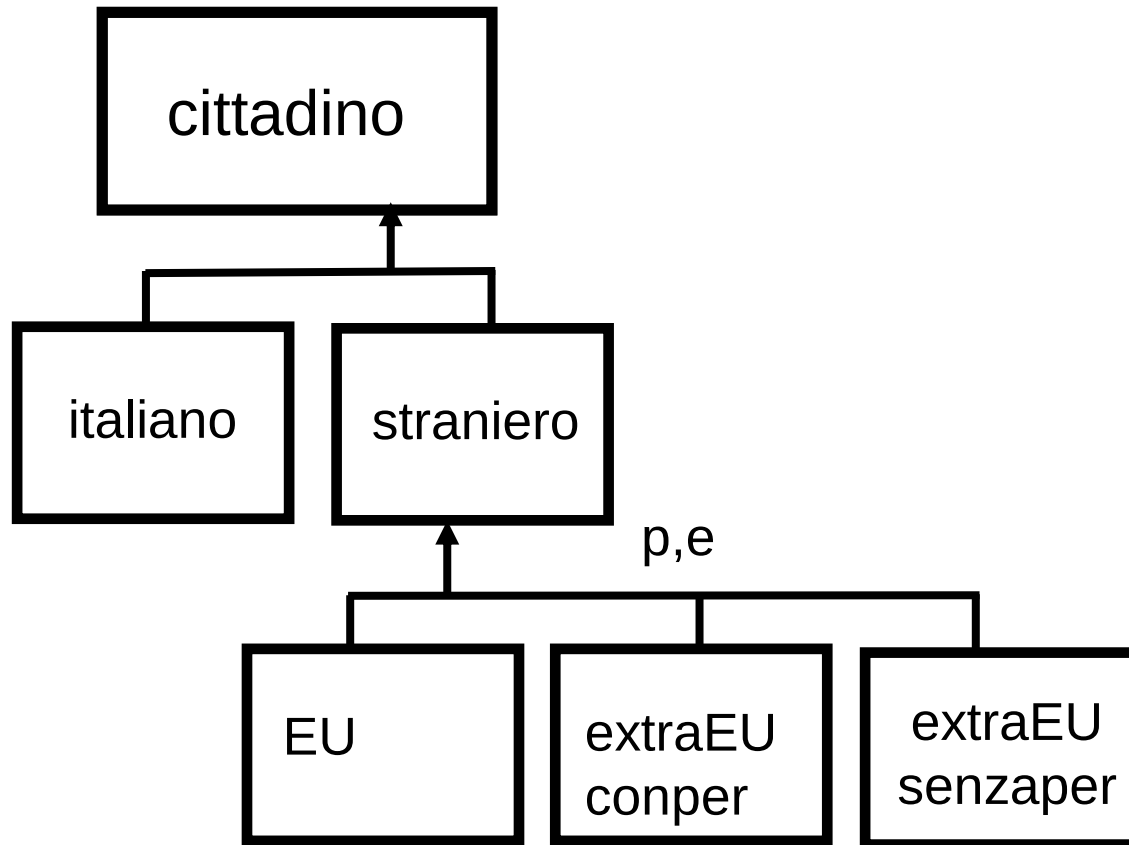
Valutazione della strategia "a macchia d'olio"

- La tecnica è adatta a tradurre pian piano una descrizione testuale in un diagramma
- Pur essendo bottom-up, il progettista analizza le specifiche in modo "stratificato" e le aggiunge progressivamente ad un unico schema, perciò i conflitti sono meno probabili

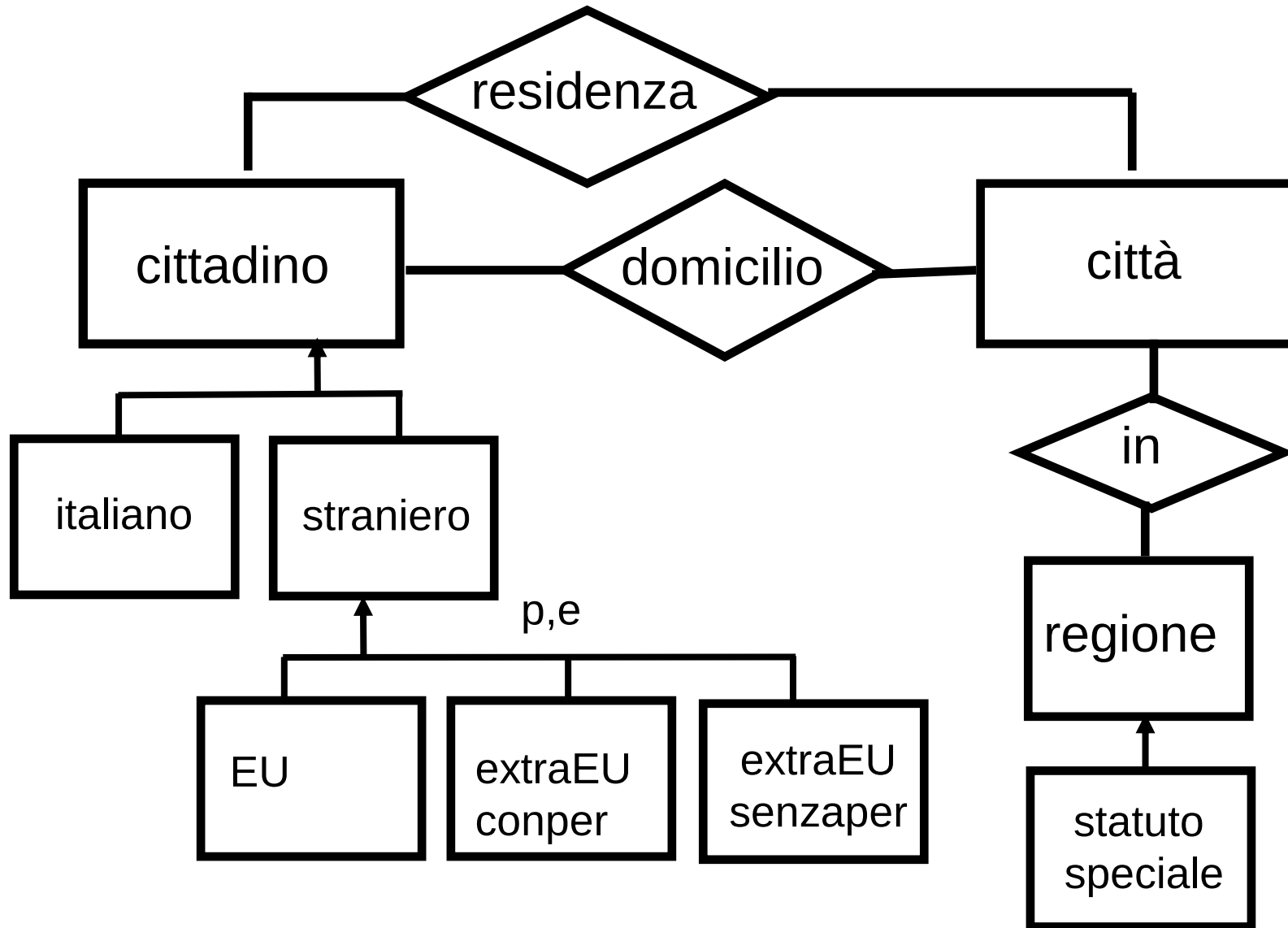
Strategia Mista

- Adatta di fronte a progetti ampi
- Si suddividono le specifiche in parti (ad esempio: le funzioni amministrazione, personale, marketing, vendita, produzione di una azienda)
- Si realizzano top-down le varie parti
- Si realizza (bottom-up) l'integrazione delle varie parti sviluppate

Sviluppo di due sottoschemi



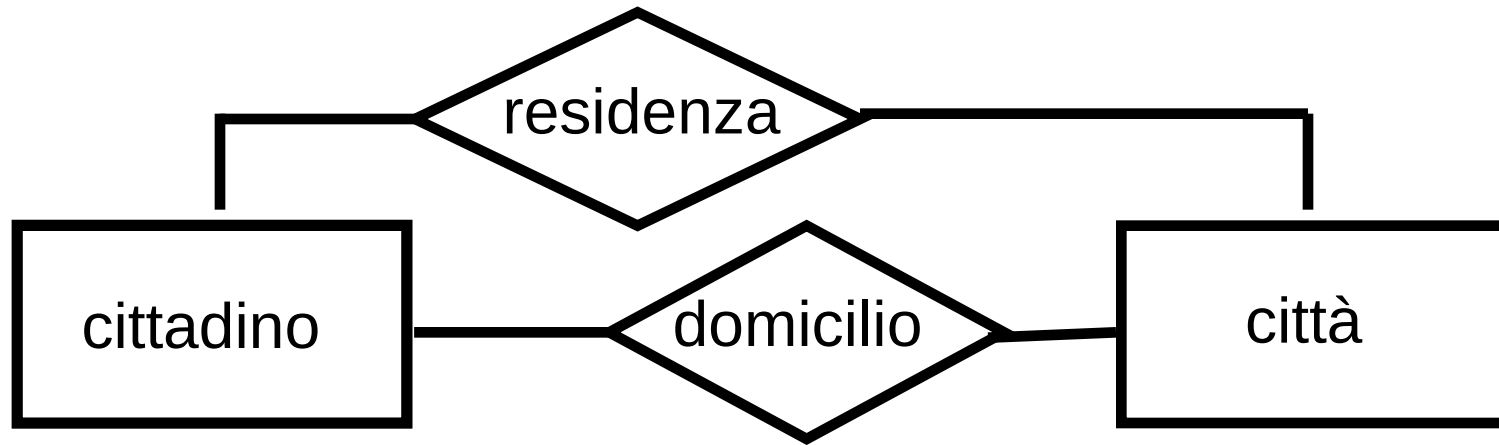
Integrazione



Valutazione della strategia mista

- Vantaggi:
 - diversi progettisti elaborano gli schemi parziali, il singolo progettista ha una visione più precisa del proprio settore
- Problema:
 - conflitti e difficoltà di [integrazione](#)
- Soluzione possibile:
 - sviluppare un piccolo schema dei soli concetti principali (schema scheletro) in modo top-down e attenersi alle scelte presenti nello schema scheletro in tutti gli altri schemi

Schema Scheletro



Sintesi

- Un progettista esperto procede (inconsiamente) sia in modo top-down che in modo bottom-up
- Per affrontare gli esercizi, la tecnica a macchia d'olio viene usata spesso
- In ogni caso, è possibile (e molto conveniente)
DOCUMENTARE un progetto in modo top-down a posteriori

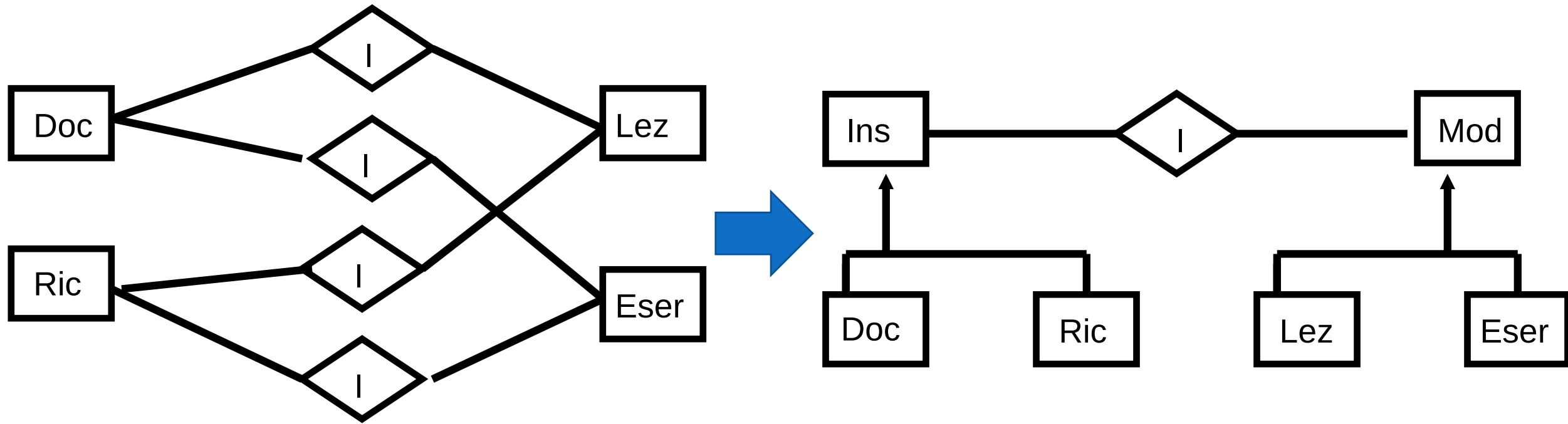
Qualità di uno schema concettuale

- Completezza
- Correttezza
- Leggibilità
- Minimalità
- Auto-Esplicatività

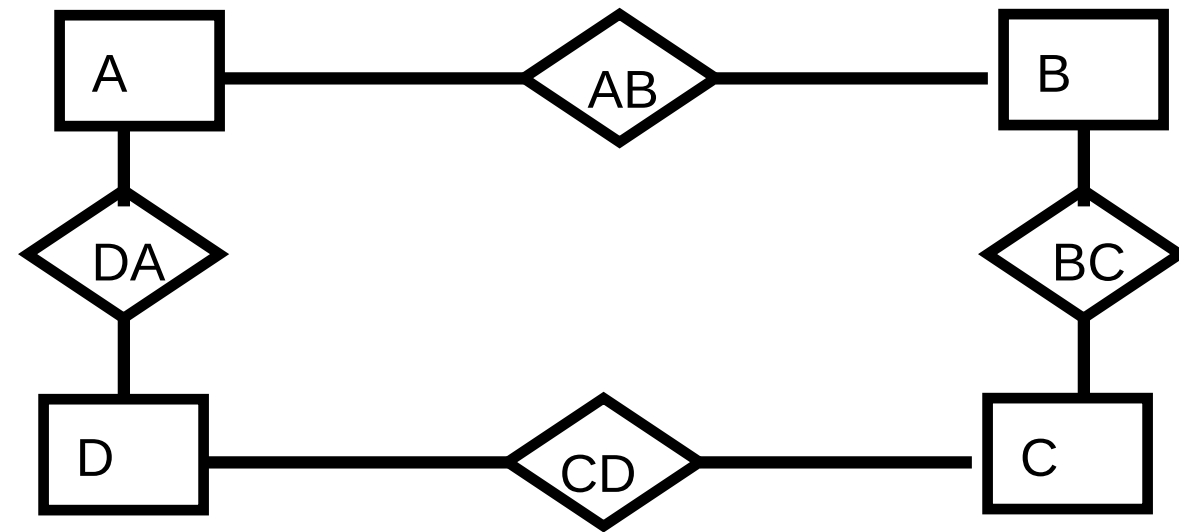
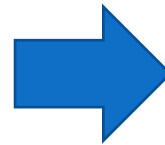
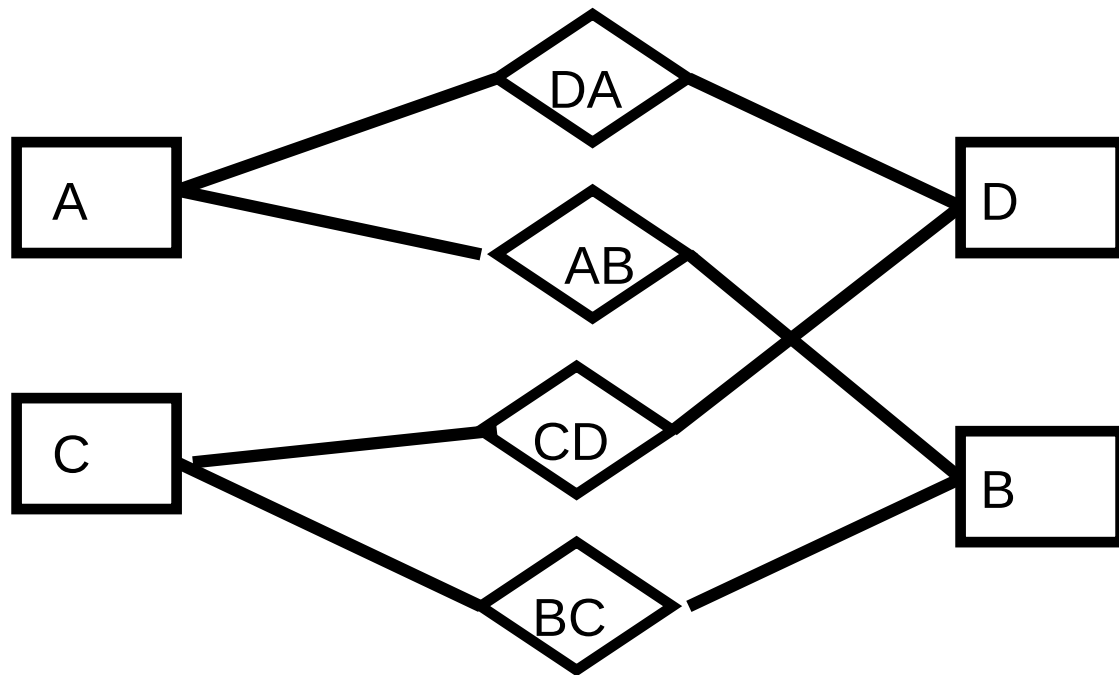
Completezza e correttezza

- Rappresentare in modo completo e corretto i requisiti
- Sono proprietà ovvie ma
 - Per la completezza: assicurarsi che i dati consentano di eseguire tutte le applicazioni
 - Per la correttezza: assicurarsi che sia possibile popolare la base di dati anche con informazione parzialmente incompleta durante fasi iniziali della sua evoluzione

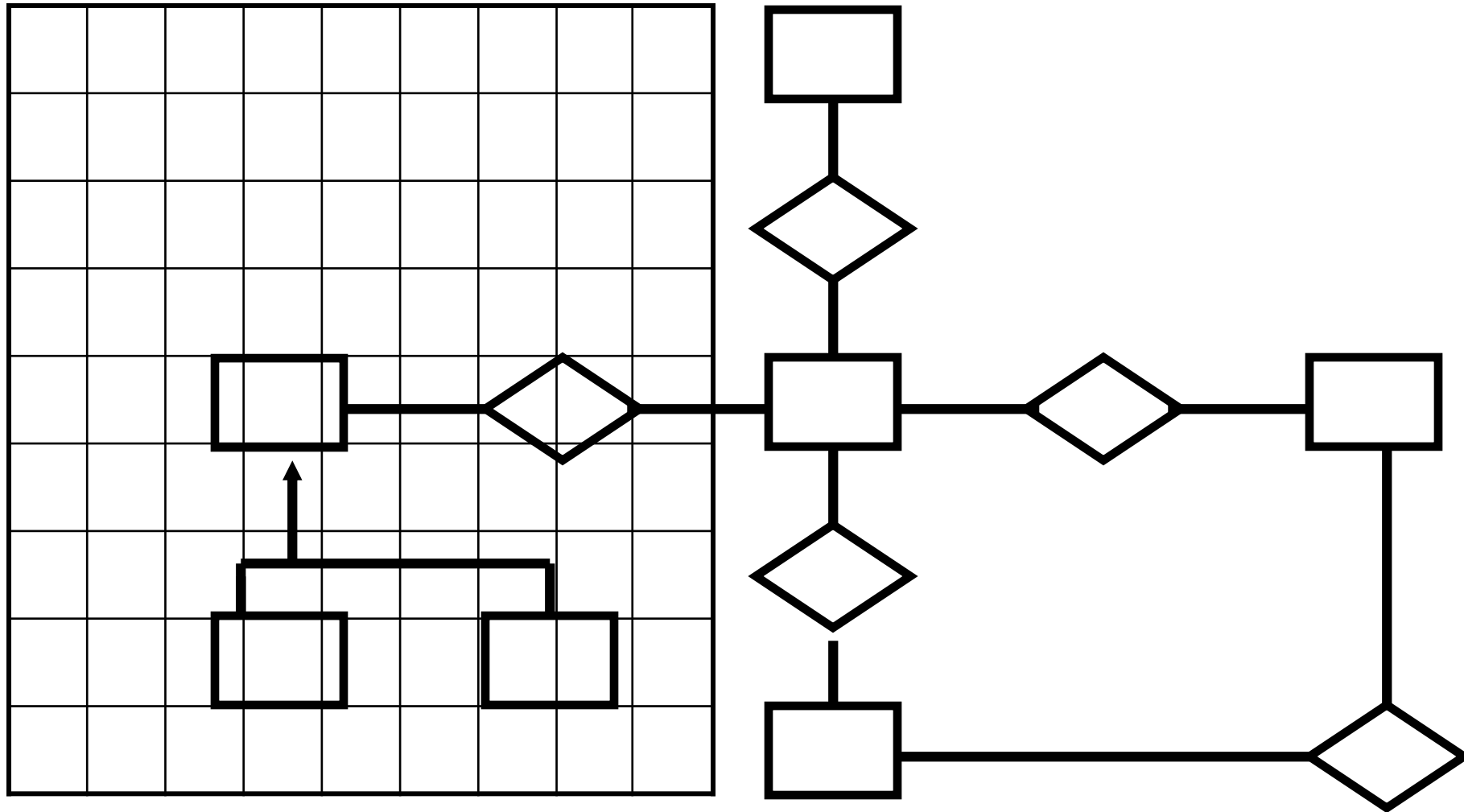
Leggibilità concettuale



Leggibilità grafica



Disegnare in una griglia



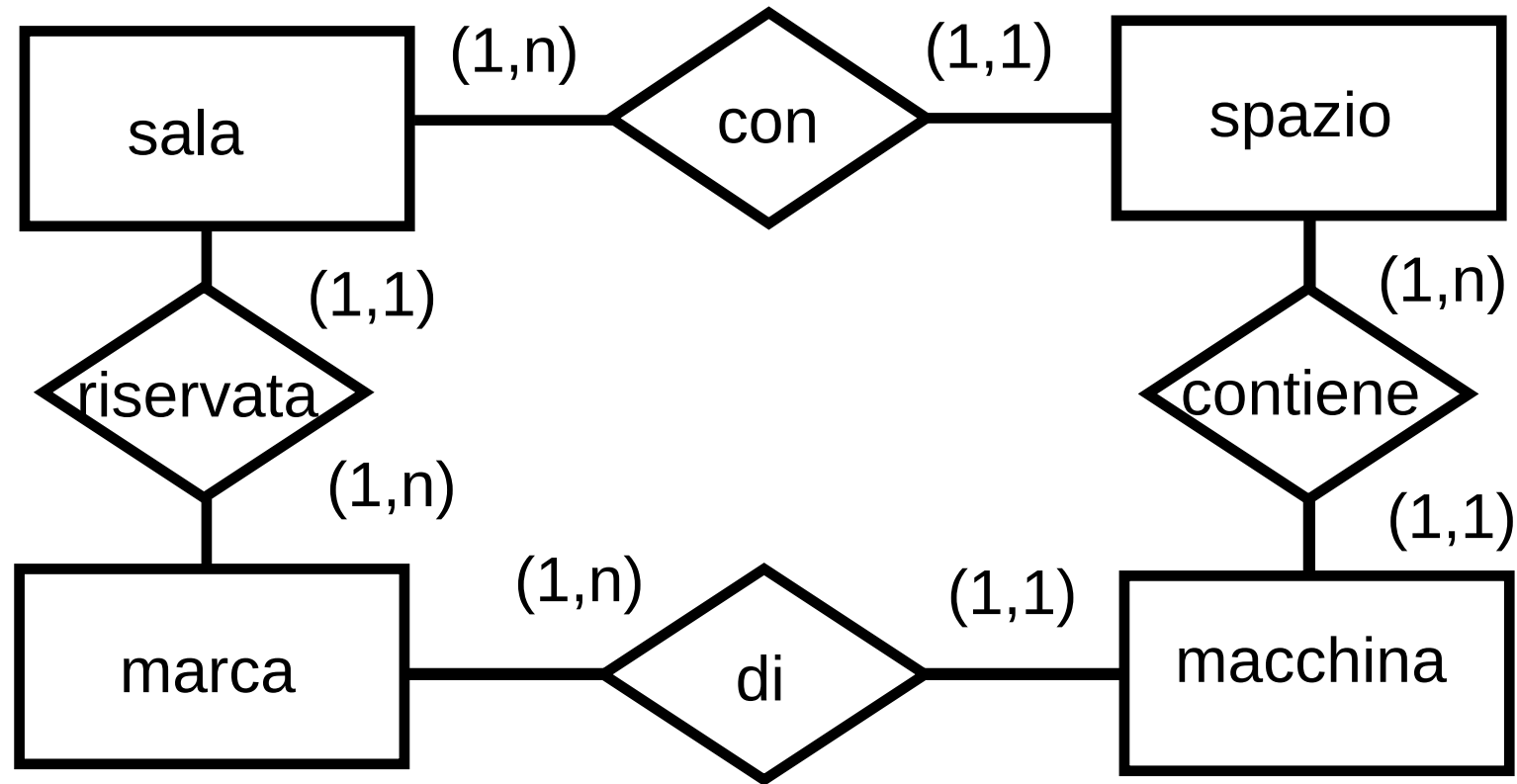
Ridondanze negli schemi

- Vanno individuate e documentate, ma non necessariamente eliminate
- Esempio: SALDO nella tabella CONTO CORRENTE, ricavabile a partire dal SALDO TRIMESTRALE CONSOLIDATO e dalla sommatoria di tutti gli AMMONTARI delle successive transazioni

Un esempio di possibile ridondanza

- Una società gestisce delle **sale** di esposizione
- le sale di esposizione sono **riservate** a **marche** di macchine
- le sale **comprendono** **spazi** di esposizione
- gli spazi **contengono** **macchine**
- le macchine **appartengono** ad una certa marca

Ridondanze negli schemi

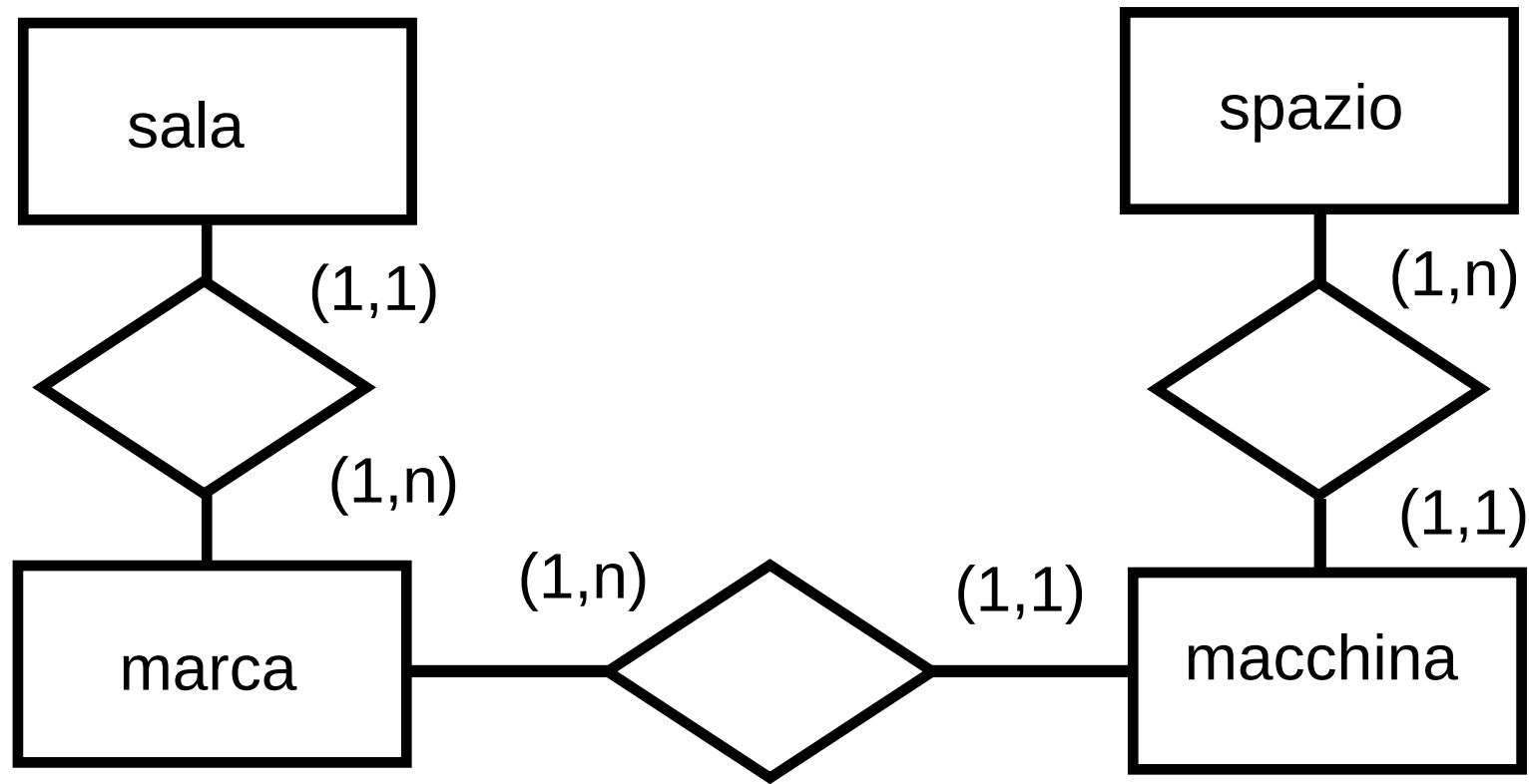


potrebbe esserci una ridondanza !

Ridondanze negli schemi

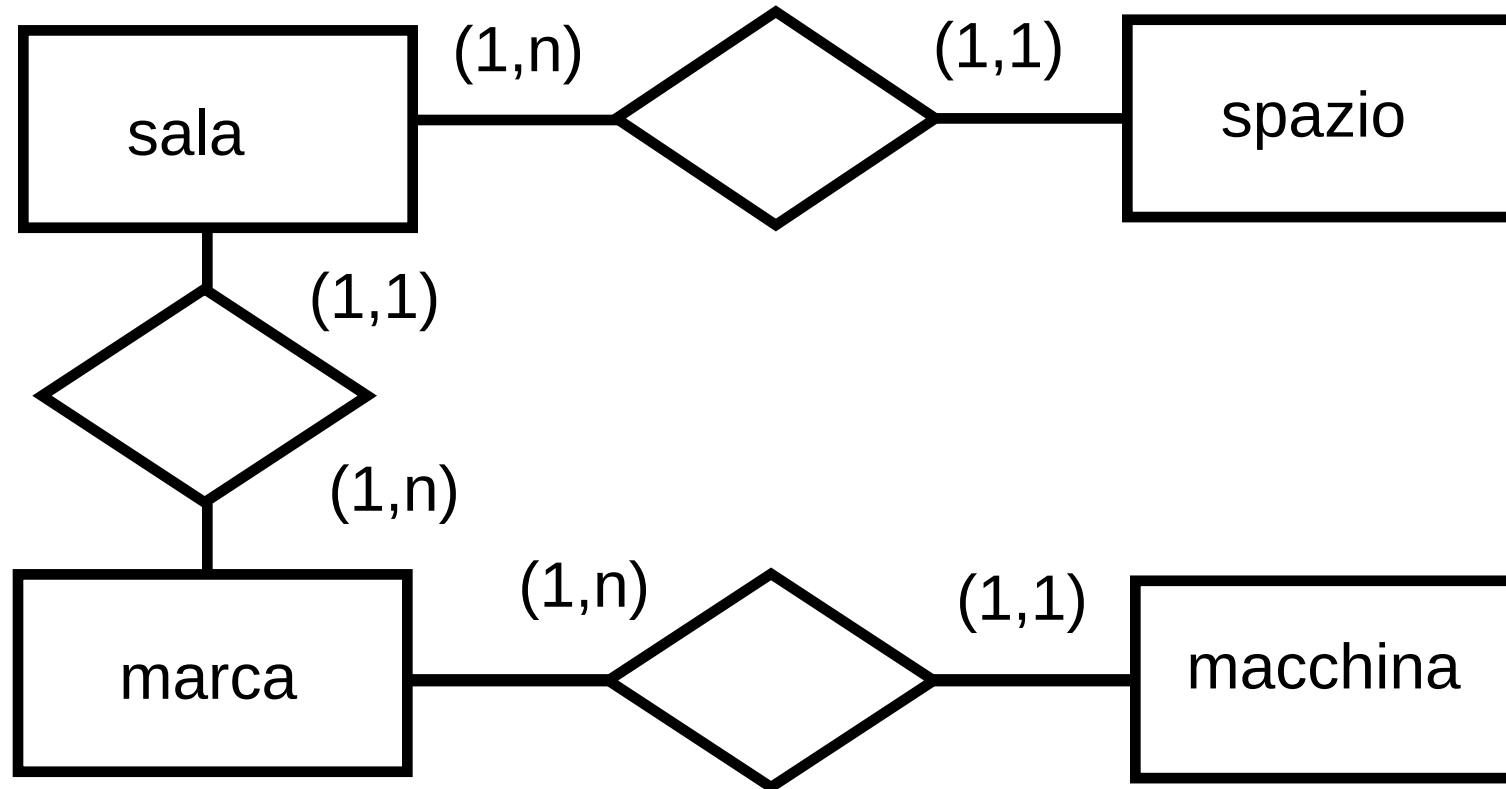
- Il ciclo è **ridondante** se la sistemazione delle macchine negli spazi viene effettuata nel rispetto del vincolo che **una sala sia assegnata per intero ad una sola marca**
- Come eliminiamo il ciclo?
 - Proviamo ad **eliminare** le 4 associazioni a turno e verificare il rispetto delle specifiche

Eliminazione di: comprendere



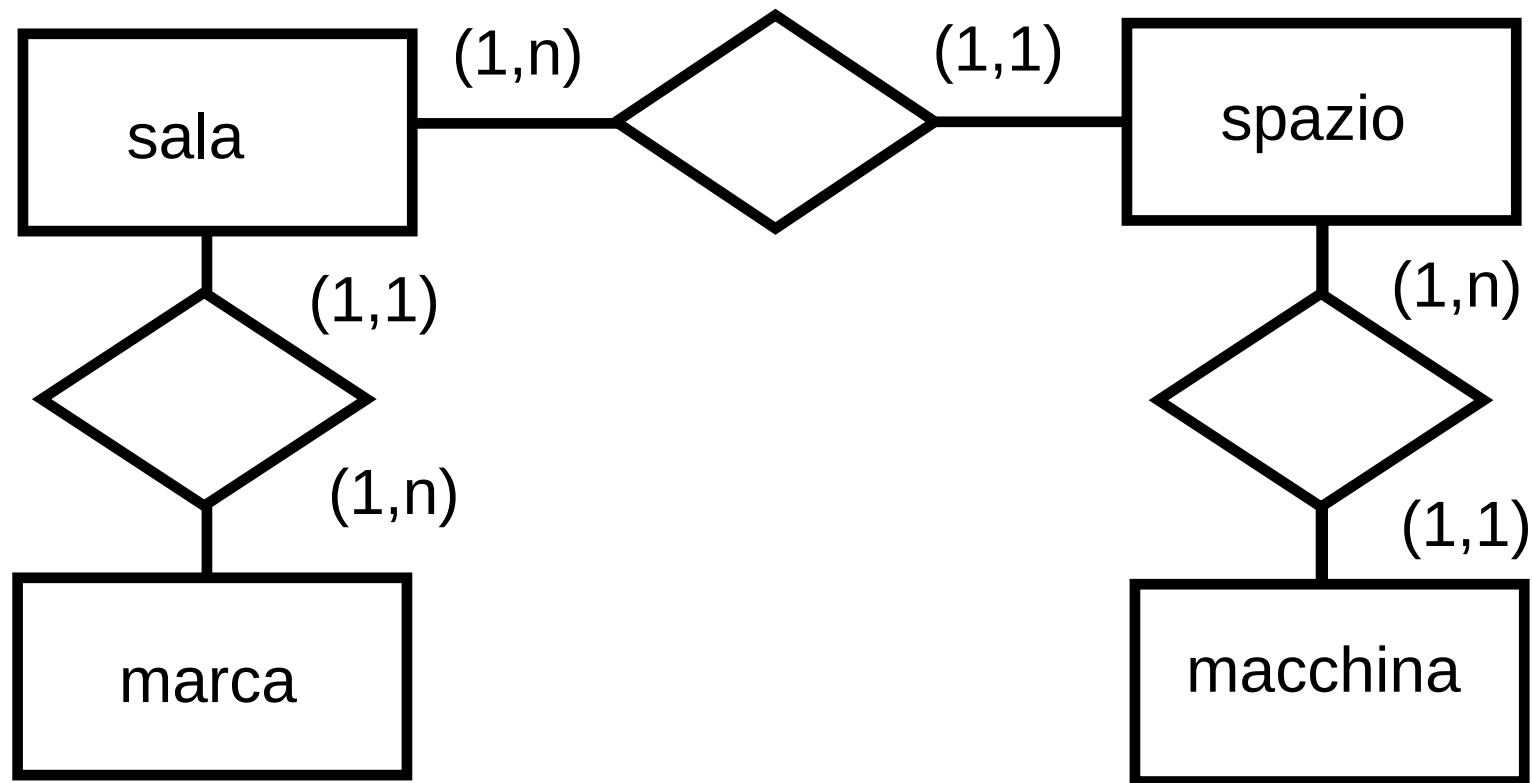
Dato uno spazio non è possibile risalire
alla sala che lo comprende

Eliminazione di: contenere



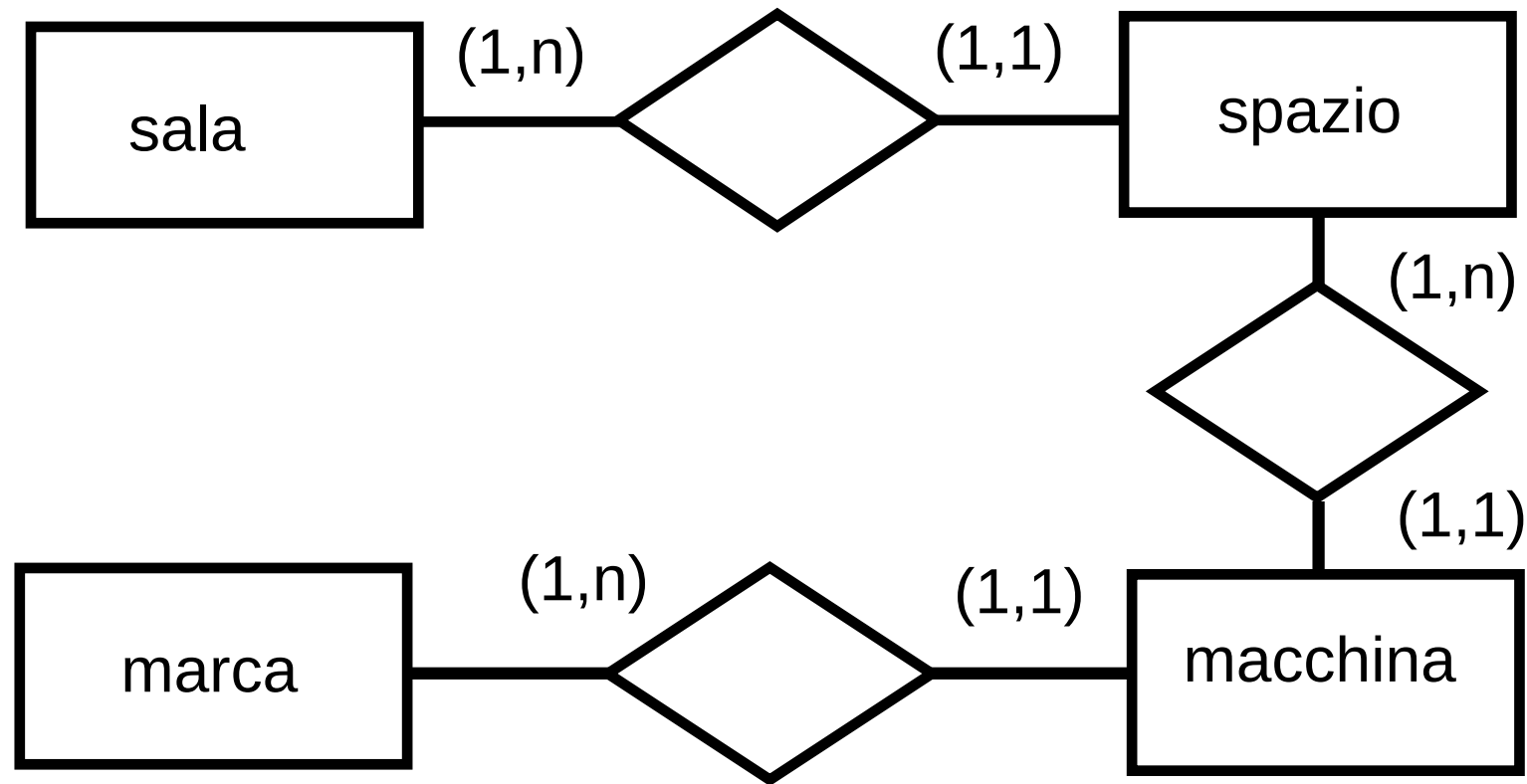
Data una macchina non è possibile risalire
allo spazio che la contiene

Eliminazione di: appartenere



Non viene persa informazione

Eliminazione di: riservare



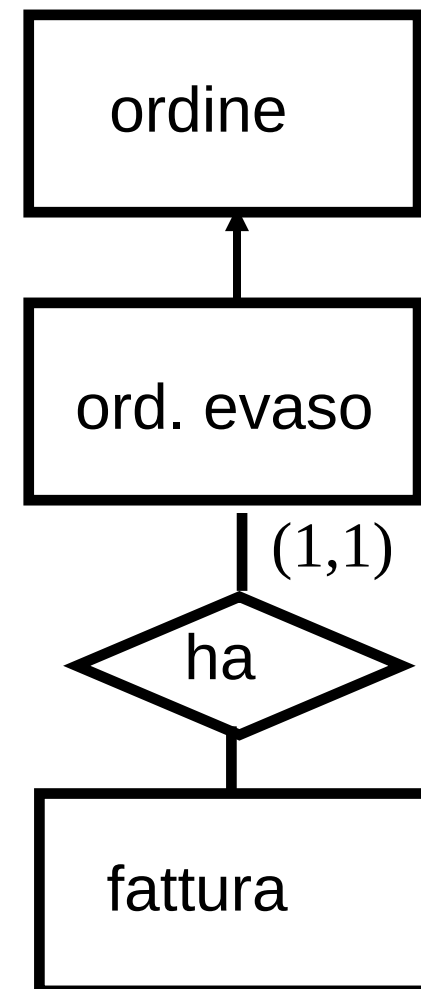
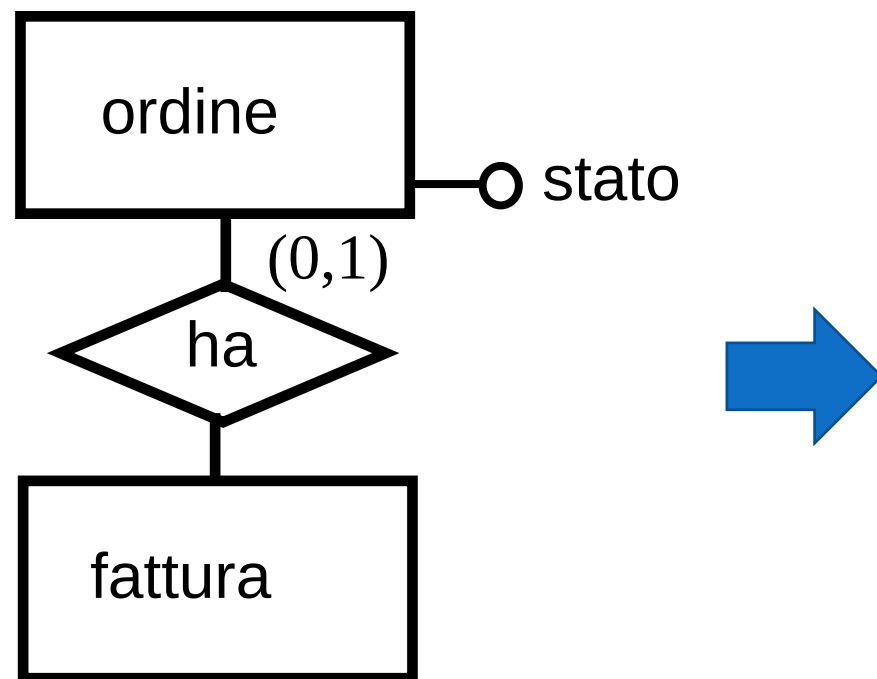
Non viene persa informazione (ma è meno ovvio)

Discussione

- Piuttosto irragionevole eliminare informazione perché poi è necessario ricorrere a join complessi
- E in ogni caso: occorre conoscere bene il significato delle associazioni!

Auto-esplicitività

Fare in modo che lo schema rappresenti esplicitamente il massimo di conoscenza sulla realtà



A questo punto....

- Il progetto è stato condotto (o documentato) top-down
- Il progetto risponde ai requisiti di qualità

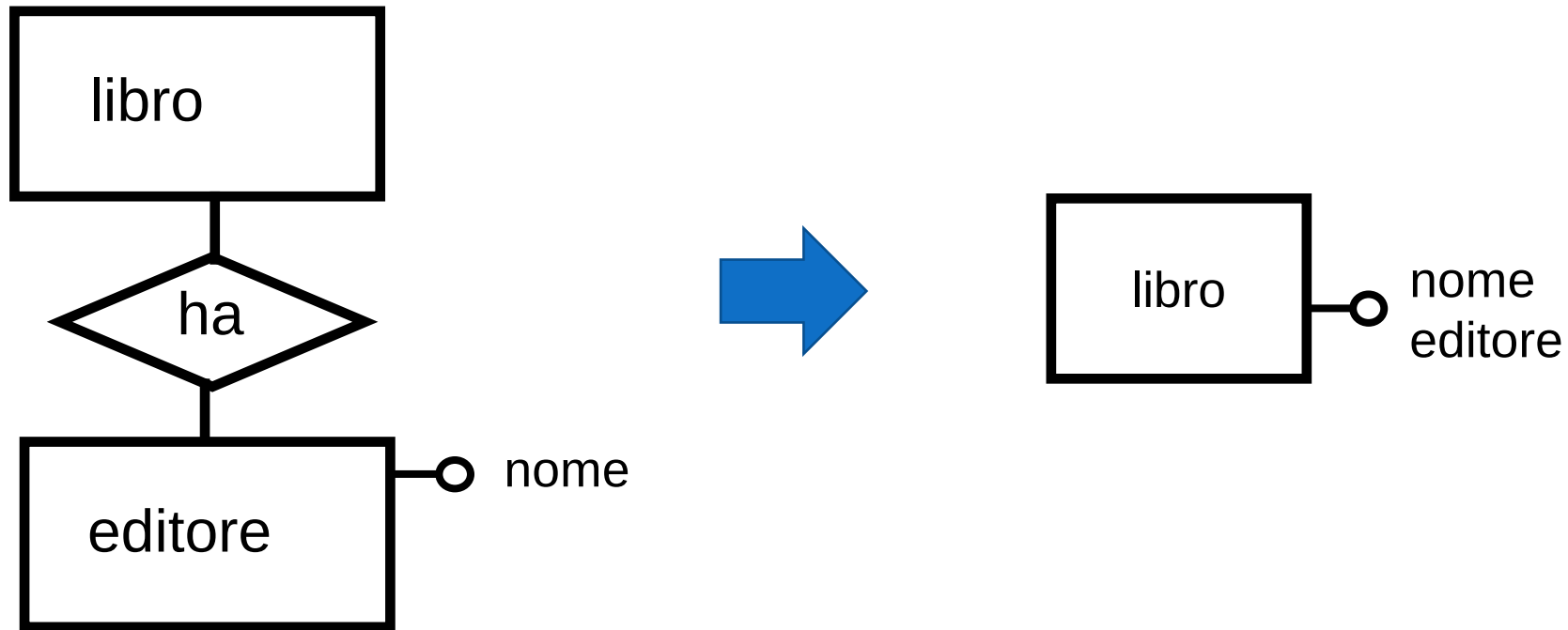
Ultimo passo: post-processing

Post-processing

Verificare che:

- Tutte le entità abbiano un identificatore
- Tutte le associazioni abbiano cardinalità ben definite
- Le entità siano significative (consentano di rappresentare più di una attributo o siano collegate ad altre entità tramite associazioni)
- Le generalizzazioni siano utili (consentano di ereditare proprietà)

Esempio di entità inutile



Esempio di gerarchia inutile

