

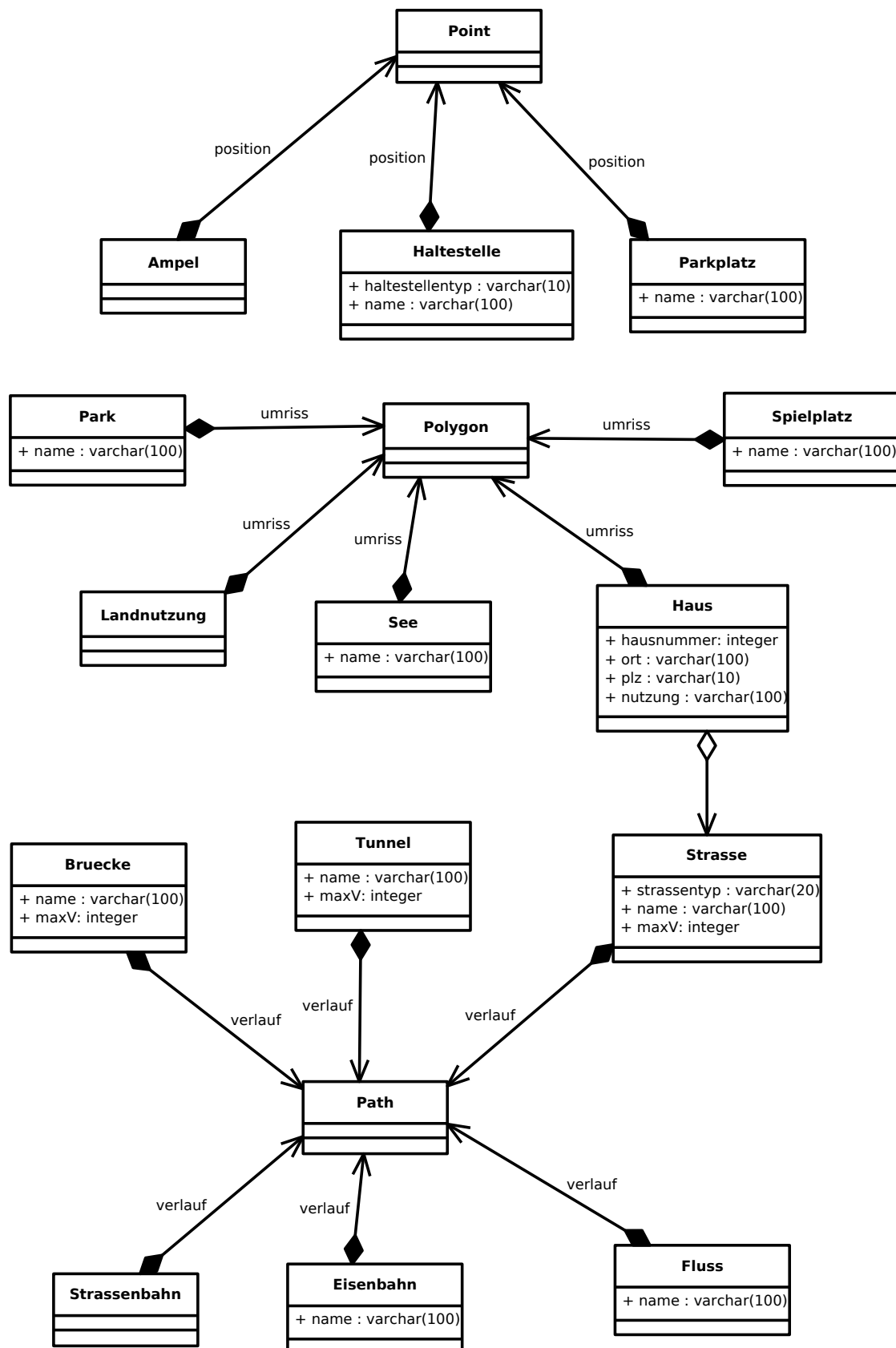
Praktikum „Integritätsbedingungen“, Phase 1

Henning Basold

Igor Zerr

26. April 2011

1 Modell



2 Relationsschemata

```
CREATE TABLE Ampel (position Point);
CREATE TABLE Haltestelle (position Point, haltestellentyp varchar(10),
    name varchar(100));
CREATE TABLE Parkplatz (position Point, name varchar(100));

CREATE TABLE Park (umriss Polygon, name varchar(100));
CREATE TABLE Spielplatz (umriss Polygon, name varchar(100));
CREATE TABLE Landnutzung (umriss Polygon);
CREATE TABLE See (umriss Polygon, name varchar(100));
CREATE TABLE Haus (umriss Polygon, hausnummer int, ort varchar(100),
    plz varchar(10), nutzung varchar(100), strassenId REFERENCES
    Strasse(id) );

CREATE TABLE Bruecke (verlauf Path, name varchar(100), maxV int);
CREATE TABLE Tunnel (verlauf Path, name varchar(100), maxV int);
CREATE TABLE Strasse (id int PRIMARY KEY, verlauf Path, strassentyp
    varchar(20), name varchar(100), maxV int);
CREATE TABLE Strassenbahn (verlauf Path);
CREATE TABLE Eisenbahn (verlauf Path, name varchar(100));
CREATE TABLE Fluss (verlauf Path, name varchar(100));
```

3 Assertions

1. (innere) Punkte disjunkt bei
 - (a) HOUSE_STREET_DISJOINT: Haus \longleftrightarrow Straße
 - (b) HOUSE_HOUSE_DISJOINT: Haus \longleftrightarrow Haus
 - (c) HOUSE_LAKE_DISJOINT: Haus \longleftrightarrow See
 - (d) STREET_LAKE_DISJOINT: Straße \longleftrightarrow See
 - (e) STREET_LANDUSE_DISJOINT: Straße \longleftrightarrow Landnutzung
 - (f) STREET_PLAYGROUND_DISJOINT: Straße \longleftrightarrow Spielplatz
2. NO_STANDALONE_STOP: An einer Haltestelle verläuft eine Straße, Straßenbahn- oder Eisenbahnlinie in höchstens Abstand r
3. WATER_CROSSING: Wenn eine Straße, Straßenbahn- oder Eisenbahnlinie und ein Fluss sich kreuzen, ist dort eine Brücke oder ein Tunnel über/unter dem Fluss
4. TRAFFIC_LIGHT_AT_STREET: Ampeln stehen an Straßen mit höchstens Abstand r
5. PARKING_REACHABLE: Zu einem Parkplatz führt eine Straße oder an diesem in höchstens Abstand r entlang

```
CREATE FUNCTION approx_circle(point, double precision)
    RETURNS path AS $$
    SELECT path(polygon(256, circle($1, $2)))
    $$ LANGUAGE SQL IMMUTABLE STRICT ;
```

```
CREATE ASSERTION HOUSE_STREET_DISJOINT CHECK (
    NOT EXISTS ( SELECT * FROM
        Haus AS h, Strasse AS s
```

```

        WHERE path(h.umriss) ?# s.verlauf
    )
);

```

```

CREATE ASSERTION HOUSE_HOUSE_DISJOINT CHECK (
    NOT EXISTS ( SELECT * FROM
        Haus AS h1, Haus AS h2
        WHERE h1.umriss && h2.umriss
    )
);

```

```

CREATE ASSERTION HOUSE_LAKE_DISJOINT CHECK (
    NOT EXISTS ( SELECT * FROM
        Haus AS h, See AS l
        WHERE h.umriss && l.umriss
    )
);

```

```

CREATE ASSERTION STREET_LAKE_DISJOINT CHECK (
    NOT EXISTS ( SELECT * FROM
        Strasse AS s, See AS l
        WHERE s.verlauf ?# path(l.umriss)
    )
);

```

```

CREATE ASSERTION STREET_LANDUSE_DISJOINT CHECK (
    NOT EXISTS ( SELECT * FROM
        Strasse AS s, Landnutzung AS l
        WHERE s.verlauf ?# path(l.umriss)
    )
);

```

```

CREATE ASSERTION STREET_PLAYGROUND_DISJOINT CHECK (
    NOT EXISTS ( SELECT * FROM
        Strasse AS s, Spielplatz AS p
        WHERE s.verlauf ?# path(p.umriss)
    )
);

```

```

CREATE ASSERTION NO_STANDALONE_STOP CHECK (
    NOT EXISTS ( SELECT * FROM
        Haltestelle AS b

```

```

WHERE NOT EXISTS ( SELECT * FROM
                    Strasse AS s
                    WHERE approx_circle(b.position, r) ?# s.verlauf
                )
            )
        );

```

```

CREATE ASSERTION WATER_CROSSING CHECK (
    NOT EXISTS ( SELECT * FROM
                Strasse AS s
                JOIN Fluss AS r ON s.verlauf ?# r.verlauf
                WHERE NOT EXISTS ( SELECT * FROM
                                    Bruecke AS b
                                    WHERE (b.verlauf && (s.verlauf # r.verlauf)
                                )
                )
    );

```

```

CREATE ASSERTION TRAFFIC_LIGHT_AT_STREET CHECK (
    NOT EXISTS ( SELECT * FROM
                Ampel AS t
                WHERE NOT EXISTS ( SELECT * FROM
                                    Strasse AS s
                                    WHERE approx_circle(t.position, r) ?# s.verlauf
                                )
                )
    );

```

```

CREATE ASSERTION PARKING_REACHABLE CHECK (
    NOT EXISTS ( SELECT * FROM
                Parkplatz AS p
                WHERE NOT EXISTS ( SELECT * FROM
                                    Strasse AS s
                                    WHERE approx_circle(p.position, r) ?# s.verlauf
                                )
                )
    );

```

3.1 Bemerkungen

Leider konnten wir die jeweiligen Radien noch nicht feststellen. Dazu müssen wir erst die Daten analysieren, um einen vernünftigen Wert zu erhalten.

Die Funktion „approx_circle“ stellt einen Kreis angenähert als Pfad dar. Dies ist nötig, da kein Schnitt zwischen Kreisen und Pfaden in Postgresql direkt möglich ist.