Axiomatic Minkowski Spacetime in Isabelle/HOL

Richard Schmoetten

Master of Science

Informatics

School of Informatics

University of Edinburgh

2020

Abstract

In an effort to establish verified foundations for relativistic physics, this MSc project furthers [34, 35] the development of Minkowski spacetime from a system of axioms devised by Schutz [42]. This enterprise occurs in the interactive theorem prover Isabelle/HOL, which facilitates trusted certification of proofs, and provides automated tools for assistance. We begin by reviewing the mechanisation of axioms and definitions of Palmer and Fleuriot [35], and our completion of their mechanised axiom system. We then present newly mechanised proofs for several theorems found in Schutz' monograph [42] concerning temporal order. We highlight differences between the prose of the original, and our more verbose – but precise – mechanisation. Several auxiliary results not present in Schutz [42] are presented throughout, with comments on why they are necessary or desirable.

Acknowledgements

I would like to thank Jacques Fleuriot and Jake Palmer for their help and encouragement. Their support with the details of Isabelle/HOL, their knowledge of relevant theories and approaches, and their copious feedback were invaluable. I was astonishingly lucky to have two supervisors of their calibre.

Table of Contents

Chapter 1

Introduction

The special theory of relativity (SR) [8] is a canonical part of modern physics. The theory as taught at universities today is founded in the formulation first established by Minkowski in 1908 [29], merging space and time into a four-dimensional spacetime. This way of thinking was received with considerable mistrust by many of Minkowski's contemporaries [9]: it shifts the focus from the material world (particles, collisions, forces) to imply that the geometric structure of the universe influences the physics within it. Minkowski spacetime becomes the central feature of SR. It is Einstein's eventual acceptance of the spacetime formalism that enables the development of general relativity (GR) [10].

The key to understanding SR, of interest both paedagogically and philosophically, is thus to reduce Minkowski spacetime to its essentials, and glean intuition from these building blocks, and their use. This is the aim of three decades of work by John Schutz [40, 41] (and many others [30, 12]), culminating in a categorical system of fifteen independent axioms [42]. Schutz constructs the geometry of spacetime and relativistic kinematics over the course of seven chapters, then demonstrates how his system is isomorphic to the standard model on $\mathbb{R}^4$ used by physicists, and finally proves independence of his system.

Our central aim is to provide a machine-checked formalisation of Schutz' system of axioms [42, chap. 2], as well as the theory of temporal order on paths [42, chap. 3]. This formalisation occurs in the proof assistant Isabelle using higher-order logic (HOL), and is a direct continuation of a prior MSc project [34, 35]. This includes a formalisation of the primitives, axioms, definitions of derived objects, and proofs of results. We have completed the mechanisation of the axioms. Analysing the notion of chains in particular, we have obtained several results related to them that are not

present in Schutz. Compared to Schutz' original text [42], we have discovered and corrected several flaws in the original theorems 10 and 11 and their proofs.

Chapter 2 of the present document will outline briefly the status of formalisation in physics, particularly SR. We give several examples of formal systems in geometry, especially where these have been implemented in Isabelle. Isabelle/HOL itself is introduced in slightly more depth in Section 2.3. We discuss our Isabelle formalisation of the axioms of Schutz [42] in Chapter 3, examining the notions of chains, path dependence, and bounds used in the axioms. The main derivations and results are presented in Chapter 4. Some more speculative discussion precedes the conclusion in Chapter 5.

Chapter 2

Background

2.1 Formalisation in Physics and Special Relativity

Formal foundations are a newly re-emerging trend in modern physics. While philosophical, mathematical, and empirical studies were inseparably entwined in antiquity, formal mathematics and physical science drifted apart in the eighteenth and nineteenth centuries [47].

The mathematical deduction employed for example in Ptolemy's *Harmonics* is taken to be almost divine. Thus he considers "arithmetic and geometry, as instruments of indisputable authority" [4, pp.507]. In contrast, the main physical theories of the twentieth century were developped as physics first, and retro-fitted with proper mathematical foundations later. An example particularly relevant to our project is that of SR, briefly outlined in Sec. 1. The comprehensive mathematical treatment given by Minkowski [29] was at first dismissed as unnecessarily complicated [9]. Early work on axiomatising SR (e.g. Robb [38]) went largely unnoticed by the physical research community.

But the search for a formal foundation to modern physics gained support in the second half of the twentieth century. Philosophical essays [47], the successes of the new mathematical quantum and relativity theories [39, 5], and increasing interest by the mathematical community, all contributed to works such as differential geometry and GR, and the Wightman axioms in particle physics [46].

2.2 Axiomatic Geometries

Geometry is arguably the oldest discipline to have seen successful axiomatisation in the form of Euclid's *Elements* [19]. Over two millenia later, Hilbert's *Grundlagen der Geometrie*[20] built on Euclid to propose a new, self-contained system of axioms using modern logical concepts such as undefined notions (in contrast to Euclid's primitive definitions). Many alternative Euclidean systems have been postulated and examined. Schutz acknowledges clear parallels between several theorems this thesis is concerned with, and theorems of Veblen [50], whose axioms for Euclidean geometry replace Hilbert's primitives to use only points and a single relation. Tarski's system of elementary Euclidean geometry [49] is influential too: points as well as two undefined relations are his only primitive notions. His axioms can be formulated in primitive notions only, using first-order logic (with identity and using an axiom schema). Schutz [42] similarly strives for simplicity, though his continuity axiom is second-order, and while a line-like primitive exists, only a single undefined relation is required.

Several axiom systems have been proposed for Minkowski spacetime. An early approach is that of Robb [38], continued by Mundy [30, 31]. A first-order alternative to Schutz is given by Goldblatt [12, 13], who relies on a relation of orthogonality in addition to the betweenness Schutz employs. Systems formulated by Szekeres [48] and Walker [51] influence Schutz' work. A recent extension of Tarski's Euclidean ideas to Goldblatt's approach to Minkowski spacetime is given by Cocco and Babic [6]. They extend their first-order system with a second-order continuity axiom in order to show the usual four-dimensional Minkowski spacetime is a model. A flexible first-order system of axioms that goes some way towards GR is given by Andréka et al. [3, 2].

2.3 Isabelle/HOL

Computer-based theorem proving, verification, and exploration is the dominant area of automated reasoning today. A breakthrough development for the field was Scott's work on LCF [43], a typed version of the λ-calculus, and the subsequent construction of an interactive theorem prover of the same acronym by Gordon, Milner and Wadsworth [14]. Isabelle is a generic proof assistant which continues the LCF-style of automated reasoning [52, 37]. Its generic meta-logic (the type system responsible for validity checking) supports multiple instances of object logic: we will be using higher order logic (HOL), but instances for e.g. first-order logic (FOL) and ZFC set theory exist.

Several axiomatic approaches to geometry have been (at least partially) formalised in Isabelle/HOL, including Hilbert's *Grundlagen* by Meikle, Scott and Fleuriot [28, 45, 44] and Tarski's geometry by Makarios [26]. Geometric formalisations also exist in other proof assistants, such as Coq [32, 25] or Mizar [16].

The Archive of Formal Proofs[1] (AFP) is an online repository for several hundred Isabelle/HOL theories. The contents of the AFP can be downloaded and used, and it provides a prime vector for dissemination of formal work done in Isabelle.

2.3.1 Automation and Readability

A proof is a repeatable experiment in persuasion. (Jim Horning)

Considering the above quote, the advantage of computer assistance in logical and mathematical proof is clear. Using Isabelle (for example), we can write a proof of any theorem, and provided our readers are convinced of the soundness [2] of Isabelle's logical kernel and reduction techniques, they can take the theorem as fact without manually verifying the proof.[3] A famous and well-popularised example is the *Flyspeck* project [17], a twelve-year formalisation effort resulting in a formal proof of the Kepler conjecture, accepted to a mathematical journal in 2017. Despite four years of work, referees for the pre-*Flyspeck* proof submitted in 1998 had been unable to fully verify the proof.

However, it is often instructive to read through proofs. One may identify methods to be used in similar problems, or generalised to unrelated areas of inquiry; intuition is built for the behaviour of the mathematical entities manipulated throughout the proof; and ultimately, understanding of a subject is often linked to what one is capable of doing with it, which must be learnt. Readability is therefore important, particularly for proofs as verbose as those often found in mechanisations. Isabelle provides us with the language Isar (**I**ntelligible **s**emi-**a**utomatic **r**easoning), which can be used for proofs that are both human-readable and supported by automatic solvers. Isar proofs merge the assumptions-forward style of reasoning common in mathematical texts and natural for human readers to follow, and the result-backwards style often useful in exploring possible avenues for a proof to be completed.

[1] website

[2] Consistency of automated provers is its own research field [23]

[3] Computer hardware and software programmers both remain fallible, as do the author and the reader of theorem statements.

Several tools for proof discovery come with the Isabelle distribution. In particular, the umbrella tool *sledgehammer* [36] automatically chooses a range of (several hundred) facts to pass to different first-order solvers (both resolution and smt provers), and, if successful, provides a reconstruction of the automatic proof using only needed results. In practice, automatic discovery is useful, but often struggles to justify steps that seem obvious to the reader, or returns proofs relying on highly unexpected facts. This may be due to the complexity of some of our definitions, or difficulty in reductions to first-order logic.

2.3.2 Working in Isabelle: Proofs and Induction

Working in Isabelle/HOL (and Isar) is a mix of meta- and object-level reasoning. This is best looked at through an example: we use one of our auxiliary lemmas, `finite_path_has_ends`. We are only interested in the formalism and method for now, refer to Sec. 4.1 for discussion.

```
lemma finite_path_has_ends:
  assumes path_X: "X∈𝒫" and min_n: "n≥0"
    shows "∀Q⊆X. finite Q ∧ card Q = n+3
      ⟶ (∃a∈Q. ∃b∈Q. a≠b ∧ (∀c∈Q. (a≠c ∧ b≠c) ⟶ [[a c b]]))"
```

Meta-logic in Isabelle can be part of the inner (e.g. ⟦...⟧ for assumptions and ⟹ for meta-implication) or outer syntax (e.g. **assumes**, **shows**). We announce the statement of a fact requiring proof with keywords such as **theorem**, **lemma**. This is followed by a unique name, as well as the fact statement in inner syntax (`P∈𝒫` ⟹ `infinite P`) or in the more legible Isar style as above. In this case, we still have a large conclusion in object logic.

We start a proof with the keyword **proof**. We can supplement **proof** with an initial method to use (e.g. a three-way case split `rule disjE3` or the general method `safe`, which splits and rewrites goals; or `induct` as below). A successful proof ends with **qed**. Two other keywords can terminate a proof: **sorry** and **oops**. Both signify a proof that is not complete, or cannot be done, but while **oops** means that Isabelle will refuse to allow use of the unproven fact, **sorry** allows an unproven statement to be used legitimate proofs of other propositions. Thus **sorry** can be quite dangerous (see Sec. 2.3.3 for an alternative).

The example lemma above is proved by induction. Induction is a recurring scheme in this project, and comes with particularities in Isabelle. The method `induct` takes

an induction parameter[4] and splits the proof into two goals. Isabelle provides shorthand notation for the usual first lines of both split cases. The *base case* (`case 0`) sets the induction variable to 0 throughout the conclusion for a goal. The induction case (`case (Suc n)`) fixes n, assumes the lemma's conclusion for n, and sets the goal to the conclusion for $n+1$ (i.e. `Suc n`). This assumption for n is called the induction hypothesis (IH).

The long conclusion is necessary for us to use the IH in the induction case. Had we moved more of the conditions into the premises, the variables these conditions applied to would be fixed, rather than quantified over, and the induction hypothesis unusable for us. We begin proving the induction case by stripping off the universal quantifier, and assuming all the conditions (Isabelle's `safe` method saves some work here).

```
case (Suc n)
  show "∀Q⊆X. finite Q ∧ card Q = Suc n + 3
    ⟶ (∃a∈Q. ∃b∈Q. a ≠ b ∧ (∀c∈Q. a ≠ c ∧ b ≠ c ⟶ [[a c b]]))"
  proof (safe)
    fix Q
    assume events_Q: "Q⊆X"
    assume fin_Q: "finite Q"
    assume "card Q = Suc n+3"
    hence card_Q: "card Q = Suc n+3" by simp
    show "∃a∈Q. ∃b∈Q. a ≠ b ∧ (∀c∈Q. a ≠ c ∧ b ≠ c ⟶ [[a c b]])"
```

In the second-last line, `hence` declares a fact, followed by name and fact statement, and proved by an invocation of the method `simp`. The proof `by simp` could be (for harder facts) replaced by calls to automatic theorem provers (ATPs) like `metis` or `blast`, or an entire subproof `proof ... qed`.

Notice the IH is quantified over: it applies to arbitrary sets Q meeting certain conditions. If we had, for example, induced directly on the cardinality of Q, the IH would apply only to this (already named) set, provided it also satisfies the object-logic conditions of the IH. Parts of the script below are omitted (`<proof>` and `...`), see Sec. 4.

```
  case (Suc n) ...
    show "∃a∈Q. ∃b∈Q. a ≠ b ∧ (∀c∈Q. a ≠ c ∧ b ≠ c ⟶ [[a c b]])"
    proof -
      obtain x where "x∈Q" <proof>
      obtain P where def_P: "P=Q-{x}" by simp
```

The central part to any induction is to obtain an object (here: `P`) that the IH applies to, and that is related to the one of interest (here: `Q`).

```
      have ind: "card P = card Q - 1" <proof>
      have "∃a∈P. ∃b∈P. a ≠ b ∧ (∀c∈P. a ≠ c ∧ b ≠ c ⟶ [[a c b]])"
        using card_P Suc ... by simp
```

[4] for us, it is always of type `nat`

In this last step, it is `Suc` that contains the IH. Isabelle automatically names it when we declare **`case`** `(Suc n)`. The base case has similar unwrapping to the induction case presented here. We leave the actual proof context for Sec. 4.1.

2.3.3 Working in Isabelle: Locales

One very useful feature, particularly for sizeable axiom systems such as ours, is Isabelle's **`locale`**. One can think of a locale as a parameterised context: it names an "arbitrary but fixed" parameter, and assumes some initial properties. In our case, these are undefined notions and axioms. Since the formulation of axioms often changes as proofs are attempted, and axioms found wanting (e.g. axiom I6, see Sec. 4.6), we try to limit the amount of logic that is affected and possibly invalidated by such a change. Containing small groups of axioms in their own separate locales makes explicit the scope of their influence. A similar purpose is served by our locale `MinkowskiDense`, which contains an assumption (in this case a proxy for an unproven result) that we do not want to spill outside the locale, see Sec. 4.6. This is a safer alternative to **`sorry`**.

Locales have additional practical benefits: they are augmented by each theorem proven inside them, they can extend other locales, and they can be interpreted. The latter is like an explicit example to an abstract algebraic concept (e.g. $SO(3)$, 3D-rotations, form a concrete instance of a group).[5] This means that if we want to find a model of our system, we can do so in steps: showing some interpretation $\mathcal{M}$ satisfies our locale `MinkowskiChain` gives us immediate acces to that locale's theorems (e.g. `collinearity2`), and those of any locales it extends. These theorems may then be used to prove $\mathcal{M}$ satisfies the additional requirements of a locale extending `MinkowskiChain`.

Since model proofs are out of scope for us, locales serve mostly an organisational purpose. See Chap. 3.

[5] Or, for object-oriented programmers, like an instantiation of an abstract class.

Chapter 3

Schutz' Axioms in Isabelle

Schutz proves several properties of his axiomatic system in his monograph [42]: consistency (relative to the real numbers), categoricalness, and independence. He insists upon independence: the search for it made him rework his earlier attempts, and he considers the resulting system to be cleaner and more intuitive. Independence means that no axiom can be derived from any combination of the other axioms.

Consistency means that one cannot derive both a statement and its negation from our axioms, [1] i.e. they are not just a convoluted way of assuming `False`. Categoricalness is a property that is in some way complementary to independence: it states that essentially only one model exists (up to isomorphism). A model is defined as an algebraic structure (a set, together with a collection of finitary operations and relations) that satisfies the axioms in question. In our case, it provides concrete sets for events, paths, and a concrete definition of the betweenness relation (see below). It might be thought of as an instantiation of the abstract undefined objects and relation.

Some of the axioms as we encode them in Isabelle initially are subtly different from Schutz' statements. We give all of the original axioms and definitions in App. A for easy reference. The consequences of our changes are discussed later.

3.1 Primitives and Simple Axioms

Schutz lays out his axioms in two main groups: order and incidence. The former relate betweenness to events and paths, and establish a kind of plane geometry with axiom O6. The axioms of incidence deal with the relationships between events and paths, and also contain statements regarding unreachable subsets, which make a Euclidean/-

[1] for a discussion and pointers regarding alternative notions of consistency, see this SEP article

Galilean model impossible. In contrast to Schutz, we present axioms according to their specificity to Minkowski spacetime. In particular, our main comparison is with Hilbert's *Grundlagen der Geometrie* [20], which introduced the separation of incidence and order axioms.

Formalising any system of axioms in Isabelle often requires amendments. In our case, these are due to Isabelle's functions being total on types, not sets, as well as a few design choices and/or simplifications that are handy in interactive proofs. Some of these choices are explained here even though they were made in a prior project: a lot of the formalisations in this section are due at least in part to Palmer [34].

Since several definitions of derived objects are required for stating some axioms, we construct our system as a hierarchy of locales (Sec. 2.3.3), defining objects in the locale they make most sense in, and often just before they are needed. The first axioms, introduced in the locale `MinkowskiPrimitive` together with the primitive notions of events and paths (which use the keyword **`fixes`**), are similar to examples found in many other geometric axiom systems, notably Hilbert [20]. Schutz numbers them I1, I2, I3. As a comparison to our locale below, Schutz' I3 reads "For any two distinct events, there is at most one path which contains both of them" [42, p. 13]. Contrast this with the many premises of `eq_paths`, and its avoidance of the negative existential statement in "there is at most one". We require one axiom Schutz does not have: `in_path_event`, which excludes the possibility of non-event objects of the appropriate type being in a path (see below).

```
locale MinkowskiPrimitive =
  fixes 𝓔 :: "'a set"
    and 𝓟 :: "('a set) set"
  assumes in_path_event [simp]: "⟦Q ∈ 𝓟; a ∈ Q⟧ ⟹ a ∈ 𝓔"
      (* I1 *)
      and nonempty_events [simp]: "𝓔 ≠ {}"
      (* I2 *)
      and events_paths:
        "⟦a ∈ 𝓔; b ∈ 𝓔; a ≠ b⟧
        ⟹ ∃R∈𝓟. ∃S∈𝓟. a ∈ R ∧ b ∈ S ∧ R ∩ S ≠ {}"
      (* I3 *)
      and eq_paths [intro]:
        "⟦P ∈ 𝓟; Q ∈ 𝓟; a ∈ P; b ∈ P; a ∈ Q; b ∈ Q; a≠b⟧ ⟹ P = Q"
```

Nothing defines $\mathcal{E}$ apart from the type of its elements, yet we do not assume that $\mathcal{E}$ is the universal set of type `'a`. We hope this may lead to easier model instantiations in the future: for example, it allows building a model from a subset of natural numbers without defining an extra datatype. In Isabelle's simple typesystem (which has no subtypes), the alternative might be painful. The set of paths $\mathcal{P}$ is always en-

visaged as a strict subset of the powerset of $\mathcal{E}$ – otherwise the unreachable axioms introduced later (see Sec. 3.3) lose all relevance. The axiom `in_path_event` guarantees $\mathcal{P}$ is in the powerset of $\mathcal{E}$, not the universal set. Our final undefined notion, the ternary relation of betweenness, is defined on events. It is introduced in the second locale-layer, `MinkowskiBetweenness`, which also contains the first five axioms of order (O1 - O5). We give three of them in Schutz' words for later comment before presenting the locale. Schutz denotes betweenness as $[\_\ \_\ \_]$, but since that notation is used for lists in Isabelle, we define it to be `[[_ _ _]]` below.

Axiom O3
For events $a, b, c \in \mathcal{E}$,

$$[a\ b\ c] \longrightarrow a, b, c \text{ are distinct.}$$

Axiom O4
For distinct events $a, b, c, d \in \mathcal{E}$,

$$[a\ b\ c] \text{ and } [b\ c\ d] \longrightarrow [a\ b\ d]\ .$$

Axiom O5
For any path $Q \in \mathcal{P}$ and any three distinct events $a, b, c \in Q$,

$$[a\ b\ c] \text{ or } [b\ c\ a] \text{ or } [c\ a\ b] \text{ or } [c\ b\ a] \text{ or } [a\ c\ b] \text{ or } [b\ a\ c]\ .$$

(Schutz [42, p. 10])

```
locale MinkowskiBetweenness = MinkowskiPrimitive +
  fixes betw :: "'a ⇒ 'a ⇒ 'a ⇒ bool" ("[[_ _ _]]")
      (* O1 *)
  assumes abc_ex_path: "[[a b c]] ⟹ ∃Q∈𝒫. a ∈ Q ∧ b ∈ Q ∧ c ∈ Q"
      (* O2 *)
      and abc_sym: "[[a b c]] ⟹ [[c b a]]"
      (* O3 *)
      and abc_ac_neq: "[[a b c]] ⟹ a ≠ c"
      (* O4 *)
      and abc_bcd_abd: "⟦[[a b c]]; [[b c d]]⟧ ⟹ [[a b d]]"
      (* O5 *)
      and some_betw:
        "⟦Q ∈ 𝒫; a ∈ Q; b ∈ Q; c ∈ Q; a ≠ b; a ≠ c; b ≠ c⟧
              ⟹ [[a b c]] ∨ [[b c a]] ∨ [[c a b]]"
```

Three of them have mild changes compared to Schutz: O3 and O5 are slightly weaker (having weaker conclusions) as the original statements can be proven in the same locale the axiom is assumed in. In O4 we drop the condition that $a \neq d$. This is so we can have proofs by contradiction that involve obtaining a relation of the form $[a\ b\ a]$. These axioms are left as in Palmer [34].

The final axiom of order is used primarily in proofs we inherited from Palmer [34]. We give it here for completeness . Schutz gives this axiom as an alternative of the axiom of Pasch common in geometric systems.

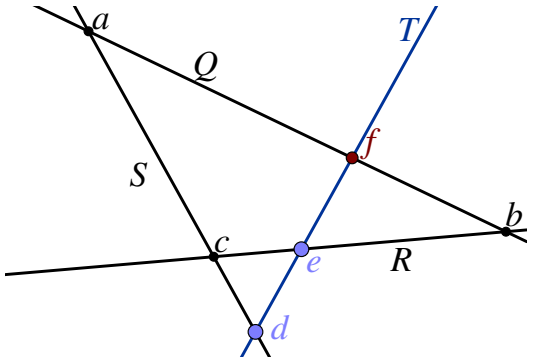

Figure 3.1: Intuitive visualisation of axiom O6. A path T that meets S external to the triangle QRS (in d) and meets R internally (in e), must meet the third side of the triangle internally (in f).

```
locale MinkowskiChain = MinkowskiBetweenness +
  assumes O6:
    "⟦Q ∈ 𝒫; R ∈ 𝒫; S ∈ 𝒫; T ∈ 𝒫; Q ≠ R; Q ≠ S; R ≠ S;
      a ∈ Q∩R ∧ b ∈ Q∩S ∧ c ∈ R∩S;
      ∃d∈S. [[b c d]] ∧ (∃e∈R. d ∈ T ∧ e ∈ T ∧ [[c e a]])⟧
    ⟹ ∃f∈T∩Q. ∃X. [[a..f..b]X]"
```

Although the statement is technical, the intention of O6 (or Pasch's axiom) is simple. Using some intuition from Euclidean geometry, a rough translation is: if three paths meet in a triangle, then a fourth path which intersects one side of the triangle externally, and another internally, must meet the third side internally as well (see Fig. 3.1). Such an intuitive understanding can be justified by noting that similar axioms occur e.g. in Hilbert's *Grundlagen* [20]; it is not O6 that makes our system non-Euclidean.

3.2 Chains of Events

Before giving the more interesting axioms of Schutz' system, we define chains of events. The formalisation of chains in Palmer [34] is different from Schutz in several ways. While Palmer's formal definition is (provably) equivalent to Schutz' prose, this leads to problems, for example in theorem 10 (Sec. 4.4). Instead of replacing Palmer's definition, we define two additional kinds of chains, and prove several results on equivalence. All of our chains differ from Schutz in that they use sets instead of his sequences (cf Sec. 4.3), and that while he assumes chains to lie on paths, we prove this as a theorem (`chain_on_path`). All of them use the notion of a ternary ordering defined through a total indexing function $\mathbb{N} \to \mathcal{E}$.

The end goal is to have a set of different approaches to proofs involving chains, and theorems to allow conversion between defferent kinds of chains. This would give us flexibility in stating axioms and results, while giving assurance that we do not stray from Schutz' intention too far.

3.2.1 Prose to Isabelle

Schutz' definition is slightly informal, and defining and working with chains in Isabelle/HOL was a major part of Palmer's work [34] as well as the present project. Quoting from the base text, chains are defined as follows.

> *A sequence of events* $Q_0, Q_1, Q_2, \ldots$ *(of a path* Q*) is called a chain if:*
>
> *(i) it has two distinct events, or*
>
> *(ii) it has more than two distinct events and for all* $i \geq 2$*,*
> $$[Q_{i-2}\ Q_{i-1}\ Q_i]\ .$$
>
> (Schutz [42, p. 11])

This is hard to represent in Isabelle because of the notion of a sequence as an indexed set. The informal naming convention of using a label Q_i for an event encodes two pieces of information: that the event lies on path Q, and that several betweenness relations hold with other events indexed by adjacent natural numbers. Palmer [35] uses this insight (after Scott [44, p. 110]) to explicitly give a function $I \to Q$ (with $I \subseteq \mathbb{N}$) that is order-preserving, and use it to define chains.

```
definition ordering ::
  "(nat ⇒ 'a) ⇒ ('a ⇒ 'a ⇒ 'a ⇒ bool) ⇒ 'a set ⇒ bool"
  where "ordering f ord X
    ≡ (∀n. (finite X ⟶ n < card X) ⟶ f n ∈ X)
        ∧ (∀x∈X. (∃n. (finite X ⟶ n < card X) ∧ f n = x))
        ∧ (∀n n' n''. (finite X ⟶ n'' < card X) ∧ n<n' ∧ n'<n''
                ⟶ ord (f n) (f n') (f n''))"
definition chain_with ::
  "'a ⇒ 'a ⇒ 'a ⇒ 'a set ⇒ bool" ("[[.. _ .. _ .. _ ..]_]")
  where "chain_with x y z X
    ≡ [[x y z]] ∧ x ∈ X ∧ y ∈ X ∧ z ∈ X ∧ (∃f. ordering f betw X)"
```

Notice that Palmer's chains are strictly stronger than Schutz', in that they assume preservation of a total order (using the condition $n < n' < n''$), while Schutz must *prove* this holds (in theorem 2, cf. Sec. 4.3) given his definition using only a local order $(n-2, n-1, n)$. This is as in Scott's treatment of ordered geometry [44, p. 110], but causes problems for us later (Sec. 4.4). We call this kind of chain *total*, like its ordering, as opposed to a *local* chain (as in Schutz). We will focus largely on finite chains.

```
definition finite_chain_with3 ::
  "'a ⇒ 'a ⇒ 'a ⇒ 'a set ⇒ bool" ("[[_ .. _ .. _]_]")
  where "finite_chain_with3 x y z X
    ≡ [[..x..y..z..]X] ∧ ¬(∃w∈X. [[w x y]] ∨ [[y z w]])"
```

Schutz' definition of a finite chain only assumes finiteness of the sequence of events in the chain, and does not involve any explicit betweenness condition (`¬∃w∈X. [[w x y ]] ∨ [[y z w]]`) as found in `finite_chain_with3`. To Schutz, the final line above would therefore be a result to be proven.

3.2.2 Local and Index-Chains

As an alternative, closer to Schutz, we define total index-chains, which move the condition of finiteness to the set of events, and keep track of the function ordering them.

```
definition short_ch :: "'a set ⇒ bool"
  where "short_ch X ≡
    ∃x∈X. ∃y∈X. path_ex x y ∧ ¬(∃z∈X. z≠x ∧ z≠y)"
    (* path_ex implies x≠y *)

definition long_ch_by_ord :: "(nat ⇒ 'a) ⇒ 'a set ⇒ bool"
  where "long_ch_by_ord f X ≡
    ∃x∈X. ∃y∈X. ∃z∈X. x≠y ∧ y≠z ∧ x≠z ∧ ordering f betw X"

definition ch_by_ord :: "(nat ⇒ 'a) ⇒ 'a set ⇒ bool"
  where "ch_by_ord f X ≡
    short_ch X ∨ long_ch_by_ord f X"

definition ch :: "'a set ⇒ bool"
  where "ch X ≡ ∃f. ch_by_ord f X"
```

Notice also that we split the definition here between chains of two events, and chains with at least three events, as Schutz does. This is one way in which our new definition is more precise and closer to Schutz', and allows us to clarify several of his prose statements (Sec. 4.4). Again, finite chains are the most important for our proofs.

```
definition fin_long_chain ::
  "(nat⇒'a) ⇒ 'a⇒'a⇒'a ⇒ 'a set ⇒ bool" ("[_[_ .. _ ..  _]_]")
  where "fin_long_chain f x y z Q ≡
    x≠y ∧ x≠z ∧ y≠z ∧ finite Q ∧ long_ch_by_ord f Q
    ∧ f 0 = x ∧ y∈Q ∧ f (card Q - 1) = z"
```

We point out the notation: a `fin_long_chain` is denoted `[f[a..b..c]X]` while the `finite_chain_with3` omits the explicit reference to the indexing function `f`. The final chain we define involves a local ordering, but is otherwise similar to total index-chains. Local index-chains are as close as we could get to Schutz' definition in Isabelle/HOL. We still use sets, not sequences; totality of the indexing function leads to technical conditions in `ordering`, `ordering2`; and we do not assume chains lie on paths.

```
definition ordering2 ::
  "(nat ⇒ 'a) ⇒ ('a ⇒ 'a ⇒ 'a ⇒ bool) ⇒ 'a set ⇒ bool"
  where "ordering2 f ord X
    ≡ ... (∀n n' n''.
      (finite X⟶n'' < card X) ∧ Suc n = n' ∧ Suc n' = n''
        ⟶ ord (f n) (f n') (f n''))"

definition long_ch_by_ord2 ::
  "(nat ⇒ 'a) ⇒ 'a set ⇒ bool"
  where "long_ch_by_ord2 f X
    ≡ ∃x∈X. ∃y∈X. ∃z∈X. x≠y ∧ y≠z ∧ x≠z ∧ ordering2 f betw X"
```

For completeness, we state several results about various chains below. The proofs are omitted. Together, these results allow for (restricted) changing between different kinds of chains. Expanding these, particularly removing the finiteness requirement from `equiv_chain_3_fin`, is a needed step towards using axioms with guaranteed independence without rewriting (too many) existing proofs. This result also depends on theorem 2 (Sec. 4.3). The first result below is part of Schutz' definition of chains.

```
lemma chain_on_path:
  assumes "ch_by_ord f X"
  shows "∃P∈𝒫. X⊆P"

lemma equiv_chain_1:
  "(∃f. ch_by_ord f X ∧ a∈X∧b∈X∧c∈X ∧ a≠b∧a≠c∧b≠c ∧ [[a b c]])
    ⟷ [[..a..b..c..]X]"

lemma (in MinkowskiSpacetime) equiv_chain_2:
  "∃f. [f[a..b..c]X] ⟷ [[a..b..c]X]"

lemma equiv_chain_3_fin:
  assumes "finite X"
  shows "long_ch_by_ord2 f X ⟷ long_ch_by_ord f X"
```

3.3 Unreachability

While the axioms of Sec. 3.1 establish a geometry, nothing in them excludes a Euclidean space with Galilean relativity [42, p. 12]. The next group of axioms (I5-I7) specifies existence and basic properties of unreachable sets, a concept tightly linked to the lightcones often used in relativistic physics. In fact, if we pre-empt significantly, and hypothesise our undefined paths to relate to observer worldlines, one can glean the notion of an ultimate speed limit hidden in the condition that certain regions of spacetime should not be connected by paths. Ultimately, saying that nothing can move faster than some speed c is merely the statement that certain histories or trajectories through space and time should not occur.

```
definition unreachable_subset ::
  "'a set ⇒ 'a ⇒ 'a set" ("∅ _ _" [100, 100] 100)
  where "unreachable_subset Q b
    ≡ {x∈Q. Q ∈ 𝒫 ∧ b ∈ ℰ ∧ b ∉ Q ∧ ¬(∃R∈𝒫. b ∈ R ∧ x ∈ R)}"
definition unreachable_subset_via ::
  "'a set ⇒ 'a ⇒ 'a set ⇒ 'a ⇒ 'a set"
    ("∅ _ from _ via _ at _" [100, 100, 100, 100] 100)
  where "unreachable_subset_via Q Qa R x
    ≡ {Qy. [[x Qy Qa]] ∧ (∃Rw∈R. Qa ∈ ∅ Q Rw ∧ Qy ∈ ∅ Q Rw)}"
```

We begin by defining different unreachable sets. The first is simple enough: it collects all the events x of a path Q that cannot be connected (by a path) to another event $b \notin Q$. The second is more complex : if Q meets R at x, `∅ Q from Qa via R at x` collects all events $Q_y \in Q$ that are on the side of the intersection x given by Q_a, and where some event on R be connected neither to Q_a nor Q_y.

Axiom I5 says that if there is a point b not on a path Q, then there are two unreachable events on Q. Together with I1 this implies that no path is empty (App. B.2).

```
locale MinkowskiUnreachable = MinkowskiChain +
    (* I5 *)
  assumes two_in_unreach:
    "⟦Q ∈ 𝒫; b ∈ ℰ; b ∉ Q⟧ ⟹ ∃x∈∅ Q b. ∃y∈∅ Q b. x ≠ y"
```

Schutz calls axiom I6 "Connectedness of the Unreachable Set". Indeed, given two unreachable (from b) events Q_x, Q_z on a path Q, it essentially states that any two points between Q_x, Q_z must be unreachable too. This is phrased in terms of a finite chain with endpoints Q_x, Q_z: any event of the chain is unreachable (second line of the conclusion), and any event between consecutive chain events is unreachable (third line of the conclusion). Notice the extra clause for short chains: if we have only two events, ternary ordering is meaningless, thus so is f. We use it in Sec. 4.6.

```
      and I6:
    "⟦Q ∈ 𝒫; b ∉ Q; b ∈ ℰ; Qx ∈ (∅ Q b); Qz ∈ (∅ Q b)⟧
      ⟹ ∃X. ∃f. ch_by_ord f X ∧ f 0 = Qx ∧ f (card X - 1) = Qz
        ∧ (∀i∈{1 .. card X - 1}. (f i) ∈ ∅ Q b
          ∧ (∀Qy∈ℰ. [[(f(i-1)) Qy (f i)]] ⟶ Qy ∈ ∅ Q b))
        ∧ (short_ch X ⟶ Qx∈X ∧ Qz∈X
          ∧ (∀Qy∈ℰ. [[Qx Qy Qz]] ⟶ Qy ∈ ∅ Q b))"
```

Axiom I7, "Boundedness of the Unreachable Set", provides a property reminiscent of the Archimedean property. Given two events Q_0, Q_y of an unreachable set, it states that one can find a finite chain $[Q_0 \ldots Q_n] \ni Q_y$ where Q_n is not unreachable; i.e. one can "leave" the unreachable set in finitely many "steps".

```
      and I7:
    "⟦Q ∈ 𝒫; b ∉ Q; b ∈ ℰ; Qx ∈ Q - ∅ Q b; Qy ∈ ∅ Q b⟧
      ⟹ ∃g X Qn. [g[Qx..Qy..Qn]X] ∧ Qn ∈ Q - ∅ Q b"
```

3.4 Symmetry and Continuity

The final two axioms, symmetry and continuity, both receive their own locale. Neither of them has been used in proofs to date, but formalising them was part of the scope of this project. The axiom of symmetry is a hefty statement that, according to Schutz [42] serves as a replacement of an entire axiom group in geometries such as Hilbert's *Grundlagen* [20]. Continuity is simple to state, but relies on mechanised definitions of bounds and closest bounds, which we formalise as well. We begin by stating the axiom of symmetry, explaining as we go along.

```
locale MinkowskiSymmetry = MinkowskiUnreachable +
  assumes Symmetry:
    "⟦Q ∈ 𝒫; R ∈ 𝒫; S ∈ 𝒫; Q ≠ R; Q ≠ S; R ≠ S;
    x ∈ Q∩R∩S; Q_a ∈ Q; Q_a ≠ x;
    ∅ Q from Q_a via R at x = ∅ Q from Q_a via S at x⟧
```

The first two lines essentially say that Q, R, S are distinct paths in `SPRAY x` (see Sec. 3.5), and obtain an event $Q_a \neq x$ on Q. The third states that the unreachable sets of Q from the source x via R and S are the same.

Thinking about paths as proto-observers, and unreachable sets as projections that define the relation between observers (much like complements of lightcones), we can try to reverse engineer a physical relativistic spacetime. We can see the symmetry between this special set of events dependent on R, S should, provided R, S are somehow thought of as inertial, "straight lines", identify the entire paths as far as Q is concerned.

We split up the conclusion of the axiom, reproducing Schutz' prose [42, p. 16] for each of the parts (i)-(iv); notice the first line below quantifies the entire conclusion.

(i) there is a mapping $\theta : \mathcal{E} \longrightarrow \mathcal{E}$

```
    ⟹ ∃θ::'a⇒'a.
```

(ii) which induces a bijection $\Theta : \mathcal{P} \longrightarrow \mathcal{P}$

Schutz doesn't give an explicit form for Θ. Since the set of paths is contained in the powerset of events, taking the direct image under θ to be the induced bijection seems the only choice.

```
                    bij_betw (λP. {z. y∈P ∧ (z = θ y)}) 𝒫 𝒫
```

(iii) the events of Q are invariant, and

```
                    ∧ (y∈Q ⟶ θ y = y)
```

(iv) $\Theta : R \longrightarrow S$

```
                    ∧ (λP. {z. y∈P ∧ (z = θ y)}) R = S
```

Schutz' statement is not completely clear on whether he means Q to be invariant under θ or Θ. We settled on the stronger version, involving θ-invariance: it is stronger than the alternative only by also preserving the ordering of the events on Q. Since this ordering affects unreachable sets, not preserving it seemed to go against the premise of the axiom. See App. B.1 for a discussion of totality in this context.

The axiom of continuity is easily stated, but relies on the additional notion of bounds of chains. It compares to the property of least upper bounds on the real numbers, and indeed, theorem 14 (entitled "Continuity"; unproven due to time constraints), the first to use this axiom, deals with sets that look similar to Dedekind cuts [7]. Bounds are defined by Schutz only for infinite chains.

Since we did not use bounds or continuity in our proofs, but completing the mechanisation of the axioms was part of the project, we give only an outline. The listing below defines what it means to say that Q_b is a bound for the infinite chain Q indexed by f: namely all chain elements are ordered as if Q_b has the largest index.

```
definition is_bound_f :: "'a ⇒ 'a set ⇒ (nat⇒'a) ⇒ bool" where
  "is_bound_f Q_b Q f ≡
    ∀i j ::nat. [f[(f 0)..]Q] ∧ (i<j ⟶ [[(f i) (f j) Q_b]])"
```

A closest bound is one which is between all other bounds and any chain event. The axiom of continuity is now so simple that the Isabelle locale below is easily readable.

```
definition closest_bound :: "'a ⇒ 'a set ⇒ bool" where
  "closest_bound Q_b Q ≡ ∃f. is_bound_f Q_b Q f ∧
    (∀ Q_b'. (is_bound Q_b' Q ∧ Q_b' ≠ Q_b) ⟶ [[(f 0) Q_b Q_b']])"

locale MinkowskiContinuity = MinkowskiSymmetry +
  assumes Continuity: "bounded Q ⟶ (∃Q_b. closest_bound Q_b Q)"
```

3.5 Path Dependence and Dimension

The final axiom we introduce is that of dimension. It comes last in our hierarchy of locales because spacetimes in different numbers of dimensions could be constructed. Thus we found it sensible to have an easily replaceable top layer that specifies only the axiom least critical to the general Minkowski spacetime structure, in case one wants to explore other dimensions. However, this axiom has a hidden purpose much more fundamental than we first realised: it is the only axiom that excludes a singleton set of events with an empty set of paths from being a model. In this way, the axiom of dimension turns out to be crucial to several fairly basic proofs involving geometric construction of several paths (that without it could not be guaranteed to exist), and we

ended up working inside the full `MinkowskiSpacetime` locale for many more proofs than originally expected (notably, any proofs requiring the overlapping ordering lemmas presented in Sec. 4.2).

Defining dimensionality in linear algebra requires the idea of linear dependence and independence. We need a more primitive notion, namely an idea of paths depending on other paths. This relation is defined only for a set of paths that all cross in one point: to simplify this discussion, Schutz defines the SPRAY at $x \in \mathcal{E}$ to be $\text{SPRAY}[x] := \{R : R \ni x, R \in \mathcal{P}\}$. Path dependence in a SPRAY is defined first for a set of three paths (we replace Schutz' notation *SPR* with our SPRAY in quotes below):

> *A subset of three paths of a SPRAY is dependent if there is a path which does not belong to the SPRAY and which contains one event from each of the three paths: we also say any one of the three paths is dependent on the other two. Otherwise the subset is independent.* (Schutz [42, p. 13])

```
definition SPRAY :: "'a ⇒ ('a set) set"
  where "SPRAY x ≡ {R∈𝒫. x ∈ R}"

definition dep3_event :: "'a set ⇒ 'a set ⇒ 'a set ⇒ 'a ⇒ bool"
  where "dep3_event Q R S x
    ≡ Q ≠ R ∧ Q ≠ S ∧ R ≠ S
      ∧ Q ∈ SPRAY x ∧ R ∈ SPRAY x ∧ S ∈ SPRAY x
      ∧ (∃T∈𝒫. T ∉ SPRAY x
        ∧ (∃y∈Q. y ∈ T) ∧ (∃y∈R. y ∈ T) ∧ (∃y∈S. y ∈ T))"
```

To obtain path dependence for an arbitrary number of paths, we extend the base case above by induction, following Schutz:

> *We next give recursive definitions of dependence and independence which will be used to characterize the concept of dimension. A path T is dependent on the set of n paths (where $n \geq 3$)*
>
> $$S = \left\{ Q^{(i)} : i = 1, 2, \ldots, n;\ Q^{(i)} \in SPRAY[x] \right\}$$
>
> *if it is dependent on two paths $S^{(1)}$ and $S^{(2)}$, where each of these two paths is dependent on some subset of $n-1$ paths from the set S. We also say that the set of $n+1$ paths $S \cup \{T\}$ is a dependent set. If a set of paths has no dependent subset, we say that the set of paths is an independent set.* (Schutz [42, p. 14])

```
inductive dep_path :: "'a set ⇒ ('a set) set ⇒ 'a ⇒ bool"
  where
    dep_two: "dep3_event T A B x ⟹ dep_path T {A, B} x"
  | dep_n: "⟦S ⊆ SPRAY x; card S ≥ 3; dep_path T {S1, S2} x;
      S' ⊆ S; S'' ⊆ S; card S' = card S - 1; card S'' = card S - 1;
      dep_path S1 S' x; dep_path S2 S'' x⟧
        ⟹ dep_path T S x"
```

This definition uses the keyword **inductive**, which allows us to give a non-recursive base case and induction rules, to create the minimal set of triplets T, S, x such that `dep_path T S x`. Notice that we keep track of the (source of the) SPRAY that the paths exist in explicitly, while Schutz keeps this implicit, referring to it as and when needed. This leaves us with only the job of transforming this constructive definition into an analytical one, such that a set of paths can be examined and found dependent or not, rather than being able only to construct such sets to measure.

```
definition dep_set :: "('a set) set ⇒ bool"
  where "dep_set S ≡ ∃x. ∃S'⊆S. ∃P∈(S-S'). dep_path P S' x"
definition indep_set :: "('a set) set ⇒ bool"
  where "indep_set S ≡ ¬(∃T ⊆ S. dep_set T)"
```

Now the axiom of dimension is given as follows, with a final definition:

> *A SPRAY is a 3-SPRAY if:*
>
> *(i) it contains four independent paths, and*
>
> *(ii) all paths of the SPRAY are dependent on these four paths.*
>
> **Axiom I4** (Dimension)
> *If $\mathcal{E}$ is non-empty, then there is at least one 3-SPRAY.*
>
> (Schutz [42, p. 14])

Formalising the 3-SPRAY in Isabelle/HOL is long because we need to introduce the four distinct paths, all of them in a SPRAY. The final two lines of the definition are the interesting ones.

```
definition three_SPRAY :: "'a ⇒ bool" where
  "three_SPRAY x ≡ ∃S1∈𝒫. ∃S2∈𝒫. ∃S3∈𝒫. ∃S4∈𝒫.
    S1 ≠ S2 ∧ S1 ≠ S3 ∧ S1 ≠ S4 ∧ S2 ≠ S3 ∧ S2 ≠ S4 ∧ S3 ≠ S4
    ∧ S1 ∈ SPRAY x ∧ S2 ∈ SPRAY x ∧ S3 ∈ SPRAY x ∧ S4 ∈ SPRAY x
    ∧ (indep_set {S1, S2, S3, S4})
    ∧ (∀S∈SPRAY x. dep_path S {S1,S2,S3,S4} x)"

locale MinkowskiSpacetime = MinkowskiContinuity +
  (* I4 *)
  assumes ex_3SPRAY: "ℰ ≠ {} ⟹ ∃x∈ℰ. three_SPRAY x"
```

Chapter 4

Formal Proofs in Isabelle

The following chapter presents several of Schutz' theorems, fully mechanised and verified in Isabelle/HOL. A large part of the proofs are much longer than in the original text, and/or rely on results not mentioned by Schutz [42]. While we make no claim to have found particularly short proofs, and in many instances believe shorter mechanisations or alternative definitions are possible, we believe this is due in part to the temptation of the working mathematician to presume more of the geometric intuition they are trying to construct than is justified. For example, several theorems of Schutz' had to be restated in more detail to include short chains, a case often disregarded completely. Similarly, our definition of chains (Sec. 3.2)is far more formal, less geometrical, and often requires additional steps when deployed for deduction.

We try to present proof procedures at a comfortable ratio of detail to length. Fairly often, extra steps required in Isabelle are obvious to the inspecting reader; often their omission goes unnoticed. We therefore employ snipping rather freely. We denote by `<proof>` a proof that was cut, but exists in the associated proof script. The notation `...` is used for cutting away multiple, not necessarily related lines, or even just a part of a line. This relaxation is possible because we trust the Isabelle verification of our proof: if one wanted to verify all the statements in this thesis, one could simply make sure they exist in the file, identify the introduced axioms, and let Isabelle check the entire file. No `sorry` or `oops` keywords are cut. All cut parts pass, make the surrounding script pass, and rely on no `sorry` statements.

The first two results presented are not full theorems: one is only the second statement of theorem 6 (Sec. 4.1), the next is a lemma given to prove theorem 9 (and enabling many other proofs, Sec. 4.2). Both of these were left out in Palmer's treatment, with the lemma left assumed to prove theorem 9. The next two results (Secs. 4.3 and

4.4) combine to give a proof of theorem 10. We then discuss proofs of theorems 11 and 13, with additional results.

4.1 Infinity of Paths: Theorem 6(ii)

We complete Palmer's proof of theorem 6, proving part (ii) by induction as suggested by Schutz. The main difficulty lies in formalising Schutz' list of results to use to a level understood by Isabelle. This involves thinking about how to translate from induction on a natural number to infinity, what exactly the induction variable should be, and dealing with the rigidity in applying induction hypotheses in Isabelle (see Sec. 2.3). We begin by stating Schutz' theorem and proof, and our formalisation.

> **Theorem 6** *(Prolongation)*
>
> *(i) If a, b are distinct events of a path Q, then there is an event $c \in Q$ such that $[a\ b\ c]$.*
>
> *(ii) Each path contains an infinite set of distinct events.*
>
> Proof [...]
>
> *(ii) By the preceding theorem any path Q has at least two distinct events. Now by part (i), Theorem 1, and induction, the path Q contains an infinite set of distinct events.* q.e.d.
>
> (Schutz [42, p. 21])

```
theorem (*6ii*) infinite_paths:
  assumes "P∈𝒫"
  shows "infinite P"
```

This corresponds to Schutz' statement almost exactly, with the difference that we make use of paths being sets of events (there is no need to talk about an infinite subset) to simplify the conclusion. The proof proceeds by assuming `finite P` to show `False`. The main work is abstracted into the helper lemma `path_card_nil`, which states that the cardinality of the path is no larger than nil. Together with `¬(P={})`, the only way for a finite set to have cardinality nil, this gives `False`.

For the inductive method, we note only that it is used indirectly to prove this result. We prove by induction that it is possible to choose two elements a, b of a set of events on a path, such that all other elements of that set are between a and b. Their prolongation from part (i) of Th. 6 then gives a contradiction. A detailed account of the mechanised proof is appended (App. B.4); see also Sec. 2.3.2, where this lemma's proof by induction is used as an example.

4.2 Overlapping orderings

Schutz introduces several lemmas that extend betweenness relations between overlapping sets of events [42, pp. 23–25], of which we give the first below.

```
lemma abc_abd_bcdbdc:
  assumes abc: "[[a b c]]"
      and abd: "[[a b d]]"
      and c_neq_d: "c ≠ d"
  shows "[[b c d]] ∨ [[b d c]]"
```

This particular result is proved using a lengthy geometric construction, and a crucial invocation of Theorem 8. However, together with the axioms of order, the lemma `abc_abd_bcdbdc` makes it easy to derive all the other lemmas about overlapping orderings we will need, so it truly is a fundamental result. It is needed for the proof of Theorem 9, which Palmer [34] formalised, but had to assume `abc_abd_bcdbdc` for. It also comes into many other proofs, such as Theorem 10 (Sec. 4.4) and equivalence proofs for different chains (Sec. 3.2).

We define a *kinematic triangle* △ `a b c` as a *set of three distinct events* $\{a,b,c\}$ *such that each pair of events belongs to one of three distinct paths* (i.e. the set of vertices of a triangle of paths). Theorem 8 then states that no path can cross all three sides of a kinematic triangle internally. Of course, this description relies on geometric imagination, which is replaced in Isabelle by betweenness and paths alone. Palmer [35] provides a mechanised proof of the Th. 8 stated as follows:

```
theorem (*8*) (in MinkowskiChain) tri_betw_no_path:
  assumes tri_abc: "△ a b c"
      and ab'c: "[[a b' c]]"
      and bc'a: "[[b c' a]]"
      and ca'b: "[[c a' b]]"
  shows "¬(∃Q∈𝒫. a'∈Q ∧ b'∈Q ∧ c'∈Q)"
```

To prove `abc_abd_bcdbdc`, we follow Schutz fairly closely, with the top layer being a proof by contradiction together with $\neg[dbc] \rightarrow [bcd] \vee [bdc]$, which is obtained by noting that path uniqueness (axiom I3) and `abc_ex_path` (axiom O1) imply that b,c,d all lie on the same path, and thus must be in some betweenness relationship (axiom O5). We here omit this high-level structure, and give only a proof of $\neg[dbc]$.

We assume $[dbc]$ and derive a contradiction. Schutz (and we) does this by constructing several kinematic triangles, whose interaction with each other leads to a contradiction with theorem 8 (`tri_betw_no_path`).

```
    assume dbc: "[[d b c]]"
    obtain ab where path_ab: "path ab a b"
      using abc_abc_neq abc_ex_path_unique abc by blast
```

```
  obtain S where path_S: "S ∈ 𝒫"
             and S_neq_ab: "S ≠ ab"
             and a_inS: "a ∈ S"
    using ex_crossing_at path_ab
    by auto
```

We follow Schutz in obtaining the basic geometric ingredients: first a path containing a and b. Given a path ab and an event a on it, Th. 5 (`ex_crossing_at`, App. B.3) provides a different path S. Schutz' next step is slightly nontrivial to Isabelle, and requires a sub-proof for existence before we can obtain, citing Schutz, "$e \in S \setminus \{a\}$ and a path be"[1]. Apart from axiom I5 (`two_in_unreach`) and theorem 4 (`unreachable_set_bounded`), which Schutz gives as justification for this step, our mechanisation relies on finding an explicit path connecting two elements reachable from each other (`reachable_path`).

```
  have "∃e∈S. e ≠ a ∧ (∃be∈𝒫. path be b e)" <proof>
  then obtain e be where e_inS: "e ∈ S"
                     and e_neq_a: "e ≠ a"
                     and path_be: "path be b e"
    by blast
```

The difficulty of translating Schutz' approach to Isabelle is in his conditional assignment of events to the variables he calls $c*, d*, f*$; they become `c'`, `d'`, `f'` in our proof script since the $*$-affix is reserved in Isabelle. We abstract this difficulty into lemmas called `exist_c'd'` and `exist_f'`. Several case splits need to be considered, but have no further importance outside of these lemmas: thus we separate them from the main proof. Notice that `exist_c'd'` and `exist_f'` are trivial in a highly non-obvious fashion: since they are to be used inside a proof by contradiction, their assumptions already imply `False`, which implies anything. This implication, however, is complex enough not to be detected by Isabelle's automatic tools, nor by us upon inspection. The assumptions on both lemmas are equivalent to the obtained facts in the main proof at the point of their use; we omit them here for brevity.

```
lemma exist_c'd':
  assumes ...
  shows "∃c' d'. ∃d'e c'e. c'∈ab ∧ d'∈ab
                         ∧ [[a b d']] ∧ [[c' b a]] ∧ [[c' b d']]
                         ∧ path d'e d' e ∧ path c'e c' e"
```

Schutz' proof (text in App. A.5) considers nested case splits "in parallel", jumping between cases for each statement in the flow of the main proof. We, instead, just abstract proofs of existential propositions with all the properties we need into the lemmas `exist_c'd'` and `exist_f'`, and require no case splits in the main proof. A structural out-

[1] At this stage, we already know that a path is uniquely determined by two points, so a path containing b and e can safely be called be.

line is provided for the proof body of `exist_c'd'`, but most of the individual steps are omitted. Notice the use of the helper-lemma `unreachable_bounded_path`, which replaces Schutz' more vague statement of "the Boundedness of the Unreachable Set (Th.4) implies". While it relies on Theorem 4 and the assumptions of `exist_c'd'` only (excluding definitions), several steps are needed in Isabelle to derive this result (App. B.5). The lemma `exist_f'`, following Schutz in the same way, is omitted from the main text, but can be found in App. B.5.

```
proof (cases)
  assume "∃de. path de d e"
  then obtain de where "path de d e" by blast
  hence "[[a b d]] ∧ d∈ab"
    using abd betw_c_in_path path_ab by blast
  thus ?thesis
  proof (cases)
    assume "∃ce. path ce c e"
    thus ?thesis ...
  next
    assume "¬(∃ce. path ce c e)"
    obtain c' c'e where "c'∈ab ∧ path c'e c' e ∧ [[b c c']]"
      using unreachable_bounded_path
    ...
  qed
next
  assume "¬ (∃de. path de d e)"
  thus ?thesis
  proof (cases)
    assume "∃ce. path ce c e" ...
  next
    assume "¬(∃ce. path ce c e)" ...
  qed
qed
```

From here on, the proof follows Schutz, who in turn follows Veblen [50, p.357]. The idea is to find three events on the path `f'b`, obtained from `exist_f'`, that lie on different sides of the triangle `\<trianlge> e a d'`. This gives a contradiction to Th.8: no path can cross all three sides of a kinematic triangle. A more detailed account is in App. B.5.

4.3 Local Chains: Theorem 2 Revisited

Schutz proves theorem 2 to strengthen his definition of chains. It allows him to define chains by imposing betweenness relations on any chain events with adjacent indices, but obtain betweenness relations between chain events with any index.

Theorem 2 *(Order on a finite chain)*
On any finite chain $Q_0 \ldots Q_n$, there is a betweenness relation for each ordered truple; that is

$$0 \leq i < j < l \leq n \Longrightarrow [Q_i\ Q_j\ Q_l\ .]$$

Furthermore all events of a chain are distinct. (Schutz [42, p. 18])

Let us note that since we use sets of events as our basis for defining chains, not sequences, distinctness of the events is a given. This theorem justifies our set-based approach. Palmer's original definition of `ordering` is strictly stronger than Schutz' definition of chains (see Sec. 3.2). This leads to a proof of theorem 2 that is essentially just an unfolding of the definition.

Given this theorem (for Schutz' local chains), and restricting our attention to finite chains, we can justify the stronger definition: locally ordered chains can be made total using the axioms. A stronger definition may, however, cause problems later on: e.g. Schutz' system may have models ours does not. Refer to Sec. 4.4 for the example that motivated us to re-prove this theorem for the more Schutz-like local index-chains.

```
theorem order_finite_chain2:
  assumes chX: "long_ch_by_ord2 f X"
      and finiteX: "finite X"
      and ordered_nats: "0 ≤ (i::nat) ∧ i < j ∧ j < l ∧ l < card X"
  shows "[[(f i) (f j) (f l)]]"
```

The proof of this theorem is omitted, but the interested reader is referred to App. B.6. The mechanisation follows Schutz' prose, up to a few clarifications of case splits and inductions; the overall structure is the same (original text in App. A.6).

4.4 Theorem 10 (Subpaths are Chains)

```
context MinkowskiSpacetime begin
```

Mechanising Theorem 10 was a major undertaking, and hit more obstacles than any other of our (proven) results. One problem was due to the definition of `ordering` we have been using since Palmer's work [34]: it is stronger than Schutz'. As shown in Sec. 4.3, this leads to a free proof of Th. 2. But such things always come with a price: Schutz' proof of Th. 10 only aims at a local chain. If we want to be consistent with our other mechanisations (which use total chains), we need this to become a total chain, which essentially means going through all the steps of Schutz' proof for Th. 2. This is

why we defined a new local `ordering2`, and proved `order_finite_chain2` in Sec. 4.3.

> **Theorem 10** *(based on Veblen (1904), Theorem 10)*
> *Any finite set of distinct events of a path forms a chain. That is, any set of n distinct events can be represented by the notation* $a_1, a_2, \ldots, a_n$ *such that*
>
> $$[a_1\ a_2 \ldots a_n]\ .$$
>
> (Schutz [42, p. 24])

Our statement differs from Schutz in another way (besides the different kinds of chain used): he forgets the condition that any chain needs to have at least two elements (by definition): thus it isn't *every* finite set of events that qualifies.

```
theorem (*10*) path_finsubset_chain:
  assumes finite_X: "finite X"
      and path_Q: "Q ∈ 𝒫"
      and events_X: "X ⊆ Q"
      and at_least_two: "card X ≥ 2"
    shows "ch X"
```

The proof is by induction, as in Schutz [42]. To encapsulate this induction, we use the lemma `path_finsubset_chain_induction`, and as before (Sec. 2.3.2) we reformulate the conclusion into the required object-level induction hypothesis.

```
lemma (*for 10*) path_finsubset_chain_induction:
  assumes path_Q: "Q ∈ 𝒫" and nat_N: "n≥0"
  shows "∀X⊆Q. finite X ∧ card X = n+2 ⟶ ch X"
```

Notice Schutz uses a four-element chain as the base case (original text in App. A.7), so our explicit proof has to provide two (simple) extra cases: two- and three-element sets. A two-element chain is just a set of two points on a path, thus a two-event set X satisfies the definition of chains immediately. A set X with three events a, b, c, all of them on a path, must be a chain because a, b, c are in some betweenness relation by axiom O5. Both of these are omitted from the listing, and we move on to the induction.

The base case of $|X| = 4$ follows directly from Th. 9: it states that a set of four events on a path forms a chain. Schutz' induction proceeds by assuming a chain of n events, and adds an extra event. This is not possible for us: we have already fixed the number of elements of the set we require to be a chain to the successor of the induction variable $n+4$ (where we have introduced the shift of 4 because Isabelle induction starts at $n = 0$, see Sec. 2.3.2). Thus we obtain a new set by removing an element, and argue this new set must be a chain by the induction hypothesis `IH`. We remove some overall indentation for legibility.

```
fix n::nat
assume IH: "∀X⊆Q. finite X ∧ card X = n+4 ⟶ ch X"
```

```
show "∀X⊆Q. finite X ∧ card X = Suc n+4 ⟶ ch X"
proof (safe)
  ...
  hence "card X ≥ 5" by (simp add: fin_X)
  then obtain b Y where Y_def: "X = Y ∪ {b} ∧ b∉Y" <proof>
  hence "ch Y"
    using IH events_X fin_X Un_infinite card_X
    by auto
  ...
  have "card Y ≥ 4" <proof>
  then obtain f where f_def: "long_ch_by_ord f Y" <proof>
```

This places us in the setting of Schutz' proof: we have a chain Y, indexed by f, of at least four events, and a set containing one extra event in addition to this chain. We now itroduce variable names that agree with those of Schutz. In terms of our indexing function, the subscripts of those variables are shifted, but it allows us to reproduce his prose more exactly. This difference is due to Schutz now using base-1 indexing, not base-0 as in his definition of chains.

```
obtain a_1 a a_n where long_ch_Y: "[f[a_1..a..a_n]Y]"
  using get_fin_long_ch_bounds Y_def f_def fin_X
  by fastforce
hence bound_indices: "f 0 = a_1 ∧ f (card Y - 1) = a_n"
  by (simp add: fin_long_chain_def)
```

The remaining proof is structured into the same three cases Schutz considers. We obtain the three possible betweenness relations the three events above can be in, and consider each in turn.

```
have betw_cases: "[[b a_1 a_n]] ∨ [[a_1 b a_n]] ∨ [[a_1 a_n b]]"
  using some_betw path_Q by (meson abc_sym)
show "ch X" using betw_cases
proof (rule disjE3)
  (* case (i) *)
  assume "[[b a_1 a_n]]"
  obtain g where "g=(λj::nat. if j≥1 then f (j-1) else b)"
    by simp
  hence "[g[b..a_1..a_n]X]"
    using chain_append_at_left_edge ... by blast
  thus "ch X"
    unfolding ch_def ch_by_ord_def using fin_long_chain_def by auto
```

The main proof steps needed for this case are inside `chain_append_at_left_edge`. Schutz' prose for this case is given below.

> *Case (i): By the inductive hypothesis and Theorem 2 we have* $[a_1\ a_2\ a_n]$*, so the previous theorem (Th.9) implies that* $[b\ a_1\ a_2\ a_n]$ *which implies that* $[b\ a_1\ a_2]$*. Thus b is an element of a chain* $[a_1^*\ a_2^*\ \ldots\ a_{n+1}^*]$ *where* $a_1^* = b$ *and (for* $j \in \{2,\ldots,n+1\}$*)* $a_j^* := a_{j-1}^*$. (Schutz [42, p. 25])

Since we cannot informally extend our notation for ordering to arbitrary numbers of elements, we skip the step involving $[b\ a_1\ a_2\ a_n]$, employing instead an alternative ordering relation `abd_bcd_abc: "⟦[[a b d]; [[b c d]]⟧⟹[[a b c]]"` that is not given in Schutz, but which follows readily from the ones provided. We could have formulated a four-element chain with an explicit indexing function to effectively yield Schutz' result, but since even that is somewhat removed from the prose, and requires multiple extra entities to be defined, we decided this way was better. We give a heavily cut listing of the proof below.

```
lemma (*for 10*) chain_append_at_left_edge:
  assumes long_ch_Y: "[f[a_1..a..a_n]Y]"
      and Y_def: "X = Y ∪ {b}" "b∉Y"
      and fin_X: "finite X"
      and bY: "[[b a_1 a_n]]"
      and g_def: "g = (λj::nat. if j≥1 then f (j-1) else b)"
    shows "[g[b .. a_1 .. a_n]X]"
proof -
  ...
  hence "[[a_1 (f 1) a_n]]"
    using order_finite_chain fin_long_chain_def long_ch_Y
    by auto
  hence "[[b a_1 (f 1)]]"
    using bY abd_bcd_abc by blast
```

Schutz' final sentence implies an indexing function that is equal to our pre-obtained `g`, and his statement requires manual proofs of multiple chain properties regarding indexing and betweenness in Isabelle. Notice that this is where Th. 2 comes in for us: Schutz only shows that a single betweenness relation holds between b and adjacent elements. It is Th. 2 (Sec. 4.3) that allows us to extend this to betweenness relations involving any events on the (finite) chain, and obtain a total chain. For details on different orderings and chains, see 3.2.

```
  have "ordering2 g betw X" <proof>
  hence "long_ch_by_ord2 g X"
    using points_in_chain ... by blast
  hence "long_ch_by_ord g X"
    using equiv_chain_3a_fin fin_X by blast
```

We now go back to Theorem 10's induction. Two cases remain: b being the middle element (ii), and b being on the right (iii). Case (iii) is symmetric with case (i), and Schutz doesn't give a written-out proof for it. Instead of copy-pasting the entire proof for `chain_append_at_left_edge`, we therefore choose to use a different result, `chain_sym`, to give a more interesting, shorter proof using symmetry.

```
lemma chain_sym:
  assumes "[f[a..b..c]X]"
  obtains g where "[g[c..b..a]X]" and "g=(λn. f (card X - 1 - n))"
```

```
lemma (*for 10*) chain_append_at_right_edge:
  assumes long_ch_Y: "[f[a_1..a..a_n]Y]"
      and Y_def: "X = Y ∪ {b}" "b∉Y"
      and fin_X: "finite X"
      and Yb: "[[a_1 a_n b]]"
      and g_def: "g= (λj::nat. if j ≤ (card X - 2) then f j else b)"
    shows "[g[a_1 .. a_n .. b]X]"
proof -
  ...
  obtain f2 where f2_def: "[f2[a_n..a..a_1]Y]" "f2=(λn. f (card Y -
   1 - n))"
    using chain_sym long_ch_Y by blast
  obtain g2 where g2_def:
    "g2 = (λj::nat. if j≥1 then f2 (j-1) else b)"
    by simp
  have "[[b a_n a_1]]"
    using abc_sym Yb by blast
```

The functions `f2` and `g2` can be thought of as reversed versions of `f` and `g`: if `f` indexes a chain "left-to-right", `f2` counts "right-to-left". We can show `g2` orders `X` into a chain using `chain_append_at_left_edge`, and then reverse it again using `chain_sym` to get `g1`, which thus orders `X`. Finally, we show `g1=g`, here in mathematical notation:

$$
\begin{aligned}
g_1(n) = g_2(|X|-1-n) &= \begin{cases} f_2(|X|-2-n) & \text{if } |X|-1-n \geq 1 \\ b & \text{otherwise} \end{cases} \\
&= \begin{cases} f(|Y|+1-|X|+n) & \text{if } |X|-2 \geq n \\ b & \text{otherwise} \end{cases} \\
&= g(n)
\end{aligned}
$$

Case (ii) of this proof can be found in detail in App. B.7. Here we just comment on our difficulty following Schutz' prose exactly. His easy statement "Let k be the smallest integer such that $[a_1\ b\ a_k]$" requires a nontrivial existence proof. Neither did we manage to split the remainder of case (ii) according to the same conditions seen in Schutz' proof. We argue this is because he restricts his attention to a handful of events only, trusting his reader's intuition to convince them that everything else "stays the same". We, on the other hand, need to show explicitly that the new way of indexing given by `g` satisfies the definition of a chain *everywhere* on X, i.e.:

```
have "∀n n' n''.
      (finite X ⟶ n'' < card X) ∧ Suc n = n' ∧ Suc n' = n''
          ⟶ [[(g n) (g (Suc n)) (g (Suc (Suc n)))]]"
```

This means splitting according to the value of n and its two successors, in order to fix the (conditional) form of g. We do mirror his case splits in the following results, which are all used in different cases according to (the successors of) n.

```
have b_middle: "[[(f (k-1)) b (f k)]]" <proof>
have b_right: "[[(f (k-2)) (f (k-1)) b]]" if "k ≥ 2" <proof>
have b_left: "[[b (f k) (f (k+1))]]" if "k+1 ≤ card Y - 1" <proof>
```

It may be argued that one could force Schutz' case split, but since our definition of `ordering2` explicitly requires universal quantification over indices, and g is defined piecewise, the case split we employ would still have to be made.

4.5 Theorem 11 (Segmentation)

```
context MinkowskiSpacetime begin
```

The final result of Schutz' section 3.6 (Order on a path), this theorem allows us to use any finite subset of a path in order to split it into disjoint regions. Schutz provides a three-line argument by analogy with the proof of Theorem 10, arguing this result is a direct consequence of Theorems 10 and 1, with a transparent case split. We found that Schutz' statement is unprovable at the point of his stating it.

Schutz defines the *segment* between distinct events a, b of a path ab as the set $(ab) = \{x : [a\,x\,b],\ x \in ab\}$. Similarly, define the *interval* $|ab|$ as $(ab) \cup \{a, b\}$, and the *prolongation* of (ab) beyond b as $\{x : [a\,b\,x],\ x \in ab\}$.

> **Theorem 11** *(after Veblen (1904), Theorem 11)*
> *Any finite set of N distinct events of a path separates it into $N-1$ segments and two prolongations of segments.*
>
> Proof *As in the proof of the previous theorem (Th. 10), any event distinct from the a_i $(i = 1, \ldots, N)$ belongs to a segment (Case (ii)) or a prolongation (Cases (i) and (iii)). Theorem 1 implies that the $N-1$ segments and two prolongations are disjoint.* q.e.d. (Schutz [42, p. 27])

While this sounds natural enough to the geometric intuition, taking a path to be somehow line-like, the part of the statement regarding the number of segments is impossible to prove at this point. Given two events a and b on a path P, Theorem 6 (on prolongation, Sec. 4.1) guarantees the existence of $c \in P$ such that $[abc]$ (or, by symmetry, such that $[cab]$), but we can guarantee the existence of an element c such that $[acb]$ only after Th. 17, which states exactly that. The problem is that formally, Th. 17 relies on Th. 13, which in turn requires Th. 11, so we cannot just postpone this result. Since no such element can be guaranteed to exist, segments can be empty. Then since they are defined as sets, all empty segments are equal (to the empty set), and this degeneracy can reduce the number of segments that exist in the segmentation.

One could fix this problem by taking intervals instead of segments. By definition, no interval is empty, fixing their number as Schutz suggests – but the intervals would overlap at their endpoints, losing disjointness. Alternatively, a segment could be defined as a set plus two endpoints, which would also dispose of the degeneracy. We surmise that one could also prove that there are *at most* $N-1$ segments. We simply choose to omit the number-of-segments part of the theorem: so far we do not need it.

Ultimately, the problem is not fatal: our statement of Theorem 11 is slightly weaker, but suffices for a proof of Theorem 13. Isabelle also requires us to change Schutz' statement further, taking as input his set of events, and returning it as a chain. His numbering the events in his proof may indicate that was what he had in mind all along. In a similar fashion, we explicitly obtain the set of segments and the two prolongations Schutz leaves unspecified, but their form is not surprising, and these objects can be obtained by Isabelle without manual existence proofs. The disjointness of the segmentation is also added as a conclusion, while Schutz only mentions it in his proof.

4.5.1 Without additional assumptions

In the end, our statement looks quite different from the original, but we believe it captures Schutz' intention while being both more useful and fully verified.

```
theorem (*11*) segmentation:
  assumes path_P: "P∈𝒫"
      and Q_def: "finite (Q::'a set)" "card Q = N" "Q⊆P" "N≥2"
  obtains S P1 P2 f a b where
    "(∀x∈S. is_seg x)" "P = ((⋃S) ∪ P1 ∪ P2 ∪ Q)"
(* The union of segments, prolongations & separators is the path. *)
    "P1∩P2={} ∧ (∀x∈S.(x∩P1={} ∧ x∩P2={} ∧ (∀y∈S. x≠y ⟶ x∩y={})))"
(* The prolongations and all the segments are disjoint. *)
    "S = (if N>2 then {s. ∃i<(N-1). s = segment (f i) (f (i+1))}
          else {segment a b})"
(* the segments are consecutive and adjacent *)
    "[f[a..b]Q]" "P1 = prolongation b a" "P2 = prolongation a b"
(* two prolongations go outwards from the set of events *)
```

Notice that the definition of `S` follows our division between short and long chains, and so must the proof. Our first step is to obtain the quantities `P1`, `P2`, `S`, which is possible thanks to Theorem 10; this gives us the indexing function `f` for the segment definitions and the two edge elements `a,b` for the prolongations. For `N≥3` we prove they satisfy the conditions laid out above one by one in helper lemmas. The case of `N=2` is simple: all individual required results are deriveable by Isabelle's *sledgehammer* with the exception of `P = ⋃{s} ∪ P1 ∪ P2 ∪ Q`, which we prove by translating `x ∈ P` (with `Q={a,b}`) into `[[a x b]] ∨ [[b a x]] ∨ [[a b x]] ∨ x=a ∨ x=b`.

```
lemma int_split_to_segs:
  assumes Q_def: "finite (Q::'a set)" "card Q = N" "N≥3"
      and f_def: "[f[a..b..c]Q]"
      and S_def: "S={s. ∃i<(N-1). s = segment (f i) (f (i+1))}"
  shows "interval a c = (⋃S) ∪ Q"
```

The proof is lengthy, but the mechanisation details are largely uninspiring, so we refer the interested reader to App. B.8.

Similar lemmas exist for the remaining conclusions of Theorem 11, but we omit their proofs here. The main result is the segmentation of the interval: the prolongations just act as a two-sided catch-all for any other element. Furthermore, disjointness of the segments, defined as `segment (f i) (f(i+1))`, follows from the ordering of finite chains, and obtaining a chain from a finite subset of a path is easy using Theorem 10.

4.5.2 Assuming path density

Since Schutz omitted so many of the conclusions of our own `segmentation` from his Theorem 11, but did insist on the number of segments, we created an additional locale, called `MinkowskiDense`, to contain an assumed version of Schutz' Th. 17. This is safer than a sorried theorem (see Sec. 2.3) – the assumption `path_dense` will never be used accidentally, as long as we never work in the locale `MinkowskiDense`, and never build a locale on top of it. We prove that the cardinality of the set `S` of segments in the theorem `segmentation` is indeed $N-1$ if path density is assumed.

```
locale MinkowskiDense = MinkowskiSpacetime +
  assumes path_dense: "path ab a b ⟹ ∃x. [[a x b]]"
begin

lemma segment_nonempty:
  assumes "path ab a b"
  obtains x where "x ∈ segment a b"
  using path_dense by (metis abc_abc_neq seg_betw assms)
```

The number-of-segments statement is obviously only interesting if $N \geq 3$, which simplifies the definition of `S`. The remaining conditions are those of the helper lemmas for Theorem 11. Schutz' "$N-1$ segments" turns into a proposition on the cardinality of the set of segments `S`.

```
lemma (*for 11*) number_of_segments:
  assumes path_P: "P∈𝒫"
      and Q_def: "finite (Q::'a set)" "card Q = N" "Q⊆P" "N≥3"
      and f_def: "a∈Q ∧ b∈Q ∧ c∈Q" "[f[a..b..c]Q]"
      and S_def: "S = {s. ∃i<(N-1). s = segment (f i) (f (i+1))}"
      and P1_def: "P1 = prolongation b a"
      and P2_def: "P2 = prolongation b c"
    shows "card S = (N-1)"
```

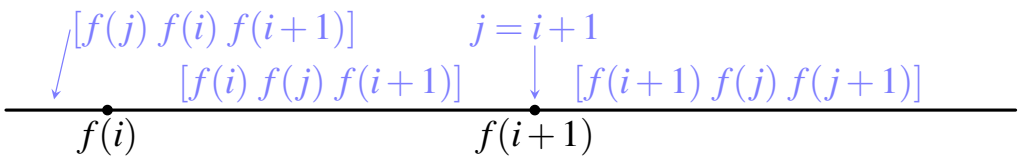

Figure 4.1: Four case splits in the proof of `number_of_segments` for `MinkowskiDense`'s theorem 11, according to the index j, respectively the event $f(j)$.

We can show two sets have equal cardinality if a bijection exists between them.[2] To this end we define a function $i \to (Q_i Q_{i+1})$, denoted `?f`, and prove it is a bijection between the sets `{0..N - 2}` and `S`. With Isabelle's functions being total on data types, this statement is not equivalent to `bij ?f`, and the proof cannot proceed via the injectivity-surjectivity schema expected from mathematics. Rather, it utilises the result `inj_on_imp_bij_betw`, found in the Isabelle/HOL theory of functions (`Fun.thy`). This effectively obtains a restriction of `?f`, so restricted injectivity and a statement about the image of `?f` on `{0..N-2}` are sufficient.

```
proof -
  let ?f = "λ i. segment (f i) (f (i+1))"
  have "?f ` {0..N-2} = S" <proof>
  moreover have "inj_on ?f {0..N-2}" <proof>
  ultimately have "bij_betw ?f {0..N-2} S"
    using inj_on_imp_bij_betw by auto
  thus ?thesis
    using assms(5) bij_betw_same_card numeral_2_eq_2 numeral_3_eq_3
    by (metis (no_types, lifting) ...)
qed
```

Diving briefly into the collapsed proof of injectivity, we show where path density comes into play. Injectivity is proven as usual, that for $i, j \in \{0 \,..\, N-2\}$, we have $f(i) = f(j) \implies i = j$. This is shown by contradiction (`assume "i≠j"`), then split into the cases seen in Fig. 4.1. Notice this is almost the case split of Theorem 10. Picking $[f(i+1)\ f(i)\ f(j)]$ as an example, we use `segment_nonempty`, which is just a segment-themed wrapper for `path_dense`, to obtain an element `e` that satisfies the contradictory orderings $[e\ f(j)\ f(j+1)]$ and $[f(j)\ e\ f(j+1)]$.

```
          assume "[[(f(i+1)) (f i) (f j)]]"
          then obtain e where "e∈?f i" using segment_nonempty
            by (metis ...)
          hence "[[e (f j) (f(j+1))]]"
            using ⟨[[(f(i+1)) (f i) (f j)]]⟩ <proof>
          moreover have "e∈?f j"
            using ⟨e ∈ ?f i⟩ asm(3) by blast
```

[2] This is generally taken as a definition in Mathematics (e.g. Liebeck [24, p. 185]. Isabelle's definition is more technical, but the proof strategy still applies.

```
        ultimately show False
          by (simp add: abc_only_cba1 seg_betw)
```

The other cases run similarly. The case of $[f(i)\ f(j)\ f(i+1)]$ proceeds in one step, since the assumption $f(i) = f(j)$ then gives

$$\begin{aligned} f(j) \in \text{segment}(f(i),\ f(i+1)) &\longrightarrow f(j) \in \text{segment}(f(j),\ f(j+1)) \\ &\longrightarrow [f(j)\ f(j)\ f(j+1)] \\ &\longrightarrow f(j) \neq f(j)\ . \end{aligned}$$

4.6 Theorem 13 (Connectedness of the Unreachable Set)

```
context MinkowskiSpacetime begin
```

To try and validate the development of Schutz' theory after a weakened Theorem 11 (cf Sec. 4.5.1), we prove his Theorem 13, which relies on `segmentation`. We give his statement below, using his notation $Q(b,\emptyset)$ for the unreachable set, but replacing his non-strict ordering with the more explicit $[Q_{i-1}\ Q_y\ Q_i]$ or $Q_y = Q_i$.

> **Theorem 13** *(Connectedness of the Unreachable Set)*
> *Given any path Q, any event $b \notin Q$, and distinct events $Q_x, Q_z \in Q(b,\emptyset)$, then*
>
> $$[Q_x\ Q_y\ Q_z] \Longrightarrow Q_y \in Q(b,\emptyset)\ .$$
>
> Proof *By axiom I6 there is a finite chain $[Q_0\ Q_1\ \ldots\ Q_{n-1}\ Q_n]$ (where $Q_0 = Q_x$ and $Q_n = Q_z$) so Theorem 11 implies that for some $i \in \{1,\ldots,n\}$, $[Q_{i-1}\ Q_y\ Q_i]$ or $Q_y = Q_i$ whence axiom I6 implies that $Q_y \in Q(b,\emptyset)$.* q.e.d.
> (Schutz [42, p. 11])

We represent this as $[Q_x\ Q_y\ Q_z] \Longrightarrow Q_y \in Q(b,\emptyset)$. Schutz' notation hides the assumption that $Q_i \in Q$ for all i, as stated in his exposé of the axioms [42, p. 9]; we do not need this assumption and therefore do not state it.

```
theorem (*13*) unreach_connected:
  assumes path_Q: "Q∈𝒫"
      and event_b: "b∉Q" "b∈ℰ"
      and unreach: "Q_x ∈ ∅ Q b" "Q_z ∈ ∅ Q b" "Q_x ≠ Q_z"
      and xyz: "[[Q_x Q_y Q_z]]"
    shows "Q_y ∈ ∅ Q b"
```

We follow Schutz' proof at the start, obtaining the chain from Axiom I6. This requires some bookkeeping: we deduce his notational convention $Q_i \in Q$ from the betweenness relation, as well as assuring Q_y is an event using our additional axiom `in_path_event`.

```
have in_Q: "Q_x∈Q ∧ Q_y∈Q ∧ Q_z∈Q"
  using betw_b_in_path path_Q unreach(1,2,3) unreach_on_path xyz
  by blast
hence event_y: "Q_y∈ℰ"
  using in_path_event path_Q by blast
obtain X f where X_def: "ch_by_ord f X"
  "f 0 = Q_x" "f (card X - 1) = Q_z"
  "(∀i∈{1 .. card X - 1}. (f i) ∈ 𝟘 Q b
    ∧ (∀Qy∈ℰ. [[(f (i - 1)) Qy (f i)]] ⟶ Qy ∈ 𝟘 Q b))"
  "short_ch X ⟶ Q_x ∈ X ∧ Q_z ∈ X
    ∧ (∀Q_y∈ℰ. [[Q_x Q_y Q_z]] ⟶ Q_y ∈ 𝟘 Q b)"
  using I6 assms unfolding fin_chain_def by blast
```

The next step is to make certain Q_x, Q_z are indeed the bounding events of the chain. It is only at this stage that we realised I6 had to be modified to account for the case of short chains, in which e.g. `f 0 = Q_x` does not imply `Q_x ∈ X`.

```
obtain N where "N=card X" "N≥2"
  using X_def(2,3) unreach(3) by fastforce
have in_X: "Q_x∈X ∧ Q_z∈X"
proof (cases)
  assume "N=2"
  show "Q_x ∈ X ∧ Q_z ∈ X"
    using X_def(1) X_def(5) ⟨N = 2⟩ ⟨N = card X⟩ short_ch_card_2
    by auto
next
  assume "N≠2" hence "N≥3" using ⟨2 ≤ N⟩ by auto
  show "Q_x ∈ X ∧ Q_z ∈ X" <proof>
qed
```

At this point, we split the proof into two layers of cases. The first is according to the cardinality of the chain (i.e. short chains and otherwise), to determine whether the index function `f` has meaning and must be used.

```
show ?thesis
proof (cases)
  assume "N=2"
  thus ?thesis
    using X_def(1,5) xyz ⟨N = card X⟩ event_y short_ch_card_2
    by auto
next
  assume "N≠2"
  have "N≥3" using ⟨N≠2⟩ ⟨2 ≤ N⟩ by auto
```

The second layer of case splitting occurs only in the case of `N≥3`, and is given in the fact `y_cases` on the first line below. Schutz absorbs this split into defining a new notation for *non-strict ordering* [42, p. 27], i.e. $[Q_{i-1}\ Q_y\ Q_i]$ or $Q_y = Q_i$. He then relies on his reader to consider both cases and to dispense with the (often degenerate) `Q_y∈X` case. Isabelle would not accept such an implicit approach, so this non-strict notation is not formalised yet; this would be more useful when one attempts a proof of Th. 12, where it condenses large intermediate propositions.

Now that we have dealt with short chains, it is also time to do as Schutz suggests, and use our `segmentation` theorem. It is here that the major flaw in his proof becomes obvious. Theorem 10 is stated with a *set of events* in the premise. Thus the chain that results is generic over this set. Similarly, Th. 11 on `segmentation` is stated in terms of a *set of events*, but again, the return format is a chain over these events. Schutz implicitly identifies the two, which is not *a priori* legitimate: In fact, we showed very early on during this project that any chain gives rise to a reversed alternative (`chain_sym`). The two chains may in fact be different orderings, and thus not equal as chains.

```
    have y_cases: "Q_y∈X ∨ Q_y∉X" by blast
    obtain S P1 P2 g a d where all_defs:
      "Q = ⋃ S ∪ P1 ∪ P2 ∪ X" "Ball S is_seg"
      "P1 ∩ P2 = {} ∧ (∀x∈S. x ∩ P1 = {} ∧ x ∩ P2 = {}
        ∧ (∀y∈S. x ≠ y ⟶ x ∩ y = {}))"
      "S=(if 2 < N then {s. ∃i<N - 1. s = segment (g i) (g (i + 1))}
        else {segment a d})"
      "[g[a .. d]X]" "P1 = prolongation d a" "P2 = prolongation a d"
      using path_Q fin_X ⟨N = card X⟩ X_in_Q ⟨2 ≤ N⟩
      using segmentation [where P=Q and Q=X and N=N]
      by blast
    show ?thesis using y_cases
    proof (rule disjE)
      assume "Q_y∈X" ...
```

We omit the proof for `Q_y∈X` above: if Q_y is an event of the chain, I6 immediately implies `Q_y ∈ ∅ Q b` (this is fact `X_def(4)`). To continue the proof, we need a way to identify the chains `f` and `g`. This led to our proof of a result that is rather interesting by itself. We give two statements below: a first one that is easy to read and think about, and a second one that is more precise and useful, but verbose.

```
lemma (in MinkowskiSpacetime) chain_unique_upto_rev:
  assumes "[f[a..c]X]" "[g[x..z]X]" "card X ≥ 3"
  shows "i < card X ⟶ (f i = g i ∨ f i = g (card X - i - 1))"

lemma (in MinkowskiSpacetime) chain_unique_upto_rev_cases:
  assumes ch_f: "[f[a..c]X]" and ch_g: "[g[x..z]X]"
      and card_X: "card X ≥ 3" and valid_index: "i < card X"
  shows "((a=x ∨ c=z) ⟶ (f i = g i))
        ∧((a=z ∨ c=x) ⟶ (f i = g (card X - i - 1)))"
```

Notice this lemma has to exclude two-element chains again, because no order exists within them. Alternatively, the result for short chains is trivial: any function that assigns one element to index 0 and the other to 1 can be replaced with the (unique) other assignment, without destroying any (trivial, since ternary) "ordering" of the chain.

It is now clear we need yet another case split: the one for equality `f=g` (implied by Schutz' proof) and one for `f` and `g` being reversed. We give the cases in the final fact below.

```
      assume "Q_y∉X"
      have "card X ≥ 3" using ⟨3≤N⟩ ⟨N = card X⟩ by blast
      have "[f[Q_x..Q_z]X]" <proof>
      have "a=Q_x ∧ d=Q_z ∨ a=Q_z ∧ d=Q_x"
        using ⟨card X ≥ 3⟩ all_defs(6) ⟨[f[Q_x..Q_z]X]⟩
        using chain_bounds_unique2
        by simp
```

We can obtain the index i that determines the element of the segmentation containing Q_y, and prove our goal in each of the cases given above. What follows is just a listing of the most salient statements of the remaining proof.

```
show ?thesis
proof -
  have "a = Q_x ∧ d = Q_z ⟹ Q_y ∈ ∅ Q b"
  proof -
    assume "a = Q_x ∧ d = Q_z" ...
    hence "[[(f(i-1)) Q_y (f i)]]"
      using chain_unique_upto_rev_cases <proof>
    thus "Q_y ∈ ∅ Q b"
      using X_def(4) ⟨i∈{1..(card X)-1}⟩ ⟨Q_y∈ℰ⟩ by blast
  qed

  moreover have "a = Q_z ∧ d = Q_x ⟹ Q_y ∈ ∅ Q b"
  proof -
    assume "a = Q_z ∧ d = Q_x" ...
    hence "[[(f(card X - i - 1)) Q_y (f (card X - i))]]"
      using chain_unique_upto_rev_cases abc_sym <proof>
    show "Q_y ∈ ∅ Q b" <proof>
  qed

  ultimately show ?thesis
    using ⟨a = Q_x ∧ d = Q_z ∨ a = Q_z ∧ d = Q_x⟩ ...
    by auto
qed
```

The completion of this proof demonstrates several benefits of mechanisation of formal mathematics. First, we have found a significant level of depth that was utterly missed by the original three-line proof in prose. Second, resolving this problem led to a proof of a result not found in the original text, namely `chain_unique_upto_rev`, which nonetheless is of similar form and necessity as many fundamental uniqueness theorems throughout mathematics. Thirdly, we were able to reconcile a follow-on result with a necessarily weaker version of a required theorem (`segmentation`), regaining faith in the soundness and scope of Schutz' undertaking.

Chapter 5

Concluding Remarks

We have obtained and clarified a formalisation in Isabelle/HOL of the axioms of Schutz [42] for Minkowski spacetime. Most of the axioms have been verified by using them in mechanised proofs, barring symmetry (which appears only later in Schutz' monograph) and continuity (which would come into the very next theorem to be mechanised). We believed it more useful to verify that a necessary weakening of Theorem 11 would not impact the further progression of Schutz' results. Overall, the only theorem where the prose proof could not be properly followed was Theorem 10. For the other proof mechanisations, it was often necessary to obtain additional lemmas or steps, but the broad structure is very similar. Case splits in Theorem 2 and the lemma `abc_abd_bcdbdc` had to be clarified, and the sequence of statements changed, but without altering the logical outline.

We plan on continuing work on formalising Schutz' third chapter, filling in the remaining Theorems 12 and 14. This would lend credence to our mechanisation of the axiom of continuity, used in proving Theorem 12. Several useful results about path dependence have been stated but left unproven, as this definition was not useful in proofs to date. Similarly, comparing with the discussion of Sec. 3.5, some further experiments with axioms and locales are in order. In particular, we would look for a way of asserting the existence of a number of paths without relying on the full axiom of dimension. Then results over a wide range of Schutz' monograph [42] could be proven in a weaker locale than `MinkowskiSpacetime`. This would almost certainly have implications for independence again. Also, we believe an equivalence proof for infinite total and local chains is possible. This has not been needed, because we have barely encountered infinite chains, but will be important in the same way `equiv_chain_3_fin` was (Sec. 4.4).

One very interesting property of this axiom system has not been explored at length. In Palmer's attempted proof of Theorem 6(ii), which introduces a binary order defined on the basis of betweenness, one needs to fix a basis of two elements. One would not be sufficient to fix a direction, yet the comparison to $<$ or $>$ on natural numbers requires one direction to be singled out. It is well-known that any poset gives rise to the dual poset , and we argue this duality is absorbed into the ternarity of betweenness. This symmetry (axiom O2) between two valid orders on the natural numbers, which we use to order our chains, may have practical consequences.

While we have obtained results such as `chain_sym` and `chain_unique_upto_rev`, and used them to enable or simplify proofs (Secs. 4.4 and 4.6), no study of their use has been undertaken. It may be possible to identify in general terms the properties of a predicate function P that are necessary for the proposition $P(f)$ to imply $P(g)$, where f is the indexing function to a chain X, and g its reverse. This would give a powerful meta-theorem, allowing us to avoid proof splits such as in Theorem 13 (Sec. 4.6). This work could be prepared by a similar theorem for betweenness using Axiom O2 (`abc_sym`), which is probably easier. This work would go in the direction of WLOG[1]-implementations [44, p.218], [18].

Apart from results of Schutz, we have proved in Isabelle multiple properties of different kinds of chains. This new focus on chains, and the interaction between local and total order on chains, adds a flavour to Schutz' theory that is closer to lattice- and order-theoretical approaches to foundational physics. While such ideas have long existed (e.g. Feynman's checkerboard [11]), one modern take is given by Knuth et al. [22]. Here one starts with just events (no paths) and a binary partial order, but adds the assumption that certain totally ordered subsets exist (which take the place of paths as inertial observers). Interestingly, this approach seems to include some quantum effects [21, 15]. Unreachable sets are equally comparable between these approaches. It is interesting that all of these theories single out inertial observers. A comparison with GR, perhaps using work on first-order axiom systems [1], appears fruitful.

[1] Without loss of generality: often cited in mathematical proof when invoking an obvious symmetry to generalise a proof of a single case to a class of cases (e.g. fixing an orientation, choosing a coordinate origin, ...)

Bibliography

[1] Hajnal Andréka, Judit Madarász, István Németi, and Gergely Székely. A logic road from special to general relativity. *Synthese*, 186:633–649, May 2012.

[2] Hajnal Andréka, Judit X. Madarász, István Németi, and Gergely Székely. An Axiom System for General Relativity Complete with respect to Lorentzian Manifolds. *arXiv:1310.1475 [gr-qc]*, October 2013.

[3] Hajnal Andréka, István Németi, Judit X. Madarász, and Gergely Székely. On Logical Analysis of Relativity Theories. *arXiv:1105.0885 [gr-qc, physics:math-ph]*, May 2011.

[4] Alain Bernard. The significance of Ptolemy's Almagest for its early readers. *Revue de Synthèse*, 131(4):495–521, December 2010.

[5] M. Born, W. Heisenberg, and P. Jordan. Zur Quantenmechanik. II. *Zeitschrift für Physik*, 35(8):557–615, August 1926.

[6] Lorenzo Cocco and Joshua Babic. A system of axioms for Minkowski spacetime. http://philsci-archive.pitt.edu/17669/, July 2020.

[7] Richard Dedekind. *Essays on the Theory of Numbers : I. Continuity and Irrational Numbers. II. The Nature and Meaning of Numbers*. Dover Publications, New York, 1963.

[8] A. Einstein. Zur Elektrodynamik bewegter Körper. *Annalen der Physik*, 322(10):891–921, 1905.

[9] A. Einstein and J. Laub. Über die elektromagnetischen Grundgleichungen für bewegte Körper. *Annalen der Physik*, 331(8):532–540, 1908.

[10] Albert Einstein. Die Feldgleichungen der Gravitation. *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften (Berlin), Seite 844-847.*, 1915.

[11] H. A. Gersch. Feynman's relativistic chessboard as an Ising Model. *International Journal of Theoretical Physics*, 20(7):491–501, July 1981.

[12] Robert Goldblatt. First-Order Spacetime Geometry. In Jens Erik Fenstad, Ivan T. Frolov, and Risto Hilpinen, editors, *Studies in Logic and the Foundations of Mathematics*, volume 126 of *Logic, Methodology and Philosophy of Science VIII*, pages 303–316. Elsevier, January 1989.

[13] Robert Goldblatt. *Orthogonality and Spacetime Geometry*. Springer Science & Business Media, December 2012.

[14] Michael Gordon, Robin Milner, and Christopher Wadsworth. Edinburgh LCF. A mechanised logic of computation. *Lecture Notes in Computer Science*, 78, 1979.

[15] Philip Goyal, Kevin H. Knuth, and John Skilling. Origin of complex quantum amplitudes and Feynman's rules. *Physical Review A*, 81(2):022109, February 2010.

[16] Adam Grabowski. Tarski's geometry modelled in Mizar computerized proof assistant. In *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 373–381, September 2016.

[17] Thomas Hales, Mark Adams, Gertrud Bauer, Dat Tat Dang, John Harrison, Truong Le Hoang, Cezary Kaliszyk, Victor Magron, Sean McLaughlin, Thang Tat Nguyen, Truong Quang Nguyen, Tobias Nipkow, Steven Obua, Joseph Pleso, Jason Rute, Alexey Solovyev, An Hoai Thi Ta, Trung Nam Tran, Diep Thi Trieu, Josef Urban, Ky Khac Vu, and Roland Zumkeller. A formal proof of the Kepler conjecture. *arXiv:1501.02155 [cs, math]*, January 2015.

[18] John Harrison. Without Loss of Generality. In Stefan Berghofer, Tobias Nipkow, Christian Urban, and Makarius Wenzel, editors, *Theorem Proving in Higher Order Logics*, volume 5674, pages 43–59. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[19] Thomas Little Heath. *The Thirteen Books of Euclid's Elements*. Courier Corporation, 1956.

[20] David Hilbert. *The Foundations of Geometry*. The Open Court Publishing Company, 1950.

[21] Kevin H. Knuth. Understanding the Electron. In Ian T. Durham and Dean Rickles, editors, *Information and Interaction: Eddington, Wheeler, and the Limits of Knowledge*, The Frontiers Collection, pages 181–207. Springer International Publishing, Cham, 2017.

[22] Kevin H. Knuth and Newshaw Bahreyni. A Potential Foundation for Emergent Space-Time. *Journal of Mathematical Physics*, 55(11):112501, November 2014.

[23] Ondřej Kunčar and Andrei Popescu. Comprehending Isabelle/HOL's Consistency. In Hongseok Yang, editor, *Programming Languages and Systems*, Lecture Notes in Computer Science, pages 724–749, Berlin, Heidelberg, 2017. Springer.

[24] Martin Liebeck. *A Concise Introduction to Pure Mathematics*. CRC Press, third edition, 2011.

[25] Nicolas Magaud, Julien Narboux, and Pascal Schreck. Formalizing Projective Plane Geometry in Coq. In Thomas Sturm and Christoph Zengler, editors, *Automated Deduction in Geometry*, Lecture Notes in Computer Science, pages 141–162, Berlin, Heidelberg, 2011. Springer.

[26] T. J. M. Makarios. *A Mechanical Verification of the Independence of Tarski's Euclidean Axiom*. Master's thesis, Victoria University of Wellington, 2012.

[27] Jiri Matousek and Bernd Gärtner. *Understanding and Using Linear Programming*. Springer Science & Business Media, July 2007.

[28] Laura I. Meikle and Jacques D. Fleuriot. Formalizing Hilbert's Grundlagen in Isabelle/Isar. In David Basin and Burkhart Wolff, editors, *Theorem Proving in Higher Order Logics*, Lecture Notes in Computer Science, pages 319–334, Berlin, Heidelberg, 2003. Springer.

[29] Herrman Minkowski. Die Grundgleichungen für die elektromagnetischen Vorgänge in bewegten Körpern. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, pages 53–111, 1908.

[30] Brent Mundy. Optical Axiomatization of Minkowski Space-Time Geometry. *Philosophy of Science*, 53(1):1–30, 1986.

[31] Brent Mundy. The Physical Content of Minkowski Geometry. *The British Journal for the Philosophy of Science*, 37(1):25–54, March 1986.

[32] Julien Narboux. Mechanical Theorem Proving in Tarski's Geometry. In Francisco Botana and Tomas Recio, editors, *Automated Deduction in Geometry*, Lecture Notes in Computer Science, pages 139–156, Berlin, Heidelberg, 2007. Springer.

[33] Tobias Nipkow. Linear Quantifier Elimination. *Journal of Automated Reasoning*, 45(2):189–212, August 2010.

[34] Jake Palmer. *Formal Axiomatisation of Minkowski Spacetime*. Master's thesis, School of Informatics, The University of Edinburgh, 2017.

[35] Jake Palmer and Jacques D Fleuriot. Mechanising an Independent Axiom System for Minkowski Space-time. In *Proceedings of the 12th International Conference on Automated Deduction in Geometry*, pages 64–79, 2018.

[36] Lawrence Paulson and Jasmin Blanchette. Three Years of Experience with Sledgehammer, a Practical Link between Automatic and Interactive Theorem Provers. In *International Workshop on the Implementation of Logics (IWIL-2010)*, 2010.

[37] Lawrence C. Paulson, Tobias Nipkow, and Makarius Wenzel. From LCF to Isabelle/HOL. *arXiv:1907.02836 [cs]*, July 2019.

[38] Alfred A. Robb. *Geometry of Time and Space*. Cambridge University Press, 1936.

[39] E. Schrödinger. An Undulatory Theory of the Mechanics of Atoms and Molecules. *Physical Review*, 28(6):1049–1070, December 1926.

[40] John W. Schutz. *Foundations of Special Relativity: Kinematic Axioms for Minkowski Space-Time*, volume 361 of *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1973.

[41] John W. Schutz. An axiomatic system for Minkowski space–time. *Journal of Mathematical Physics*, 22(2):293–302, February 1981.

[42] John W. Schutz. *Independent Axioms for Minkowski Space-Time*. CRC Press, October 1997.

[43] Dana S. Scott. A type-theoretical alternative to ISWIM, CUCH, OWHY. *Theoretical Computer Science*, 121(1):411–440, December 1993.

[44] Phil Scott. *Ordered Geometry in Hilbert's Grundlagen Der Geometrie*. PhD thesis, The University of Edinburgh, June 2015.

[45] Phil Scott and Jacques Fleuriot. An Investigation of Hilbert's Implicit Reasoning through Proof Discovery in Idle-Time. In Pascal Schreck, Julien Narboux, and Jürgen Richter-Gebert, editors, *Automated Deduction in Geometry*, Lecture Notes in Computer Science, pages 182–200, Berlin, Heidelberg, 2011. Springer.

[46] Raymond F. Streater and Arthur S. Wightman. *PCT, Spin and Statistics, and All That. Corr. 3rd Print. of the 1978 Ed.* Princeton University Press, Princeton, NJ, corr. 3rd print. of the 1978 ed. edition, 2000.

[47] Patrick Suppes. The Desirability of Formalization in Science. *The Journal of Philosophy*, 65(20):651–664, 1968.

[48] G. Szekeres. Kinematic geometry; an axiomatic system for Minkowski space-time: M. L. Urquhart in Memoriam. *Journal of the Australian Mathematical Society*, 8(2):134–160, May 1968.

[49] Alfred Tarski. What is Elementary Geometry? In Leon Henkin, Patrick Suppes, and Alfred Tarski, editors, *Studies in Logic and the Foundations of Mathematics*, volume 27 of *The Axiomatic Method*, pages 16–29. Elsevier, January 1959.

[50] Oswald Veblen. A system of axioms for geometry. *Transactions of the American mathematical society*, 5(3):343–384, 1904.

[51] A. G. Walker. Axioms for Cosmology. In Leon Henkin, Patrick Suppes, and Alfred Tarski, editors, *Studies in Logic and the Foundations of Mathematics*, volume 27 of *The Axiomatic Method*, pages 308–321. Elsevier, January 1959.

[52] Makarius Wenzel, Lawrence C. Paulson, and Tobias Nipkow. The Isabelle Framework. In Otmane Ait Mohamed, César Muñoz, and Sofiène Tahar, editors, *Theorem Proving in Higher Order Logics*, Lecture Notes in Computer Science, pages 33–38, Berlin, Heidelberg, 2008. Springer.

Appendix A

Original Text (Schutz, 1997)

All content in this appended chapter is taken from Schutz [42] for convenient reference.

A.1 Axioms of Order, Chains

Axiom O1
For events $a, b, c \in \mathcal{E}$,

$$[abc] \implies \exists Q \in \mathcal{P} : \ a, b, c \in Q \ .$$

Axiom O2
For events $a, b, c \in \mathcal{E}$,

$$[abc] \implies [cba] \ .$$

Axiom O3
For events $a, b, c \in \mathcal{E}$,

$$[abc] \implies a, b, c \ \textit{are distinct} \ .$$

Axiom O4
For distinct events $a, b, c, d \in \mathcal{E}$,

$$[abc] \ \textit{and} \ [bcd] \implies [abd] \ .$$

Axiom O5
For any path $Q \in \mathcal{P}$ *and any three distinct events* $a, b, c \in Q$,

$$[abc] \ \textit{or} \ [bca] \ \textit{or} \ [cab] \ \textit{or}$$
$$[cba] \ \textit{or} \ [acb] \ \textit{or} \ [bac] \ .$$

10

The axiom of collinearity (Axiom O6) This axiom is a kinematic analogue of the geometric axiom of plane order given by Veblen (1904, 1911) and Moore (1908) and makes it possible to discuss "rectilinear motion" in terms of "collinear sets" of events and paths (as in Figure 1 which appears after the statement of the axiom). If a template is made by cutting a narrow slit in a sheet of paper, the paths may be observed "in motion" by moving the template gradually across the diagram.

A strict analogue of the geometric "Axiom of Pasch" (as stated by Veblen (1904) and Moore (1908))[5] would result in a non-independent system of axioms, so we state the corresponding Axiom O6 in terms of the concept of a "finite chain" in order to have an independent system of axioms. Accordingly we make the definition:
A sequence of events

$$Q_0,\ Q_1,\ Q_2,\ \cdots$$

(of a path Q) is called a *chain* if:

(i) it has two distinct events, or

(ii) it has more than two distinct events and for all $i \geq 2$,

$$[Q_{i-2}\ Q_{i-1}\ Q_i]\ .$$

A *finite chain* is denoted by writing $[Q_0\ Q_1\ Q_2\ \cdots\ Q_n]$ and an *infinite chain* is denoted by writing $[Q_0\ Q_1\ Q_2\ \cdots]$ (note that the concept of a "chain" used by Veblen and Young (1908) for the discussion of projective geometries is entirely different from the concept defined above). Sometimes, for ease of reading, we will denote a chain or a relation of betweeness with commas to separate the events; thus, for example $[a, b, c]$ has the same meaning as $[abc]$. It will transpire (as a consequence of Theorem 1) that the concept of betweenness applies to any appropriately ordered triple of events of a finite chain, but note that this property is not being postulated in the axioms.

Axiom O6

If Q, R, S are distinct paths which meet at events $a \in Q \cap R$, $b \in Q \cap S$, $c \in R \cap S$ and if:

(i) *there is an event $d \in S$ such that $[bcd]$, and*

(ii) *there is an event $e \in R$ and a path T which passes through both d and e such that $[cea]$,*

then T meets Q in an event f which belongs to a finite chain $[a \cdot\cdot f \cdot\cdot b]$.

A.2 Axioms of Incidence, Path Dependence, SPRAYs, Unreachable Sets

Axiom I1 (Existence)
$\mathcal{E}$ is not empty.

Axiom I2 (Connectedness)
For any two distinct events $a, b \in \mathcal{E}$ there are paths R, S such that $a \in R$, $b \in S$ and $R \cap S \neq \emptyset$.

xiom I3 (Uniqueness)
'or any two distinct events, there is at most one path which contains both of them.

In the subsequent development we will frequently be discussing the properties of sets of paths which meet at a given event. We will call any such set a *SPRAY of paths*, or more concisely a *SPRAY*, where the upper case letters indicate that we are referring to a set of paths rather than to a set of events: given any event x, we define

$$SPR[x] := \{R:\ R \ni x,\ R \in \mathcal{P}\}\ .$$

The corresponding set of events is called a *spray*, where the lower case letters indicate a set of events. We define

$$spr[x] := \{R_y:\ R_y \in R,\ R \in SPR[x]\}\ .$$

A subset of three paths $\{Q, R, S\}$ of a SPRAY is *dependent* if there is a path which does not belong to the SPRAY and which contains one event from each of the three paths: we also say that any one of the three paths is *dependent on* the other two. Otherwise the subset is *independent*.

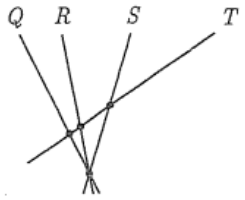

Figure 2 The subset $\{Q, R, S\}$ is dependent

We next give recursive definitions of dependence and independence which will be used to characterize the concept of dimension. A path T is *dependent on* the set of n paths (where $n \geq 3$)

$$\mathcal{S} = \{Q^{(i)} : \ i = 1, 2, \ldots, n; \ Q^{(i)} \in SPR[x]\}$$

if it is dependent on two paths $S^{(1)}$ and $S^{(2)}$, where each of these two paths is dependent on some subset of $n-1$ paths from the set $\mathcal{S}$. We also say that the set of $n+1$ paths $\mathcal{S} \cup \{T\}$ is a *dependent set*. If a set of paths has no dependent subset, we say that the set of paths is an *independent set*.

We now make the following definition which enables us to specify the dimension of Minkowski space–time. A SPRAY is a *3-SPRAY* if:

(i) it contains four independent paths, and

(ii) all paths of the SPRAY are dependent on these four paths.

Axiom I4 (Dimension)
If $\mathcal{E}$ is non–empty, then there is at least one 3-SPRAY.

Given a path Q and an event $b \notin Q$, we define the *unreachable subset of Q from b* to be

$$Q(b, \emptyset) := \{x : \text{ there is no path which contains } b \text{ and } x, \ x \in Q\} .$$

That is, the unreachable subset of Q from b is the subset of events of Q which can not be joined to b by a single path. If two events can not be connected by any path, we say that each is *unreachable* from the other; otherwise each is *reachable* from the other.

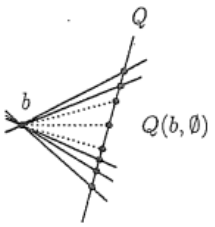

Figure 3 Pairs of events which can not be joined by a path will be indicated by a dotted line between them

Axiom I5 (Non-Galilean)
For any path Q and any event $b \notin Q$, the unreachable set $Q(b, \emptyset)$ contains (at least) two events.

Axiom I6 (Connectedness of the Unreachable Set)
Given any path Q, any event $b \notin Q$, and distinct events $Q_x, Q_z \in Q(b, \emptyset)$, there is a finite chain $[Q_0 \cdots Q_n]$ with $Q_0 = Q_x$ and $Q_n = Q_z$ such that for all $i \in \{1, 2, \ldots, n\}$,

$$\text{(i)} \quad Q_i \in Q(b, \emptyset)$$
$$\text{(ii)} \quad [\, Q_{i-1}\ Q_y\ Q_i \,] \Longrightarrow Q_y \in Q(b, \emptyset) \ .$$

Axiom I7 (Boundedness of the Unreachable Set)
Given any path Q, any event $b \notin Q$ and events $Q_x \in Q \setminus Q(b, \emptyset)$ and $Q_y \in Q(b, \emptyset)$, there is a finite chain

$$[\, Q_0 \cdots Q_m \cdots Q_n \,]$$

with $Q_0 = Q_x$, $Q_m = Q_y$ *and* $Q_n \in Q \setminus Q(b, \emptyset)$.

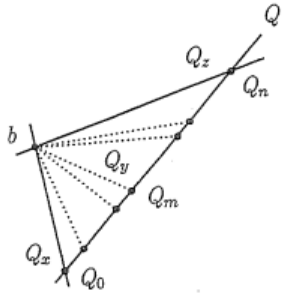

Figure 4 Boundedness of the Unreachable Set (Axiom I7)

A.3 Axiom of Symmetry, Unreachable Set via a Path

2.4 Axiom of isotropy or symmetry

Compared to the absolute geometries, Minkowski space–time has the additional structure provided by the existence and properties of unreachable sets (Axioms I5, I6, I7). These properties, together with the single property of isotropy of the following axiom, are sufficient to take the place of all the axioms of congruence and the axiom of uniqueness of parallels used by Hilbert (1899), Moore (1908) and Veblen (1911) for Euclidean geometry.

For any two distinct paths Q, R which meet at an event x, we define the *unreachable subset of Q from Q_a via R* to be

$$Q(Q_a, R, x, \emptyset) := \{Q_y : [x\ Q_y\ Q_a] \text{ and } \exists R_w \in R \text{ such that } Q_a, Q_y \in Q(R_w, \emptyset)\ \}\ .$$

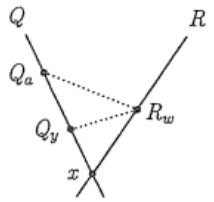

Figure 5 There is no path joining Q_a to R_w and there is no path joining Q_y to R_w

Axiom S (Symmetry or Isotropy)
If Q, R, S are distinct paths which meet at some event x and if $Q_a \in Q$ is an event distinct from x such that

$$Q(Q_a, R, x, \emptyset) = Q(Q_a, S, x, \emptyset)$$

then

(i) *there is a mapping $\theta : \mathcal{E} \longrightarrow \mathcal{E}$*
(ii) *which induces a bijection $\Theta : \mathcal{P} \longrightarrow \mathcal{P}$*

such that

(iii) *the events of Q are invariant, and*
(iv) $\Theta : R \longrightarrow S$.

The mapping θ is called a *symmetry mapping* or an *isotropy mapping*, with Q as the *invariant path*[6,7].

A.4 Axiom of Continuity, Bounds

2.5 Axiom of continuity

This final axiom resembles the geometric axiom of the same name in the axiom systems of Hilbert (1899), Veblen (1904, 1911) and Moore (1908).

Given a path $Q \in \mathcal{P}$ and an infinite chain $[\ Q_0\ Q_1\ \cdots\ ]$ of events in Q, the set

$$\mathcal{B} = \{Q_b :\ i < j \Rightarrow [Q_i\ Q_j\ Q_b];\ Q_i, Q_j, Q_b \in Q\}$$

is called the *set of bounds* of the chain: if $\mathcal{B}$ is non–empty we say that the chain is *bounded*. If there is a bound $Q_b \in \mathcal{B}$ such that for all $Q_{b'} \in \mathcal{B} \setminus \{Q_b\}$,

$$[Q_0\ Q_b\ Q_{b'}]$$

we say that Q_b is a *closest bound*.

Axiom C (Continuity)
Any bounded infinite chain has a closest bound.

A.5 Overlapping Ordering Lemma

Lemma 1 *If* $[abc]$ *and* $[abd]$ *and* $c \neq d$ *then either* $[bcd]$ *or* $[bdc]$.

Proof (based on Veblen (1904), Lemma 2, p.357). By Theorem 1, it is sufficient to show that the supposition $[dbc]$ leads to a contradiction.

By the Existence Theorem 5 there is a path S distinct from ab passing through a (see Figure 7, p.24). Axiom I5 together with Theorem 4 imply the existence of an event $e \in S \setminus \{a\}$ and a path be. If there is a path de we let $d^* := d$. Otherwise the Boundedness of the Unreachable Set (Th.4) implies the existence of both an event $d^* \in ab$ and a path d^*e such that $[bdd^*]$, which with $[abd]$ implies $[abd^*]$ by Axiom O4.

Similarly if there is a path ce we let $c^* := c$. Otherwise there is an event $c^* \in ab$ and a path c^*e such that $[bcc^*]$ which with $[abc]$ (and $[dbc]$) implies $[abc^*]$ (and $[dbc^*]$).

By Theorem 1 $[abc^*]$ implies $[c^*ba]$ and $[dbc^*]$ implies $[c^*bd]$ so, in the case where $d^* \neq d$ we have $[bdd^*]$ so Axiom O4 implies $[c^*bd^*]$, and in the case where $d^* = d$, we also have $[c^*bd^*]$ (from $[c^*bd]$).

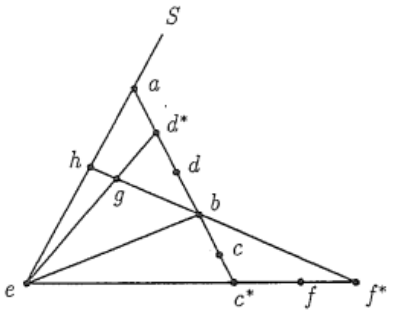

Figure 7

By the Prolongation Theorem (Th.6) there is an event $f \in ec^*$ such that $[ec^*f]$. If there is a path bf we let $f^* := f$. Otherwise the Boundedness of the Unreachable Set (Theorem 4) implies the existence of an event $f^* \in ec^*$ and a path bf^* such that $[c^*ff^*]$, which with $[ec^*f]$ implies $[ec^*f^*]$ by Axiom O4. Thus for both cases there is an event $f^* \in ec^*$ and a path bf^* such that $[ec^*f^*]$.

The remainder of this proof is based on a proof of Veblen (Veblen (1904), Lemma 2, p357, with the symbols e, f^*, g, h taking the place of Veblen's symbols O, P, Q, R respectively). For the kinematic triangle $\triangle aec^*$ with $[ec^*f^*]$ and $[c^*ba]$, the Second Collinearity Theorem (Th.7) implies that f^*b meets ae in an event h such that $[ahe]$ and $[f^*bh]$, while the same theorem applied to the kinematic triangle $\triangle d^*ec^*$ with $[ec^*f^*]$ and $[c^*bd^*]$ implies that f^*b meets d^*e in an event g such that $[d^*ge]$ and $[f^*bg]$. Since both g and h lie on f^*b, it follows that b, g, h belong to the same path: however b, g, h are events of segments of the kinematic triangle $\triangle ead^*$ since $[abd^*]$, $[d^*ge]$ and $[ahe]$. This contradicts the result of Theorem 8, and completes the proof of Lemma 1.

A.6 Theorem 2 (Proof only)

Proof (i) We first show that

$$[abc] \quad \text{and} \quad [bcd] \Longrightarrow [acd] \tag{1}$$

Axiom O2 implies that $[dcb]$ and $[cba]$, so by Axiom O4 it follows that $[dca]$, whence Axiom O2 implies $[acd]$.

(ii) For convenience in this proof we will omit the symbol "Q" and simply indicate the suffices. The integers i, j, k, l will satisfy the order relations

$$0 \leq i < j < n \quad \text{and} \quad 0 < k < l \leq n$$

We define

$$P(i, j-1) := [i, \overline{j-1}, j]$$

Now for a finite chain of $n+1$ events we have $[\overline{j-1}, j, \overline{j+1}]$, so by part (i) above

$$P(i, j-1) \Longrightarrow P(i, j)\,.$$

Furthermore, by definition of a chain, we have $P(i, i+1)$, so by induction we have, for all i and j,

$$i < j \Longrightarrow P(i, j) = [i, j, \overline{j+1}]\,. \tag{2}$$

Then as in the preceding paragraph, but with induction for a decreasing integer sequence, we have

$$k < l \Longrightarrow [\overline{k-1}, k, l]\,. \tag{3}$$

If we now let $k-1 = j$ then we have both $[i, j, j+1]$ and $[j, j+1, l]$ whence Axiom O4 implies $[i, j, l]$ for $i < j < l-1$. The case where $j = l-1$ has already been established in (2).

The first part of the theorem has now been established. Axiom O3 now implies that all events are distinct. *q.e.d.*

A.7 Theorem 10 (Proof only)

Proof (By induction). The previous theorem applies to the case where $n = 4$. We will make the inductive hypothesis that the result applies to a set of n distinct events $\{a_1, a_2, \ldots, a_n\}$ and demonstrate that this implies the result for the case of $n+1$ distinct events. We denote the $(n+1)$–th event as b. Then Axiom O5 implies that either:

(i) $[ba_1a_n]$ or (ii) $[a_1ba_n]$ or (iii) $[a_1a_nb]$

Case (i): By the inductive hypothesis and Theorem 2 we have $[a_1a_2a_n]$ so the previous theorem (Th.9) implies that $[ba_1a_2a_n]$ which implies that $[ba_1a_2]$. Thus b is an element of a chain $[a_1^*a_2^*\ldots a_{n+1}^*]$ where $a_1^* := b$ and (for $j \in \{2, \ldots, n+1\}$) $a_j^* := a_{j-1}$.

Case (ii): Let k be the smallest integer such that $[a_1ba_k]$. Then the previous theorem (Th.9) implies either that $[a_1a_{k-1}ba_k]$, or that $k = 2$ so that $[a_{k-1}ba_k]$. If $k-2 \geq 1$ we have $[a_{k-2}a_{k-1}a_k]$ which with $[a_{k-1}ba_k]$ implies $[a_{k-2}a_{k-1}ba_k]$ by the previous theorem, while if $k+1 \leq n$ we have $[a_{k-1}a_ka_{k+1}]$ which with $[a_{k-1}ba_k]$ implies $[a_{k-1}ba_ka_{k+1}]$; that is we have now shown that $[a_{k-2}a_{k-1}b]$ (if $k-2 \geq 1$) and $[a_{k-1}ba_k]$ and $[ba_ka_{k+1}]$ (if $k+1 \leq n$) so that b is an element of a chain $[a_1^*a_2^*\ldots a_{n+1}^*]$ where

$$a_j^* = \begin{cases} a_j, & j \leq k-1 \\ b, & j = k \\ a_{j-1}, & j > k\ . \end{cases}$$

Case (iii): The proof for this case is similar to that for Case (i).

Thus in each case the inductive hypothesis for the case of n distinct events implies the result for the case of $n+1$ distinct events and the result has been established in the previous theorem for the case $n = 4$: the proof is now complete. *q.e.d.*

Appendix B

Additional Proofs and Listings

B.1 Symmetry Axiom: Bijectivity and Totality of Functions

We first proposed a version with a different statement for bijectivity of Θ:

```
bij (λP. {z. y∈P ∧ (z = θ y)})
∧ (T∈𝒫 ⟶ (λP. {z. y∈P ∧ (z = θ y)}) T ∈𝒫)"
```

The current version fixes the image of Θ explicitly to all of the set of paths $\mathcal{P}$, rather than possibly only a subset, which we want. On the other hand, the version just above requires Θ to be a bijection everywhere, not just on $\mathcal{P}$ – a strong but unnecessary property. This example makes clear a danger in the (typed) λ calculus: Schutz' functions are defined on the set of events (θ), and that of paths (θ). Our functions are defined on the set of all the things that have the same type as events (θ), and the set of all sets of those things (θ). The former may pose minor problems depending on what model one works in, but it is easy to think up models where no problem arises (e.g. $\mathbb{R}^4$). Notice categoricalness relieves this problem once all axioms are considered together: no model other than $\mathbb{R}^4$ exists.

Even if we consider only $\mathbb{R}^4$, defining Θ is more of a challenge: any Θ we define must be defined for sets of events that we have no real interest in. One way to do that would be to define the desired Θ on $\mathcal{P}$, and set it to be the identity everywhere else, to avoid interaction between $\mathcal{P}$ and $2^{\mathcal{E}} \setminus \mathcal{P}$.

B.2 Paths are non-empty

A proof listing for `no_empty_paths`, which relies on I1 and I5 to show that no path can be the empty set.

```
lemma no_empty_paths [simp]:
  assumes "Q∈𝒫"
  shows "Q≠{}"
proof -
  obtain a where "a∈ℰ" using nonempty_events by blast
  have "a∈Q ∨ a∉Q" by auto
  thus ?thesis
  proof
    assume "a∈Q"
    thus ?thesis by blast
  next
    assume "a∉Q"
    then obtain b where "b∈∅ Q a"
      using two_in_unreach ⟨a ∈ ℰ⟩ assms
      by blast
    thus ?thesis
      using unreachable_subset_def by auto
  qed
qed
```

B.3 Theorem 5

Schutz states Th. 5 as follows:

> **Theorem 5** *(First Existence Theorem)*
> *Given a path Q and an event $a \in Q$, there is*
>
> *(i) an event $b \in Q$ with b distinct from a, and*
>
> *(ii) an event $c \notin Q$ and a path ac (distinct from Q).*
>
> (Schutz 1997 [42, p. 20])

Palmer's mechanisation splits this theorem into two parts, one of which is given two alternative forms; `ex_crossing_at` is the one used in the main text.

```
theorem (*5i*) ge2_events:
  assumes path_Q: "Q ∈ 𝒫"
      and a_inQ: "a ∈ Q"
  shows "∃b∈Q. b ≠ a"

theorem (*5ii*) ex_crossing_at:
  assumes path_Q: "Q ∈ 𝒫"
      and a_inQ: "a ∈ Q"
  shows "∃ac∈𝒫. ac ≠ Q ∧ (∃c. c ∉ Q ∧ a ∈ ac ∧ c ∈ ac)"
```

```
lemma (*5ii_alt*) ex_crossing_at_alt:
  assumes path_Q: "Q ∈ 𝒫"
      and a_inQ: "a ∈ Q"
    shows "∃ac. ∃c. path ac a c ∧ ac ≠ Q ∧ c ∉ Q"
```

B.4 Theorem 6ii: Details

We proceed with the proof of `path_card_nil`. As Schutz states, the case `card P = 1` is excluded by theorem 5, which is proven by Palmer. By Theorem 6(i), two-element paths are impossible.

```
lemma path_card_nil:
  assumes path_Q: "Q∈𝒫"
    shows "∀n::nat.  n≥1 ⟶ card Q ≠ n"
proof
  fix n::nat
  show "n≥1 ⟶ card Q ≠ n"
  proof
    assume "n≥1"
    have "n=1 ∨ n=2 ∨ n≥3"
      using ⟨1 ≤ n⟩ by linarith
    thus "card Q ≠ n" <proof>
    proof (rule disjE3)
      assume "n=1" ...
    next
      assume "n=2" ...
```

The third case, `n≥3`, is the crux of the matter, and where induction is needed. Schutz seems to envisage it as an informal constructive proof, i.e. two-element paths must have three elements, three-element paths must have four elements, etc.Isabelle's induction requires a more analytical approach, to apply to any cardinality. What we need is a way to always pick two elements of a finite event-set on a path so that all other events of the set are between our picks. Then their prolongation via theorem 6(i) must be outside the set, which gives us our contradiction. This way of picking we call `obtain_fin_path_ends`, and it is the proof of this that requires induction.

```
    next
      assume "n≥3"
      show "card Q ≠ n"
      proof
        assume asm: "card Q = n"
        ...
        then obtain a b where "a∈Q" and "b∈Q" and "a≠b"
            and acb: "∀c∈Q. (c≠a ∧ c≠b) ⟶ [[a c b]]"
          using obtain_fin_path_ends card_Q fin_Q path_Q
          by metis
        then obtain x where "[[a b x]]" and "x∈Q"
          using prolong_betw2 path_Q by blast
```

```
        thus False
          by (metis acb abc_abc_neq abc_only_cba)
      qed
    qed
```

The result `obtain_fin_path_ends` is a convenient wrapper for the corresponding existential statement `finite_path_has_ends`. We will only briefly outline the proof, see Sec.2.3 for a discussion of induction and a demonstration of the proof schema.

```
lemma finite_path_has_ends:
  assumes path_X: "X∈𝒫" and min_n: "n≥0"
    shows "∀Q⊆X. finite Q ∧ card Q = n+3
      ⟶ (∃a∈Q. ∃b∈Q. a≠b ∧ (∀c∈Q. (a≠c ∧ b≠c) ⟶ [[a c b]]))"
proof (induct n)
```

The base case is trivial: the cardinality of Q is known to be 3, so we name all elements. They are on a path, so must be in some ordering relation by axiom O5, `some_betw`, and in each case we can identify the two events that have the third between them as a witness pair for the existential proposition.

The induction step's logic is developed again in Sec. 2.3. Once a single element x is removed from Q, the IH applies to $Q \setminus x$, so two 'edge' events a, b can be found for it. Now split according to the three possible orderings of a, b, x as in the base case. In either case, the 'edge' events of the chosen `betw` proposition qualify as a witness pair.

B.5 Details for abc_abd_bcdbdc

We begin here by listing `unreachable_bounded_path`.

```
lemma unreachable_bounded_path_only:
  assumes d'_def: "d'∉ ∅ ab e" "d'∈ab" "d'≠e"
      and e_event: "e ∈ ℰ"
      and path_ab: "ab ∈ 𝒫"
      and e_notin_S: "e ∉ ab"
  shows "∃d'e. path d'e d' e"
proof (rule ccontr)
  assume "¬(∃d'e. path d'e d' e)"
  hence "¬(∃R∈𝒫. d'∈R ∧ e∈R ∧ d'≠e)"
    by blast
  hence "¬(∃R∈𝒫.  e∈R ∧ d'∈R)"
    using d'_def(3) by blast
  moreover have "ab∈𝒫 ∧ e∈ℰ ∧ e∉ab"
    by (simp add: e_event e_notin_S path_ab)
  ultimately have "d'∈ ∅ ab e"
    unfolding unreachable_subset_def using d'_def(2)
    by blast
  thus False
    using d'_def(1) by auto
qed
```

```
(* in Schutz' proof of 3.6 – Lemma 1, the d' we obtain is d* *)
lemma unreachable_bounded_path:
  assumes S_neq_ab: "S ≠ ab"
      and a_inS: "a ∈ S"
      and e_inS: "e ∈ S"
      and e_neq_a: "e ≠ a"
      and path_S: "S ∈ 𝒫"
      and path_ab: "path ab a b"
      and path_be: "path be b e"
      and no_de: "¬(∃de. path de d e)"
      and abd:"[[a b d]]"
    obtains d' d'e where "d'∈ab ∧ path d'e d' e ∧ [[b d d']]"
proof -
  have e_event: "e∈ℰ"
    using e_inS path_S by auto
  have "e∉ab"
    using S_neq_ab a_inS e_inS e_neq_a eq_paths path_S path_ab by
    auto
  have "ab∈𝒫 ∧ e∉ab"
    using S_neq_ab a_inS e_inS e_neq_a eq_paths path_S path_ab
    by auto
  have "b ∈ ab - ∅ ab e"
    using cross_in_reachable path_ab path_be
    by blast
  have "d ∈ ∅ ab e"
    using no_de abd path_ab e_event ⟨e∉ab⟩
      betw_c_in_path unreachable_bounded_path_only
    by blast
  have  "∃d' d'e. d'∈ab ∧ path d'e d' e ∧ [[b d d']]"
  proof -
    obtain d' where "[[b d d']]" "d'∈ab" "d'∉ ∅ ab e" "b≠d'" "e≠d'"
      using unreachable_set_bounded ⟨b ∈ ab - ∅ ab e⟩ ⟨d ∈ ∅ ab e⟩
    e_event ⟨e∉ab⟩ path_ab
      by (metis DiffE)
    then obtain d'e where "path d'e d' e"
      using unreachable_bounded_path_only e_event ⟨e∉ab⟩ path_ab
      by blast
    thus ?thesis
      using ⟨[[b d d']]⟩ ⟨d' ∈ ab⟩
      by blast
  qed
  thus ?thesis
    using that by blast
qed
```

The next full listing is of `exist_f'`, which is similar to the `exist_c'd'` outlined in the main text.

```
lemma exist_f':
  assumes path_ab: "path ab a b"
      and path_S: "S ∈ 𝒫"
      and a_inS: "a ∈ S"
      and e_inS: "e ∈ S"
      and e_neq_a: "e ≠ a"
      and f_def: "[[e c' f]]" "f∈c'e"
      and S_neq_ab: "S ≠ ab"
```

```
      and c'd'_def: "c'∈ab ∧ d'∈ab
            ∧ [[a b d']] ∧ [[c' b a]] ∧ [[c' b d']]
            ∧ path d'e d' e ∧ path c'e c' e"
    shows "∃f'. ∃f'b. [[e c' f']] ∧ path f'b f' b"
proof (cases)
  assume "∃bf. path bf b f"
  thus ?thesis
    using ⟨[[e c' f]]⟩ by blast
next
  assume "¬(∃bf. path bf b f)"
  hence "f ∈ ∅ c'e b"
    by (metis S_neq_ab ⟨f ∈ c'e⟩ a_inS abc_abc_neq betw_events c'd'
    _def e_inS e_neq_a eq_paths
        path_S path_ab unreachable_bounded_path_only)
  moreover have "c' ∈ c'e - ∅ c'e b"
    using c'd'_def cross_in_reachable path_ab by blast
  moreover have "b∈ℰ ∧ b∉c'e"
    using ⟨f ∈ ∅ c'e b⟩ betw_events c'd'_def same_empty_unreach by
    auto
  ultimately obtain f' where f'_def: "[[c' f f']]" "f'∈c'e" "f'∉ ∅ c
    'e b" "c'≠f'" "b≠f'"
    using unreachable_set_bounded c'd'_def
    by (metis DiffE)
  hence "[[e c' f']]"
    using ⟨[[e c' f]]⟩ by blast
  moreover obtain f'b where "path f'b f' b"
    using ⟨b ∈ ℰ ∧ b ∉ c'e⟩ c'd'_def f'_def(2,3)
    unreachable_bounded_path_only
    by blast
  ultimately show ?thesis by blast
qed
```

Now we continue the proof of the main text. Using the lemmas `exist_c'd'` and `exist_f'`, we obtain `c'`, `d'`, `f'` (and `f`, more easily). The only major extra result required is theorem 7, the second collinearity theorem, which was mechanised by Palmer [34]:

```
theorem (*7*) (in MinkowskiChain) collinearity2:
  assumes tri_abc: "△ a b c"
      and bcd: "[[b c d]]" and cea: "[[c e a]]"
      and path_de: "path de d e"
  shows "∃f∈de. [[a f b]] ∧ [[d e f]]"
```

From here on, the proof follows Schutz, who in turn follows Veblen [50, p.357]. This part of the script closely follows the incomplete proof left to us by Palmer.

```
    obtain ae where path_ae: "path ae a e"
      using a_inS e_inS e_neq_a path_S by blast
    have tri_aec: "△ a e c'" ...
    then obtain h where h_in_f'b: "h ∈ f'b"
                    and ahe: "[[a h e]]"
                    and f'bh: "[[f' b h]]"
      using collinearity2 [where a = a and b = e and c = c'
        and d = f' and e = b and de = f'b]
```

```
      using f'_def c'd'_def f'_def by blast
    have tri_dec: "△ d' e c'" ...
    then obtain g where g_in_f'b: "g ∈ f'b"
                    and d'ge: "[[d' g e]]"
                    and f'bg: "[[f' b g]]"
      using collinearity2 [where a = d' and b = e and c = c'
        and d = f' and e = b and de = f'b]
      using f'_def c'd'_def by blast
```

After obtaining the events and paths proved to exist in `exist_c'd'`, `exist_f'`, we follow Schutz through the construction of multiple kinematic triangles (`tri_aec`, `tri_dec` , and `△ e a d'`). We can now construct the two events we need for our contradiction using Theorem 8, `g` and `h`. Their defining statements place them on different sides of the triangle `△ e a d'`, with `b` on the third side (`[[a b d']]` in the conclusion of `exist_c 'd'`) and theorem 8 is our formal way of stating that this implies they cannot all be aligned – contradicting the fact that they all lie on the path `f'b` (`h_in_f'b`, `g_in_f'b` and `b` is in `f'b` by definition of the latter).

```
    have "△ e a d'" ...
    thus False
      using tri_betw_no_path ...
      by blast
```

B.6 Theorem 2 for Local Chains

Schutz begins by proving a lemma about overlapping orderings. It is essentially an alternative to axiom O4, and we extract it into a lemma for future use. The proof is simple and follows Schutz precisely, using axioms O2 (symmetry) and O4. It is omitted from the main text for brevity and coherence.

```
lemma abc_bcd_acd:
  assumes abc: "[[a b c]]"
      and bcd: "[[b c d]]"
  shows "[[a c d]]"
proof -
  have cba: "[[c b a]]" using abc_sym abc by simp
  have dcb: "[[d c b]]" using abc_sym bcd by simp
  have "[[d c a]]" using abc_bcd_abd dcb cba by blast
  thus ?thesis using abc_sym by simp
qed
```

Schutz' prose for this proof is somewhat confusing (original text in App. A.6). He identifies indices with their chain elements, while our chain indexing functions are explicit, and have to be used explicitly. Furthermore, he indicates proofs by induction and case splits spontaneously, without relying on any sort of organised schema, so working out the formal setting is left to us.

We absorb both of his subproofs by induction into helper lemmas `thm2_ind1`, `thm2_ind2b`, which we shall discuss later. They provide proof that indices i, j, l satisfying the theorem assumptions, and $0 < k < l$, obey the ordering relations $[Q_i\ Q_j\ Q_{j+1}]$ and $[Q_{k-1}\ Q_k\ Q_l]$.

```
theorem (*2*) order_finite_chain2:
  assumes chX: "long_ch_by_ord2 f X"
      and finiteX: "finite X"
      and ordered_nats: "0 ≤ (i::nat) ∧ i < j ∧ j < l ∧ l < card X"
    shows "[[(f i) (f j) (f l)]]"
proof - ...
  have e2: "[[(f i) (f j) (f (j+1))]]"
    using thm2_ind1 Suc_eq_plus1 chX finiteX ord1
    by presburger
  have e3: "∀k. 0<k ∧ k<l ⟶ [[(f (k-1)) (f k) (f l)]]"
    using thm2_ind2b chX finiteX ordered_nats
    by blast
```

The case split we mentioned is for the two possible situations of a non-strict ordering. We will come back to such a non-strict ordering split when we discuss theorem 13 (Sec. 4.6). Here, we just distinguish between $j = l - 1$, when the fact `e2` applies, and the strict version $j < l - 1$, when both `e2` and `e3` are needed.

```
  have "j<l-1 ∨ j=l-1"
    using ordered_nats by linarith
  thus ?thesis
  proof
    assume "j<l-1"
    have  "[[(f j) (f (j+1)) (f l)]]"
      using e3 abc_abc_neq ordered_nats
      using ⟨j < l - 1⟩ less_diff_conv by auto
    thus ?thesis
      using e2 abc_bcd_abd
      by blast
  next
    assume "j=l-1"
    thus ?thesis using e2
      using ordered_nats by auto
  qed
qed
```

We examine only the lemma `thm2_ind1` in any detail. Note the base case $j = 0$ is trivial: there is no natural number i for which $i < j = 0$. We omit it below.

```
lemma thm2_ind1:
  assumes chX: "long_ch_by_ord2 f X"
      and finiteX: "finite X"
    shows "∀j i. ((i::nat) < j ∧ j < card X - 1) ⟶ [[(f i) (f j) (
   f (j + 1))]]"
proof
  ...
    case (Suc j)
    show ?case
```

In the induction step, we employ the same split of a nonstrict ordering ($i \leq j$) into equality ($i = j$) and strict ordering ($i < j$) that we saw in `order_finite_chain2`. We only use the induction hypothesis `Suc.hyps` in the strict case. The case $i = j$ is again trivial: translating the `Suc` function to numerals, the goal is $[f(i)\ f(i+1)\ f(i+2)]$, which holds by definition of local chains. The remaining case $i < j$ is, at the top level, reduced to (decidable) linear arithmetic. Instead of using general solvers such as `blast` or `metis`, these statements are proven using the decision procedures `presburger` [33] and `linarith`[1].

```
    proof (clarify)
      assume asm: "i < Suc j" "Suc j < card X - 1"
      have pj: "?P j (Suc j)" <proof>
      have "i<j ∨ i=j" using asm(1) by linarith
      thus "?P i (Suc j)"
      proof
        ...
        assume "i<j"
        have "j < card X - 1"
          using asm(2) by linarith
        thus "?P i (Suc j)"
          using  Suc.hyps asm(1,2)
            ⟨i<j⟩ chX finiteX Suc_eq_plus1 abc_bcd_acd pj
          by presburger
      qed
    qed
```

The proof of `thm2_ind2` works in the same way, except we induct on decreasing indices. For Isabelle, that means reformulating in terms of an increasing (dummy) induction variable `m`. The lemma `thm2_ind2b` wraps this result in the form given by Schutz, using $k = l - m$, and removes the dummy variable.

```
lemma thm2_ind2:
  assumes chX: "long_ch_by_ord2 f X"
      and finiteX: "finite X"
    shows "∀m l. (0<(l-m) ∧ (l-m) < l ∧ l < card X)
      ⟶ [[(f ((l-m)-1)) (f (l-m)) (f l)]]"
```

B.7 Theorem 10: case (ii)

This appendix gives some detail on the proof of case (ii). Schutz begins by letting "k be the smallest integer such that $[a_1\ b\ a_k]$". This harmless statement requires a lengthy proof of existence in Isabelle.

```
lemma (*for 10*) smallest_k_ex:
  assumes long_ch_Y: "[f[a_1..a..a_n]Y]"
```

[1] Internally, `linarith` relies on Fourier-Motzkin elimination [27, pp. 100-104].

```
      and Y_def: "b∉Y"
      and Yb: "[[a_1 b a_n]]"
    shows "∃k>0. [[a_1 b (f k)]] ∧ k < card Y ∧ ¬(∃k'<k. [[a_1 b (f
   k')]])"
```

The proof is not instructive in detail, so we merely note it proceeds by obtaining the set of all indices of chain elements between a_1 and b: `{i::nat. [[a_1 (f i) b]] ∧ i < card Y}`. We can the obtain its maximum m (provided the set is not empty) using the `Max` operator, and show that $k = m + 1$ satisfies the properties we are looking for.

```
  (* case (ii) *)
  assume "[[a_1 b a_n]]"
  obtain k where k_def: "[[a_1 b (f k)]]" "k < card Y"
      "¬(∃k'. (0::nat)<k' ∧ k'<k ∧ [[a_1 b (f k')]])"
    using smallest_k_ex ...
  obtain g where "g = (λj::nat. if (j≤k-1) then f j else (if (j=k)
   then b else f (j-1)))"
    by simp
  hence "[g[a_1 .. b .. a_n]X]"
    using chain_append_inside k_def fin_X Y_def long_ch_Y ⟨[[a_1 b a
   _n]]⟩
    by auto
  thus "ch X"
    using ch_by_ord_def ch_def fin_long_chain_def
    by auto
```

Once we obtain this index k, Schutz' prose for case (ii) remains somewhat confusing, and we clarify his jumping between different cases by working backwards from the conclusion. While in the edge cases (i) and (iii), we only needed to prove one betweenness relation to obtain a (local) chain, we now need two or three: two if b is close enough to either edge that one cannot find two events "on one side" of it (i.e. it is adjacent to the event of case (i) or (iii)), and three otherwise. In either scenario, we require a relation $[f(k-1)\ b\ f(k)]$.

```
lemma (*for 10*) chain_append_inside:
  assumes long_ch_Y: "[f[a_1..a..a_n]Y]"
      and Y_def: "X = Y ∪ {b}" "b∉Y"
      and fin_X: "finite X"
      and Yb: "[[a_1 b a_n]]"
      and k_def: "[[a_1 b (f k)]]" "k < card Y"
        "¬(∃k'. (0::nat)<k' ∧ k'<k ∧ [[a_1 b (f k')]])"
      and g_def: "g = (λj::nat. if (j≤k-1) then f j else (if (j=k)
   then b else f (j-1)))"
    shows "[g[a_1 .. b .. a_n]X]"
proof -
  ...
  have b_middle: "[[(f (k-1)) b (f k)]]" <proof>
  have b_right: "[[(f (k-2)) (f (k-1)) b]]" if "k ≥ 2"
  proof -
    have "[[(f (k-2)) (f (k-1)) (f k)]]" <proof>
    thus "[[(f (k-2)) (f (k-1)) b]]"
```

```
      using ⟨[[(f (k - 1)) b (f k)]]⟩ abd_bcd_abc by blast
  qed
  have b_left: "[[b (f k) (f (k+1))]]" if "k+1 ≤ card Y -1"
  proof -
    have "[[(f (k-1)) (f k) (f (k+1))]]"
      using ⟨k ≠ 0⟩ f_def fin_Y order_finite_chain that by auto
    thus "[[b (f k) (f (k+1))]]"
      using ⟨[[(f (k - 1)) b (f k)]]⟩ abc_acd_bcd by blast
  qed
```

The conditional facts above mirror Schutz' thinking. However, we did not manage to split the remainder of the proof according to the same conditions. We argue this is because Schutz restricts his attention to a handful of events only, trusting his reader's intuition to convince them that everything else "stays the same". We, on the other hand, need to show that the new way of indexing given by `g` satisfies the definition of a chain *everywhere* on X, explicitly.

```
have "ordering2 g betw X"
  proof -
    have "∀n. (finite X ⟶ n < card X) ⟶ g n ∈ X"
      <proof>
    moreover have "∀x∈X. ∃n. (finite X ⟶ n < card X) ∧ g n = x"
      <proof>
    moreover have "∀n n' n''.
      (finite X ⟶ n'' < card X) ∧ Suc n = n' ∧ Suc n' = n''
          ⟶ [[(g n) (g (Suc n)) (g (Suc (Suc n)))]]"
    proof (clarify)
      fix n n' n'' assume  a:"(finite X ⟶ (Suc (Suc n)) < card X)"
      have cases_n: "n≤k-1 ∨ n≥k+1 ∨ n=k"
        by linarith
      have  cases_sn: "Suc n≤k-1 ∨ Suc n=k" if "n≤k-1"
        using ⟨k ≠ 0⟩ that by linarith
      have  cases_ssn: "Suc(Suc n)≤k-1 ∨ Suc(Suc n)=k"
        if "n≤k-1" "Suc n≤k-1"
        using that(2) by linarith

      show "[[(g n) (g (Suc n)) (g (Suc (Suc n)))]]" using cases_n
      proof (rule disjE3)
        assume "n≤k-1" show ?thesis using cases_sn
```

The first two statements are technical conditions on our definition of `ordering2`, and not present in the monograph [42]. The third statement is a more general version of what Schutz' text aims to prove: index orderings translating to event orderings. Instead of splitting this subproof according to the position of k on the set X indexed by g, we consider different positions of n relative to k, in order to unwrap the definition of g. Interestingly, a single long *smt* call is able to solve the proof at this point, without referencing `b_left` or `b_right`. This solver is not allowed in submissions to the AFP[2],

[2] submission guidelines

and since additionally it clearly diverges from the base text by not requiring results Schutz mentions explicitly, we settle for two further layers of case splits.

```
        proof (rule disjE)
          assume "Suc n ≤ k - 1"
          show ?thesis using cases_ssn
          proof (rule disjE)
            ...
            assume "Suc (Suc n) ≤ k - 1"
            thus ?thesis <proof>
          next
            assume "Suc (Suc n) = k"
            thus ?thesis
              using b_right g_def by force
          qed
```

The first case is straightforward, but requires a long list of referenced results: if all three fixed values are in the given region relative to k, `g` reduces to `f` and the proof is by definition of `f`. The second case places us in one of the cases considered by Schutz, adjacent to k, and uses `b_right`.

```
        next
          assume "Suc n = k"
          show ?thesis
            using b_middle ... by auto
        ...
        qed

      next assume "n≥k+1" show ?thesis <proof>
      next assume "n=k"
        show ?thesis
          using ⟨k ≠ 0⟩ ⟨n = k⟩ b_left g_def Y_def(1) a assms(3)
  fin_Y
          by auto
      qed
    qed
    ultimately show "ordering2 g betw X"
      unfolding ordering2_def
      by blast
  qed
```

We now encounter the second case that can make use of the results given by Schutz, `b_middle`. Going one layer of case splits up again, we continue with the case of $n \geq k+1$. This is similar to $n+2 \leq k-1$ earlier, except now `g i` becomes `f (i+1)`, which leads to extra lines. Finally, the case $n = k$ concludes the proof by using the remaining result `b_left`. Conversion to a total chain using `ordering` works as before.

B.8 Theorem 11

```
lemma int_split_to_segs:
  assumes Q_def: "finite (Q::'a set)" "card Q = N" "N≥3"
      and f_def: "[f[a..b..c]Q]"
      and S_def: "S={s. ∃i<(N-1). s = segment (f i) (f (i+1))}"
  shows "interval a c = (⋃S) ∪ Q"
proof ...
  let "?i = ?u" = "interval a c = (⋃S) ∪ Q"
  show "?i⊆?u"
```

The proof of `int_split_to_segs` splits naturally into two subset inclusion statements, and we point out two key results needed. If we let p be the customary "arbitrary but fixed" element of the set `?i - Q`, we obtain `[[a p c]]`. At this point, we want to find the segment `?s ∈ S` that contains `p` to show that `p ∈ ⋃S`. This is accomplished using a result not present in Schutz, and very similar in style to the lemma `smallest_k_ex` (Sec. 4.4)

```
lemma get_closest_chain_events:
  assumes long_ch_Y: "[f[a_0..a..a_n]Y]"
      and x_def: "x∉Y" "[[a_0 x a_n]]"
    obtains n_b n_c b c
      where "b=f n_b" "c=f n_c" "[[b x c]]" "b∈Y" "c∈Y"
            "n_b = n_c - 1" "n_c<card Y" "n_c>0"
            "¬(∃k < card Y. [[(f k) x a_n]] ∧ k>n_b)"
            "¬(∃k < n_c. [[a_0 x (f k)]])"
```

We now easily show that the elements obtained from this lemma form a segment in `S` and complete proof of the inclusion `?i⊆?u`. The reverse statement `?u⊆?i` follows in a reasonably straightforward manner mathematically, but again the many different objects involved force the mechanisation into several layers of unwrapping and case splits. For `p∈Q`, the statement is reduced to noting that all elements of the chain `Q` are between its edge elements `a, c` (formally, this fact is proved as `ch_all_betw_f`). For `p∈⋃S`, we obtain an index `y` such that `p ∈ segment (f y) (f (y+1))` and derive `[[a p c]]`:

```
    next assume "p∈⋃S"
      then obtain s where "p∈s" "s∈S" by blast
      then obtain y where "s = segment (f y) (f (y+1))" "y<N-1"
        using S_def by blast
      ...
      have "y=0 ∨ y=N-2 ∨ ([[a (f y) c]] ∧ [[a (f (y+1)) c]])"
        by linarith
      hence "[[a p c]]"
      proof (rule disjE3) ...
        assume "[[a (f y) c]] ∧ [[a (f(y+1)) c]]"
        thus "[[a p c]]" using abe_ade_bcd_ace ... by auto
      qed
```

We omit the edge cases: if the index of `y` is known, the proof is an exercise in identifying the elements associated with certain indices and using overlapping orderin lemmas (cf Sec. 4.2). The only interesting result used here is in the final step,

`abe_ade_bcd_ace`. This is another ordering lemma readily derived from our existing ones (again, Sec. 4.2), but not given by Schutz. It is somewhat special because it covers the cases $[abde]$ and $[adbe]$ both, implicitly.

```
lemma abe_ade_bcd_ace:
  assumes abe: "[[a b e]]" and ade: "[[a d e]]" and bcd: "[[b c d]]"
    shows "[[a c e]]"
```